USING USERS' TOUCH DYNAMICS BIOMETRICS TO ENHANCE AUTHENTICATION ON MOBILE DEVICES

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Science and Engineering

2019

By
Pin Shen Teh
School of Computer Science

Contents

Word Count: 54072

List of Tables

List of Figures

List of Algorithms

Abbreviations

AA	Accumulative Adaptation method
AIU	App Interface Unit
API	Application Programming Interface
ADMU	Authentication Decision-Making Unit
AF	Authentication Factor
CA	Cluster Analysis
DF	Decision Fusion
DSU	Data Storage Unit
DTR	Decision Tree
DET	Detection Error Trade-off
DM	Distance Measure
DT	Dwell Time
E-ToDiTA	Enhanced ToDiTA system
EF	Ensemble Fusion
EER	Equal Error Rate
FAR	False Acceptance Rate
FRR	False Rejection Rate
FCU	Feature Construction Unit
FF	Feature Fusion
FLF	First-Level Features
FT	Flight Time
IT	Input Time
IDE	Integrated Development Environment
KNN	K-Nearest Neighbour
MBK-E	MBK Dataset-Easy
MBK-S	MBK Dataset-Strong
MBK-LS	MBK Dataset-Logically Strong

mRMR	minimum-Redundancy-Maximum-Relevance
MAU	Model Adaptation Unit
MTU	Model Training Unit
MFA	Multi-Factor Authentication
NN	Neural Network
NA	No Adaptation method
OCC	One-Class Classifier
OCKNN	One-Class K-Nearest Neighbour
OTP	One-Time Password
OFS	Optimal Feature Subset
PIN	Personal Identification Number
PFS	Preliminary Feature Set
PS	Pressure Size
PM	Probabilistic Modelling
PA	Progressive Adaptation method
RBF	Radial Basis Function
RDAU	Raw Data Acquisition Unit
ROC	Receiver Operating Characteristic
SF	Score Fusion
SLF	Second-Level Features
SFA	Single-Factor Authentication
ST	Statistical
SVDD	Support Vector Data Description
SVM	Support Vector Machine
TAP	Touch Action Press
TAR	Touch Action Release
TCC	Two-Class Classifier
ToDiTA	Touch Dynamics based Two-factor Authentication system

Abstract

Using Users' Touch Dynamics Biometrics to Enhance Authentication on Mobile Devices

Pin Shen Teh

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy, 2019

Mobile devices have become a popular platform for users to access information and digital services, and stay connected. The increased usage and reliance on these devices also imply that they increasingly handle, manage, and process private and sensitive data. As more and more sensitive data are stored in, or accessible from, mobile devices, the risk and cost of losing these data are becoming higher, particularly given the fact that mobile devices are much more vulnerable to theft or loss in comparison with conventional computing devices such as workstations and laptops. Therefore, more stringent security services should be embedded into mobile devices. One of these services is user authentication, i.e. how to verify a claimed identity.

Our initial literature study in the topic area of user authentication on mobile devices shows that authentication on mobile devices should be strengthened and touch dynamics biometrics offers the most usable additional authentication factor than other biometrics alternatives. This has motivated us to investigate how to best exploit the use of users' touch dynamics biometrics to strengthen authentication on mobile devices. To this end, the thesis has made the following three novel contributions.

Firstly, this thesis has presented a thorough investigative study on touch dynamics biometrics authentication on mobile devices. The investigative study has led to the discovery that existing studies mainly focus on improving accuracy performance of the authentication system. The characteristics of mobile devices and the way they are typically used are not given due consideration in these studies. As a result, those systems may have high accuracy performances in a research setting, but may not be realistically usable in practice, limiting the scale of their

deployment.

Secondly, the thesis has proposed and evaluated a novel touch dynamics based two-factor authentication (ToDiTA) system to support user authentication on mobile devices in a secure and usable manner. In proposing this system, we have carried out comprehensive studies of different parameter value settings, different ways of extracting features and different machine learning techniques used to classify the features. The purpose of the studies is to increase accuracy (thus making the system more secure), while, at the same time, reduce overhead costs introduced (thus making the system more usable) as much as possible. The studies have led us to take the following measures in the design: (i) integrating touch dynamics biometrics into a PIN-based authentication method that has a wide social acceptance (improving security and usability); (ii) using descriptive statistical methods to extract additional features from the already-acquired features instead of using features that have to be captured by using additional device sensors (improving usability); (iii) reducing the number of features by selecting and using the most important set of features (improving accuracy and efficiency); and (iv) reducing the touch dynamics data required to train the model by using one-class classification approach (improving usability). In addition, we have used a more comprehensive dataset to evaluate the ToDiTA system so that the conclusion drawn from the evaluation results are more conclusive. The evaluation of ToDiTA showed that by integrating the touch dynamics authentication method into the PIN-based authentication method, along with the above mentioned measures, the ability to counter impersonation attacks is greatly enhanced. For example, if a PIN is compromised, the success rate of an impersonation attempt is drastically reduced from 100% (if only a 4-digit PIN is used) to 9.9% (if both the PIN and the touch dynamics are used).

Thirdly, the thesis has proposed and evaluated an enhanced ToDiTA (E-ToDiTA) system by adding a novel learning capability into the ToDiTA system. The E-ToDiTA system can adapt itself to any changes in a user's touch dynamics pattern. This is achieved by capturing the user's new touch dynamics data as soon as it becomes available and use the new data to update the authentication model. To minimise any additional overhead cost introduced by this addition, we have used a feature spooling process to reduce the number of times required to carry out the model adaptation processes and a progressive adaptation method which uses a fewer number of samples to update the model. The performance of the E-ToDiTA system has been evaluated and compared against that of the ToDiTA system. The results show that, on average, the E-ToDiTA system has improved the accuracy performance by 33.53% in comparison with the ToDiTA system, but with virtually zero increase in overhead cost.

These results indicate that touch dynamics biometrics is a viable option for strengthening user authentication on mobile devices, particularly the capturing of such biometric data does not require additional efforts from a device user; the data can be captured while the user carries out their normal device-using activities.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

First of all, thanks to all the blessings, support and the chance for learning.

I would like to express my sincerest gratitude to my supervisor Dr Ning Zhang for her time, dedication and advice throughout this research. Her helpful discussions, comments and feedback were invaluable, and I really appreciate that. This research would not have been possible without her whole-heartedly support.

I am also grateful to my co-supervisor Dr Ke Chen for providing the support, guidance and valuable feedback. Also, to Prof Andrew Teoh, for all his endless help and support for more than 12 years.

I would also like to thank my wife Jin Yee, for her love, sacrifice and patience during the period. Without her support, this research would have been much difficult to complete.

Most of all I would like to express my deepest gratitude and appreciation to my mother who spent and sacrificed so much of her life to see the moment of me getting a PhD, to my father who takes the most pride in me coming this far. I am very thankful for your love and endless support. Many thanks go to my sister Lyne and other family members for being always helpful and supportive.

This research has been supported financially by The University of Manchester President Doctoral Scholarship Award.

Dedication

To my beloved grandmother, Choo Lee Seah @ Choo Lye Har, who passed away in 2018.

Chapter 1

Introduction

1.1 Background

This section provides the background information for the work reported in this thesis, namely mobile cloud computing, authentication on mobile devices and touch dynamics biometrics.

1.1.1 Mobile Cloud Computing

The use of mobile devices in handling our routine activities is becoming very common. This is particularly the case with the rapid growth and widespread use of smartphones and digital tablets. The processing capability of these devices has advanced to the point where most digital activities that can be accomplished on workstations or laptops can also be performed on mobile devices. Routine activities such as personal and corporate e-mail communications, online banking transactions, accessing paperless prescription services, route navigation, etc. can be carried out ubiquitously with these devices.

Mobile cloud computing technologies enable mobile devices to run more complex applications by offloading the more computationally intensive modules of a mobile application to the cloud. These modules are instead executed by the computing resources in the cloud (e.g. servers, storage, networks, applications and services), reducing the computational requirements to run the application [1]. In this way, mobile devices can run more complex applications for use in various application areas such as those discussed as follows:

Healthcare and Wellness. Mobile medical monitoring applications allow healthcare providers to collect, monitor and diagnose patients' latest medical conditions remotely over the internet [1]. With such applications, patients can receive timely diagnoses and effective treatments.

Education. The use of mobile devices to support e-learning enhances the learning experience. With mobile learning applications, such as Blackboard App [2], Coursera Mobile App [3] and Udacity for Mobile [4], learners and educators can access content-rich learning materials (e.g. videos, blogs, forums and interactive quizzes) at any time and from almost anywhere.

Business and Banking. Business activities such as shopping, booking, ticketing and advertising can be seamlessly performed on-the-go with the use of custom-made mobile applications catered for different types of businesses. Financial institutions can provide their customers with hassle free and secure banking services (e.g. money transfer, bill payment and balance enquiry) on-the-go with the use of mobile banking applications.

Social Media and Information Sharing. According to a report [5], there are 3.028 billion active social media users globally in 2017, in which 2.78 billion (or 91.8%) of them accessed their social media accounts via mobile devices. The increasing use of mobile devices to access social media contents revolutionised the way we receive, process and share information.

Disaster Relief and Recovery. In the event of a severe natural disaster, people trapped in, or having access to, the disaster areas could use the camera embedded on their mobile devices to capture photos of their surroundings and upload the photos to the cloud. These photos can be quickly processed and analysed to generate valuable information (e.g. severity of the damage or possible access route) that can facilitate and expedite rescue mission [6].

Gaming. According to Newzoo, a video games research firm, the worldwide mobile games revenue will account for 50% of the $180.1 billion generated from total worldwide gaming market in 2018 [7]. This statistic shows that mobile devices are increasingly used for game entertainment purposes.

Based on the discussion above, it is clear that the use of mobile devices has brought benefits to several existing application areas, and they have promised

to bring further benefits to other application areas. This also means that the use of mobile devices is experiencing continuous and rapid growth. According to a forecast by Cisco, there will be approximately 11.6 billion mobile-connected devices by 2021, and the global mobile data traffic will increase nearly sevenfold between 2016 and 2021, reaching 49 exabytes per month by 2021 [8]. These staggering statistics show our increasing reliance on mobile devices and also imply that our private and sensitive data will increasingly be handled, managed and processed by these devices. Therefore, the security of accessing mobile devices and accessing data, services and other resources through these devices are of prime concern. More stringent security services should be embedded on mobile devices. One of these services is user authentication, i.e. how to securely verify a claimed identity. In the next section, we further discuss user authentication in the context of mobile device.

1.1.2 Authentication on Mobile Devices

Authentication is the first-line defence in any computer system or device as it is a pre-requisite for several other security services such as authorisation and accountability. In a mobile device context, authentication is typically achieved via a knowledge-based authentication method, and with this method, a user proves their identity by demonstrating the knowledge of a secret. This secret could be a character-based passcode, a digit-based passcode (also referred to as Personal Identification Number (PIN)) or a graphical-based passcode.

The use of passcodes is vulnerable to a number of security attacks. Studies have shown that an attacker can easily reproduce or derive a passcode from smudge (oily residues left on a touchscreen surface) [9], shoulder spoofing [10] and brute force attacks [11]. Some studies even suggested that an attacker could infer a passcode by analysing data recorded from the accelerometer and gyroscope sensors embedded in more recent mobile devices [12, 13]. Wrongful user usage behaviour such as selecting easy to remember passcode, sharing of a passcode and using the same passcode across multiple systems makes these attacks easier to perform and worsened its consequences. Even with the presence of these attacks, knowledge-based authentication methods are still the primary methods used to authenticate mobile users [14]. Therefore, to thwart these attacks or to make these attacks harder to succeed, a more robust authentication method is needed.

An alternative to knowledge-based authentication methods is biometrics-based

authentication methods. The latter identifies a person based on his/her physiological or behavioural characteristics. Physiological biometrics refers to the physical features of a human body such as fingerprint, facial characteristics and iris pattern. Behavioural biometrics refers to the traits acquired from human behaviour or habits like signature, voice, gait and touch dynamics.

To ensure that a biometrics authentication method is effective, the method should be not only secure but also usable. This is supported by the findings reported in two separate studies [15, 16]. In the first study, the authors conducted a survey to investigate how participants ranked the perceived security protection and the usability (willingness to use the method) of different types of biometrics authentication methods on mobile devices. The survey reported that the iris and voice biometrics authentication methods were ranked the highest in terms of the perceived security protection factor, but the lowest in terms of the usability factor.

The findings in the second study enforced the findings of the first study. The authors reported that one of the important factors influencing the participants' choice as to which biometrics authentication method they prefer to use on a mobile device is the usability factor. They identified two primary usability issues that may put off participants from using a biometrics authentication method: (i) slow authentication speed and inconvenience, and (ii) social awkwardness. For (i), in the case of face biometrics, participants felt that it was time-consuming and difficult to align the face correctly in front of the device's camera. In the case of fingerprint biometrics, the participants felt that it was hard to scan a fingerprint properly when the fingers were too oily or dry, or when the device was covered with a protective case, and this often leads to a false negative authentication result [17]. For (ii), participants felt that it was awkward to hold a device in front of a face to perform an authentication task in public. This is more so the case in the context of mobile devices, where the use of these devices in public is very common and frequent.

Furthermore, using a single biometrics authentication method to form an authentication system introduces a number of limitations that may hinder usability. For example, a face-based authentication method imposes a higher privacy risk than other biometrics-based authentication methods as it can be more likely used in a covert manner [18]. A touch dynamics-based authentication method suffers

from a higher false rejection rate than other physiological-based biometrics authentication method. A hand geometry-based authentication method is less practical to be used in a mobile device application context because the physical size of a hand geometry is large [19]. A voice-based authentication method is sensitive to a number of factors such as background noise as well as the emotional and physical state of the user [20].

One of the possible measures to make the authentication service more secure, while, at the same time, without hindering usability, is to integrate a biometric-based (e.g. touch dynamics) authentication method with a knowledge-based authentication method to form a so-called two-factor authentication system. Such system can make unauthorised accesses to mobile devices harder, thus strengthening the security protection level of mobile devices, while, at the same time, still be able to maintain usability as high as possible. The existing knowledge-based authentication method has a wide social acceptance, and the touch dynamics authentication method is also expected to be widely acceptable by the general public [21]. The following section introduces touch dynamics biometrics.

1.1.3 Touch Dynamics Biometrics

Touch dynamics is a behavioural biometrics, which captures the way a person touches on a touchscreen mobile device. When a human interacts with such a device, a digital signature is generated. The signatures generated are believed to be rich in discriminative properties, which are relatively unique to each individual and can be used to identify a person/user [22].

Touch dynamics authentication method can be implemented by employing sensors already available in most mobile phones, digital tablets and other touchscreen devices, making the implementation of this authentication method comparatively cheaper than other biometrics-based authentication methods such as fingerprint-based and iris-based authentication method where additional or specialised hardware is required. The availability of higher resolution sensors in recently released mobile devices provides added opportunities to the development of touch dynamics biometrics, as these sensors allow the extraction of more discriminative features. Also, the acquisition of touch dynamics features is less sensitive to external factors such as lighting conditions and/or background noise levels, making it more usable and reliable in a mobile context. Mobile devices usually operate in an on-the-go manner, so their background noises and lighting

may change continuously. These external factors may cause problems to some biometrics authentication method. For example, a voice-based authentication method is more affected by the background noise level [23]. A face-based authentication method using the visible lighting approach would not work well in weak lighting conditions [24]. A face-based authentication method using active invisible lights may work well in weak lighting conditions [25], but the approach requires the use of special sensors incorporated into a device, increasing cost.

In addition, touch dynamics authentication method can operate in parallel with a person's normal mobile device usage activities and is non-intrusive [26]. For example, different from the case for a fingerprint-based or iris-based authentication method where the acquisition of biometrics features has to be done via separate authentication related input activities, touch dynamics features can be acquired whenever a user uses his/her device. In other words, the acquisition of touch dynamics features can be carried out during a user's normal (i.e. non-authentication related) input activities, requiring little extra interactions by the user. This way of acquiring biometrics features brings two further benefits. Firstly, the authentication method can verify the identity of a user at any point of the user's interaction with his/her device, and do so in a non-intrusive and transparent manner (i.e. without explicit attention from the user). Secondly, the authentication method is less vulnerable to privacy risks, as touch dynamics data acquired at different times exhibit a degree of variability, and the higher the variability, the lower the privacy risk [27].

While using touch dynamics to identify a user has a number of advantages, it also introduces challenging issues. The next section describes the research motivation and highlights the challenging issues.

1.2 Research Motivation and Challenges

From our discussions above, it is clear that authentication on mobile devices should be strengthened, and touch dynamics biometrics offers a desirable additional authentication factor that can be used to strengthen the authentication. Through our literature study of the state-of-the-art in the topic area of touch dynamics biometrics authentication, it was discovered that existing studies mainly focus on improving accuracy performance of the authentication system. The characteristics of mobile devices and the way they are typically used are not given

due consideration in these studies. This observation has motivated the research conducted in this thesis.

This research is aimed at investigating how to best (in terms of security and usability) strengthen user authentication on mobile devices by using touch dynamics biometrics. We answer this question by designing and evaluating a touch dynamics authentication system to strengthen user authentication on a mobile device. In doing so, the following challenging issues are identified:

- **Maximising Accuracy:** The accuracy performance of a touch dynamics authentication system is relatively lower than other physiological biometrics authentication system (e.g. fingerprint and iris). This is because touch dynamics data acquired at different occasions are likely to exhibit a certain degree of variations due to external factors, such as fatigue, mood or distraction. Therefore, consideration should be given as how to increase the accuracy performance of a touch dynamics authentication system in the design.

- **Minimising Computation and Communication Costs:** Computational capabilities of mobile devices are typically lower than desktop computers. This means that certain criteria, such as algorithm complexity, should be considered with the aim of reducing communication cost and authentication delays. In other words, computational and communication costs introduced as the result of deploying the authentication system should be as low as possible.

- **Minimising Energy Consumption:** Mobile devices, unlike their desktop counterparts, are typically battery powered, so the less the energy an application consumes, the longer the device can operate. Though communication is the primary source of power consumption of a device [28], other factors such as the number and usage frequencies of various sensors embedded in a device, which are used to extract touch dynamics data, also have a direct impact on the mobile device battery consumption level. Therefore, how to reduce the energy consumption of a mobile device should be taken into consideration in the design.

- **Adaptive Learning Capability:** Usually, human behavioural characteristics change more frequently than their physiological characteristics. Similarly, a user's touch dynamics pattern may change gradually as the user is

getting more familiar with the passcode, input method, device and other external factors. A robust touch dynamics authentication system should be capable of adapting itself to any changes in a user's touch dynamics pattern.

1.3 Research Aim and Objectives

This research aims to investigate how to best exploit the use of users' touch dynamics biometrics to strengthen authentication on mobile devices. This aim is supported by the following objectives.

1. To investigate and analyse the characteristics of authentication on mobile environment and the authentication factors used in such environment to identify the security threats associated with the use of the authentication factors.

2. To analyse and specify requirements to guide the selection of authentication factors used for the design of a secure and usable authentication solution for mobile device based on the threat analysis performed in Objective 1.

3. To investigate and critically analyse related work in the topic area of touch dynamics biometrics authentication on mobile device with the aim of identifying gaps in knowledge and investigating novel measures and ideas to improve existing solutions in terms of accuracy performance and usability.

4. To design a touch dynamics based two-factor authentication (ToDiTA) system to strengthen user authentication on mobile devices with as high accuracy, and as low cost, as possible. In designing this system, we also investigate the effects of various parameter settings on the performance of the designed system.

5. To extend the ToDiTA system for it to have an learning capability so that it can also adapt itself to any changes in a user's touch dynamics pattern. This extended system is called an enhanced ToDiTA system, i.e. the E-ToDiTA system.

6. To evaluate and compare the performances of both designed systems using simulation.

1.4 Research Methodology

The research methodology utilised in this research consists of four key components: literature survey and critical analysis of the related work, system design, and implementation and evaluation.

1.4.1 Literature Review

The first task carried out in this research was to study in-depth the related work in the literature. We started by investigating the topic area of authentication on mobile environment. The purpose of this study was to become familiar with the characteristics of authentication on mobile environment and the current authentication factors used in such environment, and to identify the security threats associated with the use of these authentication factors. Upon completion of this study, it becomes apparent that touch dynamics biometrics is the preferable authentication factor that we could use to design our solution. The next step was to critically analyse the existing solutions to identify their strengths and limitations, with the aim of building our solution on the strengths of existing solutions, but overcoming their limitations. The insights gained from the analyses have led us to the design of our secure and usable two-factor authentication system. Reviewing relevant literature was carried on throughout the duration of this research. As new work was published, it was reviewed, and necessary findings were taken into account. Performing the literature review and critically analysing the literature satisfies Objectives 1, 2 and 3.

1.4.2 System Design

Following the literature review, a number of research gaps were identified. Based on the identified gaps and the research aim, measures and ideas were formed to design systems that address the gaps. The system measures and ideas were repeatedly refined by considering input from existing work and by our progressively insightful thinking towards the research problem. Considerations were given to reduce any additional overheads imposed to the proposed systems. At the conclusion of this phase, two novel systems were designed. The first system was the ToDiTA system. The second system was an enhanced version of ToDiTA, i.e. the E-ToDiTA system. The design of these two systems satisfies Objective 4 and 5.

1.4.3 Implementation and Evaluation

The next stage of our research was to implement and evaluate the designed systems. The performances of the designed systems were evaluated using simulation method. The implementation of the simulation-based evaluations was carried out using MATLAB programming platform. Before the implementation, it was necessary first to perform two tasks: (i) define the evaluation metrics, and (ii) design the evaluation methods. Following these tasks, the simulations were implemented using the evaluation methods to provide accurate measures of the metrics. The simulations were carried out in two stages. In the first stage, the simulation was run under various parameter settings to evaluate the impacts of different parameter value settings on the performance of the ToDiTA system. In the second stage, the simulation was carried out to compare the ToDiTA system with the E-ToDiTA system to demonstrate the enhancement of E-ToDiTA over ToDiTA. The collected results from the simulation runs were plotted into graphs and recorded into tables, and these graphs and tables were then used to analyse and evaluate the performances of the proposed systems and to compare with the most relevant work to demonstrate the merits of the proposed systems over the related work. The analyses and evaluations provided satisfy Objective 6. Conclusions were drawn from the evaluations of the designed systems, and directions for future research were identified.

1.5 Novel Contributions

The research work presented in this thesis has led to the following novel contributions.

Novel Contribution 1: An investigative study of touch dynamics biometrics authentication on mobile devices [29]
Novel contribution 1 has two parts. In the first part, an initial literature study has been conducted on authentication on mobile environment and the types of authentication factors used in such environment. Following this, we have identified the security threats imposed on such environment and specified a set of requirements to guide the selection of authentication factors for the design of authentication system that counters the identified threats. By analysing these

authentication factors against the specified requirements, touch dynamics biometrics was identified as the most desirable for the system. This has led to the second part of novel contribution 1. In the second part, a thorough investigative study has been conducted on touch dynamics biometrics authentication on mobile devices to gain insights and comparative analysis on the current state-of-the-art, including the raw data acquisition protocols, feature data representations, feature data classification techniques, evaluation criteria, as well as experimental settings and evaluations. The investigative study has led to the identification of the research gaps described in Section 3.10 in Chapter 3. Based on these gaps, recommendations for a way forward were made. These recommendations have led to the generation of the measures and ideas used in the designs of the secure and usable authentication system proposed in this thesis, which is the second and third novel contributions in this thesis, described below.

Novel Contribution 2: A secure and usable system for strengthening user authentication on mobile devices [30, 31]

A touch dynamics based two-factor authentication (ToDiTA) system has been proposed. This system integrates touch dynamics biometrics into existing PIN-based authentication method to form a two-factor authentication system. In this two-factor authentication system, the touch dynamics biometrics serves one factor, and the touch PIN-based authentication method serves the other factor. It requires additional effort to successfully bypass the authentication control of such system, as, in this case, to successfully launch an impersonation attack, an impersonator would have to compromise both authentication factors. Experimental results show that by integrating touch dynamics biometrics into PIN-based authentication method, the ability to counter impersonation attacks is greatly enhanced. The results indicate that the idea of using touch dynamics biometrics to strengthen user authentication on a mobile device or application context is effective. To this end, the benefits of the ToDiTA system are clear, yet, the system not effective when a user's touch dynamics pattern changes. This limitation has led to the design of an enhanced system (the E-ToDiTA system), which is the third novel contribution in this thesis, described below.

Novel Contribution 3: An enhanced touch dynamics based two-factor authentication system with adaptive learning capability [32]
The E-ToDiTA system extends the ToDiTA system to provide adaptive learning capability, making it effective in adapting itself to any changes in a user's touch dynamics pattern. This is achieved by updating the authentication model with new touch dynamics data as soon as the new data becomes available. In the design of the E-ToDiTA system the following ideas have been used:

- A feature screening process to examine whether the acquired new data are suitable for use to update the model, reducing the likelihood that the updated model may incorrectly deviate from the owner's real touch dynamic pattern.
- A feature spooling process to limit the number of times required to carry out the model adaptation processes, reducing computational overhead and improving usability.
- A progressive adaptation method which uses a fewer number of samples to update the model, reducing the storage overhead required to store the samples and the risk of overfitting the model.

The E-ToDiTA system is evaluated against the ToDiTA system. The evaluation results show that E-ToDiTA performs better in two ways: it is more effective in improving the accuracy performance of the system when a user's touch dynamics pattern changes and do so without introducing excessive overhead to the mobile device.

1.6 Publications and Industry Collaboration

Parts of the research presented in this thesis have led to the following journal and conference publications, poster presentation and industry collaboration.

Journal Papers

1. **Pin Shen Teh**, Ning Zhang, Ke Chen, and Qi Shi, "A Touch Dynamics Based Authentication System for Mobile Devices with Adaptive Learning Capability", *Transactions on Mobile Computing, IEEE*, in progress.

2. **Pin Shen Teh**, Ning Zhang, Ke Chen, and Qi Shi, "Strengthening User Authentication on Touchscreen Mobile Devices with User's Touch Dynamics Pattern", *International Journal of Human-Computer Studies, Elsevier*, submitted and under review.

3. **Pin Shen Teh**, Ning Zhang, Andrew Beng Jin Teoh, and Ke Chen, "A Survey On Touch Dynamics Authentication In Mobile Devices", *Computers & Security, Elsevier*, vol.59, pp.210-235, 2016, doi:10.1016/J.COSE.2016.03.003.

4. **Pin Shen Teh**, Ning Zhang, Andrew Beng Jin Teoh, and Ke Chen, "TDAS: A Touch Dynamics Based Multi-factor Authentication Solution for Mobile Devices", *International Journal of Pervasive Computing and Communications, Emerald*, vol.12, no.1, pp.127-153, 2016, doi:10.1108/IJPCC-01-2016-0005.

Conference Papers

1. Nian Chi Tay, Connie Tee, Thian Song Ong, Michael Kah Ong Goh, and **Pin Shen Teh**, "A Robust Abnormal Behavior Detection Method Using Convolutional Neural Network", *In Computational Science and Technology, Lecture Notes in Electrical Engineering, Springer*, vol.481, pp. 37-47, 2019, doi:10.1007/978-981-13-2622-6_4.

2. **Pin Shen Teh**, Ning Zhang, Andrew Beng Jin Teoh, and Ke Chen, "Recognizing Your Touch: Towards Strengthening Mobile Device Authentication via Touch Dynamics Integration", *In Proceedings of the 13th International Conference on Advances in Mobile Computing and Multimedia (MoMM)*, pp. 108-116, 11-13 Dec. 2015, Brussels, Belgium. ACM, doi:10.1145/2837126.2837127. Acceptance rate: 30.2%.

Poster Presentation

The author was selected to present the potential of this research in The University of Manchester Postgraduate Summer Research Showcase 2018. In the showcase, an interactive poster was presented with the title "Enhancing Mobile Device Authentication by Using Touch Dynamics Biometrics". Only twelve researchers across the University were selected.

Industry Collaboration

Parts of this research have been put into practical application. With the support of funding by the University (Higher Education Innovation Funding: Connecting Capability Fund), the author has implemented the two-factor authentication system for potential exploitation by Worldpay (a leading global payments provider) to support secure payment applications. This implementation is a working prototype in the form of a mobile application. The application analyses users' touchscreen input patterns to enhance PIN security. The project has been completed successfully, and Worldpay has expressed interest in working on a new proposal for the next stage of funding.

1.7 Thesis Structure

The remainder of this thesis is organised as follows (the structure of the thesis is further illustrated in Figure 1.1).

Chapter 2 provides a background about authentication on mobile environment and the authentication factors used in such environment. The chapter identifies the security model involved and analyses the security threats imposed on the model. It also specifies the requirements to guide the selection of authentication factors for the design of secure and usable authentication solutions that counter the identified threats.

Chapter 3 presents a detailed investigative study of the related work in the topic area of touch dynamics biometrics. Based on the identified strengths and weaknesses of the existing work, the chapter identifies research gaps in these work and makes recommendations for a way forward. The analyses provided in this chapter constitute the first novel contribution of this research.

Parts of this research have been published in peer-reviewed journal paper: "A Survey on Touch Dynamics Authentication in Mobile Devices" [29].

Chapter 4 presents the design and evaluation of the second novel contribution of this research, i.e. the ToDiTA system. The measures and ideas used in the design of the ToDiTA system are based on the recommendations made in Chapter 3. The ToDiTA system is evaluated under different parameter value settings. The shortcoming of the ToDiTA system is identified.

Parts of this research have been published in peer-reviewed journal and conference paper: "TDAS: A Touch Dynamics based Multi-Factor Authentication Solution for Mobile Devices" [30] and "Recognizing Your Touch: Towards Strengthening Mobile Device Authentication via Touch Dynamics Integration" [31], respectively.

Chapter 5 presents the design and evaluation of the E-ToDiTA system, which is an enhanced version of the ToDiTA system. This chapter presents the third novel contribution of this research. The novelty of the E-ToDiTA system is that it is capable of adapting itself to any changes in a user's touch dynamics pattern. The E-ToDiTA system is evaluated against the ToDiTA system.

Parts of this research have been submitted for a peer-reviewed journal paper: "Strengthening User Authentication on Touchscreen Mobile Devices with User's Touch Dynamics Pattern" [32]. Another journal paper is in progress. In addition, parts of this research have been implemented into a working prototype in partnership with Worldpay.

Chapter 6 concludes this thesis and suggests directions for future research.

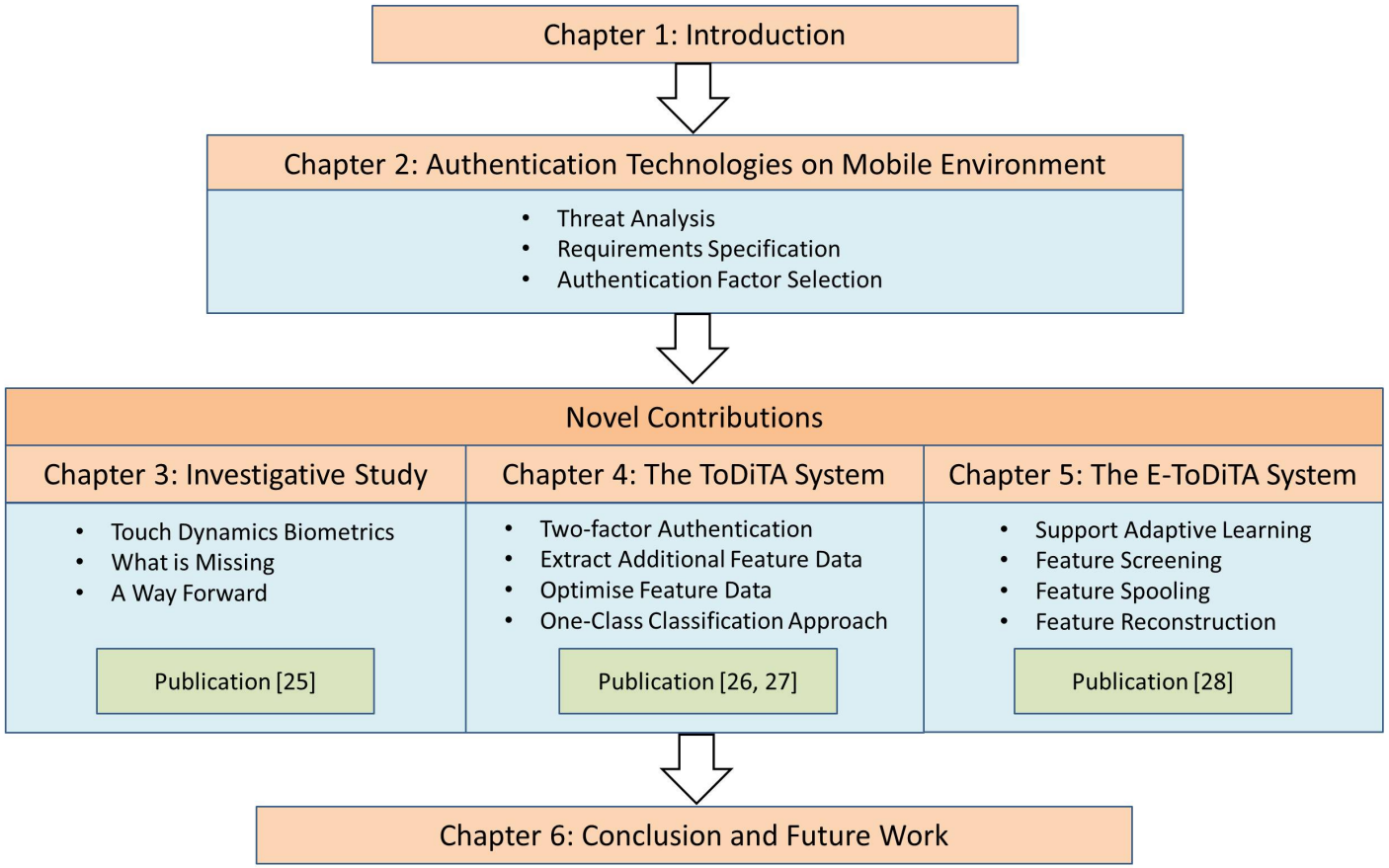

Figure 1.1: Thesis structure

Chapter 2

Authentication Technologies on Mobile Environment

2.1 Chapter Introduction

This chapter gives an overview of authentication on mobile environment and the types of authentication factors used in such environment. It also identifies the security models involved and the security threats imposed on the models. In addition, the chapter specifies a number of requirements that are used to guide the selection of authentication factors for the design of authentication system that counters the identified threats. Further, it analyses the different types of authentication factors against the specified requirements and identifies the most desirable authentication factors for the system.

In detail, Section 2.2 presents an overview of authentication in relation to our research context. Section 2.3 describes the types of authentication factors used including their strengths and weaknesses. Section 2.4 discusses the differences between single-factor and multi-factor authentication approaches. Section 2.5 identifies the entity and security models involved. Section 2.6 identifies the security threats imposed on the identified models. Section 2.7 specifies a set of requirements for the selection of authentication factors. Section 2.8 presents a comparative analysis of the authentication factors against the specified requirements. Finally, Section 2.9 summarises this chapter.

2.2 Authentication Overview

Authentication refers to the process of initiating trust in, or verifying, the identity of a user, process or device [33]. This process plays a major part in protecting private and sensitive data from unauthorised access. Depending on the application scenario, authentication can be performed using various methods. In a conventional banking scenario, such as the opening of a bank account, the primary method used to verify the identity of an individual is by comparing the valid proof of identity (e.g. passport or driving licence) against the physical presence of the individual. In a mobile application scenario, the use of physical presence to verify the identity of a mobile user is harder and less practical, as a mobile user typically access a mobile device in an on-the-go manner. As such, alternative methods for verifying the identity of a mobile user that can achieve a comparable level of trust are required. Apart from the physical presence of an individual, authentication can also be performed using different types of credentials, also known as authentication factors (AFs). In the following section, we discuss the AFs in detail.

2.3 Authentication Factors (AFs)

AFs are normally classified into three categories as follows:

- Knowledge-based AFs ("something a person knows")
- Possession-based AFs ("something a person has")
- Biometrics-based AFs ("characteristic belonging typically to a person")

In the following sections, we discuss the AFs in each category, highlighting their strengths and weaknesses. The discussion focuses on AFs used in the mobile environment.

2.3.1 Knowledge-Based AFs

Knowledge-based AFs refer to the use of a shared secret or password (also known and hereafter referred to as a passcode) as evidence for identity verification. Passcode has a number of advantages (+) and disadvantages (-) as follows:

- Easy to use (+).

- Cheaper implementation (+).
- Widely acceptable (+).
- Vulnerable to shoulder surfing and brute force attacks (when a passcode is short in length and low in complexity) (-).
- Difficult to remember (when a passcode is long in length and high in complexity) (-).

There are three different forms of passcode: (i) textual, (ii) digit, and (iii) graphical. In the following, we further discuss these three forms of passcode.

Textual passcode: A textual (also known and hereafter referred to as character) passcode consists of letters, digits, symbols or the combinations among them. It has been commonly used for access control to digital systems accessed through the use of physical keyboards equipped on desktop and laptop workstations. Mobile devices, unlike their desktop and laptop counterparts, are equipped with virtual keyboards. The use of a virtual keyboard to input a character passcode is harder and slower than on a physical keyboard for the following two reasons. Firstly, a virtual keyboard is significantly smaller than a physical keyboard, making it harder to input the character keys. Secondly, due to the limited space on a virtual keyboard, the layout of the character keys is spread over different virtual keyboard screens. Therefore, to complete the input of a passcode, users have to toggle between different screens, making passcode entry slower and inconvenient. This is more so the case when a passcode consists of the combination of letters, digits and symbols.

Digit passcode: A digit passcode (or a PIN), consists of numerical symbols ranging from 0 to 9. A PIN, typically 4-digit in length, is the most commonly used AF on mobile devices [34]. The PIN has two advantages over a character passcode. Firstly, the sizes of digit keys are larger on a virtual keyboard screen, making it easier to input the intended key. Secondly, it is easier to remember a PIN that a character passcode. However, a PIN has a lower entropy than a character passcode (when both passcodes have the same length), making the PIN more vulnerable to shoulder surfing and brute force attacks.

Graphical passcode: A graphical passcode can be classified into three categories [35]: recall-based, cued recall-based and recognition-based. A recall-based

passcode consists of either a drawing (e.g. DooDB [36]) or a sequence of connected dots on a virtual grid interface (e.g. Pass-Go [37]). A cued recall-based passcode consists of a sequence of hidden points in an image (e.g. PassPoints [38]). A recognition-based passcode consists of a selection of images from a list of images (e.g. Awase-E [39]). A graphical passcode has two advantages over a character passcode and a digit passcode in terms of usability. Firstly, the human brain is more capable of remembering graphical information than character and digit sequences [40]. Secondly, the process of inputting a graphical passcode is easier and faster than a character passcode and a digit passcode [41]. However, a graphical passcode has three weaknesses. Firstly, a recall-based passcode is vulnerable to smudge attack [42]. Secondly, a recognition-based passcode requires much more storage space than a character passcode [43]. Thirdly, the time required to input a recognition-based passcode is longer than a character passcode and a digit passcode [35].

2.3.2 Possession-Based AFs

Possession-based AFs refer to the use of a valid possession of an object as evidence for identity verification. Examples of possession-based AFs are a key to a lock, a key fob, a photographic identity card and a One-Time Password (OTP) generator or paper list. OTP is one of the commonly used possession-based AF on mobile devices [44]. OTP, as its name suggest, is a passcode that can only be used once. The content of the passcode is a randomised string generated by a predefined algorithm stored in a token. There are two different types of token: hardware and software. A hardware token is a small device that can be conveniently carried. A software token is an application that can be installed on a mobile device. There are three disadvantages of using a token. Firstly, the possession of a valid token alone is not sufficient to guarantee that the individual who possesses the token is the legitimate user (e.g. the token may be stolen from the legitimate user). Secondly, the production, distribution and maintenance of a token incur additional cost to the service provider. Thirdly, it is inconvenient for users to carry or manage multiple tokens as, normally, different service providers use different tokens [45].

2.3.3 Biometrics-Based AFs

Biometrics refer to human traits or characteristics that are uniquely associated with an individual. Biometrics-based AFs have four properties that make them useful as a personal identifier [46]: (i) unique among different individuals, (ii) harder to be transferred or shared, (iii) less likely to be forgotten or lost, and (iv) difficult to forged or duplicated. However, unlike knowledge-based and possession-based AFs, the use of biometrics-based AFs may result in false positive (i.e. wrongly identify an impersonator as a legitimate user) and false negative (i.e. wrongly identify a legitimate user as an impersonator) authentication decisions [44]. In general, biometrics are divided into two categories: physiological and behavioural. In the following sections, we discuss the two biometrics categories in detail.

2.3.3.1 Physiological Biometrics

Physiological biometrics refer to the physical characteristics or measurement of parts of the human body [47], such as fingerprint, palmprint, face and iris. Physiological biometrics are highly unique among different individuals and remain relatively stable or less likely to change drastically over a human's lifetime. These two properties make using physiological biometrics AFs more accurate than behavioural biometrics AFs when used as a personal identifier. While physiological biometrics AFs are more accurate at identifying an individual, when compromised, imposes greater security and privacy risk as physiological biometrics data are not revocable (i.e. cannot be altered or replaced) [48]. In the context of mobile device, four types of physiological biometrics have been used: fingerprint, face and iris. They are further discussed in the following.

Fingerprint: Human fingerprint consists of pattern of ridges, valleys and minutiae points that makes it unique among a wide population. No two fingerprints are identical. It is the most mature and establish AF used compared to the other biometrics AFs [49]. Fingerprint has been used in areas such as law enforcement, forensic investigation and access control. In recent years, fingerprint has been increasingly used for authentication on mobile devices. For example, Apple Touch ID [50], enables fingerprint to be used as an alternative to knowledge-based AFs for activities that require proof of identity (e.g. unlock a device, authorise a transaction and access to a mobile application). This AF is subject to two

drawbacks. Firstly, it requires the use of a dedicated fingerprint scanner to capture and extract fingerprint pattern. This means that it incurs additional cost to users, hindering the widespread adoption and deployment of the authentication system. Secondly, the acquisition of fingerprint pattern is affected by external factors such as sweat, cuts or dry skin on the finger [50], preventing the proper capture of fingerprint pattern.

Face: Human face consists of features that can be analysed, compared and used for identity verification. Face recognition is widely used for border control and crime investigation [51]. Face image can be captured via camera built-in on most mobile devices, keeping the cost of implementation low. Capturing good quality face images is essential for this AF to work effectively. The quality of face images are affected by two factors [52]: (i) lighting condition (e.g. under or over-exposed lighting condition), and (ii) image shape, size and orientation (e.g. distance and angle of a captured face image). These two factors are often difficult to control in a mobile device usage scenario, making it harder to capture good quality face images.

Iris: Human iris is formed and developed during gestation (i.e. the period between conception and birth of a baby), and once developed remains permanent throughout a human's lifetime. Iris pattern has a very complex structure which is unique among different individuals, is difficult to replicate or forge, and when used as an AF, can achieve a high level of accuracy performance [53]. There are two concerns when implementing an iris-based AF on a mobile device. Firstly, acquiring iris image requires the use of a specialised infrared camera, and most commercial off-the-shelf mobile devices are not equipped with such a camera [54]. This means that the implementation of this AF on a mobile device requires the use of additional hardware, increasing the implementation cost and limiting the large-scale deployment of this AF. Secondly, the processing of iris images involves a number of computationally intensive processes. The average mobile devices do not have the required hardware components capable of executing these processes efficiently [53].

2.3.3.2 Behavioural Biometrics

Behavioural biometrics refer to the measurable patterns of human activities such as handwritten signature, voice, gait and touch dynamics. Behavioural biometrics

data can be acquired transparently without requiring users to perform additional authentication task. For example, gait data can be captured by carrying a motion sensing device. In contrast, to acquire physiological biometrics data, users are required to align their body parts with a sensor of the device. Behavioural biometrics AFs have one major drawback. Unlike physiological biometrics (with exception to face) which remain relatively stable over a human's lifetime, behavioural biometrics are less stable and may gradually change over time. These changes are caused by factors such as ageing, emotional state and health conditions, and may have a negative effect on the performance of behavioural biometrics-based authentication systems. In the context of mobile device, four types of behavioural biometrics have been used: voice, gait, touch dynamics and gesture. They are further discussed in the following.

Voice: Human voice varies in terms of vocal tone, pitch and accent [55]. These variations are unique across different individuals. Voice samples can be recorded by using microphone embedded in every mobile device, thus the implementation cost of this AF is low. In addition, voice-based AF has a higher level of user acceptability than other biometrics-based AFs (e.g. iris) [56]. This is because voice-related functionalities are very common on mobile devices (e.g. making a phone call, voice search and voice recording), suggesting that most users are familiar and comfortable with using voice as a method of interaction with their devices. The primary concern when implementing a voice-based AF is the background noise level during the voice recording process. Mobile devices are often used in an on-the-go manner, so their background noises may change continuously. Inconsistent background noise level may affect the accuracy performance of the voice-based AF.

Gait: Gait refers to the way a person walks. In the past, gait data have been collected via three different methods: (i) video recordings, (ii) floor mat embedded with motion sensors, and (iii) motion sensors attached to different body parts [57]. Today, gait data are collected via mobile device embedded with motion sensors, increasing the deployment scale of this AF. Gait data can be used to verify a user transparently and continuously with the condition that the user is constantly moving and in possession of a device equipped with motion sensors. These conditions also mean that this AF will not work in situations when the user is not moving (i.e. sitting or standing).

Touch dynamics: Touch dynamics refer to the digital signatures generated when a human interacts with a mobile device. The generated signatures are believed to be rich in discriminative properties, which are relatively unique to each individual and hold potential as personal identifier [38]. The acquisition of touch dynamics data can be carried out during a user's normal input activities, requiring little extra interactions by the user. The challenging issue in the use of this AF is that the data acquired at different instances are likely to exhibit a certain degree of variations due to external factors such as fatigue, mood and distractions. These variations may affect the accuracy performance of this AF. Section 3.2 provides further discussions on this AF.

Gesture: Gesture refers to the physical manipulation of direction and movement when a user is holding a device in the air. In the context of mobile device, gesture can be captured in two different forms: standard control and loosely-defined [58]. Standard control gesture refers to the creation of geometrical shapes, letters, words or the combination of them, similar to signing a hand-written signature [59]. Loosely-defined gesture includes tapping on any surface, tilting, flipping and shaking of a mobile device [60]. The fundamental hardware requirement for acquiring gesture data is an accelerometer sensor that measures the movement of a device along the x-, y-, and z-axis [61]. The acquisition of gesture data involves activities performed using motor skills that require little conscious thought [62], which means that the data acquisition process is non-invasive and the user acceptance level is higher. However, gesture-based AF suffers from a lower accuracy performance compared to handwritten signature-based authentication, as it is more difficult to repeat the same gesture while holding a device in the air (without physical surface support) [59].

2.4 Single Factor vs Multi-factor Authentication Approaches

Single-Factor Authentication (SFA) approach refers to the use of any one AF for identity verification [63]. SFA, if implemented properly, can provide a reasonable level of security protection for resources and services. For example, setting a set of rules to enforce the use of a strong passcode reduces the risk of shoulder surfing attacks. While proper implementation of SFA can provide a reasonable level of

security protection, they offer no further protection when the (one and only) AF has been compromised (e.g. by more complex attacks). One way to increase the level of security protection of an authentication system is to increase the overall difficulty level of compromising the security protection of the authentication system. This can be achieved by using a Multi-Factor Authentication (MFA) approach to design authentication systems.

MFA refers to the use of more than one AF for identity verification. In a MFA approach, to gain access to the protected resources and services, all the AFs concern must be presented and verified successfully. This also means that when any one of the AF has been compromised, the resources and services are still protected by the other AFs. As discussed in the previous section, each AF has its unique strengths as well as weaknesses. By using an MFA approach, we combine the strengths and compensate the weaknesses of the AFs concern, thus strengthening the overall security protection level of the authentication system.

MFA is becoming the industry best practice for providing additional account security. World-leading IT companies such as Google (2-step Verification [64]), Facebook (Login Approvals [65]), Twitter (Login Verification [66]) and Dropbox (Two-step Verification [67]), have introduced MFA to add an extra layer of protection to their customers' accounts. These MFA systems are known as different names, but they essentially work in a similar way. To successfully gain access to such systems, a two-stage verification process is used. In the first stage, users are required to provide a valid pair of username and passcode (knowledge-based AF). In the second stage, users are required to provide a time-sensitive security code generated by a token (possession-based AF).

There are three issues to consider when implementing a MFA system: (i) cost of implementation, (ii) ease of integration with the existing IT ecosystem and security infrastructure, and (iii) user acceptability and learning curve. An ideal MFA system should increase the overall security protection level of the system, and, at the same time, address the three issues mentioned above.

2.5 Security Models

To determine the security models in a mobile environment, we have taken a three-step approach as follows. Firstly, the entities involved are identified, and

they are: (i) user, (ii) device or app, and (iii) cloud service. Then, trust boundaries are created to separate the entities. Figure 2.1 shows the trust boundaries between the identified entities in a mobile environment. Finally, based on the trust boundaries, two security models are determined: (i) User-to-Device/App, and (ii) Device/App-to-Service. Each security model is associated with different security threats. In this thesis, we focus on the threats imposed on the User-to-Device/App model. In the following section, we analyse and identify the related threats imposed on this model.

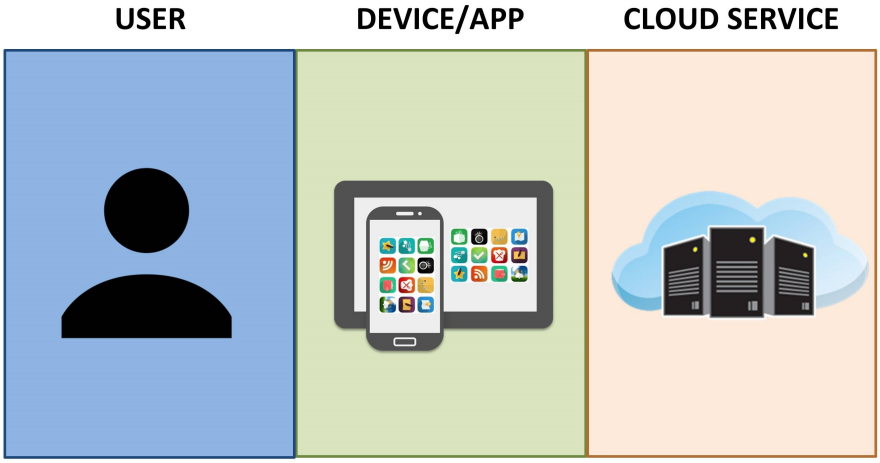

Figure 2.1: Trust boundaries between different entities in a mobile environment

2.6 Security Threat Analysis

Based on the User-to-Device/App security model above, this section identifies the threats or attacks imposed on this model.

Shoulder surfing attacks: A shoulder surfing attack refers to the use of visual observation methods, such as direct observation and video recording to capture a user's AF [68]. For example, an attacker captures the AF of a user by peeking or eavesdropping the user when the user is using the AF to perform an authentication task. Shoulder surfing attacks are most effective when performed in public. This is due to the fact that being near a person in public is common, allowing an attacker to perform such attacks less noticeably. Voice and knowledge-based AFs are vulnerable to this attack.

Social engineering attacks: A social engineering attack refers to the technique of disguising as a trusted entity to mislead a user into disclosing his/her AF [69]. For example, an attacker disguised as a technical support personnel to deceive a user into revealing personal information that can be used to infer or reproduce the user's AF. Face image, character and digit passcode that are not carefully chosen (e.g. the use of date of birth or social security number) are vulnerable to this attack.

Smudge attacks: A smudge attack refers to the technique used to reproduce an AF by observing and analysing the frequently touched area of a touchscreen mobile device [9]. For example, an attacker analyses the leftover smudges (oily residues) on a user's touchscreen mobile device to reproduce the user's passcode. Fingerprint and knowledge-based AFs are vulnerable to this attack.

Guessing attacks: A guessing attack refers to an attack in which an attacker uses repeated trial and error methods to obtain an AF [69]. Guessing attack can be conducted in three different ways: (i) pure guessing (effective against simple and/or frequently used passcodes), (ii) dictionary attack (effective against passcodes that consist of common words found in dictionary), and (iii) brute force attack (effective against passcodes with different character combinations). Knowledge-based AF are vulnerable to this attack.

Theft: This attack refers to the act of illegally gaining possession of an AF. For example, stealing a mobile device or memorising a passcode written on a paper placed on an open workspace. Token-based and knowledge-based AFs are vulnerable to this attack.

2.7 AF Requirements Specification

To design a secure and usable authentication system that counter the identified threats, the first step is to select the preferable AFs that we could use to design the system. The selection of AFs should be guided by a set of requirements. The requirements are divided into security and usability requirements. In identifying the requirements, we have taken inspiration from the highly cited and extensive survey conducted by [70]. In the survey, the authors proposed a comprehensive list of requirements used to comparatively evaluate different AFs used in the web

environment. Some of the requirements that are relevant to the mobile environment are adapted in our set of requirements. The following sections describe these requirements.

2.7.1 Security Requirements

This section specifies a set of security requirements that should be used in the selection of AFs. These requirements are chosen to counter the identified security threats discussed in Section 2.6. The requirements are as follows:

(S1) Shoulder Surfing Resistant: An AF that is resistant to shoulder surfing attacks can prevent attackers from using visual observation to capture the AF of a user.

(S2) Social Engineering Resistant: An AF that is resistant to social engineering attacks can prevent attackers from analysing the stolen personal information of a user to reproduce the AF of the user.

(S3) Smudge Resistant: An AF that is resistant to smudge attacks can prevent attackers from using the smudge traces left on a user's touchscreen mobile device to reproduce the AF of the user.

(S4) Guessing Resistant: An AF that is resistant to guessing attacks can prevent attackers from obtaining an AF through repeated trial and error methods.

(S5) Theft Resistant: An AF that is thief resistant can prevent an AF from being stolen or render a stolen AF unusable.

(S6) Privacy-Preservation: An AF that fulfils this requirement can ensure that, in the event when an AF is lost or stolen, the privacy risk inflicted on the user is minimal [71].

(S7) Infrequent False Acceptance: An AF that fulfils this requirement has a low likelihood of making a false positive authentication decision (i.e. incorrectly grant access to a illegitimate user who do not presents a valid AF).

(S8) Non-Repudiation: An AF that fulfils this requirement can ensure that a presented AF has been used by the user claiming to own the AF. In this way, the user who presents the AF cannot later deny having presented the AF.

(S9) Revocable: An AF that fulfils this requirement can ensure that, in the event when the AF has been compromised or lost, the compromised AF can be renewed or replaced.

2.7.2 Usability Requirements

This section specifies a set of usability requirements that should be used in the selection of AF. The requirements are as follows:

(U1) Easy-to-Learn: An AF that fulfils this requirement has an authentication procedure that is easy to learn and remember.

(U2) Efficient-to-Use: An AF that fulfils this requirement has a short authentication turnaround time, which includes the time taken for submitting an authentication request, processing of the request and receiving of the authentication decision.

(U3) Scalable-for-Users: An AF that fulfils this requirement allows the same AF to be used in more than one distinct authentication systems without violating security and usability requirements.

(U4) Infrequent False Rejection: An AF that fulfils this requirement has a low likelihood of making a false negative authentication decision (i.e. incorrectly deny access to a legitimate user who presents a valid AF).

(U5) Accessible: An AF that fulfils this requirement should facilitate people with disability on using the AF.

(U6) Negligible Cost: An AF that fulfils this requirement has a low or negligible deployment cost.

(U7) Seamless Integration: An AF that fulfils this requirement can integrate with existing system or device without requiring major changes or reconfigurations.

(U8) Mature: An AF that fulfils this requirement should have an established track record of successful deployment of the AF in a real-world setting (beyond a laboratory setting).

(U9) Non-Proprietary: An AF that fulfils this requirement is free from intellectual property protection. The implementation details of the AF are openly available (i.e. not protected by patents, trade secrets and copyright), and the deployment does not require the use of proprietary hardware or software.

2.8 Comparative Analysis of AFs

This section presents a comparative analysis of the different AFs (as discussed in Section 2.3) against the requirements specified above. To perform the analysis, we have introduced and used a scoring system. The scoring system provides a quantitative way of measuring how well the AFs fulfil the requirements. The scoring system uses a five-step approach as follows.

Step 1: The first AF is evaluated against the first requirement to determine the level of fulfilment of the requirement. There are three levels of fulfilment: (i) fully fulfil, (ii) partially fulfil, and (iii) do not fulfil.

Step 2: Depending on the level of fulfilment, a scoring point is allocated to the AF. The point allocation is based on the following rules:

- Fully fulfils the requirement, 2 points are allocated.
- Partially fulfils the requirement, 1 point is allocated.
- Do not fulfil the requirement, 0 points are allocated.

In some cases, a particular requirement (e.g. smudge resistant) just does not apply to a particular AF. For example, there is no leftover oily residues to analyse where the user do not need to touch the face against the touchscreen device. To make comparative analysis as simple as possible, we define the AF as fully fulfil the requirement, instead of introducing a "not applicable" value. This measure is taken in the basis that nothing can go wrong, for that AF, with respect to the corresponding threat.

Step 3: Step 2 is repeated for the remaining requirements.

Step 4: The points for each of the requirements are aggregated to form an overall score. The score indicates how well the AF concern fulfils the requirements. The

higher the score, the better the AF is at fulfilling the requirements, and thus the more desirable the AF.

Step 5: Steps 1 to 4 is repeated for the remaining AFs.

We are aware that the requirements are not all of equal weight. Some requirements are clearly more important and deserve more weight than others. The importance of any particular requirement depends strongly on the specific goal of a security system or an application scenario. For this reason, we used a more qualitative approach in our scoring system, i.e. the scoring point assigned to each AF are not a continuous value but a coarsely quantised value (i.e. fully, partially and do not fulfil). Nevertheless, we refer readers to [70] for a framework that can be adapted to extend our scoring system to use weights.

Table 2.1 gives a list of AFs used on mobile devices. For each AF, the table also gives their corresponding scoring points against each requirement and overall scores. The table offers a comprehensive overview and allows us to discover high level patterns. From the table, we can make two clear observations. Firstly, biometrics-based AFs score noticeably higher in terms of security requirements than knowledge-based AFs, indicating that the former offer a higher level of security protection than the latter. This observation is supported by the work reported in [72,73]. Secondly, character and digit passcodes are the only two AFs that fully fulfil all (except for one) the usability requirements, indicating that they are ideal for large-scale deployments. This observation is also supported by the work reported in [70].

Based on these observations, and taking into consideration that the level of security protection of a MFA system is higher than a SFA system, we have identified that touch dynamics biometrics and passcode as the two most desirable AFs that should be used in the design of a secure and usable authentication system for mobile devices. The justification for the selection of touch dynamics biometrics instead of the others biometrics-based AFs are further discussed in Section 3.2.2.

2.9 Chapter Summary

This chapter has presented an overview of authentication on mobile environment. Through a security model, the security threats are identified and analysed. Based on the analysis, we have specified a number of security and usability requirements

to guide the selection of AFs for the design of an authentication system that counters the identified threats. The chapter also presented a comparative analysis of the different types of AFs using the specified requirements. The analysis has led to the identification of two AFs, i.e. touch dynamics biometrics and passcode, as most desirable for the system.

The next chapter presents an investigative study of published work in the topic area of touch dynamics biometrics to identify the gaps in these work and propose a way forward to address the identified gaps.

AFs	(S1) Shoulder Surfing Resistant	(S2) Social Engineering Resistant	(S3) Smudge Resistant	(S4) Guessing Resistant	(S5) Theft Resistant	(S6) Privacy-Preservation	(S7) Infrequent False Acceptance	(S8) Non-Repudiation	(S9) Revocable	(U1) Easy-to-Learn	(U2) Efficient-to-Use	(U3) Scalable-for-Users	(U4) Infrequent False Rejection	(U5) Accessible	(U6) Negligible Cost	(U7) Seamless Integration	(U8) Mature	(U9) Non-Proprietary	Overall Scores
	Requirements																		
Face	●	◒	●	●	◒	○	●	●	◒	●	◒	●	◒	●	◒	◒	◒	◒	25
Touch Dynamics	◒	●	●	●	●	◒	◒	◒	●	●	◒	●	○	◒	●	◒	○	●	25
Fingerprint	●	●	◒	●	◒	○	●	●	◒	●	●	●	◒	◒	○	◒	◒	○	23
Iris	●	●	●	●	●	○	●	●	◒	●	◒	●	◒	◒	○	◒	○	○	23
Character Passcode	○	○	○	○	○	●	●	○	●	●	●	○	●	●	●	●	●	●	22
Digit Passcode	○	○	○	○	○	●	●	○	●	●	●	○	●	●	●	●	●	●	22
Voice	○	●	●	●	◒	◒	◒	◒	◒	●	◒	●	●	◒	◒	◒	◒	◒	22
Software Token	●	●	●	◒	◒	●	●	○	●	○	○	◒	●	○	◒	◒	◒	◒	21
Gait	◒	●	●	●	●	◒	◒	◒	◒	◒	◒	●	○	◒	◒	◒	○	◒	21
Graphical Passcode	○	◒	○	○	○	●	●	○	●	◒	◒	○	●	◒	●	●	●	●	20
Gesture	◒	●	●	◒	●	◒	◒	◒	●	◒	◒	●	○	◒	◒	◒	○	◒	20
Hardware Token	●	●	●	●	○	●	●	○	●	●	○	○	●	○	○	○	○	●	20

●:Fully fulfils (2 points); ◒:Partially fulfils (1 points); ○:Do not fulfil (no points)

Table 2.1: A comparative analysis of different AFs against different selection requirements

Chapter 3

An Investigative Study of Touch Dynamics Biometrics Authentication on Mobile Devices

3.1 Chapter Introduction

This chapter presents an investigative study of published work in the topic area of touch dynamics biometrics highlighting the contributions and technological advances in the topic area. It critically analyses them from a range of perspectives, leading to the identification of research gaps and outlines a possible way forward to address these gaps. The thorough investigative study of the existing work in the topic area of touch dynamics biometrics is our first novel contribution in this thesis.

In detail, Section 3.2 provides an overview of touch dynamics biometrics. Sections 3.3-3.7, respectively, compares the related work in terms of their experimental designs, data acquisition procedures, feature extraction methods, feature classification techniques and fusion approaches (the discussions in Section 3.2.4, 3.2.5, 3.3.1 and 3.3.2 are applicable to other biometrics AFs in general as well). Their performances are discussed in Section 3.8. Section 3.9 provides further discussion on the related work most relevant to ours. Section 3.10 discusses what is missing in the most related work. Section 3.11 outlines a way forward to address the missing bits. Finally, Section 3.12 summarises this chapter.

3.2 Touch Dynamics Biometrics

This section first provides an overview of touch dynamics biometrics, followed by its benefits and limitations. It then describes the primary operational processes of a touch dynamics authentication system and the evaluation criteria used to assess this system.

3.2.1 Overview

Touch dynamics biometrics refers to the digital signatures generated when a human interacts with a touchscreen mobile device. The signatures generated by different individuals are believed to be rich in distinctive properties, which hold potentials as personal identifiers [22]. The technological advancement in information and communication technology sector plays a vital role in the development of touch dynamics biometrics research. Figure 3.1 shows the timeline of touch dynamics biometrics research as influenced by technological developments in the sector.

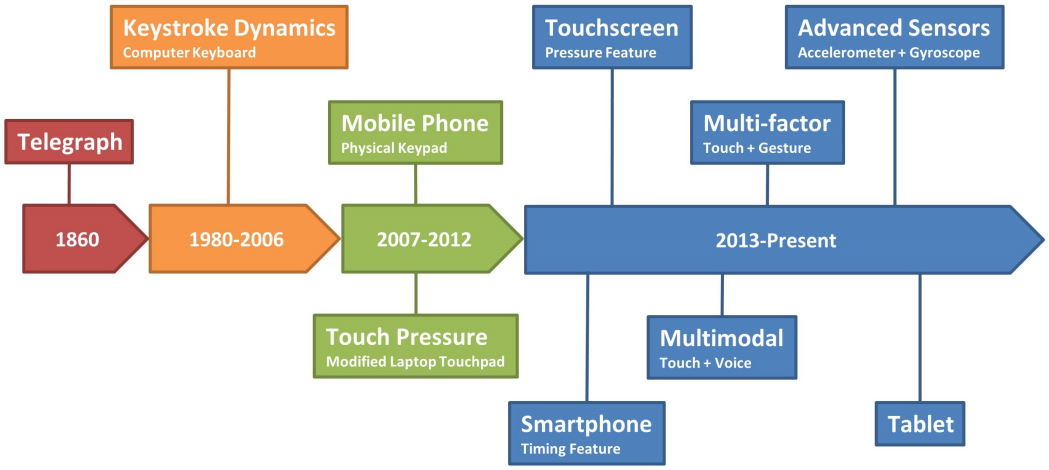

Figure 3.1: Evolution of touch dynamics biometrics research

Before mobile devices existed, there were some predecessors. In the 1860's, the primary method for long distance communication was the telegraph machines. Telegraph operators were able to 'identify' each other through ways in which they tapped on telegraph keys [74]. Today, telegraph keys have been replaced by computer keyboards, cell phone keypads and touchscreen virtual keyboards.

Before the emergence of touch dynamics, one of the earliest research work

on using keystroke dynamics (i.e. the patterns of interactions between human input and a computer keyboard) to identify users was by [75]. They attempted to recognise six professional secretaries by analysing the way they typed three passages of text consisting of 300 to 400 words each. Since then, keystroke dynamics either on workstations [76,77] or in a web-based environment [78,79] have been the major topic of research. Crawford [80], Karnan et al. [81] and Teh et al. [82] have, independently, written surveys of the published work on keystroke dynamics authentication. However, these early work largely focused on keystroke dynamics authentication on computer keyboards.

With the rapid development of mobile communication technologies, research efforts in the area have been focusing on mobile devices with physical keypads [21, 83]. Since 2007, there has been growing efforts on examining the possibility of applying the concept of keystroke dynamics to user authentication on mobile devices [82]. More recently, the research activities are largely carried out in the context of touchscreen mobile devices [84–87]. The availability of more advanced sensors in recently released mobile devices provides added opportunities to the development of touch dynamics biometrics, as these sensors allow the extraction of more feature data types.

3.2.2 Benefits

A touch dynamics authentication system can offer a number of benefits compared to the other types of biometrics-based authentication system. These benefits are as follows:

Transparency: A touch dynamics authentication system requires little or no additional interventions from a mobile device user. This is because the capturing and processing of touch dynamics data can be carried out in the background while the user is using the device. Users may not even be aware that they are being authenticated periodically or are being protected by an extra layer of authentication. This is in stark contrast to other biometrics-based authentication systems that usually require explicit alignment of a biometrics feature to a specific sensor. For example, in the case of iris-based authentication, a user needs to look straight into an intra-red camera to take an iris image, and in the case of fingerprint-based authentication, a user needs to put one of his/her fingers on the fingerprint sensor to scan a fingerprint pattern.

Familiarity: The touch dynamics data used for authentication is acquired during mobile users routine input activities. This is a process which mobile users are already familiar with, so the data acquisition process tends to have a gentler learning curve with a higher usability level.

Revocability: The touch dynamics data can be replaced should a passcode associated with a touch dynamics pattern is compromised, as a new touch dynamics template can be associated to a new passcode. This is not the case for other physiological biometrics, e.g. for iris or face biometrics, once they are compromised, there will be no replacement (unless cancelable biometrics are used [88]), and for fingerprints, the number of replacements is limited (there are only ten fingers to use after all).

Non-dependency: The data acquisition of touch dynamics is less sensitive to environmental factors. Therefore, it is more suited to, and can be more easily deployed in, a mobile device context. A mobile device is usually operated in an on-the-go manner, so the conditions of some environmental factors, such as the environmental condition and background noise level, are constantly changing. Other biometrics features such as iris or voice biometrics are sensitive to these environmental factors.

Cost Effectiveness: In contrast to other physiological-based biometrics authentication systems such as iris and fingerprint-based authentication recognitions, which typically require the use of some specialised hardware, a touch dynamics authentication system uses only built-in mobile device sensors. This can reduce device costs, and it is ideal for large-scale deployments.

3.2.3 Limitations

While a touch dynamics authentication system can offer a number of benefits, it also introduces a number of limitations. These limitations are as follows:

Lower Accuracy: The accuracy performance of touch dynamics authentication system is relatively lower in comparison to other physiological biometrics authentication system (e.g. fingerprint and iris).

Lower Permanency: Usually, human behavioural characteristics change more

frequently than physiological characteristics. Similarly, a user's touch dynamics pattern may change gradually, as the user is getting more familiar with the passcode, input method, device and other external factors.

Longer Training Time: A user's touch dynamics pattern can change over time and requires some time to become stable. Therefore, it may require more time to train an accurate authentication model.

Non-Disabled Friendly: A touch dynamics authentication system may not work effectively with users with certain disabilities (e.g. users who do not have hands, or are blind).

3.2.4 Operational Processes

Figure 3.2 shows a typical framework of a touch dynamics authentication system. From the figure, we can see that the operation of this system can largely be captured in two operational phases: user enrolment and user authentication. In the enrolment phase, a set of touch dynamics input samples (or training samples) are acquired from a user, processed and stored as a reference template. In the authentication phase, a presented touch dynamics input sample (or test sample) is compared against the stored reference template to determine whether the test sample is indeed originated from the owner of the device. The two operational phases are accomplished by a number of functional blocks, each of which performs a well-defined function or operation. These functional blocks and their respective operations are described below.

Data Acquisition: Data acquisition is an operation by which raw touch dynamics data are acquired. This is usually carried out as the first step and during the enrolment phase of a touch dynamics authentication system. The acquired raw data typically consist of repetitive input samples or collections of input samples over a specified period. These raw data will later be used for feature extraction. Section 3.4 provides further discussions on the data acquisition operation.

Feature Extraction: Feature extraction is another mandatory operation which is carried out in both the enrolment and authentication phases. The main task of this operation is to identify and extract distinctive touch dynamics features common to a user from the acquired raw data. These features will later be used

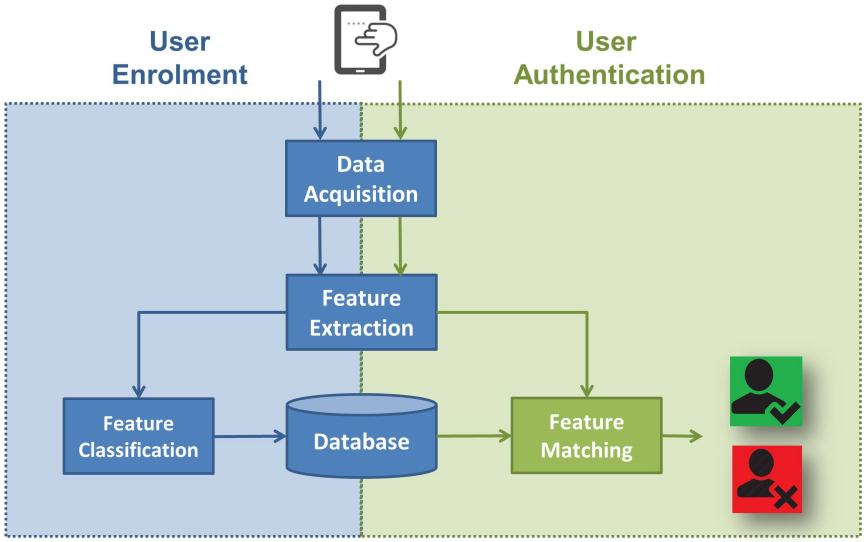

Figure 3.2: A touch dynamics authentication system framework

for feature classification. Section 3.5 provides further discussions on the feature extraction operation.

Feature Classification: This is the major operation for most biometrics systems, where the extracted features are used to create a reference template or model that uniquely represent a user's touch dynamics pattern. This unique model is stored in a database and will later be used for feature matching. Section 3.6 provides further discussions on the feature classification operation.

Feature Matching: This is an operation carried out by a touch dynamics authentication system to authenticate a user. It is to decide whether a test sample is indeed originated from the owner of the device. Before the final decision is made, a number of fusion approaches may be applied to increase the accuracy performance of the system (further discussions in Section 3.7).

3.2.5 Evaluation Criteria

To assess the suitability of a biometrics authentication system to real-world applications, two main criteria should be used to evaluate the system, system accuracy and system usability.

3.2.5.1 System Accuracy

To evaluate the accuracy performance of a biometrics authentication system, three metrics are commonly used: (i) False Rejection Rate (FRR), (ii) False Acceptance Rate (FAR), and (iii) Equal Error Rate (EER). The relationship among these metrics is shown in Figure 3.3 and their definitions are given below.

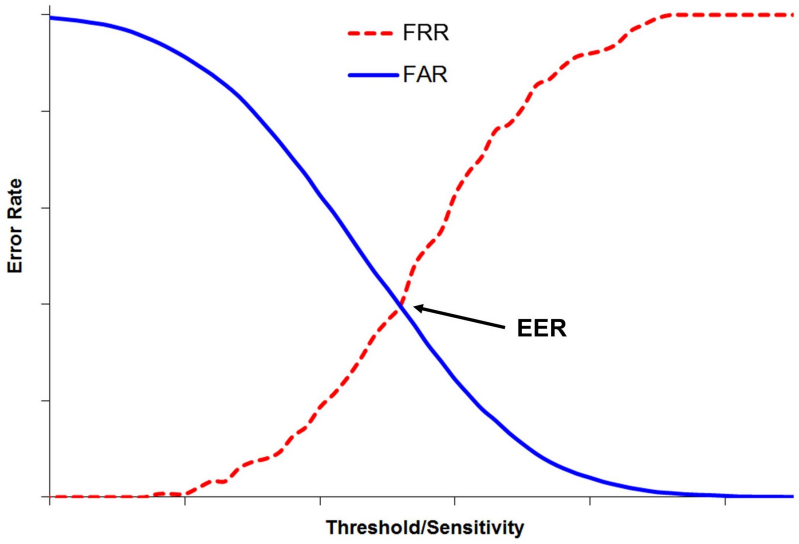

Figure 3.3: Relationship between the FRR, FAR and EER

False Rejection Rate (FRR): FRR is calculated as the ratio of the number of legitimate user trials that are falsely rejected and the total number of legitimate user trials. A lower FRR value indicates fewer legitimate user trails are falsely rejected, thus the higher the usability of the biometrics authentication system. FRR is also referred to as false negative rate [62], false alarm rate [89], false non-match rate [90] or Type II error [91].

False Acceptance Rate (FAR): FAR is calculated as the ratio of the number of illegitimate user trails that are falsely accepted and the total number of illegitimate user trials. Again, a lower FAR value indicates fewer illegitimate user trails being falsely accepted, thus the higher the security level of the biometrics authentication system. FAR is also referred to as false positive rate [62], impostor pass rate [89], false match rate [90] or Type I error [91].

Equal Error Rate (EER): EER is a single-number accuracy performance metric, which is commonly used to measure and compare the overall accuracy performance of different biometrics authentication system. It is calculated by averaging the FRR and FAR values under the condition that the absolute value of the difference between FRR and FAR is minimal [92]. In other words, EER is the interception point of two graphs, one for FRR and the other for FAR. Typically, the lower the FRR and FAR values, the lower the EER value, which in turn indicates a better accuracy performance of a biometrics authentication system. However, FRR and FAR are negatively correlated, so lowering both FRR and FAR values at the same time is not possible. Therefore, in real-life applications, FRR and FAR are usually adjusted and determined based on the usability and security requirements of the applications. In some literature [93–97], the term '*accuracy*', rather than EER, is used as an accuracy performance metric. This term is the inverse of EER. In other words, a higher accuracy value indicates a better accuracy performance of the biometric authentication system.

The accuracy performance can also be graphically visualised by using the Detection Error Trade-off (DET) curve or the Receiver Operating Characteristic (ROC) curve as shown in Figure 3.4 and Figure 3.5, respectively. To plot the DET curve of a biometric authentication system, a set of FRR and FAR values of each system is needed. The values are obtained by setting the threshold to different values. The curve is formed by plotting the FRR values on the y-axis and the FAR values on the x-axis. The optimal point of a DET curve is at the origin of the graph (0,0). In other words, the closer the curve to the bottom left corner, the better the accuracy performance of the system. Figure 3.4 shows a DET graph with three DET curves. Each of the curves represents the accuracy performance of a specific system.

The ROC curve is obtained by plotting Genuine Acceptance Rate (GAR) against FAR at different matching threshold values. GAR is the ratio between the correctly accepted legitimate user trials against the total number of legitimate user trials. It is also referred to as the inverse of FRR (100-FRR) or true positive rate [96]. The closer the curve to the top left corner of the graph (or the larger area under the curve), the better the accuracy performance of the biometric authentication system.

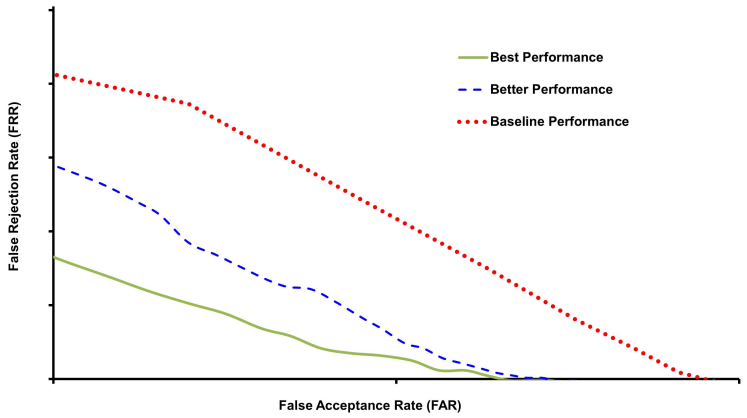

Figure 3.4: DET curves of three performance scenarios

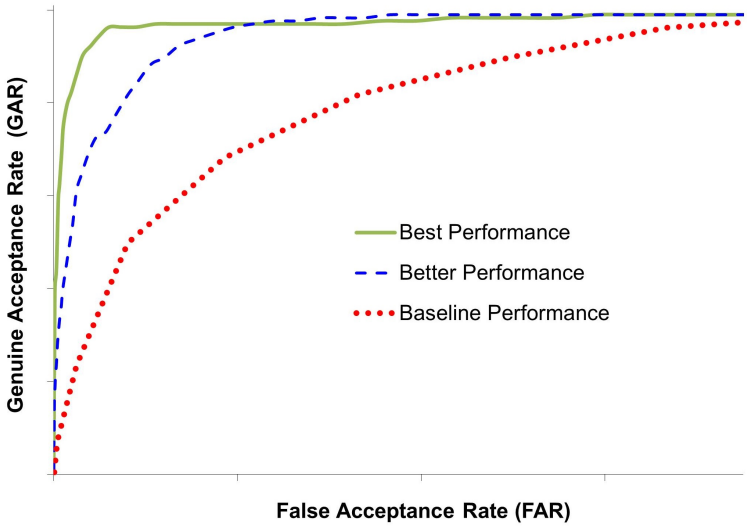

Figure 3.5: ROC curves of three performance scenarios

3.2.5.2 System Usability

The system usability (or user acceptance) of an authentication system is also an important criterion in the successful deployment of a new authentication system. System usability consists of two factors: (i) the ease of use offered by a biometrics authentication system; and (ii) the computational cost imposed by a biometrics authentication system. With regards to the first factor, a number of research efforts have investigated the adoption of authentication systems on mobile devices [98–101]. A common finding across the surveys is that users will eventually abandon or be reluctant to use any system that is tedious or slow to use, even if it can offer a higher level of security protection. Therefore, an authentication system should be convenient or easy to use. Satisfying the second factor is particularly important for computational resource-limited mobile devices. A complex authentication system may impose a higher level of computational overhead, increasing authentication delays and reducing system usability. Therefore, it is also important to design authentication systems that introduce as low computational overhead as possible.

3.3 Experimental Design

To conduct touch dynamics biometrics experiments, we need to acquire some touch dynamics data. The setup of an experiment carried out to acquire a touch dynamics dataset concerns a number of issues: (i) deployment modes, (ii) working modes, (iii) degree of control when carrying out the experiment, (iv) acquisition devices, (v) application development platform, (vi) subject size, and (vii) subject demography. They are discussed in detail in the following sections. Note that, hereafter, the discussions and statistics shown are based on our review of 69 papers accessed from reputable digital libraries, i.e. IEEE Xplore, ACM, ScienceDirect and SpringerLink.

3.3.1 Deployment Modes

A touch dynamics authentication system can be deployed in one of the two modes: identification or verification. These modes function uniquely and serve different purposes and application scenarios. The purpose served by the identification

mode is to classify or identify some unknown identity. It is used to answer questions such as "who is this person" or "is this person in the database". This mode is typically deployed for forensic investigations or intrusion detections. Its use on mobile devices is rather limited. The purpose served by verification mode, on the other hand, is to verify or prove a claimed identity. It is used to answer the question "is this person whom he/she claims to be". The authentication of a mobile device user fits into this mode.

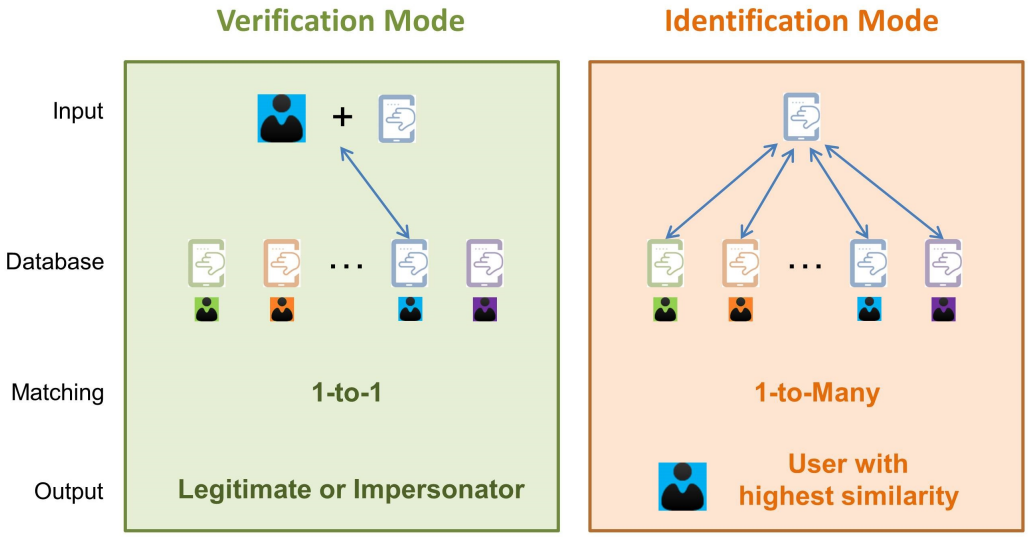

Figure 3.6: Deployment modes of a touch dynamics authentication system

As shown in Figure 3.6, the fundamental difference between the two modes is that, in the identification mode, the checking between the touch dynamics samples submitted by a user and the database is 1-to-Many, whereas, in the verification mode, this checking is 1-to-1. According to our literature survey, the number of papers published on the study of the verification mode (78%) is much higher than the identification mode (19%).

3.3.2 Working Modes

The verification deployment mode can operate in two working modes: static mode or dynamic mode. These modes are used in different application scenarios. In the following, we discuss the two working modes.

Static Mode: One application scenario of this mode is static authentication, also known as one-off authentication [90]. In static authentication, a user is verified

at the initial instance of, or at some predefined intervals during, a user-to-system interaction.

Dynamic Mode: One application scenario of this mode is dynamic authentication, also known as continuous authentication [102]. In dynamics authentication, a user may be verified at any instant of a user-to-system interaction or for every service access (i.e. continuously) throughout a service access session (in addition to the initial verification). In addition to countering unauthorised device sharing and device lost/theft, dynamic authentication is also useful in countering threats such as session hijacking.

The functions performed in both working modes are complementary, which means that they can be deployed independently, or alongside with each other to enhance the security protection level of mobile devices or the security of service access using mobile devices. The focus of this thesis is on investigating how to strengthen the authentication on mobile device at the initial instance of a user-to-system interaction, so hereafter our analyses focus on the static working mode of the verification deployment mode.

3.3.3 Degree of Control

The degree of control refers to restrictions or constraints imposed when carrying out an experiment. It covers three different aspects: (i) device selection, (ii) experimental setting control, and (iii) input string selection. Generally, the number of experiments that imposes restrictions outnumbered those that do not, and this is the case for all three aspects, as shown in Figure 3.7.

3.3.3.1 Device Selection

The devices used in experiments can be selected with one of the two approaches. One is to use a predetermined device and the other is to use a subject (hereafter, the term subject refers to a mobile device user recruited for an experiment) specific device. Based on our literature survey, a majority of the experiments published in literature (96%), with the exception of the work reported in [87,103,104], were undertaken by using the first approach. The primary reason for using a predetermined device, rather than the devices chosen by subjects, is to prevent any inconsistencies in the touch dynamics data acquired. For example, the availability

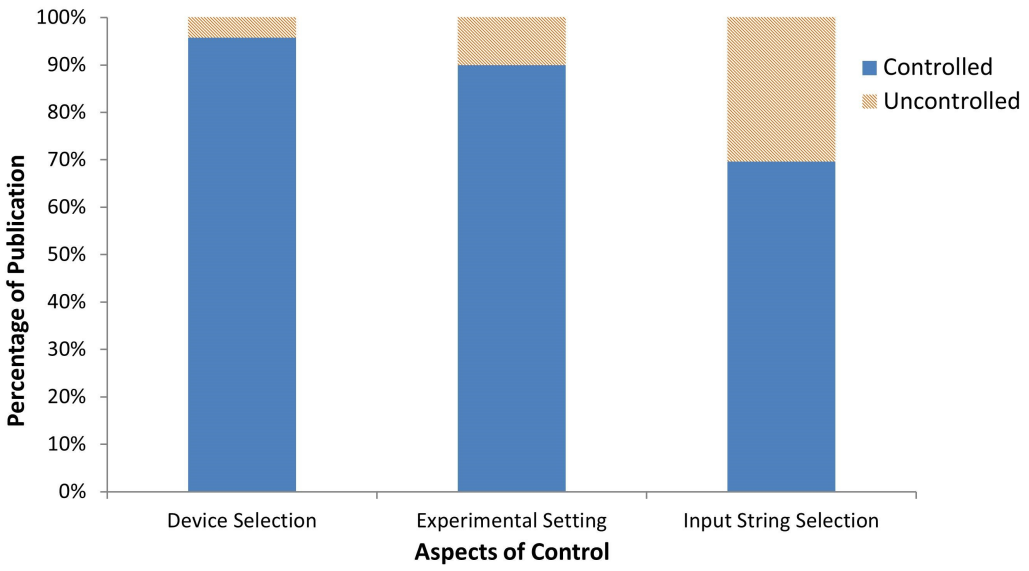

Figure 3.7: Distribution of publications on reporting the degree of control of different aspects imposed in experiments

of device sensors and the variations in sensor resolution or sensitivity between different devices may cause inconsistencies in the data acquired [105]. Additionally, as subjects are usually more familiar with their own devices, allowing subjects to use their own devices to acquire data may introduce bias in experimental results. This can be avoided by requiring subjects to use a predetermined device for the entire data acquisition operation [106].

3.3.3.2 Experimental Setting Control

The experimental setting control refers to the degree of restrictions that are imposed on an experiment during a data acquisition operation. Data acquisition operation may be conducted under two settings: (i) supervised with a stringent protocol (e.g. under a controlled laboratory environment), or (ii) unmonitored without imposing any restrictions (e.g. while the subjects carry out their activities as usual). The papers by [12, 87, 107, 108] reported experiments that were carried out under the unmonitored setting. A majority of experiments (90%) reported in literature were carried out under the supervised setting. The primary reason to control an experiment is to reduce the level of variations in touch dynamics data which may be caused by external factors such as device functionalities, cognitive load, distractions and sickness, etc. Controlling an experiment with a

stringent protocol can prevent these external factors from inflicting noise to the data acquired [109], allowing primary experimental variables (i.e. the discriminative capabilities of touch dynamics features and the classifiers) be evaluated more accurately [110]. Also, the unmonitored setting may risk the data acquired being distorted or tampered with, resulting in the reduction of data quality and inaccurate experimental results.

There are also views that touch dynamics data should be acquired in a natural manner without imposing any restrictions, or in an environment that can resemble a real-life device usage scenario [111]. The reason given by the authors was that a subject's interactivity with his/her device might differ in different circumstances, e.g. when performing a task at hand or doing a job in a controlled environment. The authors of [112] suggested that experimental results obtained under a controlled environment are over optimistic compared to those acquired in uncontrolled environments.

3.3.3.3 Input String Selection

How to select the input strings or which input strings should be used during experiments is another factor one should consider when carrying out touch dynamics related experiments. In the majority of the experiments reported in literature (70%), the subjects were asked to provide a predetermined set of input strings during a data acquisition operation. In other words, the input strings used in these experiments are identical across all the subjects. This means that the samples acquired from different subjects in the same dataset can be reused for evaluation purposes (not just for model creation), and as a result, a larger number of test samples can be acquired without acquiring them separately. However, in certain experiments, this approach is not practical. For example, the data acquisition experiments conducted by [113–115] were aimed at acquiring all the touch dynamics data over an entire interaction session with a device. Due to the nature and objective of these experiments, predetermining a set of input strings would not be practical.

3.3.4 Acquisition Devices

To acquire touch dynamics data, we need to use some data acquisition devices. Different devices may be equipped with different sensors that may have the ability to acquire different types of features. For example, a conventional mobile device with a physical keypad is only able to acquire timing feature. In contrast, a recently manufactured touchscreen mobile device is more likely to have multiple more powerful built-in sensors that can acquire more features (such as pressure and motion). So, acquisition device selection is an important experimental variable that should be considered in touch dynamics biometrics research. As the focus of this thesis is on touch dynamics biometrics, hereafter, we shall only scope our discussions on mobile devices with touchscreens. For existing work on keystroke dynamics on mobile devices with physical keypads, readers are referred to review articles such as [80, 82].

Based on our literature study, researchers mainly use smartphones (86%), rather than digital tablets (14%), as their data acquisition devices. There are two reasons why smartphones are given the priority. Firstly, a larger population of mobile device users use smartphones rather than digital tablets [116]. Secondly, smartphones are cheaper than tablets. Researchers have been using more recent and more powerful mobile devices to carry out their research work. Modern devices usually come with higher precision and resolution sensors that can capture higher quality features. Modern devices also have greater computational capabilities and resources, which can better support the use of more complex algorithms and more able sensors. Another device selection criterion is the intended application development platform associated with a mobile operating system, and this is discussed in the following section.

3.3.5 Application Development Platform

To acquire touch dynamics data, we need to use a data acquisition application (or app), and, for the development of the app, we need to choose an application development platform. Based on our literature study, Android is the most popular development platform used to develop data acquisition app, which is followed by iOS and then by Windows. When selecting a development platform, these four factors should be considered, i.e. its customisability, flexibility, cost and market shares. These four factors are further discussed in the following.

Customisability: To acquire touch dynamics data, the ability to log various touchscreen input events (or touch event) is essential. However, the native input methods, such as a virtual keyboard or a virtual numeric keypad, used by a mobile operating system do not provide any method to acquire these data [117]. This is typically part of the security measure used to protect against the easy implementation of spyware or touch-logger apps [118]. Therefore, to acquire touch dynamics data, researchers have to create their own custom input methods with the necessary functionalities. Unlike its competitors (iOS and Windows), Android has made this easier by providing open source library functions, which allows researchers to modify the app framework [113], giving them greater flexibility in their app design and customisation.

Flexibility: Flexibility in terms of cross-platform development, sideloading and file system accessibility is among the criteria that contribute to the popularity of a chosen development platform. Android supports cross-platform development, and this means that developers have the flexibility to develop apps using any operating systems and to make use of existing resources in their developments. Also, both Android and Windows allow sideloading, which means that an app can be directly installed on a device without first publishing it to the app store or marketplace. App publishing involves publishing fee, rigid procedures and could be time-consuming. Therefore, sideloading can reduce the cost, time and effort on app development and testing. Furthermore, native file explorers have been provided by both Android and Windows providers. This means that data and system files can be accessed directly without additional configurations or installations of any third party apps. This provides a convenient way for researchers to transfer acquired data files between different devices for further analysis.

Cost: The cost required for acquiring a development tool and device should also be taken into account when making the selection. The official integrated development environments (IDEs) required for app developments for Android and Windows devices are Android Studio and Visual Studio, respectively. They are both available for download free of charge. This is not the case for the Xcode IDE for iOS devices. Until recently (Xcode version 7 and later), the IDE is only available once an annual fee has been paid for [119], and the cost of the fee is shown in Table 3.1. In addition, Android has been used by a wide range of mobile devices. Among the largest manufacturers of Android-powered devices

are Samsung, Huawei, Xiaomi and LG [120]. This wider range of devices can provide researchers with a cheaper option to conduct their experiments.

Market Share: Selecting a development platform with a larger market share (used by more people) allows greater accessibility to users. To date, Android devices have the strongest end-user demand worldwide, followed by iOS and Windows [121]. The similar trend is reflected in the selection of development platforms by researchers, as shown in Figure 3.8 (N/A in the figure indicates unspecified).

Properties	**Android**	**iOS**	**Windows**
Development Tool	Free	99/annual [122]	Free
Registration Fee (USD)	25 one-off [123]		19 one-off [124]
Programming Language	Java	Swift	C#/Visual Basic
Open Source	Yes	No	No
Cross-Platform Development	Yes	No	No
Sideloading	Yes	No	Yes
File System Accessibility	Yes	No	Yes
Market Share	High	High	Low

Table 3.1: Properties of different application development platforms

3.3.6 Subject Size

The subject size refers to the number of subjects from whom data are acquired. The subject size used in an experiment is known to have an impact on the obtained results of an experiment [125, 126]. Using a larger subject size can provide more data to verify the scalability of a study [127], and to reflect the true accuracy performance of a biometrics authentication system when deployed in real-world [128].

Typically a subject size of greater than 100 subjects is regarded as a very large subject size [82]. This number is supported by the work conducted by [129]. In this work, the authors conducted a statistical power analysis to determine the optimal subject size for an experiment. The result of the analysis shows that

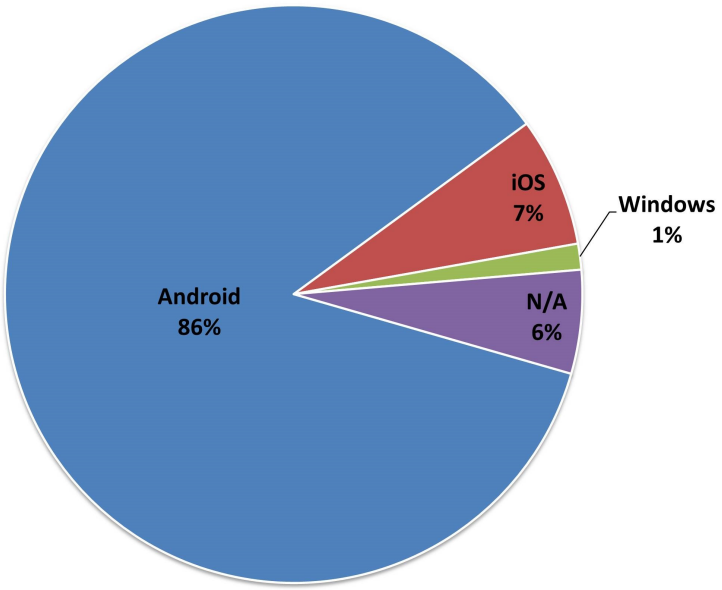

Figure 3.8: Distribution of application development platforms used in touch dynamics biometrics research

a subject size of 96 is required to detect the effect of an experiment at the 0.05 level of significance. Most experiments recruited less than 50 subjects [85,86,118], with some less than five subjects [130, 131]. Only a handful of published work [106,132,133] use a subject size greater than 100 subjects. Figure 3.9 summarises the subject sizes used in the experiments published in literature (N/A in the figure indicates unspecified).

In most of the experiments (93%), subjects were recruited on a voluntary basis, i.e. without receiving any monetary benefits. Only in a few experiments, subjects were awarded cash [86], vouchers [134] or some form of prizes [104]. The awards or prizes were used to motivate the subjects to take part in the experiments, increasing the participation rates. A data acquisition operation could be a resource-intensive process that requires dedication and effort from the subjects. To increase the participation rate or the number of subjects taking part in an experiment, the data acquisition tools can be distributed via a mobile app store [135] or via crowdsourcing platform [136]. This may be a possible way of recruiting a larger number of subjects, but, by doing so, control over the data acquisition operation will certainly become limited, and the risk of data being tampered with becomes higher.

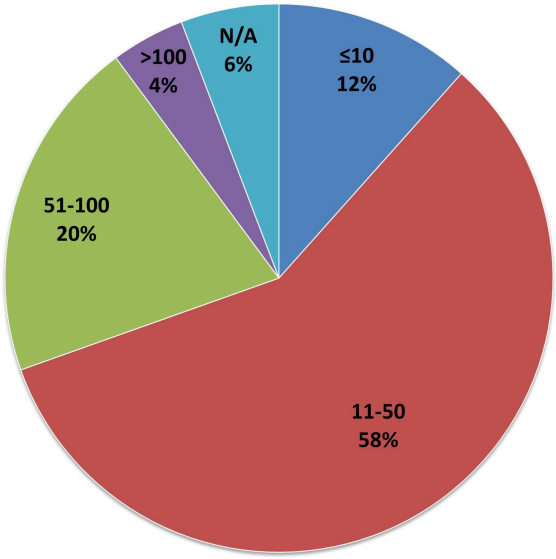

Figure 3.9: Distribution of subject sizes used in touch dynamics biometrics research

3.3.7 Subject Demography

Experimental subjects are typically selected based on some criteria. The commonly used criteria are age distribution, population mixture and profession diversity. Subjects from different age groups, with different backgrounds and/or different professions tend to use their devices at different frequencies. Therefore, if subjects are not selected properly, there may be unintentional biases in the experimental results.

Most of the published works are based on subjects that are selected from: (i) a narrow age distribution (i.e. 19-26) [85, 118, 137], (ii) confined to only people within the same organisation (i.e. within a research institute) [86, 103, 138], or (iii) restricted to limited profession (i.e. students) [113, 114, 139]. However, it may be argued that the data acquired from this special group of population may not realistically represent the wider community. The papers [135, 140–142] are among the few pieces of work we were able to find in literature, which recruited subjects from wide age groups, and diverse population groups and professions. Their goal is to diversify the dataset's subject demography so that the obtained results of the experiments can better resemble real-world scenarios.

3.4 Data Acquisition

Data acquisition operation is usually the first step in a touch dynamics biometrics research. This operation is used for acquiring subject's input samples and for extracting raw touch dynamics data from the samples. The design of this operation involves a number of issues, including determining the input string type, input sample size, number of acquisition session and session interval, and how to acquire the legitimate and illegitimate subject samples. In the following sections, we first discuss each of these issues in detail and then describe the public dataset available for touch dynamics biometrics research.

3.4.1 Input String Type

The input string type is an important experimental variable in a data acquisition operation, as the touch dynamics features used for analysis are extracted from a subject's input string. Generally, subjects are required to provide some input string during a data acquisition operation, and the input string can broadly be categorised into two types: non-keyboard-based and keyboard-based input. Non-keyboard-based input string includes a random acquisition of some continuous touch actions [114, 143], a random multi-touch gesture input [144] or touch input interactions with a common user interface element (e.g. buttons, checkboxes and sliders) [84, 135], made over a period of time. For keyboard-based input string, there are further two types, character-based (i.e. alphabetic, special character or alphanumeric string) and digit-based (i.e. only numerical string). The majority of touch dynamics experiments published required a subject to provide a keyboard-based input string as shown in Figure 3.10. As the focus of this thesis is on keyboard-based input string, hereafter we shall focus our discussions on this input string type.

A character-based string can be further categorised into short and long character strings. A short character string usually consists of a username or a password [47, 145], a random character combination [102, 137] or a set of dictionary words [85, 138]. A long character string is usually a segment of texts [117, 133] or several paragraphs of texts [22, 146]. Likewise, a digit-based string can also be classified into short and long digit strings. A short digit string is typically 4 to 6 digits long. It usually resembles a mobile device unlock code or an ATM

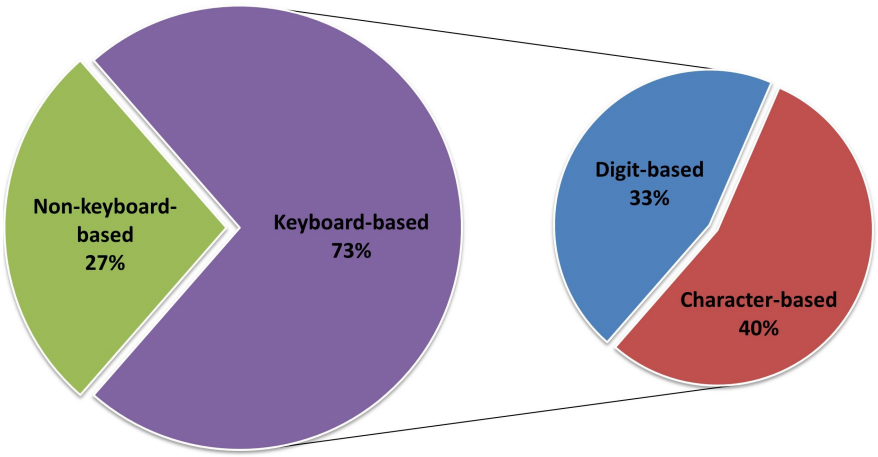

Figure 3.10: Distribution of input string types used in touch dynamics biometrics research

PIN number [147, 148]. The length of a long digit string, on the other hand, usually exceeds ten digits, similar to, e.g. a social security number [104] or a phone number [149, 150].

3.4.2 Input Sample Size

The input sample size affects the accuracy, robustness and outcome of an experiment [151, 152]. A larger number of samples used would allow us to gain a better representation of a subject's touch dynamics pattern, and, as a result, achieve a higher level of accuracy [153].

There are two ways of acquiring multiple samples in one data acquisition session. One is to acquire a fixed input string repeatedly (usually for keyboard-based input string), and the other is to continuously acquire touch events over a fixed period of time (usually for non-keyboard-based input string). According to the literature, the benchmark for the number of samples collected per subject per session is somewhere 10 to 20 repetitions for keyboard-based input string [85, 118, 132, 152]and 5 to 10 minutes for non-keyboard-based input string [113, 126].

Requesting a large number of samples from a subject in one data acquisition session is impractical. This is because subjects may not be available for a long stretch of time, or may feel uncomfortable with a lengthy acquisition session [153]. Therefore, selecting an optimal input sample size for each data acquisition session is necessary. Alternatively, instead of acquiring a large number of samples in

one lengthy acquisition session, we can carry out the data acquisition operation in multiple shorter sessions spread over a period of time. This approach can reduce the level of discomfort imposed on a subject and also better capture any inter-session variability on the data acquired (further discussions in the following section).

3.4.3 Acquisition Session and Session Interval

As mentioned above, a data acquisition operation can be carried out and completed either within a single session or spread over several sessions separated by a predefined interval. In the majority of the reported experiments (60%), data from each subject were acquired within a single session.

Though the first approach is relatively cheaper, it is not always practical to request a large number of samples from each subject in a single session. More importantly, touch dynamics biometrics, like other behavioural biometrics (i.e. voice, gait and signature), is not stable over time [126]. This implies that there may be some variations between different input samples in different sessions, even if the input samples are provided by the same subject. These variations are also known as the inter-session variability [154]. If samples are acquired in a single session, the inter-session variability may not be captured properly. This is evident in the experimental work reported by [85], where the accuracy performance evaluated using data acquired in a single session is two times better than those obtained in different sessions. This means that experimental evaluation using data acquired in single sessions may be over-optimistic and may not reflect the true accuracy of a biometrics authentication system when deployed in real-world.

Ideally, a data acquisition operation should be divided into multiple sessions separated by some intervals. In this way, inter-session variability can better be captured. This approach has been adopted in several experiments reported in literature, and, in these experiments, the selected intervals separating different sessions vary from minutes [104], hours [86], days [87,106,126] to weeks [150,153]. It is worth noting that careful considerations should be given when determining the lengths of intervals separating different sessions because this factor may influence the subject participation rate of an experiment. For example, a data acquisition operation that spans across a longer period of time is more likely to receive a lower subject participation rate (due to subjects' availabilities) or may result in a higher subject dropout rate (due to a greater commitment required of

the subjects) [155]. Therefore, there is a balance between better capturing subjects' inter-session variability and preserving the subject participation rate and commitment to the data acquisition operation.

3.4.4 Legitimate and Illegitimate Subject Sample

The degree of accuracy of a touch dynamics authentication system is measured by using the accuracy performance metrics values discussed in Section 3.2.5.1. To compute these values, two categories of samples are required, legitimate and illegitimate samples. Acquiring the legitimate samples is a straightforward process that has been described above. However, for acquiring the illegitimate samples, there are three approaches. The first approach (denote as A1) is to partition the already acquired subjects' samples into two subsets, one used as the legitimate samples, and the other (legitimate samples) acting/used as the illegitimate samples. The second approach (denote as A2) is to acquire additional samples from a subset of the subjects involved in a dataset, and use these additional samples as the illegitimate samples. The third approach (denote as A3) is to recruit additional subjects to provide the illegitimate samples.

Based on our literature research, A1 is most frequently adopted (87%) in comparison with A2 and A3. For example, in these experiments [132,138,143,151], A1 were used to acquire the illegitimate samples. In these experiments, t out of i samples acquired were used as the illegitimate samples, with the rest used as the legitimate ones. The subset of illegitimate samples can be selected in a randomised [132, 151] or a predefined order [138, 143]. With this approach, an equal number of the legitimate and illegitimate samples can be obtained with minimal or no additional resources. However, this approach is not practical when the input string is not the same across all subjects, as it is not possible to compare the touch dynamics patterns of two input strings when they are different.

The experiments that do not use the same input string across all the subjects can use A2 to acquire the illegitimate samples. For example, the authors of [153] recruited 100 subjects to provide the legitimate samples. Then, ten subjects were randomly selected from the 100 subjects and were given the PINs of all the other subjects. Each of these ten subjects was given the additional task of providing five impersonated samples of each of the other 99 subjects. With this approach, the ten selected subjects need to devote more time and effort to the experiment, and this may discourage voluntary participation. For this reason, the number of

the illegitimate samples acquired is usually smaller than that acquired using A1.

A3 is to recruit a separate pool of subjects specifically for providing the illegitimate samples. Take the published works of [133,156,157] for example, instead of requesting the subjects recruited for providing the legitimate samples, the research team recruited additional subjects for providing the illegitimate samples. In this way, it is more likely to obtain a balanced number of the legitimate and illegitimate samples than using A2. It is worth mentioning that, with this approach, for each increment of the subject size in a dataset, two subjects should be recruited. This means that the resource and effort needed to obtain a dataset is doubled.

3.4.5 Public Dataset

The availability of a public dataset for touch dynamics biometrics research is vital. For example, with such a dataset, we could do comparisons of different algorithms on the same dataset and/or different experimental settings. The availability of a public dataset also allows researchers to focus on more challenging research issues, spending less time on data acquisitions. However, the availability of open dataset (with a large subject size) in the domain of touch dynamics biometrics is still limited. This may be due to that the creation or acquisition of such data is a time and resource consuming process. At the time of this writing, we are only able to find four public datasets in relation to touch dynamics. These datasets are summarised in Table 3.2 and described in the following.

Dataset 1: The data acquisition experiment conducted by [140] involved 51 subjects. The input is a fixed character-based string "rhu.university" entered on a window touchscreen smartphone (Nokia Lumia 920). Each subject attended three sessions with an interval of five days apart for each session. The first session was used as a practice session, so the actual data acquisition started from the second session. A total of 15 samples were acquired from each subject over the second and third sessions.

Dataset 2: Another dataset, published by [153], was based on digit-based strings. The device used for acquiring the data was an early generation smartphone running on Android 2.0.1 (Éclair) Application Programming Interface (API) level 6, which was released in December 2009. The subjects were only required to provide two samples per session, and five sessions were used with an interval of

at least one week apart for each session to capture any inter-session variability. Input strings were not predefined; subjects were allowed to freely choose a desired digit-based string, and most of the chosen strings are 4 to 8 digits long. The subject size used in this dataset is 100 subjects, the largest of the four public datasets. The age distribution of the subjects involved is biased towards young people, where 85% of the subjects have the age of 25 or younger. Different from the usual practice, illegitimate samples were collected separately from those subject who provided the legitimate samples. Ten subjects are randomly chosen to act as impersonators. These impersonators were given the PINs of every other subject and were asked to impersonate the subjects by providing five samples for each subject. Note that, at the time of writing, the dataset is no longer available for download.

Dataset 3: Another related effort on acquiring and sharing their datasets publically was made by [137]. The number of subjects involved is the smallest among the four public datasets. The entire subject population recruited in this work were students. The data acquisition in this experiment was done via the use of two types of devices. 37 subjects provided their inputs on a Nexus 7 phablet (a hybrid of both a phone and a tablet with a screen size between 5-7 inches [158]), while the remaining five via the use of a LG Optimus L7 P700 smartphone. The paper did not explain why two different device types were used and whether the use of different device types would have any performance implications. The input string used for the data acquisition was predefined as ".tie5Roanl". Additionally, the touch actions captured include not only the input string but also *shift key* (toggle between lower and uppercase characters) and *keyboard switch key* (toggle between characters and numerical keys). These secondary key events may capture valuable and distinctive information about a subject touch dynamics pattern. Also, in this dataset, most of the subjects provided their input strings 30 times each on two separated sessions in two weeks (the duration between the two sessions was not mentioned in the paper). As some invalid inputs were removed, so the resulting dataset only contains 51 input samples per subject (instead of 60 from both sessions).

Dataset 4: Another dataset based on character strings were published by [159]. Unlike their previous dataset [137], this dataset is different in four ways. Firstly,

three character-based input strings were used for the data acquisition: "kicsikuty-atarka", ".tie5Roanl", and "Kktsf2!2014". Secondly, only one type of device, a Nexus 7 phablet, was used for acquiring the data. Thirdly, more subjects were recruited in this dataset. Lastly, for each input string, the subjects provided at least 20 input samples per session, and three sessions were used with an inter-session interval of at least one week.

Properties	**Dataset 1**	**Dataset 2**	**Dataset 3**	**Dataset 4**
Input String Type	C	D	C	C
Subject Size	51	100	42	54
Acquisition Sessions	3	5	2	$\geq$3
Inter-Session Interval	3-30 Days	1 Week	-	1 Week
Input Sample Size per Session	5	2	30	$\geq$20
Total Session Duration per Subject	-	5 Weeks	2 Weeks	3 Weeks
Total Sample Size per Subject	15	10	51	$\geq$60
Illegitimate Sample Acquisition Approach	A1	A2	A1	A1

C: Character-based; D:Digit-based

Table 3.2: Four public touch dynamics biometrics datasets

3.5 Feature Extraction

Human touch dynamics pattern contains unique features that can be used to distinguish one another. In the feature extraction phase, these features are extracted by processing the raw touch dynamics data acquired from a subject. Common features discussed in literature can broadly be classified into three categories: (i) timing, (ii) spatial, and (iii) motion. Figure 3.11 summarises the research efforts made on these features. Note that this figure gives a general idea of the usage distribution of each feature against all the papers we have reviewed, there are

papers that use more than one feature. These features are further described in the following sections.

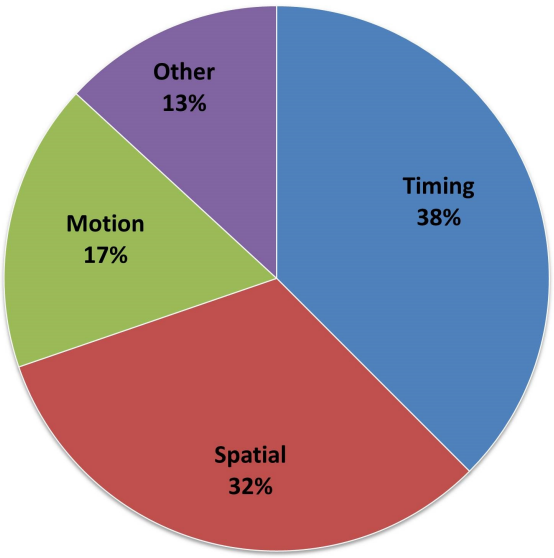

Figure 3.11: Distribution of touch dynamics feature types used in touch dynamics biometrics research

3.5.1 Timing Feature

The timing feature is the most widely used feature in touch dynamics biometrics research. A touch action (finger touching down or lifting up) on a virtual keyboard generates digital interrupts that can be detected by the mobile operating system API functions [118]. Each of these actions can be coupled with a timestamp value. Based on these timestamp values, two types of timing feature can be extracted: dwell time and flight time. Dwell time refers to the duration between the touch actions of the same key. It is also known as press [22] or hold time [147] in literature. Flight time refers to the interval between the touch actions of two or more keys. It is also known as latency [145] or interval time [139].

3.5.2 Spatial Feature

The spatial feature is a characteristic associated with physical interactions between a fingertip and a device touchscreen surface, and it can be acquired when a touch action is performed. Three most commonly reported spatial features in

literature are the touch pressure size, touch pressure intensity and touch position. A visual illustration of these features is reported in [143].

Touch Pressure Size: The touch pressure size represents an approximation of the screen area being touched in a touch action. Each touch action is associated with a touch pressure size value. The touch pressure size value captured from a subject is determined by the subject's fingertip size. For example, the authors of [160] observed that an adult male subject usually produces a larger touch pressure size value than a child or an adult female subject.

Touch Pressure Intensity: The touch pressure intensity is another feature that is often used along with the touch pressure size. Each touch action is associated with a touch pressure intensity value. A touch pressure intensity value measures the approximated force asserted on the screen upon each touch event. A touch pressure intensity value is linked to a subject's finger muscle that is unique to each subject.

Touch Position: The touch position is a two-dimensional matrix feature that captures a fingertip landing position on a device screen (or key). Each touch action can be associated with an x- and y-coordinate measured in pixel units [135]. The touch position of a key varies with a subject's fingertip size and cognitive preference. Also, by some mathematical manipulations, additional features can be derived. These include the distance [86], speed [118] or angle [106] between two touch actions. However, there is a concern with this coordinate representation of touch position values [111], that is, the coordinate system of a screen is device dependent. Using different devices, the captured touch position values are not consistent [149].

3.5.3 Motion Feature

The motion feature is the small amount of movement and/or rotation inflicted to the device during a touch action. These motions can be captured by two hardware sensors embedded in modern mobile devices: accelerometer and gyroscope. Figure 3.12 shows a graphical representation of the different motions captured by both sensors.

The accelerometer sensor measures the linear movement rate applied to a device over time. It is designed to detect the movement along the x-, y- and

z-axis in both positive and negative directions. These three values are measured in the unit of meters per second squared (m/s^2) [12]. On the other hand, the gyroscope sensor measures the rotation rate applied to a device against the three axes: (i) tilt forward and backward (Pitch), (ii) twist from side to side (Roll), and (iii) turn from portrait to landscape (Yaw). These values are measured in the unit of radian per second (rad/s) [138]. Note that the availability of these sensors and the variations in sensor resolution or sensitivity between different devices may cause inconsistencies in the data acquired may prevent these sensors from being used effectively for user authentication purposes [105].

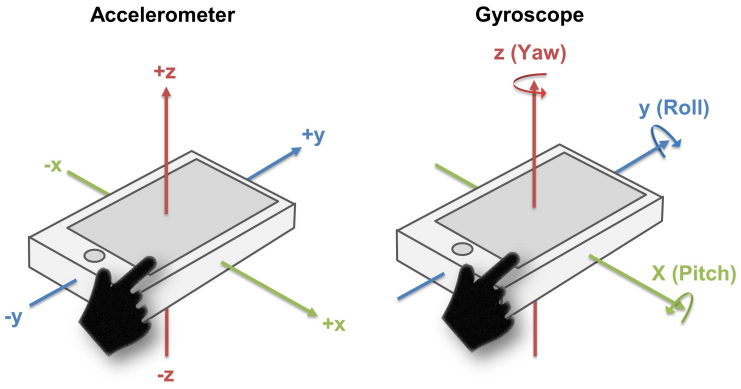

Figure 3.12: Raw motion data captured by two motion sensors

3.6 Feature Classification

Feature classification operation is the major operation for a touch dynamics authentication system. This operation is carried out to transform the extracted features into reference template or model that uniquely represent each user's touch dynamics pattern. Feature classification is usually carried out using machine learning techniques. A number of such techniques have been used in touch dynamics biometrics research reported in literature: (i) Probabilistic Modelling (PM), (ii) Cluster Analysis (CA), (iii) Decision Tree (DTR), (iv) Support Vector Machine (SVM), (v) Neural Network (NN), (vi) Distance Measure (DM), and (vii) Statistical (ST). Figure 3.13 summarises the machine learning techniques used in touch dynamics biometrics research. Note that this figure gives a general idea of the usage distribution of each technique against all the papers we have

reviewed, there are papers that use more than one technique. We describe each technique in the following.

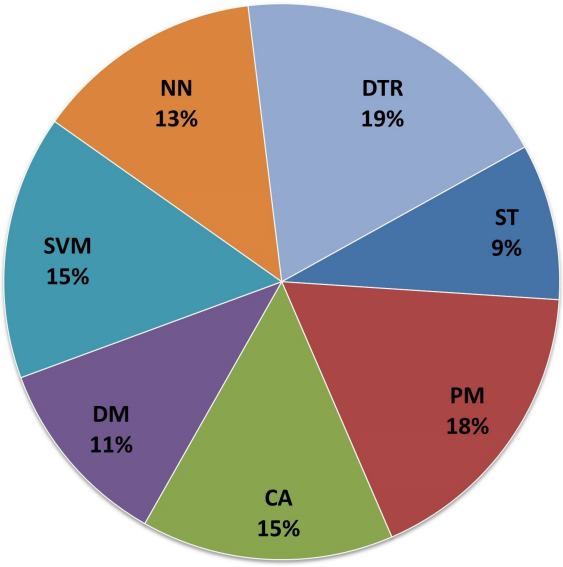

Figure 3.13: Distribution of machine learning techniques used in touch dynamics biometrics research

Probabilistic Modelling (PM): The main idea behind the PM technique is to predict the likelihood of a given test sample belonging to a particular subject using the prior probability calculated from training samples. One widely used PM technique is the Bayesian Network [22, 161]. It uses an acyclic graph model to find the probabilistic relationship between the parent and child node [137]. For example, feature data from a reference template will be used as the parent node and the associated subject identity as a child node. Then, given a test sample, the intended child node is determined by the probability of the parent node [162]. Other variants of the PM technique include the Naive Bayes [56, 148] and the Gaussian Probability Density Function [47].

Cluster Analysis (CA): The CA technique assumes that samples belonging to the same subject have similar properties [113]. The aim is to group sample with similar properties to form a homogeneous cluster. Then, the class label of a test sample is decided by the degree of proximity towards a cluster [137]. Samples from different clusters are highly dissimilar but very similar among the samples in the same cluster. There are variants of the CA technique, including the K-means [132, 163], K-Star [152, 164] and k-Nearest Neighbours [138, 139].

Decision Tree (DTR): The concept of the DTR technique is to create a tree-like model that predicts the class label of a given test sample based on previously known training samples. A decision tree is constructed by continuously splitting feature data into subsets represented as tree nodes so that the information gain ratio at each node is maximised. This iterative process stops when a node has only one single label, or when further splitting a tree node no longer provides additional information gain. The DTR technique such as the J48 [84, 131, 164] and the Random Forest [22,118,165] are widely used in touch dynamics biometrics research.

Support Vector Machine (SVM): The SVM is another technique commonly used in many studies [146,149,163]. The fundamental concept of this technique is first to determine how two classes of feature data differ from each other and then create a boundary that best separates them. Having this boundary, subsequent test samples can be classified as either legitimate or illegitimate according to which side of the boundary they are located. The search for this boundary can be performed within a 2-dimensional hyperplane using a linear kernel function or a higher dimensional feature space using a non-linear kernel function such as Radial Basis Function [106, 163].

Neural Network (NN): The NN technique simulates the information processing structure of biological neurons. Typically, a NN architecture consists of three interconnected layers: input, hidden and output layer. To start with, the feature data from all subjects are fed into the input layer of the network as a set of neurons. An activation function is used to assign weights to each neuron. Then, the information of the activated neurons is passed from one to another within the hidden layer. This process iterates until an output is produced. Finally, based on the output values, a learning process is used to update the weights of each neuron in the hidden layer to improve the network. Some commonly used NN techniques are Radial Basis Function Networks [113, 164] and Multi-Layer Perceptron [106, 131, 152]. A NN technique generally produces a higher level of accuracy in identifying a subject but is more computationally expensive [114] and more time consuming to be used [166]. According to [118], running this technique on mobile devices with less than 512MB of memory is impractical.

Distance Measure (DM): The DM technique calculates a dissimilarity or similarity score between a test sample and the training sample of a given subject.

The score is then compared against a threshold to determine if the test sample belongs to the target subject. Most frequently used DM techniques include Euclidean [56, 166, 167], Manhattan [104, 150, 168], Mahalanobis [137, 138] and Bhattacharyya [151].

Statistical (ST): Several ST techniques have been used in touch dynamics biometrics research. These techniques include the mean and standard deviation [153, 169] and the deviation tolerance [145, 170].

The output of feature classification operation is normally associated with a matching score, and this score is used for feature matching. Feature matching is an operation responsible for deciding whether or not the test sample is indeed originated from the target user. The decision is usually made by comparing the score against a predefined threshold. Before the final decision is made, a number of fusion approaches may be applied to improve the accuracy performance of a touch dynamics authentication system. These fusion approaches are further described in the following section.

3.7 Fusion

Fusion is an approach used to combine information from multiple sources to improve the accuracy performance of a biometrics authentication system. The multiple sources may be from multiple sets of features or by using multiple machine learning techniques. The information from these sources may be combined using four different approaches: (i) Feature Fusion (FF), (ii) Ensemble Fusion (EF), (iii) Score Fusion (SF), and (iv) Decision Fusion (DF).

Feature Fusion (FF): The FF approach is the most used fusion approach in touch dynamics biometrics research. FF involves concatenating more than one set of features into a superset of features and is performed before the feature classification operation. Fusion may be performed on features extracted from the same [162] or different sensors [138, 147]. FF is simple to implement and it enables the utilisation of additional properties of multiple sets of features. Despite its benefits, it can result in an overly large set of features known as the curse of dimensionality [171]. Some machine learning techniques, such as the DTR technique, may not work well with a high dimensional feature set [172].

Therefore, the number of feature set fused may influence the selection of machine learning technique.

Ensemble Fusion (EF): The EF approach, unlike the FF approach, is performed after the feature classification operation. Fusion is performed by training two or more models and then combine the structure or output of the models to improve the overall accuracy performance of the models. The Random Forest technique used in touch dynamics biometrics research is a typical example of the EF approach [22, 118, 165].

Score Fusion (SF): The SF approach, like the EF approach, is performed after the feature classification operation. For example, in the published work of [103], two different machine learning techniques were used independently on one set of features, resulting in two matching scores, one from each technique. The two scores are then combined into a single score for decision making. Methods such as the sum, weighted-sum and product rules are commonly used to combine multiple scores [156]. If the scores from different techniques are not comparable, they will need to be normalised before fusion [52].

Decision Fusion (DF): The DF approach is the least complex among the four fusion approaches. It requires minimum changes being applied to the internal structure of each machine learning techniques. Fusion is performed by combining decisions (accept or reject) made by multiple machine learning techniques using voting rules, such as the majority, AND and OR rules [156].

3.8 Analysis and Discussion

This section provides an overview of the performances achieved by related work, i.e. touch dynamics authentication solution, reported in literature. For the sake of clarity, the performances are discussed based on the working modes that they support, static or dynamic modes, input string lengths used, feature discriminative capabilities, fusion approaches used and the system overheads they each impose.

3.8.1 Static Working Mode

In the static mode, the identity of a subject is verified based on the input provided by the subject in the first instance of accessing a system. This is the first line of protection and also the most common security protection measure deployed on mobile devices. According to our literature research, character-based and digit-based passcode are the most used input string types in the static mode. In the following, we select and discuss one work from each input string type.

To test the viability of integrating touch dynamics biometrics with a character-based passcode, the authors of [118] has conducted experiments to acquire passwords from 20 subjects. The passcode is a predefined alphanumeric character string ("7q56n5ll44"). To acquire each passcode, a subject is required to complete the passcode input by touching on different keys on the touchscreen keyboard. The feature extraction was performed using two different methods, individual key-based extraction and overall key-based extraction. With the first method, the feature data of every single touched key are extracted and analysed by the classifier. With the second method, the average feature data of all the touched keys is calculated before analysed by the classifier. Experimental results show that regardless of the classifiers used, the first method always outperforms the second. This is because more fine-grained information can be captured in the feature data of individual touched key than the average feature data of all touched keys.

The work conducted by [153] attempt to use touch dynamics biometrics to enhance digit-based passcode. In the experiment, ten subjects were asked to provide 100 input samples of a predefined PIN ("1593") each. Using a NN classifier, they were able to achieve an FRR of 14%. The FAR test was conducted somewhat differently. Two additional subjects were recruited (acting as an attacker) to imitate the PIN input pattern of the all the ten legitimate subjects. To facilitate the impersonation attempt, the two attackers were given a visualisation tool. The tool is designed to reveal the correct timing and pressure information of each digit of the PIN input of the legitimate subjects. Even by deliberately exposing the timing and pressure feature data of the PIN, the authors were still able to archive a FAR of 16%.

As can be seen from Table 3.3, the work conducted by [162] is the only one that uses a symbolic passcode instead of a character-based or digit-based passcode. The symbolic passcode consisted of spade, heart, diamond and club. Each

Studies	Subject Sizes	Input Types	Input Lengths	Features	Classifiers	EERs (%)
[86]	12	C	6	TM, SP	CA	6
[164]	16	C	11	TM, SP	DTR	2.03†, 2.67#
[118]	20	C	10	TM, SP	DT	26
[138]	20	C	8-9	TM, MO	DM	0.08
[173]	25	C	34	TM	NN	9.3
[85]	28	C	6-8	TM, SP	PM	21.02
[145]	40	C	11	TM	ST	7.5
[137]	42	C	10	TM, SP	DM	12.9
[170]	56	C	10	TM, SP	ST	5.79†, 5#
[152]	10	D	4	TM, SP	NN	15.2
[148]	12	D	6	TM, SP, MO	NN	91.2*
[168]	55	D	4	TM, SP, MO	SVM	4.4†, 5.3#
[147]	80	D	4	TM, SP, MO	ST	3.65
[153]	100	D	4-10	TM, SP	ST	8.4
[174]	100	D	8	TM, SP, MO	SVM	0.556
[132]	152	D	17	TM, SP	CA	4.19†, 4.59#
[162]	10	O	4	TM, SP	NN	82.18*

C: Character-based; D: Digit-based; O: Other

TM: Timing; SP: Spatial; MO: Motion

*Accuracy; †FAR; #FRR

Table 3.3: Research work conducted in static mode

of the symbols can be represented in four different colours. A total of 16 symbols arranged in a 4x4 matrix block were designed as the input screen layout. To control the screen size inequality of different mobile devices, the data acquisition tool is developed as a web application so that the input screen layout can automatically scale to the screen size of different mobile devices. An accuracy of 82.18% was achieved by using the Bayesian Network classifier to classify the timing and spatial features of the ten subjects.

3.8.2 Dynamic Working Mode

In the dynamic mode, a subject's identity is continuously verified throughout the active session of a mobile device. To authenticate a subject in the dynamic mode, a longer input string is normally required than in the static mode. The longer input string is commonly acquired by researchers using three different ways: (i) requiring subject to input a long input string, (ii) accumulating touch actions over a predefined period of time, or (iii) setting a predefined number of touch actions to accumulate.

The authors of [22] attempt to identify a subject based on the character strings commonly used in emails and chat messages. It is noted that storing feature data of each character combination of the character string may impose system overhead on a mobile device. Therefore, only the 40 most frequently used English language character combinations were stored and used for analysis. The subjects input were analysed through a series of character strings with the lengths of 20, 40 and 60. If the touch dynamics pattern within the character string is unrecognised, the subject is declared as an impersonator and a reauthentication request will be invoked. The result shows that the best accuracy performance achieved was 1% EER by using the Random Forest classifier based on a 40 character string. The author also suggested that for authentication in the dynamic mode, maintaining high usability should be given a higher priority. This means that achieving a low FRR value is important.

The experiment conducted by [133] is rather different from the one discussed above, as in the input strings are some predefined 160 characters pangrams (words or sentences containing every letter of the alphabet at least once). One example given was "the quick brown fox jumps over the lazy dog". As shown in Table 3.4, the number of subjects recruited in the experiment is the largest (315 subjects) among the research works conducted in the dynamic mode. The FRR test was

conducted somewhat differently, among all the 315 subjects' samples, only 12 subjects' samples were used to calculate the FRR value, with the remaining 303 subjects' samples used for the FAR test. An accuracy performance of 1% FAR and 18% FRR was obtained by using a SVM classifier with the linear separating function.

Studies	Subject Sizes	Input Lengths	Input Freedom	Classifiers	EERs (%)
[135]	5	15 touch actions	Yes	PM	80*
[114]	13	6000 touch actions	Yes	NN	86*
[113]	20	10-minutes touch actions	Yes	NN	2.92
[126]	32	5-minutes touch actions	No	SVM	< 10
[102]	36	∼8974 characters	No	DTR	4.25†, 6#
[22]	40	14-53 characters	No	DTR	1
[143]	50	120 touch actions	Yes	NN	2.46
[108]	51	800 touch actions	Yes	SVM	< 8
[106]	190	80 touch actions	No	PM	∼13.8
[133]	315	160 characters	No	SVM	1†, 8#

*Accuracy; †FAR; #FRR

Table 3.4: Research works conducted in dynamic authentication mode

Data acquisition operation conducted by [143] did not use a predefined input string. Data acquired was in the form of touch actions generated during subjects' routine input activities. For example, subjects were requested to use an Android smartphone to perform their usual activity, such as text messaging and web browsing. Unlike their previous study [113], where a time-based session (ten minutes) was used to acquire touch actions, their latest study predefined a fixed number of touch actions to be acquired in each data acquisition session. The changes were made because the number of touch actions acquired by using the time-based session were unpredictable. The latter approach was able to supply a more consistent and sufficient number of touch actions for analysis, and thus improves the effectiveness of the experiment. In the experiment, the performances of five different classifiers were compared. The results show that the Radial Basis

Function Network classifier achieved the best EER of 2.46%.

3.8.3 Input String Length

Previous studies suggested that the input string length has a direct relationship with the accuracy performance of a classifier. For example, the experiment conducted by [147] compared the accuracy performance between a 4-digit and a 8-digit PIN. The dissimilarity scores between the legitimate and the illegitimate subjects have been calculated and plotted in a frequency distribution graph. The authors discovered a clear gap in the dissimilarity scores graph of the 8-digit PIN, but an overlap for the 4-digit PIN. This shows that the longer the PIN, the better it is at representing a subject's touch dynamics pattern, and the higher the accuracy performance of the classifier. The authors of [114] also discovered that by increasing the number of touch events (from 5 to 15) used for data classification, the accuracy performance could be improved by 27%. Table 3.5 shows similar observation by other research work.

3.8.4 Feature Discriminative Capability

The timing feature has been the most frequently used feature since the early stage of keystroke dynamics research (predecessor to touch dynamics). More recent mobile devices are embedded with various sensors that are capable of providing additional features that can be used to describe a subject's touch dynamics pattern. It is interesting to compare the discriminative capabilities of these new features (i.e. the spatial feature and the motion feature) against the timing feature.

In the experiment conducted by [85], the spatial feature, such as the touch pressure size, touch pressure intensity and touch position, were extracted from the capacitive sensor of a mobile device. The result shows that these spatial features always perform better than the timing feature in most of the experimental settings. The best EER was obtained using the touch position and the touch pressure size feature, and it was approximately 14% better than the timing feature.

The motion feature extracted from the motion sensor provides additional movement information that can be used to describe a touch action performed by a subject on a mobile device. The experiment conducted by [138] reported that the EER of their proposed method was reduced from 4.97% (by using the

Studies	Working Modes	Input Types	Input Lengths	EERs (%)	Improvements (%)
[118]	Static	C	10	26	+47.69
			47	13.6	
[103]	Static	C	200	90.7*	+5.51
			300	95.7*	
[147]	Static	D	4	5.98	+24.75
			8	4.5	
[175]	Static	D	6	23	+21.74
			10	18	
[22]	Dynamic	C	20	8.93†, 5.6#	+88.8, +46.43
			40	1†, 3#	
[114]	Dynamic	O	5	67.7*	+27.03
			15	86*	

C: Character-based; D: Digit-based; O: Other
*Accuracy; †FAR; #FRR

Table 3.5: Accuracy performance of short and long input string lengths

timing feature) to 0.08% after using the motion feature. Whilst the EER values may seem to be unusually low (possibly due to the small number of subjects and the higher number of training samples used in the experiment), the EER reduction did show that the motion feature provides better discriminative capability than the timing feature.

Table 3.6 clearly shows that some of the features perform better than the other. However, this does not mean that the lower performed features are not useful. Studies [85, 103] suggested that by combining multiple features (using fusion approaches), the accuracy performance of the classifier can be further improved (further discussions in the following section).

Studies	**Input Strings**	**EERs (%)**			**Better off by (%)**
		TM	**SP**	**MO**	
[103]	300 Characters	39*	38*	-	+2.56
[85]	6-8 Characters	21.75	18.65	-	+14.25
[149]	10 Digits	10.5	3.5	-	+66.67
[147]	16 Digits	16.5	11.5	15	+30.03 (TM), +23.33 (MO)
[174]	8 Digits	71.3*	69.03*	98.26*	+3.18 (TM), +29.75 (MO)
[138]	8 Characters	4.97	-	0.08	+98.39

TM: Timing feature; SP: Spatial feature; MO: Motion feature

*Accuracy

Table 3.6: Accuracy performance of different feature data types

3.8.5 Fusion

To increase the overall accuracy performance of a touch dynamics authentication system, researchers have attempted to use different fusion approaches. For example, the authors of [147] suggested that different feature captures a different aspect of a touch dynamics pattern and combining them allows a subject's touch dynamics pattern to be more precisely represented. To prove this assumption, the authors conducted four experiments using four different features: (i) acceleration, (ii) touch pressure intensity, (iii) touch pressure size, and (iv) timing. When the features are used individually, each of the features obtained the EER

of 19%, 12%, 25% and 21%, respectively, whereas, when all four features are used together, the EER has decreased to 4.5%. This shows that using the combination of all features produces a better result than using the features individually.

As has been discussed in Section 3.7, there are four different types of fusion approaches, i.e. FF, EF, SF and DF. We can use more than one of the approaches simultaneously to improve the accuracy performance of a classifier. For example, the authors of [103] use both the FF and SF approaches in their experiment. To start with, they first combined different features extracted from a 300 character input text (FF). Then, the individual score produced by two different classifiers are combined (SF) to collectively make an authentication decision. The accuracy performance was successfully improved from 55% (before fusion) to 95.7% (after fusion).

Table 3.7 summarises the research work in touch dynamics biometrics that uses different fusion approaches. All of the experiments shows an improved accuracy performance after applying fusion approach.

3.8.6 System Overhead

The accuracy performance has been the primary evaluation metric for a touch dynamics authentication system. Nevertheless, system overhead in terms of computational speed and resource consumption is also an important evaluation criterion. The lower the system overhead of a proposed authentication system, the less the impact it has on the performance of a mobile device [113]. The trade-off between these two metrics was normally not reported. This is because only a small number of research work (12%) conducted both the data acquisition and the feature classification operation on a mobile device. Most of the research work conducted the feature classification operation on a desktop computer.

To evaluate the computational speed and the resource consumption of a proposed method, one has to implement both the data acquisition and feature classification operation on a mobile device. For example, the authors of [118] conducted their experiment entirely on a Sony Ericsson Xperia smartphone. The Random Forest and the k-Nearest Neighbor classifiers were implemented to identify subjects based on a 47 character input string. The experimental result shows that the k-Nearest Neighbor classifier not only achieves a better EER but also consumes relatively the same amount of computational time, memory and CPU resource than the Random Forest classifier.

Studies	Input Strings	Fusion Approaches	EERs (%)		Improvements (%)
			Before Fusion	After Fusion	
[162]	Symbol	FF	62.64*	82.18*	+31.19
[147]	8 Digits	FF	12	4.5	+62.5
[132]	17 Digits	FF	9.28†, 6.72#	4.19†, 4.59#	+54.8, +31.7
[149]	10 Digits	FF	3.5	2.8	+20
[174]	8 Digits	FF	98.26*	99.17*	+0.93
[85]	6-8 Characters	FF	18.65	13.74	+26.33
[176]	10 Characters	FF, EF	6.6	3.1	+53.03
[22]	4 Characters	FF, EF	17.8†, 60#	10.4†, 11.1#	+41.57, +81.5
[103]	300 Characters	FF, SF	55*	95.7*	+74
[156]	10 Characters	SF, DF	4.032	0.806	+80

*Accuracy; †FAR; #FRR

Table 3.7: Accuracy performance before and after applying the fusion approach(s)

The number of features used to represent a subject's touch dynamics pattern can also influence the computational time and the resource consumption of a mobile device. The more the number of features used to represent a subject's touch dynamics pattern, the more the computational time and the resource consumption incurred by a mobile device. Ideally, one would want to minimise the number of features used and yet maintaining a reasonable accuracy performance. This has been shown in an experiment conducted by [143], where, even when the number of features was reduced from 37 to 8, the accuracy performance was maintained at a reasonable level. By reducing the number of features used, the computational time and the resource consumption has been reduced.

3.9 Further Analysis and Discussion on Most Related Work

This section provides further critical analyses of the related work most relevant to ours, i.e. touch dynamics authentication based-on digit-based input strings. Depending on the input strings used, the related work can largely be classified into two groups: i) work on digit-based input strings (or PINs), and ii) work on character-based input strings. As the digit-based authentication method has so far been the most widely used authentication method for mobile devices [177], so our further critical analysis here will focus on i). The work most relevant to ours has largely been focusing on studying the applicability of, or improving the performance in, using touch dynamics as a means of verifying subjects. Several studies have been published. We here discuss the most notable ones.

The work reported in [166] was carried out to test the applicability of verifying subjects based on touch dynamics using digit-based input strings. In their experiments, timing, touch pressure size and touch pressure intensity features were extracted by using device built-in capacitive sensor, and, in addition, linear and angular motion features were extracted by using the more advanced accelerometer and gyroscope sensors. As the features extracted by the accelerometer and gyroscope sensors have high feature dimension, the authors applied the Principal Component Analysis method to reduce the feature dimension. With lower feature dimension, feature classification consumes less computational resources. By using an Euclidean Distance classifier, they obtained an accuracy performance of 20% EER on a 4-digit PIN. However, the performance comparisons of individual

features were not reported in this paper. It would be interesting to see whether or not one feature performs better than the other or which is the best performing feature. Also, relying on the use of advanced sensors to extract some of the features means that the method can only be deployed on mobile devices that are equipped with these sensors, limiting the scale of its deployment. This is also the case for the work reported in [134], where the authors only used features extracted from a 3D touch sensor. This type of sensor is only available on certain brands of mobile devices (i.e. iPhone 6s and later releases), making their proposed method only applicable to some mobile devices.

The authors of [152] also investigated the accuracy performance of touch dynamics using a digit-based input string. In their investigation, they recruited ten subjects. Each subject was asked to provide 100 input samples of a predefined 4-digit PIN. They used the Multi-Layer Perceptron classifier to classify the legitimate subjects, and these subjects are correctly classified up to 86% of the time. The results are encouraging, but to achieve the reported level of accuracy performance, they have made use of 100 input samples per subject to train the classifier, and this sample amount is considered to be large. Acquiring such a large number of samples from the subjects is time-consuming and not always practical during an enrolment phase. It is not clear if the same level of accuracy performance could still be achieved when a smaller number of input samples are used. A similar case can be said for the work reported in papers [150, 163, 178].

The paper [153] reported an experiment carried out on a small number of input samples than those described above. Input PINs ranging from 4 to 8 digits in length were used. By using a simple statistical classifier, the authors were able to achieve an EER of 8.4%. They have also studied the time taken to perform the training and testing phase of different PINs. The average total time taken to complete both phases is less than $12ms$. This result is encouraging when considering the deployment of touch dynamics biometrics on power limited mobile devices. In addition, the experiment was carried out on 100 subjects, more than many of the experiments reported in literature. For example, the experiments reported in papers [142, 148, 152, 161, 178–180], were carried out on 15 or fewer subjects. Carrying out an experiment on a larger number of subjects allows us to draw more conclusive conclusions.

The authors of [147] conducted their experiments using 4- and 8-digit PINs. In their experiments, they employed a statistical classifier and obtained EER values

of 3.65%, 6.96% and 7.34% using three different 4-digit PIN numbers, "3244", "1111" and "5555", respectively. These results show that when a higher repetition of digits is used, the accuracy performance is reduced. They also compared the accuracy performances of two different 8-digit PINs (i.e. "12597384" and "12598416"). Surprisingly, for one of the 8-digit PIN ("12598416"), the EER value was 4.45%, marginally worse than the 4-digit PIN ("3244"). This finding is in contrary to previous studies [142, 163, 175] which have suggested that longer input strings produce a better accuracy performance.

The work reported in [175] was rather unique. The authors proposed a method to allow subjects to change their PINs without rebuilding a new authentication model. The subjects were asked to input ten different randomly selected 10-digit PINs. Based on the samples collected, they produced a table of all possible feature data for each digit. Using this method, they were able to achieve EER values of 23%, 21% and 18% on three different PINs with the string lengths of 6, 8 and 10, respectively. However, the majority of the subjects taking part in this experiment were at the age of 17-20, it is not clear whether the experimental findings apply to other age groups.

In most of the experiments reported in literature, timing features were extracted using two shortest feature lengths (further discussed in Section 4.5.2.1.1). An exception to these is the work carried out by [132]. In this work, the authors used three different feature length values (i.e. 1-graph, 2-graph and 3-graph) for extracting timing features and investigated their accuracy performances on a 17-digit PIN. The experimental results suggested that 1-graph achieved the best accuracy performance. They also remarked that, by combining the features extracted using 1-graph and 2-graph, they could achieve an even better accuracy performance. A similar observation has also been reported in papers [141, 147, 149, 153, 168]. It would be interesting to investigate the accuracy performances of timing features extracted using larger feature lengths such as those investigated in our work.

More recently, the authors of [163] investigated the accuracy performance of touch dynamics on 4-, 5- and 6-digit PINs across different operational scenarios. Unlike other related work, they used only motion features extracted from accelerometer and gyroscope sensors. The authors pointed out that the raw data recorded by these sensors could not be used directly as features to build an authentication model. To make the raw data useable as features, they computed a

set of statistical metrics (min, max, mean, variance, etc.) from the raw data and used the computed metrics as motion features. A similar method has also been used in other experiments such as [147, 161, 168]. By far, this method has only been used to extract motion features. It would be interesting to see how well this method works when used to extract other types of features, such as timing and spatial features. To investigate the effects of operational scenarios on accuracy performance, they designed three types of scenarios (i.e. hand-hold, table-hold and walk-hold) for collecting subjects touch dynamics data. The results show that the hand-hold scenario achieves the best accuracy performance. A similar observation has also been reported in [148, 178]. Also, the results of these three pieces of work consistently show that the table-hold scenario achieves the lowest accuracy performance. These results can be explained as follows. In the table-hold scenario, subjects input their PINs with the device placed horizontally on the table. With the support of the table, the amount of movement inflicted on the device while the subjects are inputting their PINs is small. When the amount of movement is small, the data recorded by the accelerometer and gyroscope sensors will have a small-value range, indicating that the features extracted from these sensors capture less information about the subjects' touch dynamics patterns, thus leading to a lower level of accuracy. This means that the features extracted from the accelerometer and gyroscope sensors may not be effective for user authentication purposes.

By far, the best accuracy performance was reported by [174]. The authors achieved an EER value of 0.56%. They trained an authentication model by using the SVM classifier using both legitimate and illegitimate subject samples. In their experiments, each subject provided four different 8-digit PINs. They discovered that the PIN with repeated digits, e.g. "11111111", resulted in a lower accuracy performance than those without, e.g. "16843752". In their experiments, they imposed a condition that the experimental subjects were selected from those who were very familiar with touchscreen smartphones. This condition was imposed to ensure that the subjects were able to complete the experiment of inputting the PIN in a natural way. This condition might have played a role in achieving the high accuracy performance. It is not clear if the same level of accuracy performance could still be achieved if this condition is removed. Also, in this work, a two-class classification approach is used in building the model, i.e. samples from both legitimate and illegitimate subjects are used to build the model. However,

in real-life, as mobile devices are very much personal devices, illegitimate subject samples may not always be available. Therefore, this approach is less practical. This is also the case for the work reported in papers [87, 168, 174].

3.10 What is Missing

There are a number of studies on the use of touch dynamics biometrics for mobile user authentication. However, these studies have largely focused on improving accuracy performance of the system, but the characteristics of mobile devices and the way they are typically used are not given due consideration. These characteristics are as follows. Firstly, both short and long input strings are used in real-life, but the short string inputs are used more often as they are easier to remember. Secondly, mobile devices are rarely shared among multiple users. Therefore the use of two-class classification techniques may not be the most efficient way of training an authentication model. Thirdly, mobile devices are typically battery powered, so the use of sophisticated sensors (e.g. accelerometer and gyroscope sensors) to capture touch dynamics data or the use of large amount of data when training an authentication model is less efficient. To make such a system more accurate as well as more usable, these characteristics should be taken into account. In addition, more work is necessary to gain a better understanding with regard to how to achieve a better accuracy performance, while, at the same time, keep the overhead costs as low as possible. For example, it is not clear whether a device with a larger screen size would allow us to capture touch dynamics features more effectively, whether using fewer but less overlapping features would allow us to achieve the same level of accuracy performance but with less cost, and whether the use of adaptive learning method would allow us to capture better users' touch dynamics patterns.

3.11 A Way Forward

To investigate how to achieve a better accuracy performance, while, at the same time, keep the overhead costs as low as possible in a touch dynamics authentication system, we have designed and evaluated a novel solution, called Touch Dynamics based Two-factor Authentication (ToDiTA) system with the following measures and ideas:

- Support the use of both short and long input string, making it easier for users to remember the string thus achieving a higher usability level.
- Use the capacitive sensor built-in on most, if not all, touchscreen mobile devices to extract raw touch dynamics data, instead of more sophisticated sensors, making the system applicable to a wider range of mobile devices, less costly and maximising the scale of its deployment.
- Use a device with larger screen size than an average mobile device (i.e. larger than 7 inches) to acquire touch dynamics data, leveraging the higher variation in touch dynamics data (as a result of the additional finger movement space afforded by the device) to train a more accurate authentication model.
- Use feature selection methods to analyse and select only the most distinctive features for model training, reducing the storage space of the features and the model training time.
- Use fewer samples for model training, reducing the amount of time and effort required by the user to complete an enrolment phase thus achieving a higher usability level as well as reducing model training time.
- Use classification techniques that can train a model with only legitimate samples, so that the system can be more efficient in a mobile device context.
- Use an adaptive learning method to update the model, so that the updated model can adapt itself to any changes in a user's touch dynamics pattern.
- Use a more comprehensive dataset to evaluate the system

3.12 Chapter Summary

This chapter has presented an investigative study of the related work in the literature in the area of touch dynamics biometrics. The chapter started by providing an overview of touch dynamics biometrics in general. It then presented a detailed comparison on the related work from a range of perspectives and discussed their performances. The chapter has also provided a further analysis and discussion of the work most relevant to ours. It has highlighted their strengths and limitations.

Finally, the chapter outlined our prospective measures and ideas in designing a touch dynamics based two-factor authentication (ToDiTA) system.

The next chapter presents the design and evaluation of ToDiTA.

Chapter 4

A Novel Touch Dynamics Based Two-Factor Authentication (ToDiTA) System

4.1 Chapter Introduction

This chapter presents the design and evaluation of a touch dynamics based two-factor authentication (ToDiTA) system. The ToDiTA system is designed to strengthen user authentication on mobile device and do so in a usable and efficient manner. The design has made use of the following five measures: (i) use an additional more usable AF, (ii) use a more cost-efficient way to the extract additional feature data without using additional sensors, (iii) use a feature selection process to select and use the most important set of features, (iv) use one-class classifiers to train the authentication model, and (v) use a more comprehensive dataset to more accurately evaluate the system. Experiments were carried out to evaluate the performance of the ToDiTA system under different parameter value settings to investigate the impacts of different settings on the performance of the system. The design and evaluation of the ToDiTA system is our second novel contribution in this thesis.

The chapter starts by describing the measures used in the ToDiTA design in Section 4.2. It then gives the design preliminaries (threat model, assumptions and notations) in Section 4.3. Section 4.4 outlines the ToDiTA architecture. Section 4.5 describes the designs and operations of core functional units of the architecture in detail. Section 4.6 presents the evaluation methodology used.

Section 4.7 analyses and discusses the evaluation results. Section 4.8 compares the ToDiTA system with the most related work. Finally, Section 4.9 concludes this chapter.

4.2 Design Measures

In this section, we discuss the measures used in the design of the ToDiTA system (as mentioned above) in detail. The remaining part of this chapter then describes how these measures are implemented, leading to the design of the ToDiTA system.

4.2.1 Using an Additional More Usable AF

The success of a biometric-based authentication system depends on how well the system is accepted by users [181]. Users are less likely to use a system that has a low level of usability even if the system offers a higher level of security protection [16]. One way to design a secure and also usable authentication system is to use two-factor authentication by integrating touch dynamics biometrics with a knowledge-based authentication method. A two-factor authentication system provides a stronger level of protection than a single factor system, as compromising two factors requires more efforts than compromising a single factor. In terms of usability, the benefit of using touch dynamics biometrics, as one of the AFs, have been discussed in Chapter 3 Section 3.2.2. With regards to the knowledge-based AF, the second AF, there are three options, a character-, a digit-, or a graphical-based passcode. Studies [182–184] have shown that the digit-based passcode is the most preferred among the three options. This is because it is more usable in the following two ways. Firstly, the input difficulty of a digit-based passcode is lower than a character-based passcode. This is because a digit-based passcode uses a smaller set of keys (i.e. only ten keys, 0 through to 9), whereas a character-based passcode requires the use of a much larger set of keys (i.e. A through to Z, 0 through to 9 and often other symbols). Having a smaller set of keys allows all the keys to be fitted into a virtual keyboard of a mobile device with each key having a larger size, making it easier for users to press the keys. Secondly, the input speed of a digit-based passcode is faster than a graphical-based passcode [185].

4.2.2 Extracting Additional Feature Data

One way to improve the accuracy performance of an authentication model is to increase the number of feature data used in training classification algorithms (or classifiers) when building an authentication model [186]. One method to increase the number of feature data is to obtain additional raw data from additional sensors and then extract additional feature data from these raw data. However, this method has three limitations. Firstly, the method is more costly as the use of mobile devices equipped with additional and more sophisticated sensors (e.g. accelerometer and gyroscope sensors) are required. Secondly, the method is more energy-consuming. For example, it consumes more energy to obtain readings from an accelerometer sensor [187]. Thirdly, the method can only be deployed on mobile devices that are equipped with those sensors, limiting the scale of its deployment. Another method that can prevent these limitations is to extract additional feature data from the already-obtained feature data, and these already-obtained feature data are extracted from the raw data captured using a basic sensor (i.e. the capacitive sensor) built-in on most touchscreen mobile devices. The second method can make the authentication system less costly, more applicable to a wider range of mobile devices, maximising the scale of its deployment. For the reasons explained above, we have used the second method to extract additional feature data.

4.2.3 Optimising Feature Data

As mentioned above, extracting additional feature data from already-obtained feature data can increase the number of feature data, and the use of a larger number of feature data may allow us to build a more accurate authentication model. However, increasing the number of feature data indiscriminately may not always be beneficial as using more feature data to train a classifier will result in more processing time and CPU overhead, and using more feature data requires more storage space. In addition, not all of the feature data offer the same degree of discriminative capability; some of the feature data provide a higher degree of discriminative capability, thus more important than the others. To build a more accurate model with as low cost as possible, we should optimise the feature data used, and this can be done by using a feature selection technique that can select the most important set of features. This set of features is optimised and used to

train the classifier.

4.2.4 Using One-Class Classifiers

A mobile device is a highly personal device and is rarely shared among multiple users, so obtaining touch dynamics data from more than one class (e.g. data from multiple users on the same device) is harder in practice. In other words, in real-life cases, only the data from the owner of a mobile device are available. Based on this observation, we have used one-class classifiers to train an authentication model.

4.2.5 Using Comprehensive Dataset

To evaluate the proposed authentication system, a touch dynamics dataset that is as comprehensive as possible is needed so that the conclusion drawn from the evaluation results are more conclusive. We have acquired the dataset taking into account the following criteria:

- Data should be acquired from a very large number of subjects. Using a larger subject size can provide more data to verify the scalability of the authentication system.
- The demography of the subjects taking part in the data acquisition process should be as diverse as possible. In other words, people from different age groups, with different device usage frequencies and working in different professions should be represented as much as possible. In this way, the evaluation results can be as unbiased as possible and can be generalised to a wider population as much as possible.
- The input strings used for data acquisition should not be chosen randomly. Instead, the strings should be carefully designed to cover as many key positioning strategies as possible. In this way, a broad variety of finger movements across the touchscreen can be captured.

The public datasets that we have identified in Section 3.4.5 did not fulfil all the above criteria. Therefore, we have acquire our own dataset (further discussions on the dataset in Section 4.5.1.1).

4.3 Design Preliminaries

This section details the threat model, assumptions and notations used in the design of ToDiTA.

4.3.1 Threat Model

The threat model we use defines an attack scenario where an attacker (or impersonator) tries to gain access to a user's mobile device in an unauthorised manner. In this attack scenario, the following assumptions are used:

- The impersonator has physical access to the device (e.g. by accessing a lost device or stealing a device from its owner).
- The device is locked by a knowledge-based authentication method (e.g. a passcode).
- The device is free of malware. Therefore, threats from malware attacks are not considered.
- The device will be disabled after a specific number of unsuccessful authentication attempts in a row. The device will be re-enabled (to allow further authentication by the user) after a specific amount of time delay. The delay should increase upon further unsuccessful authentication attempts, making it difficult for an impersonator to succeed in brute force attacks. Therefore, threats from brute force attacks are not considered.
- The ToDiTA system is designed for User-to-Device/App authentication and not for User-to-Remote-Site authentication. The user's touch dynamics data and authentication model are stored securely on the device and are not transferred over the network. Therefore, threats from network interception attacks are not considered.

The primary aim of the evaluations is to investigate and assess the added security protection level provided by the biometrics component. In this attack scenario the impersonator has unrestricted access to the device. With this access privilege, the impersonator may compromise the security and privacy of the owner of the device by performing the following actions:

- Steal any account information associated with the apps installed on the device.
- Access sensitive and private data stored on the device.
- Masquerade as the owner and engage in communication with people in the contact list on the device to perform further attacks or malicious activities.

We will demonstrate that, by integrating touch dynamics biometrics with the knowledge-based authentication method, forming a so-called two-factor authentication system, it is harder for the impersonator to successfully bypass the authentication system, as, in this case, the impersonator would have to correctly input the passcode as well as to correctly reproduce the owner's touch dynamics biometrics to perform a successful attack.

4.3.2 Assumptions

To scope our design, the following assumptions are used:

A1. The device is used solely by the owner and not shared among multiple users (e.g. family members and friends).

A2. The device is a touchscreen type of mobile device with a basic built-in capacitive sensor. The device is not equipped with other more sophisticated sensors such as the accelerometer, gyroscope and fingerprint sensors.

A3. The device has the minimum hardware specifications (i.e. 1.3GHz CPU, 1GB RAM and 8GB internal storage space) required to run standard machine learning algorithms on the device for building authentication model.

A4. The users' touch dynamics patterns are stable and do not change over time.

A5. The ToDiTA system is deployed in a verification mode, and the verification mode is operated in a static working mode.

4.3.3 Notations

The main notations used in the description of the ToDiTA design are summarised in Table 4.1. These notations are used throughout the thesis.

Notations	Definitions
4D	A 4-digit input string "5560"
16D	A 16-digit input string "1379666624680852"
m	The number of keys in an input sample
k_i	The i^{th} key in an input sample, $i \in \{1, 2, \ldots, m\}$
p	A timestamp recorded on the action of a finger pressing down on a key
r	A timestamp recorded on the action of a finger releasing from a key
ps	A touch pressure size recorded on the action of a finger pressing down on a key
n	A timing feature length
f_i	The i^{th} feature extracted from an input sample, $i \in \{1, 2, \ldots, d\}$
v	A feature vector or a set of all the f extracted from an input sample
d	The feature dimension of v
τ	A training sample or a set of all the v acquired from an enrolment attempt
$\hat{\tau}$	A testing sample or a set of all the v acquired from an authentication attempt
δ	A classsification score
$\hat{\tau}_L$	$\hat{\tau}$ classified as belonging to the owner of the device
$\hat{\tau}_I$	$\hat{\tau}$ classified as belonging to an impersonator
D_{auth}	An authentication decision
θ_{auth}	A predefined authentication threshold

Table 4.1: Notations

4.4 Architecture Design Overview

This section presents an overview of the ToDiTA system architecture. As shown in Figure 4.1, the architecture consists of six functional units: an App Interface Unit (AIU), a Raw Data Acquisition Unit (RDAU), a Feature Construction Unit (FCU), a Model Training Unit (MTU), an Authentication Decision-Making Unit (ADMU), and a Data Storage Unit (DSU). All the six units are run on a user's mobile device.

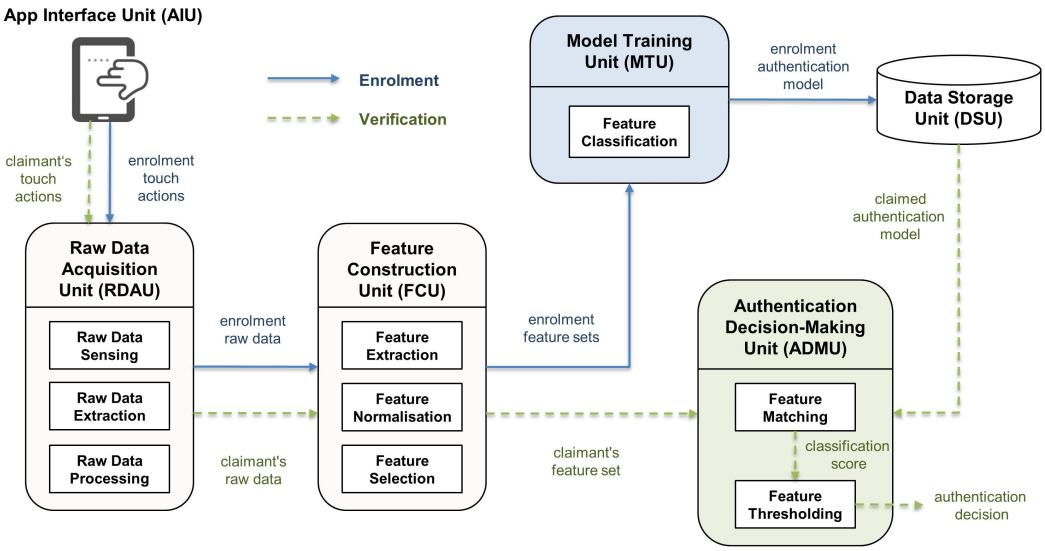

Figure 4.1: ToDiTA system architecture

AIU is an input facility for users to provide their touch dynamics input samples. DSU is a database used to store authentication model. The rest four units provide the core functions of ToDiTA. These units along with their functions are highlighted below and described in more depth in the next section.

RDAU: This unit acquires touch dynamics input samples from users via AIU and extracts raw touch dynamics data from the samples. These raw data are passed to FCU for feature construction.

FCU: This unit identifies and extracts touch dynamics features from the raw data. The extracted features are passed to MTU for model training.

MTU: This unit analyses and trains the extracted features to build an authentication model. The built model is stored in DSU and will be used by ADMU for

authentication decision-making.

ADMU: This unit makes an authentication decision by matching a claimant's touch dynamics features against the model stored in DSU.

The ToDiTA operations can broadly be captured in two phases: enrolment and verification. In the enrolment phase, the touch dynamics input samples of the owner of a mobile device are acquired, processed and transformed into an authentication model that is stored in the DSU. In the verification phase, the touch dynamics input samples of a test subject (i.e. a claimant) is acquired, processed and compared against the model retrieved from DSU to verify if the claimant is indeed whom he/she claims to be (i.e. the owner of the mobile device). The two operational phases along with the units involved are illustrated in Figure 4.1. In the next section, we describe the ToDiTA design in more detail.

4.5 Architecture Design in Detail

This section describes the ToDiTA system design in more detail, focusing on the designs of the four core functional units (i.e. RDAU, FCU, MTU and ADMU) and discusses the issues involved.

4.5.1 Raw Data Acquisition Unit (RDAU)

RDAU is responsible for extracting raw touch dynamics data from subjects' input samples. This section describes the design of this unit, giving detailed discussions with regard to how the raw data acquisition experiment is setup (Section 4.5.1.1), how input samples are acquired (Section 4.5.1.2), how raw touch dynamics data are extracted from the input samples (Section 4.5.1.3), and how the raw data are processed into a proper format for further analysis (Section 4.5.1.4).

4.5.1.1 Raw Data Experimental Setup

The setup of an experiment carried out to acquire a touch dynamics dataset concerns a number of issues, and they are: (i) defining a data acquisition procedure, (ii) determining a physical environment, (iii) recruiting subjects, (iv) selecting a data acquisition device, and (v) selecting input strings. In the following, we

discuss these issues in detail.

Defining a data acquisition procedure

The use of an improper data acquisition procedure could cause a subject's touch dynamics data to be captured improperly. Improperly captured data may have adverse effects on the quality of the data acquired, which could lead to inaccurate study results. To ensure that a subject's touch dynamics data are captured properly, a proper data acquisition procedure should be used. To design a proper data acquisition procedure, two issues should be considered.

The first issue is how to ensure data are acquired after a subject is in a stable state (i.e. after he/she is familiar with the data acquisition procedure, device and app). According to [188], a subject's input pattern, style and/or speed can change over time and they stabilise after some time. To ensure data are acquired after a subject is in a stable state, we have added a familiarisation time at the start of each data acquisition session. During this time, subjects were asked to familiarise themselves with the data acquisition procedure, device and app. They could use as much familiarisation time as necessary before inputting their data. We have observed that subjects who spent less familiarisation time are those who have used mobile devices more frequently. This correlation may be due to the fact that subjects who use mobile devices more frequently tend to be more familiar with the facility and functionality of their mobile devices, so they can familiarise with, or learn how to use, a new device and app more quickly.

The second issue is how to best capture inter-session variations for each subject. According to literature [85], if all the samples from a subject are acquired in a single session, inter-session variations in the touch dynamics pattern of the subject may not be captured properly. To overcome this limitation, some authors [140, 153] suggest breaking a single data acquisition session into multiple sub-sessions that are separated by intervals and then combine the data acquired in the sub-sessions into a single set. However, breaking a single data acquisition session into multiple sub-sessions may mean that the sessions will span across a longer period, and this may lead to a higher subject withdrawal rate or a lower subject participation rate [155]. As a result, the sample size of the dataset may be reduced.

To balance these considerations, i.e. to achieve a higher subject participation rate, while, at the same time, to better capture inter-session variations of

the subjects, we have chosen to acquire data in a single session, but to add a familiarisation time to each subject's data acquisition session. On average, each subject took less than 5 minutes to complete a single session in addition to the familiarisation time.

Determining a physical environment

Data acquisition may be conducted under two main physical environments. The first is in an uncontrolled environment, i.e. while subjects use their mobile devices as usual in an uncontrolled or a natural manner and without any restrictions. The second is in a controlled environment, i.e. under strict supervision in a controlled laboratory environment at a fixed location. These two options come with their respective advantages and disadvantages.

The first option is expensive, as due to the ubiquitous nature of mobile devices, subjects are likely to be on-the-move and following the subjects while collecting the data may not be convenient and could be costly. Also, data acquired in this manner may be of poorer quality, as there is a risk of the data being distorted or tampered with. On the positive side, data acquired naturally without imposing any restrictions may better capture a subject's touch dynamics pattern. This is because a subject's interactivity with his/her device may differ in different circumstances, e.g. when performing a task at hand or doing a task in a controlled environment.

The second option is less costly in terms of setting up and running the experiments, and data acquired may be of better quality, as the subjects are supervised, so the chances of data being tampered with are lower. Also, when data acquisition is conducted under a controlled environment, external factors that may inflict noise to data can be prevented. However, there have been suggestions that experimental results obtained from data acquired under a controlled environment may be overly optimistic compared to those acquired in an uncontrolled environment [112].

To balance between preserving data quality and better capturing subjects' touch dynamics patterns, we have made a hybrid use of the first and second option in our experiments. We have established a semi-controlled data acquisition environment and let subjects to choose their preferred locations where their touch dynamics data were acquired. The locations used included offices, homes, classrooms, inside vehicles and public areas. To prevent inconsistent inputs or

outliers mid-way into the data acquisition task, subjects were told to perform the required task continuously without breaks, avoid distractions and stay focused while performing the task. To avoid distracting the subjects and to allow them to perform the data acquisition task naturally, we left the subjects alone when they are performing the task and monitor them from a safe distance. Also, to prevent subjects from deliberately altering the way they interact with their devices and to capture their touch dynamics patterns in a natural manner, we explained the purpose of the study to the subjects only after they have completed the entire data acquisition session.

Recruiting subjects

Recruiting subjects for an experiment involves two issues. One is to determine the subject size used in the experiment. The other is to select the subjects.

The subject size refers to the number of subjects from whom data will be acquired. The subject size is known to affect the results of the experiment [125]. Typically a subject size of greater than 100 subjects is regarded as a very large size [82]. The larger the subject size, the more the data that could be acquired for the dataset and the more accurate the experimental results. Most of the studies carried out in the field of touch dynamics research used subject sizes smaller than this value, with two exceptions [106, 132], both of which recruited more than 100 subjects. However, in these two studies, the subjects recruited were restricted to a certain profession (i.e. mainly students), and the datasets were not made public. In our experiments, 150 subjects were recruited. All the subjects were recruited on a voluntary basis, i.e. without receiving any monetary benefits. Formal ethics approval was not required as the data acquisition was conducted outside Europe and all the subjects are Non-EU nationals. However, we have taken the required steps to ensure that no personally identifiable data or information were collected. This is achieved by taking the following measures:

- All subjects were requested not to use their real names but instead to use a pseudonym or random string as an identifier.
- Age-related data were collected in the form of categorical data (e.g. between 20 and 40) instead of discrete data (e.g. 28).
- No contact or personal information was collected.

- The input strings were predefined; they are unlikely to be used by the subjects in real life.
- Other demographic data collected such as hand preference, device usage frequency and affiliation were also not sufficient to reveal the identity or other private information of a particular subject.
- The raw touch dynamics data acquired are arbitrary numerical values.
- Any identifiable data are further anonymised if necessary.

With regards to subject selections, three criteria are usually used, age, affiliation and profession. In most of the published work [114, 137, 138], subjects were selected from a specific age group, confined to people within the same organisation (e.g. within the same research institute) or restricted to limited profession (e.g. students). Datasets acquired under such conditions are often considered as convenience samples, as it is relatively easier and less costly to acquire data from nearby population groups [189]. Sometimes data acquired from convenience samples may be biased; they may not properly represent the biometric features from a wider population or the conclusions drawn from convenience samples may not realistically generalise to a wider population. To ensure study results are as unbiased as possible, and can be generalised to a wider population as much as possible, the subjects recruited in our experiment are from diversified groups, from different age groups, have different device usage frequencies and are people from the general public working in different professions. Table 4.2, summarised the demography of the subjects recruited in our experiment.

Selecting data acquisition devices

Devices used in data acquisition may be selected in one of the two methods. The first is to predetermine a device and let all the subjects use this predetermined device. The second is to let subjects use their own devices. The second method imposes fewer constraints on how a data acquisition experiment should be carried out. For example, the subjects do not need to be physically present during, or be supervised throughout, the data acquisition process. Without these constraints, data acquisition can be conducted on a larger scale and can reach out to wider population groups. However, using the second method also means that the devices used by different subjects are likely to be different, and the use of different devices may have implications. Firstly, the variations in sensor resolution,

Properties	**Categories**	**Number of Subjects**	**Distributions (%)**
Age	Below 20	28	19
	Between 20 and 40	66	44
	Above 40	56	37
Device Usage Frequency	Rarely	49	33
	Average	32	21
	Often	69	46
Affiliation	Academia	18	12
	General Public	132	88
Gender	Male	45	30
	Female	105	70
Hand Preference	Right-hand	136	91
	Left-hand	14	9

Table 4.2: Subject demography of our experiments

program compatibility and functionality across different devices may introduce inconsistencies in the data acquired [105]. Inconsistent data may introduce bias into experimental results, which, in turn, may lead to inaccurate experimental observations, thus incorrect conclusions. Secondly, as subjects are usually more familiar with their own devices, allowing subjects to use their own devices in data acquisition may also introduce bias into experimental results. Based on these considerations, we have chosen to use the first method for our data acquisition experiments. This method is also commonly used in related work reported in literature [103, 104, 111].

The device used in our experiments is a Samsung Galaxy Tab (GT-P7510) with a 10.1-inch screen, a 1GHz dual-core processor, and a 1-GB RAM, and it operates on Android 4.0.4 (Ice Cream Sandwich). The device has a larger screen size than an average mobile phone (screen size below 5 inches) and phablet (screen size between 5 and 7 inches). The decision for using a device with large screen size is based on the following observations. Firstly, the average size of mobile devices in the market is getting bigger year after year [190]. Secondly, a subject's touch dynamics pattern may vary with different device sizes [84]. Thirdly, a majority of the devices used in literature have a screen size of approximately 5-inch or less [12, 85, 132, 147, 150]. The use of a device with a larger screen may allow us to discover unknown.

Selecting input strings

What input strings should be used (and their lengths) is an important variable one should consider when designing a data acquisition experiment. There are two ways of selecting input strings. The first is to predefine input strings for the subjects to use. The second is to allow the subjects to choose their own input strings. The second way imposes fewer constraints and better resembles real-life situations. However, input strings chosen by different subjects are likely to be different both in terms of the content and length, and the use of different strings by different subjects may have the following three implications. Firstly, it is harder to compare the effects of input strings. Secondly, it is more costly and time-consuming to acquire testing samples, as, in this case, there is a need to acquire testing samples for each subject separately. Thirdly, it may be more difficult to make a more unbiased analysis of the experimental results, as, with different input strings, input difficulty or familiarity experienced by subjects may

be different. Based on these considerations, we have chosen to use the first option in our experiments. This option is also commonly used in related work reported in literature [85, 147].

We have selected two PIN input strings with different lengths. The first is a 4-digit string "5560" (hereafter referred to as 4D). This input string length is chosen to represent a PIN commonly used to unlock a mobile device or a debit/credit card. The second is a 16-digit string "1379666624680852" (hereafter referred to as 16D). This input string length is chosen to resemble a debit/credit card number commonly used when making a card-based on-line payment. The use of these two PINs allows us to empirically study the effects of different input string lengths on the performance of ToDiTA. All subjects are requested to input the same PIN for each string length. Here, the "Enter" key is included as part of the input string, as we believe that the touch actions of this key also provide essential touch dynamics characteristics, which is distinct for each individual.

The two PINs are carefully selected so that a broad variety of finger movements across the touchscreen can be captured. To select these two PINs, we have used a key positioning strategy with four rules. This strategy ensures that all the rules are captured in the chosen two PINs. The strategy is graphically illustrated in Figure 4.2 and explained below:

- **Apart:** Keys that are separated by one key.
- **Repetition:** Keys that are repeated.
- **Adjacent:** Keys that are located diagonally to each other.
- **Sequence:** Keys that are located next to each other.

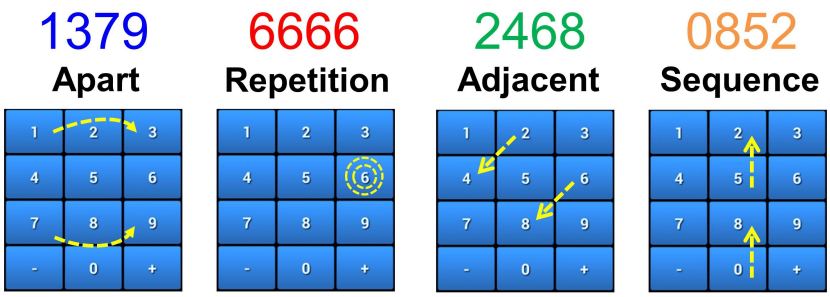

Figure 4.2: Key positioning strategy used in the selection of our input strings

Our dataset

The entire dataset we have acquired consists of 3,000 samples and 33,000 touch actions from 150 subjects. Each subject contributed a total of 20 samples (10 for the 4D string, and 10 for the 16D string) from 220 touch actions (50 for the 4D string, and 170 for the 16D string). The dataset is here at https://goo.gl/sNACU8.

4.5.1.2 Raw Data Sensing

Raw data sensing is the first-step processing applied in RDAU. This is a process in which a user's raw touch dynamics data are obtained and via user-to-device interactions. These interactions are also known as touch actions. They refer to the actions of a user pressing on and releasing from input keys on a touchscreen device. A touch action generates digital interrupts that can be captured by the touchscreen of a mobile device, and these captured interrupts are referred to as raw touch dynamics data.

To acquire the raw data we used a specially developed software with a customised keyboard, i.e. a Data Acquisition App. The App was developed using Java and Android API level 15. Figure 4.3 shows a screen capture of the App.

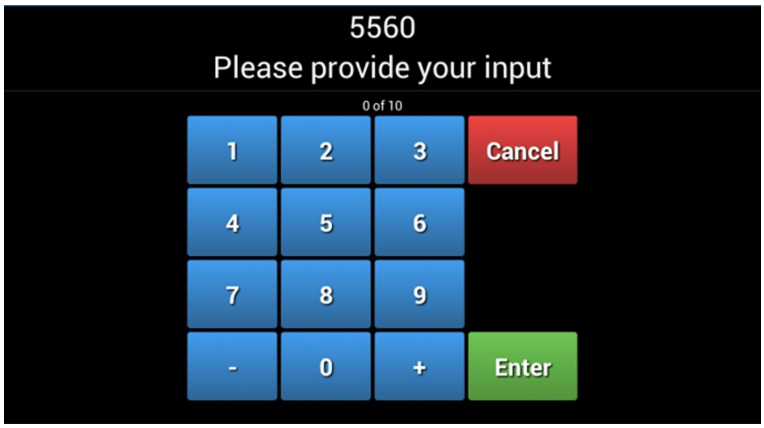

Figure 4.3: A sample Data Acquisition App screen capture

During a data acquisition session, the subjects were requested to complete two rounds of data acquisition, one using the 4D string and the other using the 16D string. In each round, the App displays the required string and a numeric keypad interface in a full-screen mode, and the subjects are asked to input the string. All subjects were required to repeat each string ten times, resulting in ten input samples per string. This number of repetitions is chosen based on the

benchmark value used in experiments reported in literature [118,152]. To help the subjects keep track of the input sample count, a counter is displayed between the string and the keypad. If a subject made an input error when keying in an input sample, that sample would be discarded, and the subject would be prompted to repeat that particular sample. This error handling approach is also commonly used in experiments reported in literature [191, 192].

For each key input of a sample, the App logs a set of raw data. The raw data are classified into two categories: timing and spatial. Both categories of raw data are captured when a touch action is performed. For timing raw data, each touch action is associated with a timestamp. The timestamp value represents the time when the action is taking place. A timestamp is recorded using the nanoTime() API function [193]. It returns a time value with the highest timing precision (up to nanoseconds precision) that is available on the device. The spatial raw data have further two variants, a touch pressure size and a touch pressure intensity (as describe in 3.5.2). The touch pressure size and the touch pressure intensity are recorded using the getSize() [194] and getPressure() [195] API functions, respectively. Both functions return a decimal value between 0 and 1.

4.5.1.3 Raw Data Extraction

Once input samples are acquired from each subject, raw touch dynamics data are extracted from the samples. Each sample consists of m number of keys, and each key, $k_i, i \in \{1, 2, \ldots, m\}$, contains two touch actions. The first is the finger pressing down on the key, referred to as touch action press (TAP). The second is the finger releasing from the key, referred to as touch action release (TAR). The raw data are associated to these two actions, i.e. a timestamp, p, and a touch pressure size, ps, are recorded on each TAP, and a timestamp, r, is recorded on each TAR. Figure 4.4 illustrates the raw data along with their respective touch actions. The raw data extracted from each key are recorded using the format $[k_i, p_i, r_i, ps_i]$, and the recorded data are stored in a separate file for each subject.

In our experiments, the touch pressure intensity raw data were not extracted due to the following reasons. First, we have noticed that the getPressure() function always returns a value of 1. We have tried to resolve this issue but with no success. We have discovered that some other devices also encounter the same problem. This function cannot be used before this problem is resolved by the mobile operating system's provider. Secondly, the getPressure() function is not

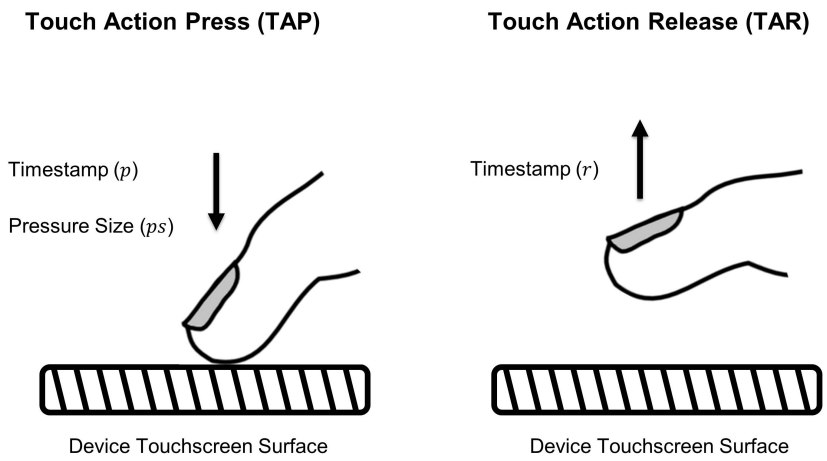

Figure 4.4: Touch actions and their associated raw touch dynamics data

fully supported by some devices in the market [148].

Up to this point, we have discovered that the raw data among subjects are different, even if the input strings used by different subjects are the same. These differences may be because different subjects tend to have different input speeds, chunking patterns (breaking up an input string into multiple smaller subsets (i.e. chunks) to make it easier to memorise [196]), fingertip sizes, finger muscles and device holding positions.

4.5.1.4 Raw Data Processing

Once the raw touch dynamics data are obtained, the data are processed into a format that is suitable for analysis and subject verification. As discussed in Section 4.5.1.2, one of the raw data, the touch action timestamp, has a time value up to nanoseconds precision. This precision might be too high to properly capture a subject's touch action speed, as a human's touch action speed is usually at a slower pace than this order. If the timestamps are not set to an appropriate precision, the accuracy performance of the timing features, which are extracted from the timestamps, may be affected. So, it is important to choose an appropriate precision for the timestamps before extracting any timing features from them. To choose an appropriate precision, we used a scaling factor, α. A default timestamp, t, can be scaled by α to produce a scaled timestamp, $\hat{t}$, with the chosen

precision, and this is done by using the following equation:

$$\hat{t} = \frac{t}{10^{\alpha}} \tag{4.1}$$

4.5.2 Feature Construction Unit (FCU)

FCU is responsible for extracting a subject's touch dynamics features from the subject's raw touch dynamics data. This section describes the design of this unit, giving detailed discussions with regard to the types of features that are extracted from the raw data and how they are extracted (Section 4.5.2.1), the normalisation of the extracted features to the same value range (Section 4.5.2.2) and the selection of a subset of optimal features from the extracted features (Section 4.5.2.3).

4.5.2.1 Feature Extraction

In our design, two categories of features are extracted, first-level features (FLF) and second-level features (SLF). The FLF features are a basic set of features extracted directly from the raw touch dynamics data. The SLF features are an extended set of features extracted from FLF features. In the following sections, we give details on how these two categories of features are extracted.

4.5.2.1.1 First-Level Feature (FLF)

This section describes the process of extracting FLF features from a subject's raw touch dynamics data and constructing a cumulative FLF feature vector for the subject. For each subject, seven FLF features are extracted, one spatial feature and six timing features.

A spatial feature is an attribute capturing the physical-related properties during a touch action. The pressure size (PS) is a spatial feature capturing the approximated size of the screen area being touched during a TAP of a key. The touch pressure size raw data (as discussed in Section 4.5.1.3) is directly used as the PS feature without further manipulation.

A timing feature is an attribute capturing a time interval between two touch actions of one or more keys. Depending on how the intervals are measured, there are three types of timing features, i.e. dwell time (DT), flight time (FT) and input time (IT), and for FT, there are further four variants, i.e. FT1, FT2, FT3 and FT4. The definitions of these timing features are illustrated in Figure 4.5.

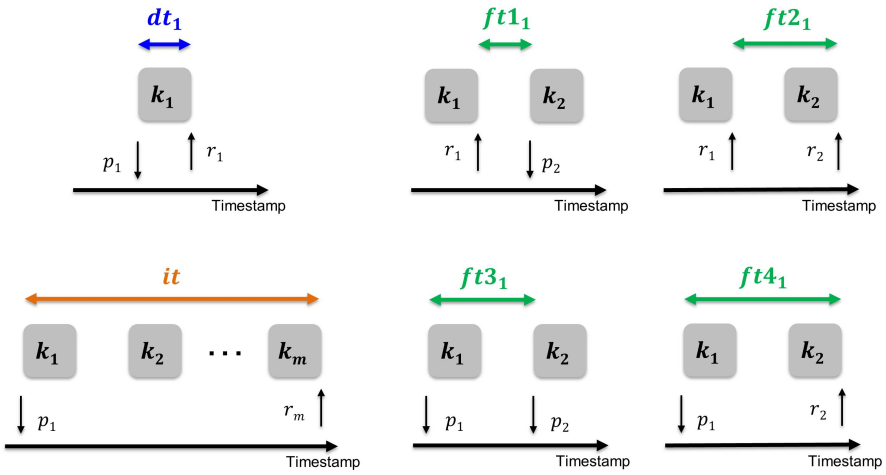

Figure 4.5: Types of timing feature

These timing features are extracted from timing raw data, i.e. timestamps. Each action is associated with a timestamp recording the time when the touch action is taking place. Timestamps do not have semantic meaning and are not readily usable as timing features. To make the timestamps usable as features, we should apply some mathematical computations on the timestamps, and the results of these computations can be used as timing features. The mathematical methods used to extract the timing features from the respective timestamps are given in Table 4.3. The variable n in the equations refers to the timing feature length (to be discussed below) used to extract timing features and m refers to the number of keys in the input string.

A timing feature can be extracted at different feature lengths. A feature length is measured in terms of the number of graphs, i.e. the number of keys involved in each measurement. It is represented in the form of n-graphs, where n denotes the number of graphs. Figure 4.6 shows the different feature lengths for a given input string. The shortest feature length is 1-graph (or uni-graph), and the subsequent feature lengths are 2-graph (or di-graph), 3-graph (or tri-graph) and so on. With 1-graph, the only timing feature that can be extracted is DT. With a larger feature length, for example, when $n \geq 2$, the timing feature from two or more keys can be extracted, i.e. FT and IT.

Table 4.4 shows the feature dimensional spaces of all the timing features that could be extracted at all possible feature lengths of a 4D string. From the table, it can be seen that the feature dimension of DT and IT features remains the

Timing Features	Descriptions	Equations
DT	The time interval between the TAP and TAR of a key.	$dt_i = r_i - p_i$
FT1	The time interval between the TAR of a key and the TAP of the next key.	$ft1_i = p_{i+(n-1)} - r_i$
FT2	The time interval between the TAR of a key and the TAR of the next key.	$ft2_i = r_{i+(n-1)} - r_i$
FT3	The time interval between the TAP of a key and the TAP of the next key.	$ft3_i = p_{i+(n-1)} - p_i$
FT4	The time interval between the TAP of a key and the TAR of the next key.	$ft4_i = r_{i+(n-1)} - p_i$
IT	The time interval between the TAP of the first key and the TAR of the last key.	$it = r_m - p_1$

Table 4.3: Descriptions and definitions of timing features

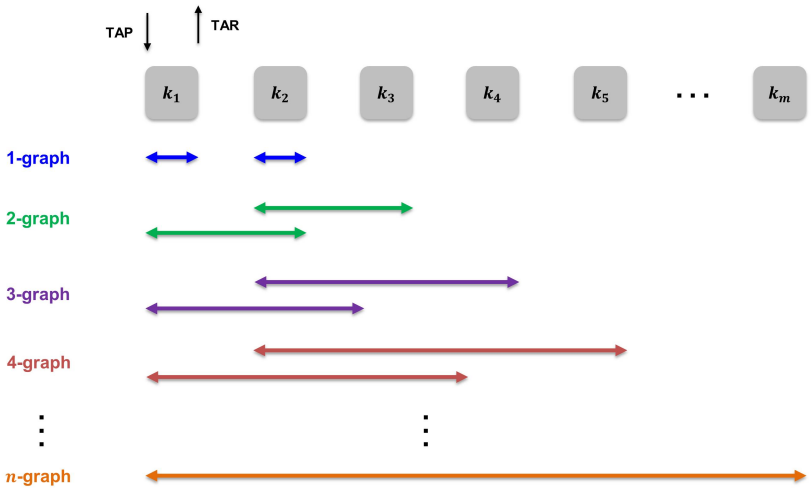

Figure 4.6: Different feature length values

Timing Features	Feature Lengths				
	1	**2**	**3**	**4**	**5**
DT	5	-	-	-	-
FT1	-	4	3	2	1
FT2	-	4	3	2	1
FT3	-	4	3	2	1
FT4	-	4	3	2	1
IT	-	-	-	-	1

Table 4.4: Feature dimensions of timing feature extracted at various feature lengths from a 4D string

same regardless of what feature length is used. For the case of DT, the feature dimension is always the same as the input string length (in the case of a 4D string, this is 5, four digits and one enter key). For the case of IT, the feature dimension always equals to 1 (the total time used to enter the whole of an input string). In other words, the feature dimensions of DT and IT feature are not correlated with the feature length. However, this is not the case for FT. For the case of FT, the larger the feature length, the smaller the feature dimensional space of FT features. The properties discussed above also apply to the 16D string. For the reasons explained above, we have carried out an experiment to examine the effects of different feature lengths on the accuracy performances of FT features. In this experiment, different feature lengths are used, but in our other experiments, a feature length of 2 is used to extract FT features.

Once FLF features, $f_i, i \in \{1, 2, \ldots, d\}$, are extracted from the raw data, they are organised into the form of an FLF feature vector, i.e. $v_{FLF}^T = [f_1, f_2, \ldots, f_d]$, where T indicates a particular type of feature, and d refers to the feature dimension of v_{FLF}^T. Table 4.5 gives a list of the types of feature considered in our experiments along with their corresponding feature vectors, IDs and dimensions. The variables n and m in the feature vectors refers to the timing feature length used to extract timing features and the number of keys in the input string, respectively. As shown in the table, two input string lengths are used. When all the feature vectors are formed for a subject, a cumulative FLF feature vector,

V_{FLF}, is generated. This is done by concatenating the FLF feature vectors, i.e. $V_{FLF} = [v_{FLF}^{DT}, v_{FLF}^{FT1}, \ldots, v_{FLF}^{PS}]$.

4.5.2.1.2 Second-Level Feature (SLF)

This section describes the process of extracting SLF features from FLF features, constructing a cumulative SLF feature vector and forming a feature set for the subject.

Some classifiers perform better with a larger number of features [186], so increasing the number of features used in training a classifier to build an authentication model can improve the accuracy performance of the model. For this reason, we extract a new category of features, known as the second-level features (SLF), from FLF features, and use both of them in training a classifier. As discussed in the previous section, FLF features extracted from raw touch dynamics data are organised into FLF feature vectors. For each of these vectors, a set of SLF features is extracted. The set consists of 19 features, and each feature represents a descriptive statistics metric of the FLF feature vector concerned. Descriptive statistics metrics are used to quantitatively summarise or describe a collection of data in a meaningful way [197]. Table 4.6 lists the 19 metrics used in our experiments. For each metric, the table also gives the metric IDs, descriptions and equations used to extract the metric. The variable v in the equations refers to a FLF feature vector and d refers to the feature dimension of v.

Metrics	**IDs**	**Descriptions**	**Equations**
Minimum	mn	The smallest value in v.	$MIN(v) = arg\ min(v)$
Maximum	mx	The largest value in v.	$MAX(v) = arg\ max(v)$
Arithmetic Mean	am	The average value of v.	$\mu(v) = \frac{1}{d}\sum_{i=1}^{d} v_i$
Quadratic Mean	qm	The average value of v giving higher weights to the larger values.	$\mu_Q(v) = \sqrt{\frac{1}{d}\sum_{i=1}^{d} \lvert v_i\rvert^2}$

Continued on next page

Table 4.6 – *Continued from previous page*

Metrics	**IDs**	**Descriptions**	**Equations**
Harmonic Mean	hm	The average value of v emphasising on the ratio between the values.	$\mu_H(v) = \frac{d}{\sum_{i=1}^{d} \frac{1}{v_i}}$
Geometric Mean	gm	The average value of v by using the product of their values.	$\mu_G(v) = (\prod_{i=1}^{d} v_i)^{\frac{1}{d}}$
Median	md	The value at the centre of a sorted v.	$MD(v) = \begin{cases} v_{(d+1)\div 2} & \text{if } d \text{ is odd} \\ \frac{v_{d\div 2} + v_{d\div 2+1}}{2} & \text{if } d \text{ is even} \end{cases}$
Range	rg	The difference between the maximum and the minimum of v.	$R(v) = MAX(v) - MIN(v)$
Variance	vr	The spread between the values in v and the mean expressed as the squared units of the mean.	$VAR(v) = \frac{1}{d} \sum_{i=1}^{d} (v_i - \mu(v))^2$
Standard Deviation	sd	The spread between the values in v and the mean expressed as the same units as the mean.	$\sigma(v) = \sqrt{\frac{1}{d} \sum_{i=1}^{d} (v_i - \mu(v))^2}$

Continued on next page

Table 4.6 – *Continued from previous page*

Metrics	**IDs**	**Descriptions**	**Equations**
Skewness	sk	This value represents how asymmetry the values of v are around the mean.	$SKEW(v) = \frac{\frac{1}{d}\sum_{i=1}^{d}(v_i-\mu(v))^3}{\sigma(v)^3}$
Kurtosis	ku	The degree of outlier-prone of the distribution of the values in v.	$KURT(v) = \frac{\frac{1}{d}\sum_{i=1}^{d}(v_i-\mu(v))^4}{\sigma(v)^4}$
First Quartile	fq	The median of the upper half values of v.	$Q_1(v) = \{\frac{d+1}{4}\}^{th}$ value in v
Third Quartile	tq	The median of the lower half values of v.	$Q_3(v) = \{\frac{3(d+1)}{4}\}^{th}$ value in v
Interquartile Range	ir	The difference between the upper quartile and the lower quartile values of v.	$IQR(v) = Q_3(v) - Q_1(v)$
Mean Absolute Deviation	ma	The average absolute difference between the values in v and the mean.	$MAD(v) = \frac{1}{d}\sum_{i=1}^{d}\|v_i - \mu(v)\|$
Median Absolute Deviation	mi	The median of the absolute difference between the values in v and the median.	$MID(v) = MD(\|v_i - MD(v)\|)$
Coefficient of Variation	cv	The relative variation between the values in v and the mean.	$CV(v) = \frac{\sigma(v)}{\mu(v)}$

Continued on next page

Table 4.6 – *Continued from previous page*

Metrics	IDs	Descriptions	Equations
Standard Error of Mean	se	This value measures how precisely the sample mean of the values in v estimates the population mean.	$SEM(v) = \frac{\sigma(v)}{\sqrt{d}}$

Table 4.6: List of descriptive statistics metrics used to extract SLF features from FLF features

Similar to the case for FLF features, SLF features, $f_i, i \in \{mn, mx, \ldots, se\}$, once extracted from FLF features, are organised into the form of a SLF feature vector, i.e. $v^T_{SLF} = [f_{mn}, f_{mx}, \ldots, f_{se}]$, where T indicates a particular type of FLF feature. When all the SLF feature vectors are formed for a subject, a cumulative SLF feature vector, V_{SLF}, is generated. This is done by concatenating the SLF feature vectors, i.e. $V_{SLF} = [v^{DT}_{SLF}, v^{FT1}_{SLF}, \ldots, v^{PS}_{SLF}]$. The cumulative SLF feature vector has a 114-dimensional feature space (6 types of FLF features multiplied by 19 descriptive statistics metrics). Unlike the FLF features, the feature dimension of SLF features is independent of the input string length used. In other words, the numbers of SLF features for both the 4D string and the 16D string are the same. Note that not all the types of FLF features are used to extract the SLF features. IT is not used, as the feature only has one value, and by using descriptive statistics metrics, no meaningful information could be extracted from this feature.

Once both FLF and SLF cumulative feature vectors are formed for a subject, they are combined into a form of a feature set, i.e. $\tau = \{V_{FLF}, V_{SLF}\}$. A feature set consists of both the FLF and SLF features extracted from the raw touch dynamics data of one input sample of a subject.

4.5.2.2 Feature Normalisation

If different features have different value ranges, the values of the features should be normalised. This is because, when features have different value ranges, they may have unbalanced weights when representing the structure of the data. For example, features with a higher value range may have a higher weight when

Types of Feature	Feature Vectors	Input String Lengths			
		4D		16D	
		Feature IDs	Feature Dimensions	Feature IDs	Feature Dimensions
DT	$v_{FLF}^{DT} = [dt_1, dt_2, \ldots, dt_m]$	1-5	5	1-17	17
FT1	$v_{FLF}^{FT1} = [ft1_1, ft1_2, \ldots, ft1_{m-(n-1)}]$	6-9	4	18-33	16
FT2	$v_{FLF}^{FT2} = [ft2_1, ft2_2, \ldots, ft2_{m-(n-1)}]$	10-13	4	24-49	16
FT3	$v_{FLF}^{FT3} = [ft3_1, ft3_2, \ldots, ft3_{m-(n-1)}]$	14-17	4	20-65	16
FT4	$v_{FLF}^{FT4} = [ft4_1, ft4_2, \ldots, ft4_{m-(n-1)}]$	18-21	4	66-81	16
IT	$v_{FLF}^{IT} = [it_1]$	22	1	82	1
PS	$v_{FLF}^{PS} = [ps_1, ps_2, \ldots, ps_m]$	23-27	5	83-99	17

Table 4.5: Feature vectors, IDs and dimensions of different types of feature

representing the structure of the data. Also, some algorithms perform better and faster if the features have the same value range [198]. For these reasons, the values of different features should be normalised to the same value range; this is usually done by using a process called feature normalisation.

In our experiments, we have conducted a feature normalisation process, and in this process, we have used a method called the min-max normalisation [199]. With this method, the values of the features are scaled so that their ranges are confined to a predefined lower and upper boundary. Let X denotes a dataset of feature sets, $\tau_{ij}, i \in \{1, 2, \ldots, a\}, j \in \{1, 2, \ldots, d\}$, represented in the form of an $a - by - d$ matrix, where a refers the number of feature sets in the dataset and d refers to the feature dimensions of the feature sets. The matrix is represented as:

$$X = \begin{matrix} \tau_{11} & \cdots & \tau_{1d} \\ \vdots & \ddots & \vdots \\ \tau_{a1} & \cdots & \tau_{ad} \end{matrix} \tag{4.2}$$

Then, the normalised dataset, $\hat{X}$, is obtained by using the following equation:

$$\hat{X} = \frac{X_{ij} - min(X_j)}{max(X_j) - min(X_j)} \times (u - l) + l \tag{4.3}$$

where l and u are, respectively, the lower and upper boundaries of the normalised features. In our experiments, the value for l is set to 0, and u is set to 1.

4.5.2.3 Feature Selection

Feature selection ensures an optimal set of features is used when training an authentication model. It may improve or maximise the accuracy performance of the model, while, at the same time, keep the cost incurred in training the model as low as possible. Feature selection involves two tasks, the selection of a feature selection method, and the implementation of a feature selection process using the selected method. The following sections describe the two tasks in detail. Before describing these tasks, we first give an overview of the feature selection process.

Typically, there are two sets of features, a preliminary feature set (PFS) and an optimal feature subset (OFS). PFS is a set of features that are directly extracted from raw touch dynamics data. OFS is a subset of features that are selected

from the features in the PFS set. Normally, different features may be different in terms of their importance in representing the structure of the data. Some may be more important or more relevant than others, while others may have similar importance or relevance, and in the latter case they are regarded as redundant. The use of redundant features increases the feature dimension of a dataset, which may lead to three problems: (i) a higher computational complexity thus a higher cost in training the model; (ii) a higher risk of overfitting the model, leading to a lower level of accuracy performance of the model [200]; (iii) a larger storage space for storing the features. To avoid these problems, feature selection is used to select the most relevant features and to remove redundant features from a PFS set. The resulting smaller set of features, i.e. an OFS set, is then used to train an authentication model.

4.5.2.3.1 Feature Selection Method

As mentioned earlier, a feature selection method is used to analyse the features in a PFS set and to select an OFS set from the PFS set. The selected OFS set should satisfy the following two criteria: (i) the subset should contain the most relevant features, and (ii) the subset should have the least number of redundant features. In other words, the features in the OFS set should have maximum relevance to the target variable (in our work, this is the identity of the subject), and, at the same time, have minimum redundancy amongst the features in the subset. To obtain an OFS set, we need a method to measure how well the features in the PFS set satisfy both criteria. In our work, we have chosen to use the minimum-redundancy-maximum-relevance (mRMR) method [201].

The mRMR method adopts the concept of mutual information [202], a variable dependence measure commonly used in the field of information theory. Mutual information measures the amount of information shared between any arbitrary pair of discrete random variables, x and y, and is defined as follows:

$$MI(x, y) = \sum_{i,j} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)p(y_j)} \tag{4.4}$$

where $p(x, y)$ is the joint probability distribution function of x and y, $p(x)$ and $p(y)$ are the marginal probability distribution functions for x and y, respectively.

The mRMR method quantifies how well a feature, f_i, satisfies the two criteria

above, and this is done by using a scoring metric, G. This metric is defined as:

$$G(f_i) = L(f_i) - D(f_i) \tag{4.5}$$

where the first term, $L(f_i)$, measures the degree of relevance between f_i and the identity of the subject, and the second term, $D(f_i)$, measures the degree of redundancy between f_i and the other features in the PFS set. The higher the value of G, the better f_i is at satisfying both criteria.

$L(f_i)$ is calculated by using the equation below:

$$L(f_i) = MI(f_i, b) \tag{4.6}$$

where $MI(f_i, b)$ represents the mutual information in common between f_i and the identity of the subject, b. The higher the value of L, the greater the amount of information that is shared between f_i and b, so the more relevant the feature. In other words, the higher the value of L, the better.

$D(f_i)$ is calculated by using the equation below:

$$D(f_i) = \frac{\sum_{j \in F} MI(f_i, f_j)}{|F|} \tag{4.7}$$

where $MI(f_i, f_j)$ is the pairwise mutual information shared between f_i and another feature, f_j, in a PFS set, F. $|F|$ refers to the total number of features in F. The lower the value of D, the less the amount of information that is shared between f_i and the other features in F, so the less redundant the feature. In other words, the lower the value of D, the better.

4.5.2.3.2 Feature Selection Process

The feature selection process is based on the mRMR method described above. The mRMR method is implemented by using the FEAST toolbox (version 1.1.4) [203]. The process consists of the following four steps.

Step 1: The feature values in the PFS set are converted to discrete values using the histogram bin counts method [204].

Step 2: Starting with the first feature, G is calculated and assigned to the feature

using equation 4.5. This step is repeated for each of the remaining features in the PFS set.

Step 3: Based on the value of G, the features are sorted into a ranked list. The features with higher scores are ranked higher in the list.

Step 4: Finally, the features ranked at the top-z% in the list are selected as the OFS set, where z refers to the feature selection size.

Table 4.7 shows the top-10 performing features, where two different input string lengths are considered. The PFS sets for both input string lengths contain all the FLF and SLF features extracted from the raw touch dynamics data. From the table, it can be seen that, even if the features are the same, the scores of the features are not the same for different input string lengths. Take PS (md) for example, the score is 1.3668 for the 4D string, but 1.4598 for the 16D string. This means that, given the same set of PFS, features may be ranked differently when different input string lengths are used. Therefore, the resulting OFS set may be different for different input string lengths. This indicates that, for a better performance (i.e. higher accuracy and/or lower costs), it may be necessary to select different sets of OFS for different input string lengths. For this reason, we have performed a feature selection process for each of the input string lengths, resulting with two different sets of OFS.

In our study carried out to examine the effects of feature selection sizes on the accuracy performances, we have used different feature selection sizes. However, in the rest of our studies, we have used a feature selection size of 20. Table 4.8 shows the respective OFS sets for the two input string lengths, where the feature selection sizes used for both cases are set to 20.

4.5.3 Model Training Unit (MTU)

MTU analyses the touch dynamics features extracted by FCU and uses them to train an authentication model. The trained model should uniquely represent the corresponding subject's touch dynamics pattern. Model training is carried out by using a process called feature classification. The input of this process is the extracted OFS features, the algorithm used is a classifier, and the output is an authentication model. Feature classification involves two tasks, the selection of the classifier and the implementation of the selected classifier. In the following sections, we discuss these tasks in detail.

Ranks	Input String Lengths			
	4D		16D	
	Features	**Scores**	**Features**	**Scores**
1	PS (qm)	2.0155	PS (qm)	2.1576
2	FT4 (qm)	1.7608	FT4 (hm)	1.7390
3	FT3 (mn)	1.4850	DT (gm)	1.5276
4	PS (27)	1.4835	PS (tq)	1.5177
5	FT4 (20)	1.4274	FT4 (71)	1.5120
6	PS (mx)	1.3831	FT3 (cv)	1.4633
7	FT4 (19)	1.3752	PS (md)	1.4598
8	PS (md)	1.3668	FT3 (26)	1.4447
9	FT1 (cv)	1.3641	PS (mx)	1.4381
10	DT (am)	1.3638	FT4 (mn)	1.4333

Table 4.7: Ranks and scores of the top-10 performing features in the PFS set of two input string lengths (with feature IDs (refer Table 4.5 and Table 4.6) given in brackets)

Input String Lengths	OFS Features
4D	DT(am,tq), FT1(hm,cv), FT2(12,mn), FT3(16,mn,fq), FT4(18,19,20,am,qm,md), PS(24,25,26,27,mx,am,qm,hm,gm,md,fq,tq,cv)
16D	DT(hm,gm,md), FT1(mn,am,fq), FT2(29,30,cv), FT3(25,26,mn,gm,cv), FT4(70,71,mn,hm,gm,md,fq,tq), PS(83,86,89,90,92,94,95,96,97,99,mn,mx,am,qm,hm,gm,md,fq,tq,ma)

Table 4.8: OFS sets for different input string lengths (with feature IDs (refer Table 4.5 and Table 4.6) given in brackets)

4.5.3.1 Classifier Selections

Depending on the data used, classifiers can be classified into two groups, one-class classifier (OCC) and two-class classifier (TCC). An OCC classifier only uses data from a single class (e.g. data from a legitimate subject). Unlike an OCC classifier, a TCC classifier uses data from two classes (e.g. data from both legitimate and illegitimate subjects).

In the mobile device context, obtaining two classes of data with a similar size is not practical. This is due to the fact that a mobile device is rarely shared among multiple users. Also, sharing a passcode with others increases data privacy risks and is not a recommended practice. For these reasons, only the data from the legitimate subject are used to train a classifier. A TCC classifier may not perform well given one class of data, as it depends on similar sized data from two classes to train a model that separates the two classes apart [205]. In addition, the time taken by a TCC classifier to train a model is longer than that by an OCC classifier, as the former uses more data in training the model. Based on these considerations, we have chosen to use OCC classifier for feature classification, although, we have, for the purpose of our comparative study, implemented both types of classifiers. In other words, for feature classification, we have implemented two OCC classifiers, the one-class k-nearest neighbour (OCKNN) [206] and the support vector data description (SVDD) [207]. For our comparative study, i.e. the study we have carried out to examine the effectiveness of using OCC classifiers versus using TCC classifiers, we have also implemented two TCC classifiers, the k-nearest neighbour (KNN) and the support vector machine (SVM) [208]. The implementation of these classifiers is described in the following section.

4.5.3.2 Classifier Implementations

The implementation of classifiers is a complex task due to the complexity in selecting their parameter values. Most classifiers have one or more parameters each [209]. The values assigned to these parameters control or influence the way the classifier works, as well as the outcome of the classification. In other words, for a given classifier, if the values of the parameters are changed, the way the classifier works will also change, and, as a result, the accuracy performance of the model will also change. In this section, we describe the classifier implementation details, including the selection of parameter values for each classifier. The parameter values will be useful for others to reproduce our experiments for comparative

studies.

The classifiers are implemented using the Matlab programming platform (version 8.5.0.197613) and two open source toolboxes, the dd_tools [210] and the PRTools [211]; the dd_tools is used to implement the OCC classifiers and the PRTools is used to implement the TCC classifiers. The implementation of the classifiers involves two phases, training (or enrolment) phase and testing (or verification) phase. In the training phase, training samples are used to train a classifier to build an authentication model, and the built model is stored in DSU. In the testing phase, a testing sample is compared against the stored model to generate a classification score. This score will then be used to make an authentication decision. In the following, we describe the implementations of the classifiers with the focus on the two OCC classifiers. The approaches used to implement the two TCC classifiers are similar to their corresponding OCC classifiers, so we will not discuss it further.

OCKNN Classifier

The assumption for the OCKNN classifier is that the testing samples will have characteristics similar to at least one or more of those training samples used to build the model [206]. In the training phase, the classifier computes the distance matrix of the training samples using the nearest-neighbour parameter, k. Based on our empirical studies, we set k=1. In the testing phase, the classifier first finds k training samples that have their distance closest to the testing sample and then computes the average squared Euclidean distance between each of the k samples and the testing sample. The result of the computation is the classification score.

SVDD Classifier

With the SVDD classifier, an authentication model is built by fitting a boundary around the training samples, and any testing sample that falls outside of this boundary is regarded as an illegitimate sample. The search for this boundary can be performed within a two-dimensional feature space using a linear kernel function. However, distinguishing touch dynamics samples of a legitimate subject from those of an illegitimate subject is nonlinear [126]. When samples are not separable by a linear boundary, the linear kernel function may not perform well, as the boundaries created by the function is too simple to optimally split both classes apart. To create more complex boundaries that can optimally split both

classes, a nonlinear kernel function such as Radial Basis Function (RBF) can be used. The RBF kernel enables the classifier to project samples onto a higher-dimensional feature space, and, as a result, it can more accurately determine which side of the feature space a testing sample belongs to.

In the training phase, the classifier first uses the RBF kernel to project the training samples onto a high-dimensional feature space, and then tries to fit a closed hypersphere boundary around the training samples. The width parameter of the RBF kernel, σ, is set to 5, and the regularisation parameter, c, is set to 0 so that the boundary contains all the training samples. In the testing phase, the classifier first projects the testing sample onto the same high-dimensional feature space as the training samples, and then computes the distance between the testing sample and the boundary around the training samples. The result of the computation is the classification score.

4.5.4 Authentication Decision-Making Unit (ADMU)

ADMU makes an authentication decision determining whether a testing sample matches with the authentication model of the owner of the device. The design of ADMU involves two processes: feature matching and feature thresholding. In the feature matching process, the testing sample acquired from an authentication attempt, $\hat{\tau}$, is matched against the stored model in DSU to obtain a classification score, δ. In the thresholding process, δ is compared to a predefined threshold, θ_{auth}, to make an authentication decision, D_{auth}, based on the following rules:

$$D_{auth} = \begin{cases} + & \text{if } \delta > \theta_{auth} \\ - & \text{otherwise} \end{cases} \tag{4.8}$$

If δ is over θ_{auth}, then D_{auth} is positive, and in this case, $\hat{\tau}$ is classified as legitimate, $\hat{\tau}_L$ (i.e. belonging to the owner of the device). Otherwise, $\hat{\tau}$ is classified as illegitimate, $\hat{\tau}_I$ (i.e. belonging to an impersonator), and D_{auth} is negative.

4.6 Evaluation Methodology

This section describes how the accuracy performance of ToDiTA is evaluated. It covers the evaluation method used and the implementation of the method.

4.6.1 Method Overview

To perform the accuracy performance evaluation of ToDiTA, the following four tasks are performed. Firstly, subjects are classified into two groups, one designated as legitimate subjects and the other as illegitimate subjects. Secondly, some of the touch dynamics samples acquired from these subjects are used as training samples that are used by MTU to train authentication models. Thirdly, some of the other samples are used as testing samples that, along with the trained models, are used by ADMU to make authentication decisions. Lastly, based on the decisions, the evaluation metrics values which are calculated indicate the accuracy performance of the model.

Four sets of samples are acquired: i) legitimate training set, ii) legitimate testing set, iii) illegitimate training set, and iv) illegitimate testing set. To acquire the first two sets, we split the samples of each subject into two subsets, one as legitimate training set and the other as legitimate testing set. To acquire the third set, we assign the samples of each subject that are not used as testing samples as the illegitimate training samples for all other subjects.

With regard to the acquisition of the fourth set of samples, i.e. illegitimate testing samples, a scenario called a zero-effort attack [212] is adapted. In this attack scenario, an impersonator attempts to compromise a biometrics authentication system by submitting his/her own biometrics samples without any knowledge of other subjects' biometrics features stored in the system. In other words, the impersonator attempts to perform a successful verification against other subjects' profiles using his/her own profile. Using this scenario, we assign the samples of each subject that are not used as training samples as the illegitimate testing samples for all other subjects. One of the advantages of this approach is that the samples acquired from different subjects can be reused for evaluation purposes (not just for use in model training), and in this way, we can acquire a larger number of illegitimate testing samples. This approach has been commonly used in previous studies [106, 139, 213].

Once the four sets of samples are acquired, they are used to evaluate the accuracy performance of the ToDiTA system. The samples in the legitimate and illegitimate training sets are used by MTU to train classifiers to build authentication models. The samples in the legitimate and illegitimate testing sets along with the generated models are used by ADMU to make authentication decisions. The decisions are made by classifying each testing sample as either belonging to

the legitimate or illegitimate subject. The decisions are then compared against the actual classes of the testing samples to formulate a false acceptance count and a false rejection count. If a legitimate testing sample is incorrectly classified as illegitimate, the false rejection count is incremented. If an illegitimate testing sample is incorrectly classified as legitimate, the false acceptance count is incremented. The false acceptance and false rejection counts are used to calculate the evaluation metrics values, i.e. FAR, FRR and EER (as described in Section 3.2.5).

4.6.2 Method Implementation

The evaluation method implementation is summarised in Algorithm 4.1 and is described below. In the description, it is assumed that the touch dynamics samples acquired from a set of subjects, B, are stored in a dataset, S.

Step 1: One of the subjects in B is assumed and assigned as a legitimate subject, b. The remaining subjects in B are regarded as illegitimate subjects.

Step 2: The samples in S are split into two subsets, S^+ and S^-. S^+ contains the samples of b, and S^- contains the samples of the illegitimate subjects.

Step 3: S^+ and S^- are each randomly split into K equal-sized folds. The value of K can be set to any integer value greater than 2. In practice, a value of 5 or 10 is typically used [214, 215]. In our evaluations, we set K=10, also known as the 10-fold cross validation.

Step 4: Here, using the samples in the folds of both subsets, a testing set, T_{ts}, and a training set, T_{tr}, are established, where T_{ts} contains the samples in the k^{th} fold, and T_{tr} contains the samples of the remaining folds.

Step 5: If the classifier, C, is an OCC classifier, the training samples of the illegitimate subjects are removed from T_{tr} and added to T_{ts}, as C only trains on the samples of b. Otherwise, the samples in both sets remain unchanged.

Step 6: T_{tr} is used to train C to build a model, M_b, for b.

Step 7: M_b is evaluated on T_{ts}, and the accuracy performance of M_b (for the k^{th} fold) is calculated and recorded as P_{fd}.

Step 8: Step 4 to 7 is repeated for each of the remaining folds in the K folds.

Step 9: The average accuracy performance of the model for b (over all folds), P_{sb}, is calculated and recorded by dividing P_{fd} by K.

Step 10: Step 1-9 is repeated, assigning each of the other subjects in B as b in turn.

Step 11: The overall accuracy performance of the model (over all subjects), P, is calculated by dividing P_{sb} by B.

Algorithm 4.1: Evaluation method for ToDiTA

Input: Dataset S with B number of subjects, $\{d_1, d_2, \ldots, d_B\}$, classifier C, folds K

Output: Accuracy performance of the model P

```
for b = 1 to B do
    S+ ← initialise the legitimate subject samples {d_b}
    S− ← initialise the illegitimate subjects samples S − {d_b}
    Randomly split S+ and S− into K disjoint folds, {s_1^+, s_2^+, ..., s_K^+} and
     {s_1^−, s_2^−, ..., s_K^−}
    for k = 1 to K do
        S_tr^+ ← initialise the legitimate training samples, S+ − {s_k^+}
        S_tr^− ← initialise the illegitimate training samples, S− − {s_k^−}
        T_tr ← initialise the training set, S_tr^+ + S_tr^−
        T_ts ← initialise the testing set, {s_k^+} + {s_k^−}
        if C is an OCC classifier then
            T_tr ← T_tr − S_tr^−
            T_ts ← T_ts + S_tr^−
        end
        Train C on T_tr to build model M_b
        P_fd ← P_fd+ (test the accuracy performance of M_b on T_ts)
    end
    P_sb ← P_sb + P_fd/K
end
P ← P_sb/B
```

4.7 Evaluation Results and Discussions

This section describes the experiments carried out to evaluate the performance of the ToDiTA system and discusses the evaluation results obtained. The results are presented in the following order, the evaluation of, RDAU (Section 4.7.1), FCU (Section 4.7.2), MTU (Section 4.7.3) and the authentication architecture as a whole (Section 4.7.4). The experiments were conducted using the evaluation methodology described in Section 4.6. Unless otherwise stated, each experiment was repeated four times, each time using one of the four classifiers (as discussed in Section 4.5.3.1) in turn, and the results reported were the average of the classifiers.

4.7.1 RDAU Evaluation

The evaluation of RDAU is carried out with different value settings for two parameters, the scaling factor and the input string lengths.

4.7.1.1 Scaling Factor

The scaling factor is used to set the timestamps to different precision levels (as discussed in Section 4.5.1.4). Choosing different scaling factor values may affect the accuracy performances and the storage requirements of the timing features, as the features are extracted from the timestamps. To investigate the effect of using different scaling factor values, we have evaluated the accuracy performances and the storage requirements of timing features by setting the scaling factor to be a range of values from 0 (no scaling) to 8 (maximum scaling) with an increment of 1. For each scaling factor value, we have extracted all the timing related FLF features from both the 4D string and the 16D string as the test cases.

Figure 4.7 shows the EER values versus different scaling factor values, where two different input string lengths are considered. As can be seen from the figure, for both input string lengths, the EER values stay flat before a threshold value is reached. For the 4D string, this threshold value is 5, and for the 16D string, it is 6. Beyond the threshold values, the EER values increase steadily for the 4D string and sharply for the 16D string. These observations can be explained as follows. When timestamps are scaled using a smaller to a medium scaling factor, the timestamps have a proper precision to capture a human's touch action speed. As a result, the timing features extracted from the timestamps contain sufficient information to properly capture a human's touch dynamics pattern, resulting in

the stable EER values as observed. However, when using a very large scaling factor (7 or 8), the timestamps become smaller. As a result, the timing features extracted from the timestamps contain less information, leading to an increase in the EER values. These observations indicate that, by scaling timestamps, the accuracy performance of the timing features extracted from the timestamps cannot be improved, but the use of an inadequate scaling factor value can worsen the accuracy performance.

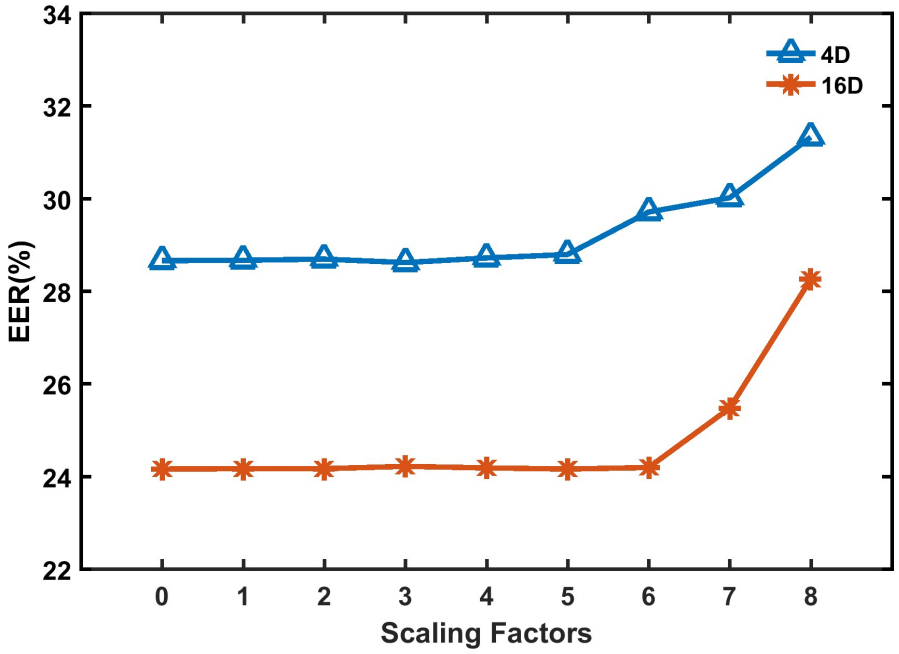

Figure 4.7: EER values versus different scaling factor values

It should also be emphasised that the use of different scaling factor values may have an effect on the amount of storage space required to store the timing features. To investigate this effect further, we have recorded and compared the storage spaces used by the timing features extracted from timestamps with different scaling factor values. The results showed that the larger the scaling factor value, the less the amount of storage space that is used to store the timing features, though the level of reduction is small. For example, when the scaling factor value is set to a value between 0 and 5, the storage space used is reduced from 0.217KB to 0.110KB, and the reduction is only 0.107KB. This level of reduction can be regarded as negligible, especially considering the fact that recently released mobile devices typically have a storage space of at least 8GB to 32GB [216], and

the space is expected to increase in newer generation devices [217].

Based on the above analyses, we can conclude that scaling timestamps does not bring many benefits. On the contrary, the scaling process introduces additional computational overhead. For these reasons, we do not scale timestamps in our design. For some applications, for example, where the primary requirement is to minimise storage space, we recommend setting the scaling factor value to 5 (and not beyond).

4.7.1.2 Input String Lengths

Using different input string lengths may also affect EER values. To examine the effect, we used two input strings with two different lengths, a 4D string and a 16D string. For both input strings, all the FLF and SLF features were extracted. For each input string, we repeated the experiment four times, and for each time, one of the four classifiers was used.

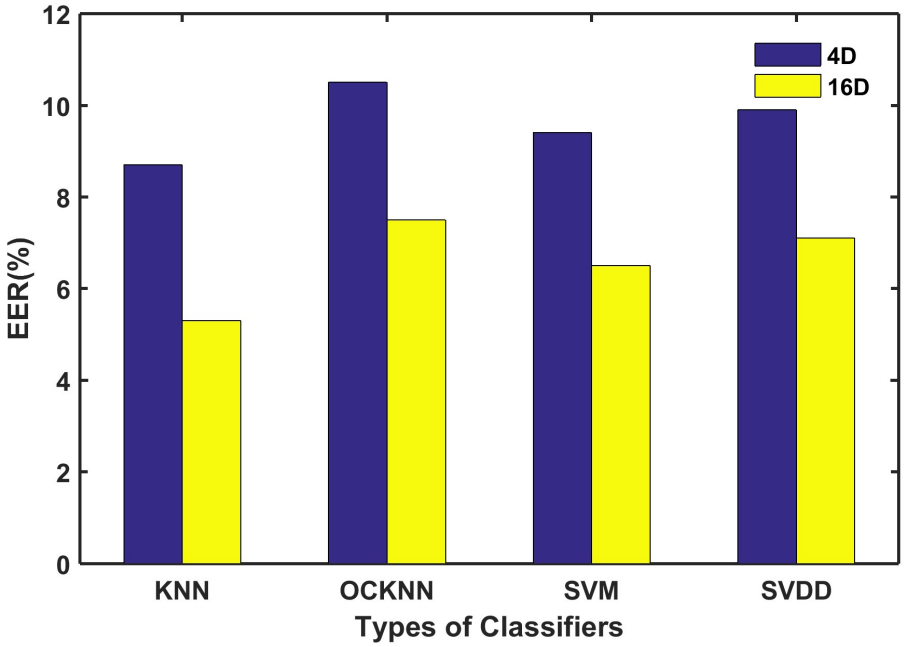

Figure 4.8: EER values versus two different input string lengths and four classifiers

Figure 4.8 shows the EER values versus two input strings and four different classifiers. As shown in the figure, for all the classifiers, using the 16D string introduces a lower EER value, indicating that the longer the input string length,

the more accurate the authentication model. The reasons for this are three-fold. Firstly, the length of the 16D string is four times longer than that of the 4D string, and so is the number of features that are extracted from the 16D string. More features mean more information about a subject's touch dynamics pattern can be captured. Therefore a more accurate model can be built out of the features. Secondly, when the input string length increases, the number of possible chunk combinations also increases (as shown in Figure 4.9), and so is the ability to better capture a subject's touch dynamics pattern. Finally, when the input string length increases, the number of illegitimate features required to match that of a legitimate subject's model will also increase, which means that the level of difficulty in impersonating a legitimate subject successfully also increases.

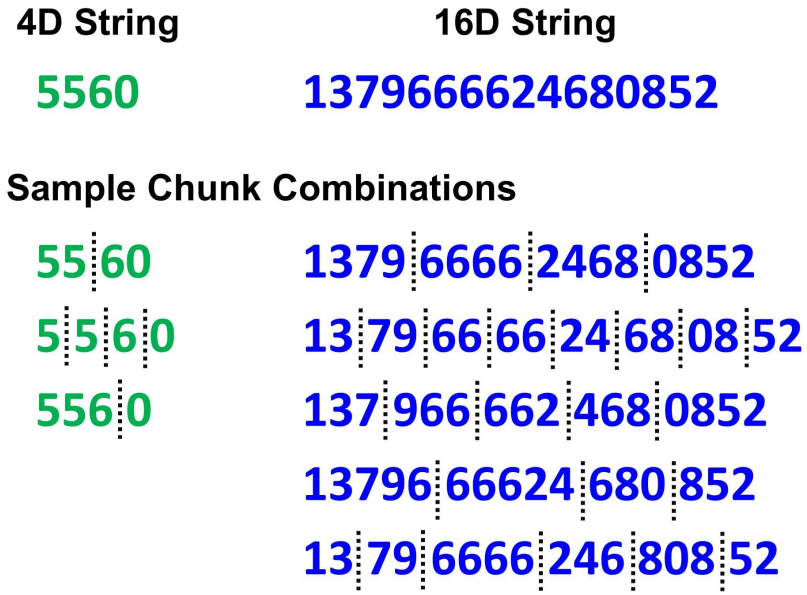

Figure 4.9: Possible chunk combinations of different input string lengths

The above results have revealed a correlation between the input string length and security. The shorter the input string length, the lower the level of authentication accuracy, indicating a lower level of security. There is also a correlation between the input string length and usability. The longer the input string, the more the number of touch actions are required to complete the input of the string, thus the harder and slower it is for the users to memorise the string, indicating a lower level of usability. A similar correlation has also been reported in [218]. In summary, the input string length influences the trade-off between security and

usability. Therefore, in real-life applications, it should be chosen based on the security and usability requirements of the apps.

4.7.2 FCU Evaluation

This section presents the evaluation of FCU with different parameter value settings, i.e. FLF features, FLF feature combinations, SLF versus FLF features, timing feature lengths, feature normalisation, and feature selection.

4.7.2.1 FLF Features

There are four types of FLF (as discussed in Section 4.5.2.1.1). Each type of FLF captures a subject's touch dynamics pattern in a different way. Some are better than the others at representing the intrinsic properties or characteristics of a subject's touch dynamics pattern, and thus achieve a higher level of accuracy. To investigate the accuracy performance of the authentication system using different types of FLF, we have extracted all four types of FLF from the 4D string as the test case.

Figure 4.10 shows the EER values of the four types of FLF. From the figure it can be seen that the EER value of PS is the lowest among the four types, and this also means that the accuracy performance of PS is better than the other (timing) features. This result can be explained as follows. The PS values are determined by several factors such as: i) the physical size of the fingertip used to perform a TAP; ii) the amount of force exerted during a TAP; iii) the fingertip position or angle during a TAP; and iv) the speed of a TAP. The combination of these factors creates a distinctive pattern, which allows PS to better capture each subject's touch dynamics pattern, and, as a result, achieves a higher level of accuracy.

With regards to the timing features, FT achieves the best accuracy performance in comparison with IT and DT. There are two reasons for this. Firstly, FT has a significantly larger feature dimensional space than IT and DT, and, as a result, more features are available for use in building the model that could better capture the subject's touch dynamics pattern. Secondly, FT has more control information and more discriminative properties than IT and DT, which may have originated from two sources. The first is the information embedded within the natural short pauses between different chunks. The second is the variability of

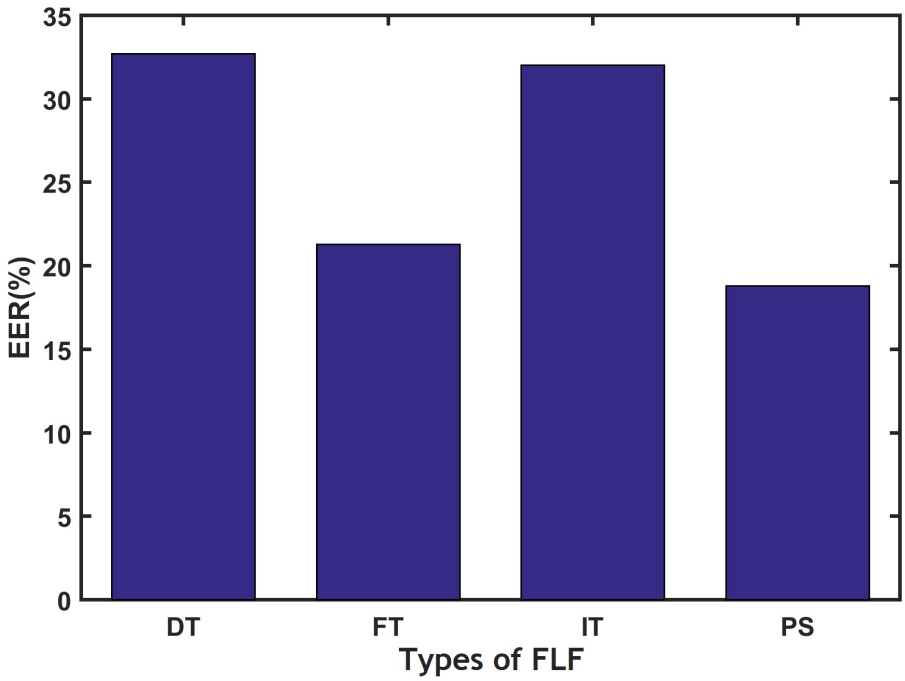

Figure 4.10: EER values for different types of FLF

chunk combinations. These can enable the classifiers to build models that better distinguish touch dynamics patterns from one subject to another. For these reasons, FT achieves a higher level of accuracy than DT.

A better way of understanding the accuracy performances achieved by different types of FLF is to visualise the feature values from different subjects graphically. Figure 4.11 shows the feature scatter plots of three types of FLF from three subjects. The subjects are randomly chosen. The x- and y-axis of each figure represents a type of FLF with the feature ID given in brackets. What is striking about the plots shown in the figure is that when PS is used (shown in Figure 4.11c), the three subjects can be clearly distinguished or separated. However, this is not the case for FT (shown in Figure 4.11b) and DT (shown in Figure 4.11a). These observations are consistent with our discussions given above, i.e. PS achieves the best accuracy performance, which is followed by FT and, then, by DT.

4.7.2.2 FLF Feature Combinations

The results in Section 4.7.2.1 show that some types of FLF perform better than others. However, this does not mean that those under-performing types of FLF

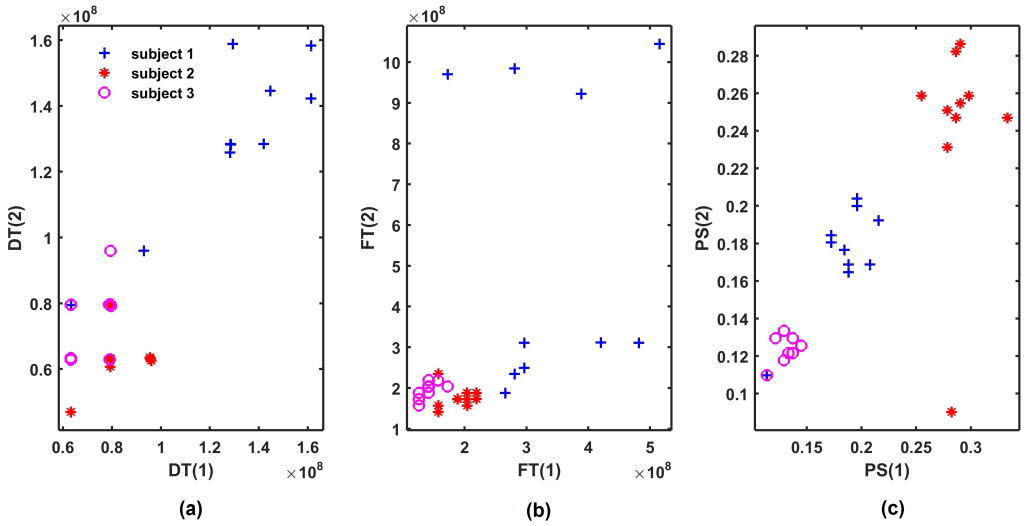

Figure 4.11: Feature scatter plots of three types of FLF: (a) DT, (b) FT and (c) PS

are not useful, as each type of FLF captures a different aspect of a subject's touch dynamics pattern. Previous studies [85,103] have shown that the accuracy performance of a model could be improved by combining multiple types of FLF. To verify this claim, we formed 15 different feature set combinations by using the four types of FLF, and grouped them into four categories, i.e. S1(4), S2(6), S3(4) and S4(1). The number in the brackets indicates the number of feature set in each category. The category number represents the number of types of FLF in each feature set of the corresponding category. Take the S3(4) category for example, there are four feature sets, and each set is formed by using three types of FLF, i.e. DT,FT,IT, DT,FT,PS, DT,IT,PS and FT,IT,PS.

Figure 4.12 shows the EER values and the feature dimensions for different feature sets. For each category, only the best-performing feature set is shown. As can be seen from the figure, the larger the category number, the lower the EER value. In other words, the more types of FLF are combined, the better the accuracy performance we may achieve. For example, when S1 is used, the EER value is the highest (18.8%). However, when S2 is used, the EER value decreases to 12.55%, which is a marked drop, and when S3 and S4 are used, the EER value drops to 11.50% and 11.45%, respectively. These results can be explained as follows. When more types of FLF are combined, more features are used in building the model, thus a more accurate model is produced. However, the accuracy performance gain between the feature set S3 and S4 is very small (only

0.5%). An explanation for this is that the features in S3 and S4 are very similar. The only difference is that S4 contains one additional feature, IT, which means that the features in S4 only capture a small amount of additional information in comparison with the features in S3, so there is only a minor improvement in the accuracy performance.

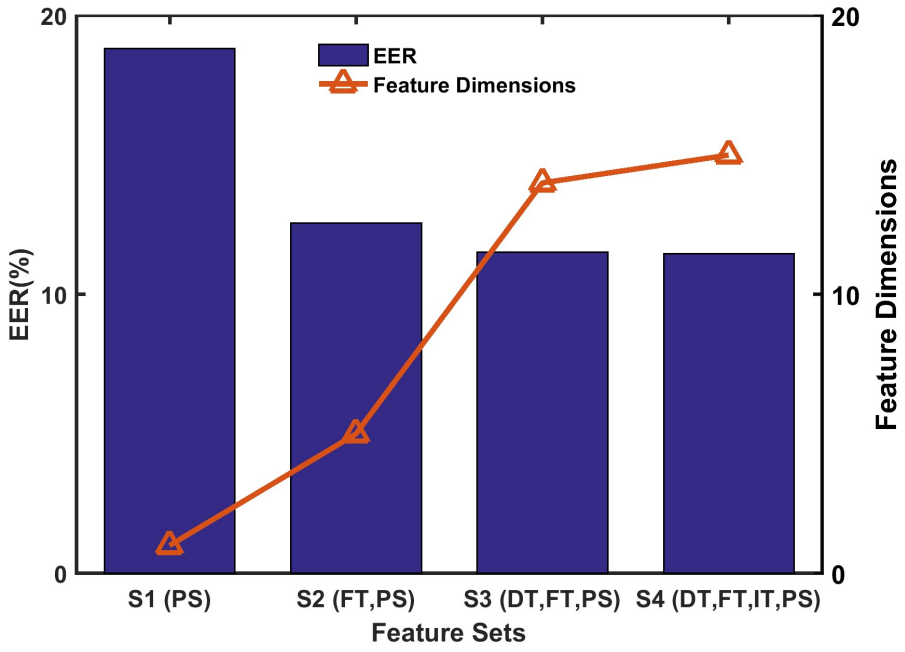

Figure 4.12: EER values and feature dimensions for different feature sets

In summary, these experimental results suggest that the accuracy performance of the model can be improved by combining different types of FLF, and the combination of low- or medium-performing types of FLF can achieve a higher level of accuracy than the best-performing type of FLF when used individually.

4.7.2.3 SLF vs FLF Features

SLF features are extracted from FLF features. To study the effectiveness of the SLF features, we have used three categories of features: i) C1, containing the FLF features; ii) C2, containing the SLF features that are extracted from the features in C1; and iii) C3, containing the features in C1 and C2. Each category consists of three feature sets formed by using DT, FT and PS, respectively. The features in the sets are extracted from the 16D string.

Figure 4.13 shows the EER values of the feature sets of three different categories. From the figure, we can see that, with the exception of C2 for FT, the EER values of C2 are higher than those of C1 in both cases of DT and PS. This may be due to that, as discussed in Section 4.5.2.1.2, the SLF features are descriptive statistics metrics, the fewer values available for use to generate the metrics, the less meaningful the metrics are at representing the trends of the values, and, as a result, achieves a lower level of accuracy. However, in the case of FT, the feature has a larger feature dimensional space than its counterparts, DT and PS, so more feature values are available when it is used to extract C2 (the SLF features) values. Therefore, we get a better accuracy performance.

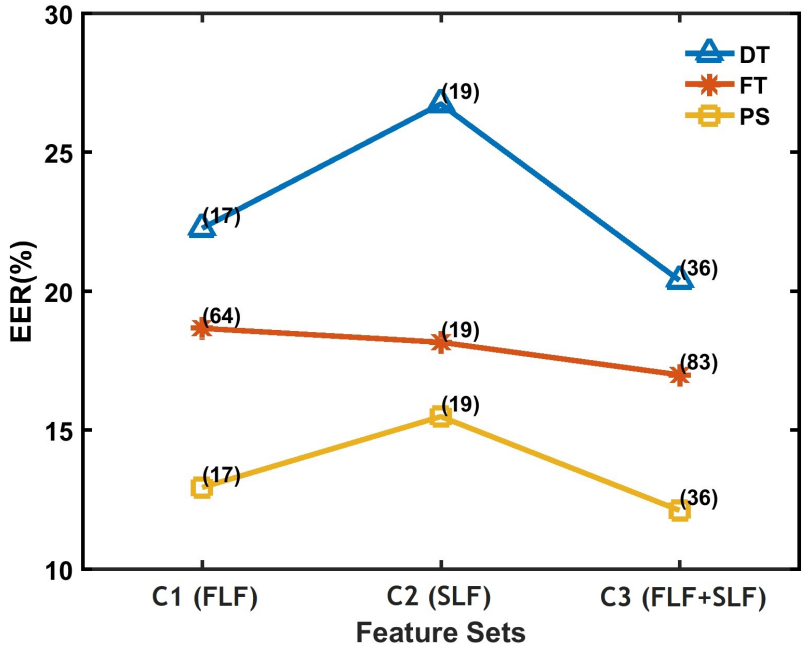

Figure 4.13: EER values for different feature sets (with the feature dimensions in brackets)

Based on the results, it seems that the accuracy performance of the authentication system with the use of C2 is not as good as that of C1, but this does not mean that they are not useful. C2 represents a subject's touch dynamics pattern in a different way from that of C1, and by combining the features in C1 and C2 using the FLF approach (as discussed in Section 3.7), the number of features available for use in training the model increases. As a result, the model has a better accuracy performance, meaning that the model could better capture the subject's touch dynamics pattern. For example, the EER value of C3 for

DT (20.4%) is lower than the corresponding values of both C1 (22.28%) and C2 (26.75%). This is also true in the case of C3 for FT and PS.

4.7.2.4 Timing Feature Lengths

FT features can be extracted with different feature lengths, and different feature lengths may have implications on the accuracy performance of ToDiTA. To study the correlation between the feature length and the accuracy performance, we set the feature length as a variable with values from 2-graph to n-graph; the minimum and maximum values correspond to the shortest and longest possible feature length used to extract FT features, respectively. In this test case, we have extracted the FT features from the 4D string.

Figure 4.14 shows the EER values and feature dimensions versus different feature length values. From the figure, it can be seen that there is a steady increase in the EER values as the value of the feature length increases. When the feature length is set to 2-graph, the EER value is the lowest at 21.28%. As the value of the feature length increases, the EER value increases steadily, and when the feature length is set to 5-graph, the EER value reaches to the highest at 32.20%. These results indicate that the accuracy performance worsens when a longer feature length value is used. This could be due to the reason that timing features expressed using a longer feature length value contain a lower level of granularity, and thus capturing less information about a subject's touch dynamics pattern, leading to a lower level of accuracy.

These results reflect those reported in [138] where the authors also found that the accuracy performances of the features extracted with a shorter feature length are better than those with a longer feature length. This could be the reason why a majority of the experiments reported in literature used a smaller feature length value to extract FT features.

4.7.2.5 Feature Normalisation

Different features extracted from raw touch dynamics data have different value ranges. We should normalise the values of different features so that they are in the same value range. If different features are in the same value range, they can be more balance in representing the structure of the data, and, as a result, may achieve a higher level of accuracy. To investigate the effectiveness of feature normalisation on the accuracy performance, we have produced and compared two

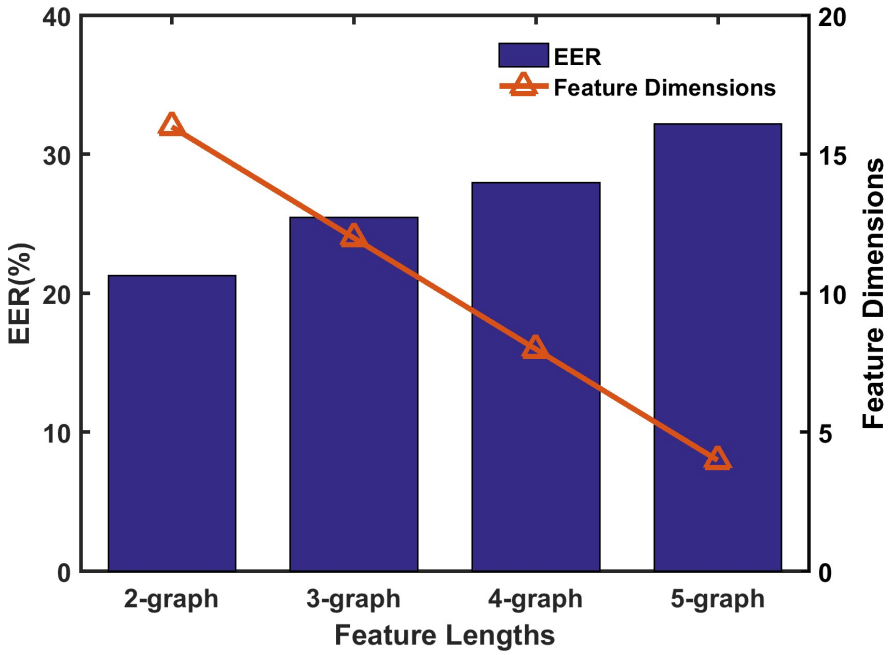

Figure 4.14: EER values of FT versus different feature length values for the 4D string

sets of features, one with and the other without normalisation. Each set consists of all types of FLF extracted from the 4D string.

Table 4.9 shows the EER values of the two sets of features along with their accuracy performance gains. It can be seen from the table that, for all types of FLF, the EER values with normalisation are lower than those without normalisation. From the table, it can also be seen that FT has the highest accuracy performance gain, markedly higher than the other types of FLF. The reason for this may be because the values of FT have a wider value range as compared to the other types of FLF. The wider the value range of a feature, the more unbalanced the feature is at representing the structure of the data, and therefore the better the accuracy performance after they are normalised. These results suggest that feature normalisation can effectively improve the accuracy performance if the features have different value ranges.

4.7.2.6 Feature Selection

In a feature selection process, the most important set of features are selected from a feature set, PFS, to form a subset of features, OFS, that could better represent the structure of the data. In comparison with the PFS set, the OFS set

Types of FLF	EERs (%)		Accuracy Performance Gains (%)
	Without Normalisation	With Normalisation	
DT	41.3	32.7	+20.82
FT	35.15	21.28	+39.47
IT	39.35	32.03	+18.61
PS	22.4	18.8	+16.07

Table 4.9: EER values of different types of FLF with and without normalisation

is better in two ways. Firstly, the accuracy performance of the model with the use of the OFS set is usually higher. Secondly, the feature dimension of the OFS set is smaller. To investigate the benefit of using feature selection, we have used 13 different sets of OFS and compared their EER values. To form these sets of OFS, we first formed a PFS set consisted of the FLF and SLF features extracted from the 16D string. Then, we applied a feature selection process to the PFS set using 13 different feature selection sizes. The values of the size ranged from 1% to 100%. The first four size values were set to 1%, 2%, 5%, 10%, and the remaining ones were set to values with an increment of 10%. The first 12 sets represented the OFS sets with different feature selection sizes, and the last set was the PFS set.

Figure 4.15 shows the EER values and feature dimensions versus the OFS sets with different feature selection sizes. As can be seen from the figure, the EER values are higher when the size is set to a very small or very large value. The EER values decrease sharply as the sizes increase from 1% to approximately 10%, remain largely unchanged from 10% to 30%, before increasing slightly as the sizes increase toward 100%. These results can be explained as follows. When the smallest size (1%) is used, the OFS set has only 2 (out of 213) features. With such a small number of features available for training the model, the model does not have sufficient information to properly represent a subject's touch dynamics pattern, and, as a result, achieves the lowest level of accuracy (EER 17.05%). When larger sizes are used, the OFS sets have more features, and with more features available for training the model, the model could better represent a subject's touch dynamics pattern, and, as a result, the EER values decrease

sharply. At a certain feature selection size, the EER value reverses the downward trend and starts gradually increase as the size increases further towards 100%. This change in trend is referred to as the *"peaking phenomenon"* [219], in which the increase in the feature dimension no longer improves, but, instead, reduces the accuracy level of the model. This phenomenon usually happens when the feature dimension is relatively large, larger than the training sample size, such that the ability of the classifier to build an accurate model is reduced because of the large set of features [220].

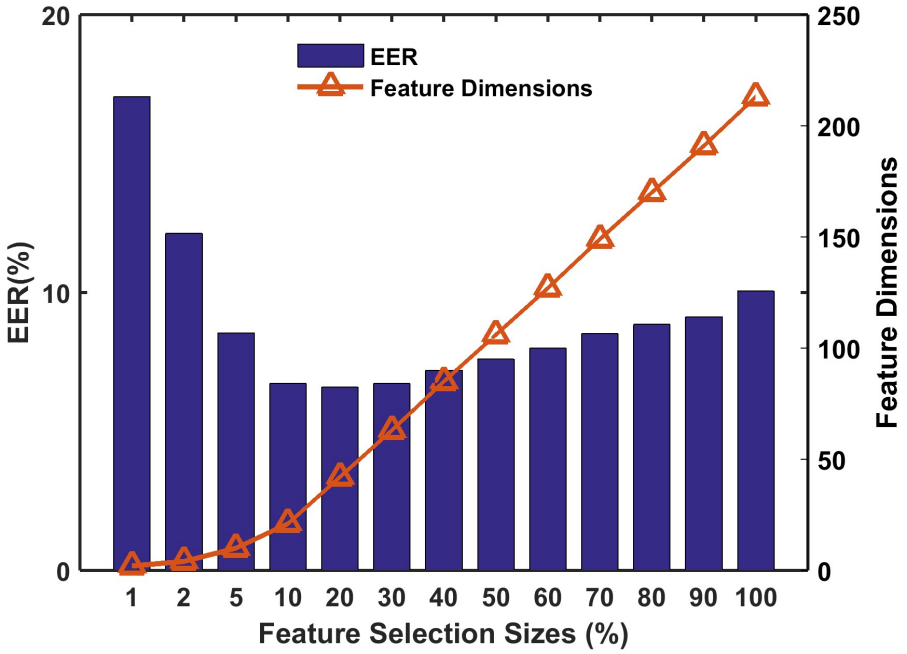

Figure 4.15: EER values and feature dimensions versus the OFS sets at different feature selection sizes

With the exception of the feature selection size of 1% and 2%, the accuracy performances for the other sizes are better than that of 100% (which is the case without feature selection or the use of the PFS set). This observation implies that, as long as the feature selection size is not set to an extremely low value, applying feature selection to the PFS set can improve the accuracy performance, and reduce the feature dimension of the PFS set, which leads to an increase in the efficiency and the robustness of the authentication system.

4.7.3 MTU Evaluation

This section evaluates MTU under different parameter value settings, including the use of different training sample sizes and classifier groups.

4.7.3.1 Training Sample Sizes

The training sample size refers to the number of touch dynamics samples that are used to train a classifier to build an authentication model. The choice of a training sample size may have implications on the accuracy performance of, and the time taken to train (i.e. training time), the authentication model. To investigate the effect of using different training sample sizes on the accuracy performances, and the training times, of authentication models, we have set the training sample size to be a range of values from 2 to 9 with an increment of 1, and investigated the accuracy performances and training times under these value settings. For this test case, we have extracted the FLF and SLF features from the 16D string.

Figure 4.16 shows the EER values and training times against different training sample sizes. From the figure, we can make two major observations: i) the EER value decreases steadily as the training sample size increases; and ii) the training time increases steadily as the training sample size increases. The reason for i) is that, as the training sample size increases, the classifier will have more data to learn, and, therefore, the classifier can build more accurate model that better represent a subject's touch dynamics pattern. As a result, the accuracy performance of the model will improve. The reason for ii) is that, the more samples are used to train the model, the more training time it takes.

As shown by the above discussions, there is a trade-off between security and usability in this authentication system. Using a larger training sample size will improve security, as the accuracy in verifying a claimed identity will be higher, but the time taken in training the model will also be higher, reducing usability. How to balance this trade-off, i.e. enhance or maintain the security level, while, at the same time, minimise the usability cost, may worth further investigation. We leave this investigation for future work.

4.7.3.2 One-Class Classifier (OCC) vs Two-Class Classifier (TCC)

The classifiers used in our experiments can be classified into two groups, OCC and TCC. The main difference between the two groups lies in the type of training

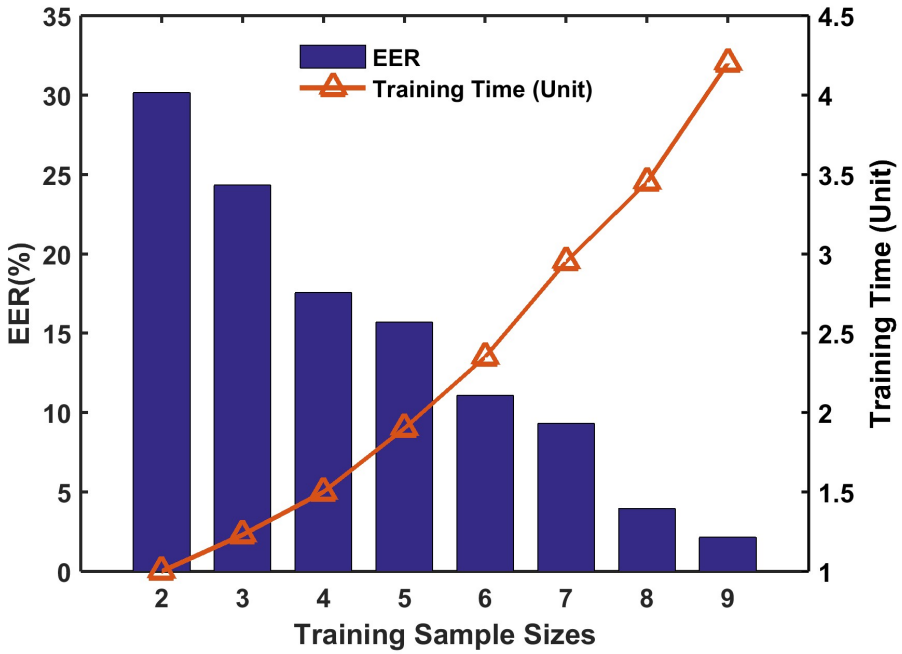

Figure 4.16: EER values and training times at different training sample sizes

samples they each use in building authentication models (as discussed in Section 4.5.3.1). Because of this, the models built by the two groups of classifiers differ in three attributes: i) accuracy performance, ii) training time, and iii) testing time. To evaluate and compare the two groups of classifiers in terms of these three attributes, we have chosen two classifiers for each group. For the OCC group, we have chosen OCKNN and SVDD, and, for the TCC group, we have chosen KNN and SVM. The input to each of these classifiers is set to be the OFS set (with the feature selection size set to 20%) extracted from the 4D string.

Table 4.10 shows the EER values, training times and testing times produced by using the classifiers, and Figure 4.17 presents the DET curves of the classifiers. As shown in the table, the EER values produced when using OCC are higher than those when using TCC. More specifically, the EER values when using OCKNN and SVDD are 10.5% and 9.9%, respectively, whereas the corresponding values when using KNN and SVM are 8.7% and 9.4%, respectively. This indicates that the accuracy performances of the models built by OCC are lower than those by TCC. This may be due to the fact that, unlike OCC, the classifiers in TCC build the models with both legitimate and illegitimate samples, which means that the models can capture more information about the subjects' touch dynamics

patterns, leading to more accurate models. However, it should be emphasised that the level of gain in the accuracy performance by using TCC is not significant. As shown in Figure 4.17, the DET curves of OCC and TCC are rather close to each other.

Unlike the case for the accuracy performance, there is no clear correlation between a particular group of classifiers, OCC or TCC, and the training time, rather the training time appears to be classifier dependent. Among the four classifiers, SVM is significantly more expensive than the other three classifiers; approximately seven times more expensive than SVDD. The second most expensive classifier is SVDD, consuming 3 units of time against 1 by OCKNN and KNN. There are two factors that influence the training time. The first is the nature (structure or approach) of a classifier, and the second is the number of samples a classifier uses to train a model. It seems that the first factor plays a dominant role in training time.

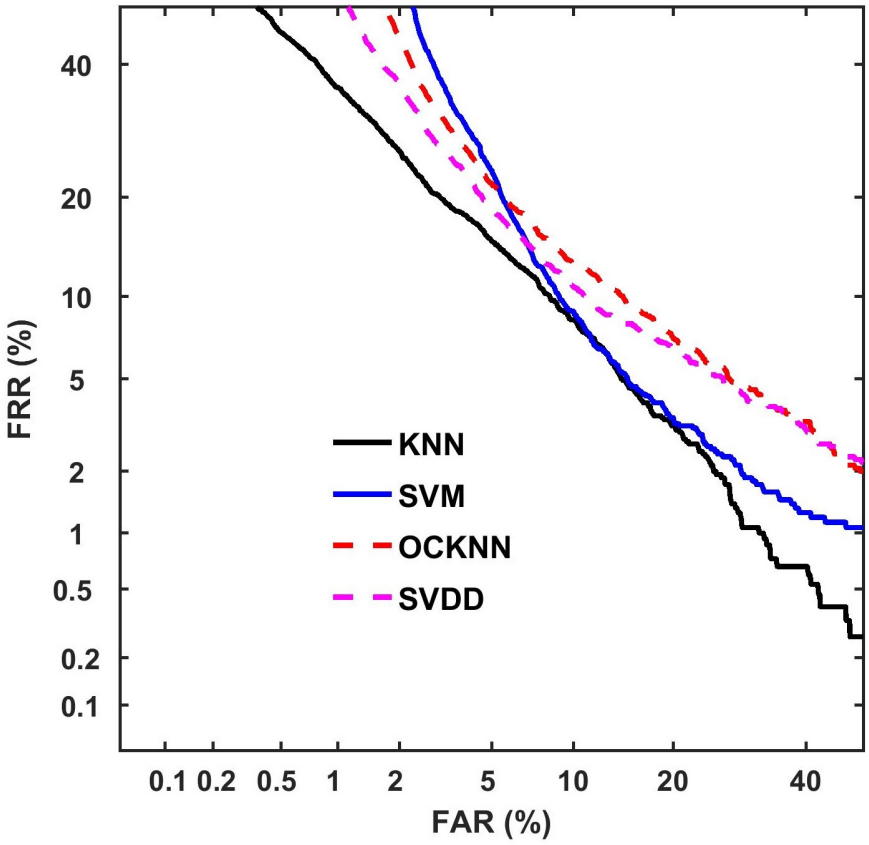

Figure 4.17: DET curves of four different classifiers

With regards to the testing time, TCC classifiers are more expensive than

OCC classifiers. KNN is the most expensive one among the four classifiers; seven times more expensive than OCKNN and SVDD. SVM is second most expensive, costing twice as much as OCKNN and SVDD. Similar to the case for the training time, two factors, the nature of a classifier, and the number of samples a classifier uses to test a model, influence the testing time. Based on the results shown in Table 4.10, it appears that the second factor plays a dominant role in testing time. As mentioned earlier, TCC classifiers use both legitimate and illegitimate samples when testing a model, i.e. TCC classifiers use more samples, therefore are more expensive.

Classifiers	**Classifier Groups**	**EERs (%)**	**Training Times (Unit)**	**Testing Times (Unit)**
OCKNN	OCC	10.5	1	1
KNN	TCC	8.7	1	7
SVDD	OCC	9.9	3	1
SVM	TCC	9.4	22	2

Table 4.10: EER, training time and testing time values of four classifiers

Based on the above results and discussions, particularly taking into consideration of the finding that, for a roughly similar level of accuracy performance, OCC classifiers are generally more efficient than TCC classifiers, both in terms of training and testing times, and that, in a mobile device context, usually only the data from the owner of a device are available for use in training the classifier to build the authentication model, and performance and usability requirements are also important, we recommend the use of OCC classifiers in building an authentication model in this application context.

4.7.4 The Architecture Evaluation

As mentioned in Chapter 3, a touch dynamics authentication method can be used as an additional factor along with an existing knowledge-based authentication method to form a so-called two-factor authentication system, which is more secure than the knowledge-based authentication method when it is used alone.

To evaluate the effectiveness and efficiency of this two-factor authentication method, we have compared two authentication systems. One using only a PIN authentication method (denote as AS1), and the other is the ToDiTA system (using both a PIN and the touch dynamics authentication method). In the evaluation, for both AS1 and ToDiTA, we have used the assumption that the PIN has already been exposed to an impersonator. Table 4.11 shows the probabilities of a successful impersonator attempt using two different authentication systems, where two different PIN lengths are considered. As shown in the table, with AS1, the probability for the impersonator to successfully gain access to the subject's device is 100%. In other words, AS1 will fail to identify any impersonation attempts. On the contrary, with ToDiTA, this probability is reduced to 9.9% and can be further reduced to 7.1% if a longer PIN, i.e. a 16-digit PIN, is used. This reduction does not apply to AS1, as, based on our evaluation assumption, even if a longer PIN is used, AS1 still fails. These reductions are significant, indicating that the two-factor authentication system can achieve a significantly higher level of security in comparison with the single-factor method.

Authentication Systems	**Probabilities of a Successful Impersonator Attempt (%)**	
	4-digit PIN	**16-digit PIN**
AS1	100	100
ToDiTA	9.9	7.1

Table 4.11: Comparison between the probabilities of a successful impersonator attempt for two different authentication systems using two different PIN lengths

To put the above results into context, assuming that there are ten impersonation attempts, then AS1 will fail to detect all ten, but ToDiTA will only fail to detect one. Of course, there is a price to pay for using ToDiTA; there is a non-zero FRR, which impedes usability. With this level of security enhancement offered by ToDiTA, 1 out of 10 legitimate login attempts may be incorrectly rejected. With AS1, the FRR is zero, as none of the login attempts will be falsely rejected as long as the PIN is entered correctly. The results show that, with an additional AF provided by using the touch dynamics biometrics, unauthorised accesses to mobile devices become harder, thus strengthening the security level of mobile devices.

The evaluation results of the ToDiTA system also indicate that the idea of using touch dynamics biometrics to support user authentication on a mobile device or application context is effective provided that users' touch dynamics patterns do not change over time. However, in many real-life cases, users' touch dynamics patterns do change over time. These changes often come from a number of factors such as: (i) cognitive factors (e.g. increasing familiarity with the operations of a device or the input string), (ii) physiological factors (e.g. finger injury), (iii) psychological factors (e.g. drunkenness or tiredness), or (iv) environmental factors (e.g. distraction or position when using the device). The changes in a user's touch dynamics pattern may cause trained model (trained using the touch dynamics samples acquired during the enrolment phase) to deviate from the user's real touch dynamics pattern, reducing the accuracy performance of the model. The ToDiTA system is not effective in such cases.

4.8 Comparison with the Most Related Work

This section presents a comparison between the ToDiTA system and the related work most relevant to ours. Conducting a like-for-like comparison with the related work is a challenging task. This is because each related work varies in terms of their system architecture, dataset, classifier, experimental settings and evaluation methodology used, making it difficult to provide a fair and direct comparison. Therefore, in the following, we provide a high-level comparison instead of a like-for-like comparison.

Table 4.12 summarises the most related work discussed in Section 3.9 and compares the related work against our work presented in this chapter. Basically, this chapter has presented a more systematic and comprehensive study of using touch dynamics biometrics for user authentication purposes. More specifically,

1. It proposes a ToDiTA system, describing the design of the system architecture and its architectural units.

2. It gives a comprehensive description of the experiment carried out to acquire a touch dynamics dataset. Our experiment involves more subjects, acquires less number of samples per subject, and uses a device with a larger screen size than many other studies. The acquired dataset is made publically available, and unlike other public datasets [140, 159], this dataset uses

numerical-based input strings.

3. It discusses the extraction of not only a basic set of timing and spatial related features (FLF) from the dataset, but also an extended set of features (SLF) from the FLF features (related work has mainly focused on the extraction SLF from motion related features, rather than timing and spatial related features).

4. It clearly describes the evaluation methodology used to critically evaluate the performance of the touch dynamics authentication method.

5. It investigates extensively how to make the most efficient use of touch dynamics biometrics in authenticating a user, in terms of optimising accuracy and efficiency performances. This includes the investigation and selection of a subset of optimal features, comparative studies of different timing feature lengths and different groups of classifier, and the evaluation of the impacts of different parameter value settings on the performances.

4.9 Chapter Summary

This chapter has presented the design and evaluation of the ToDiTA system for user authentication on mobile devices. To evaluate the effectiveness of this authentication system, we have acquired a comprehensive touch dynamics dataset. The method and process used to acquire this dataset have been clearly described and discussed. This chapter has also extensively discussed how the touch dynamics data may be used for mobile device authentication, described how raw touch dynamics data may be extracted from the dataset, and how the raw data is processed into a proper format for feature extraction. In particular, it has explained that two types of features can be extracted, FLF, a basic set of features that are extracted from the raw data, and SLF, an extended set of features that are extracted from the FLF features. The chapter also described how the features may be analysed to select a subset of optimal features, and how the features are classified using classifiers to build an authentication model. The classification can be done by using either OCC classifiers (training features solely from legitimate users, i.e. device owners) or TCC classifiers (training features from both legitimate and illegitimate users). Experimental results show that the use

Studies	Subject Sizes	Input Lengths	Sample Sizes	Classifier Groups	Device Screen Sizes	EERs (%)
[132]	152	17	10	OCC	4.65”	4.19†, 4.59#
[152]	10	4	100	TCC	3.7”	15.2
[166]	80	4,8	≥ 25	OCC	-	20
[147]	25	8	20	OCC	4.65”	4.45
[153]	100	4-8	10	OCC	3.7”	8.4
[175]	100	6	5	OCC	3.7”	23
[174]	20	8	50	TCC	-	0.56
[163]	48	5	100	OCC	6”	4.53†, 5.89#
[148]	12	6	117	TCC	5”	91.2*
[87]	95	8	30	TCC	-	0.01†, 4#
ToDiTA	150	4	10	OCC	10.1”	9.9

*Accuracy; †FAR; #FRR

Table 4.12: A summary of existing related work

of OCC classifiers is more efficient for roughly the same level of security, making the OCC-based classification approach more practical in real-world applications.

The evaluation of ToDiTA showed that by integrating the touch dynamics authentication method into the PIN-based authentication method, the ability to counter impersonation attacks is greatly enhanced. These results also indicate that the idea of using touch dynamics biometrics to support user authentication on a mobile device or application context is feasible and effective. However, ToDiTA is not effective when a user's touch dynamics pattern changes. In the next chapter, we will present the design and evaluation of an enhanced version of the ToDiTA system, i.e. E-ToDiTA system, which is designed to overcome this weakness, making it effective in adapting itself to a user's touch dynamics pattern changes as well.

Chapter 5

Enhanced ToDiTA (E-ToDiTA) System: with Adaptive Learning

5.1 Chapter Introduction

This chapter presents the design and evaluation of the E-ToDiTA system. The E-ToDiTA system is an enhanced version of the ToDiTA system in Chapter 4. It addresses the weakness of ToDiTA by adding the capability to perform adaptive learning when a user's touch dynamics pattern changes. E-ToDiTA has used three ideas. The first idea is to examine whether the new touch dynamics data acquired are suitable for use to update the authentication model. The second idea is to minimise the frequency required to carry out the model adaptation processes. The third idea is to update and improve the accuracy performance of the model with a fewer number of samples. These ideas attempt to make E-ToDiTA capable of adapting itself to any changes in a user's touch dynamics pattern and do so without introducing excessive overhead to the mobile device. The design and evaluation of the E-ToDiTA system is our third novel contribution in this thesis.

The rest of this chapter is organised as follows. Section 5.2 gives the ideas used in the E-ToDiTA system design. Section 5.3 outlines the E-ToDiTA system architecture. Section 5.4 describes the designs and operations of E-ToDiTA and its core functional unit in detail. Section 5.5 presents the evaluation methodology used. Section 5.6 discusses the evaluation results of the system. Section 5.7 concludes the chapter.

5.2 Design Ideas

As discussed in Section 4.7.4, the ToDiTA system is not effective when a user's touch dynamic pattern changes. This section presents the ideas we used to design an enhanced version of the ToDiTA system, i.e. the E-ToDiTA system, which is capable of adapting itself to any changes in a user's touch dynamics pattern. These ideas are as follows:

- Idea 1: The new touch dynamics samples acquired (i.e. during a verification phase) contains a user's most recent touch dynamics pattern. These samples can be used to retrain the classifier to build an updated authentication model. The updated model should reflect any changes in the user's touch dynamics pattern. This process is known as model adaptation. If the samples used for model adaptation do not belong to the owner, the updated model may gradually deviate from the owner's real touch dynamics pattern, resulting in an inaccurate model. To prevent this issue, the idea is to examine whether the samples are suitable before they are used for model adaptation. Here, suitable means that the samples are indeed belong to the owner of the mobile device. In this way, the likelihood of using samples that do not belong to the owner for model adaptation can be reduced, preventing the model from gradually deviating from the owner's real touch dynamics pattern.

- Idea 2: The execution of model adaptation processes incur additional computational overhead, and if carried out too frequently (e.g. carried out after every authentication attempt), the overhead costs involved may impact the device performance and battery consumption of the mobile device, reducing usability. To reduce the impact of such overhead costs and to improve usability, the idea is to execute some of the model adaptation processes only when a predefined number of samples has been acquired. In this way, there is no need for executing the processes after every authentication attempt, reducing the computational overhead costs involved, thus increasing the usability level of the system.

- Idea 3: A straightforward approach to perform model adaptation is to use the new samples and all the already-acquired samples in the training set to build an updated model. However, this approach will increase the training

set size indefinitely over time, which may then cause three problems: (i) a higher cost in retraining the model; (ii) a higher risk of overfitting the model; and (iii) a larger storage capacity to store the training set. To address these problems, the idea is to prevent the training set size from increasing indefinitely, and this can be done by using only a portion of the samples in the training set for model adaptation. The resulting smaller set of samples is then used to retrain the model. In this way, the computational cost to retrain the model, the risk of overfitting the model, and the storage requirement to store the training set, can be reduced.

5.3 Architecture Design Overview

This section gives an overview of the E-ToDiTA system architecture. Figure 5.1 shows the E-ToDiTA functional units and the interactions among them. From the figure, it can be seen that the system architecture consists of seven functional units. All the seven units are run on a user's mobile device (as were the case with ToDiTA units). One of the units is E-ToDiTA specific, i.e. the Model Adaptation Unit (MAU). The other six units are AIU, RDAU, FCU, MTU, ADMU and DSU. These six units are identical to those in ToDiTA which have been described in detail in Chapter 4, so the sections in the rest of this chapter only focus on the descriptions and discussions of MAU.

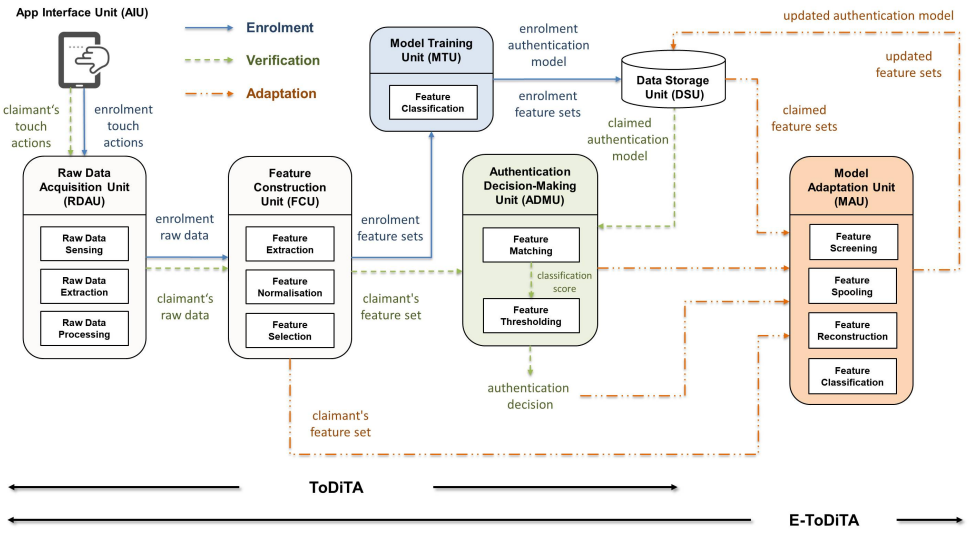

Figure 5.1: E-ToDiTA system architecture

The operation of E-ToDiTA can broadly be captured in three phases: enrolment, verification and adaptation. The enrolment and verification phases are identical to those in ToDiTA. In the adaptation phase, the most recent feature sets, as it becomes available, is used to build an updated authentication model. Figure 5.1 indicates the three operational phases along with the units involved. In the following section, we describe the design of MAU and discuss the issues involved in more detail.

5.4 Model Adaptation Unit (MAU) Design in Detail

This section describes the design of MAU in detail. MAU is the functional unit specifically designed to extend the capability of the ToDiTA system to support adaptive learning. This unit is responsible for using the most recent feature set obtained from an authentication attempt to update the stored feature sets and then retrains the classifier with the feature sets to build an updated authentication model. The updated model should adapt itself to the user's most recent touch dynamics pattern. Model adaptation is carried out by using four processes: (i) feature screening, (ii) feature spooling, (iii) feature reconstruction, and (iv) feature classification. In the following sections, we discuss these processes in detail.

5.4.1 Feature Screening

Feature screening is the first process in model adaptation. This process makes an adaptation decision, i.e. whether $\hat{\tau}$ (the testing sample acquired from an authentication attempt) is suitable for model adaptation. Model adaptation process should be carried out only when $\hat{\tau}$ is classified as $\hat{\tau}_L$ (belongs to the owner of the device), and not $\hat{\tau}_I$ (belongs to an impersonator). Otherwise, the updated authentication model may gradually deviates from the owner's real touch dynamic pattern. This effect is known as the "*creep in*" effect [221], an effect that causes a model to lose its ability to properly represent the owner's real touch dynamic pattern, resulting in an inaccurate model. So it is necessary to ensure, or at least increase the likelihood, that $\hat{\tau}_L$ is used for model adaptation.

To ensure that only the suitable $\hat{\tau}$ is used for model adaptation, a straightforward method can be used. We could simple carry out the model adaptation process when D_{auth} (the authentication decision generated by ADMU) is positive. In this case, $\hat{\tau}$ is classified as $\hat{\tau}_L$, so $\hat{\tau}$ can be used for model adaptation. However, this method is less reliable as it solely depends on ADMU to make a correct D_{auth}. In practice, there is a likelihood for ADMU to make an incorrect decision due to feature classification error described in Section 4.6. If this is the case, $\hat{\tau}$ should not be used for model adaptation, as even if ADMU outputs D_{auth} as positive, $\hat{\tau}$ is in fact $\hat{\tau}_I$, not $\hat{\tau}_L$.

To increase the reliability of this method, a feature screening process is used. This process further assesses the suitability of $\hat{\tau}$ for model adaptation. The inputs of the process are D_{auth} and δ (the classification score), both generated from ADMU. The output is an adaptation decision, D_{adap}. The two inputs along with a predefined adaptation threshold, θ_{adap}, are used to determine D_{adap} based on the following rules:

$$D_{adap} = \begin{cases} true & \text{if } (D_{auth} = +) \text{ and } (\delta > \theta_{adap}) \\ false & \text{otherwise} \end{cases} \tag{5.1}$$

If D_{auth} is positive and δ is over θ_{adap}, then D_{adap} is true, and in this case, $\hat{\tau}$ should be used for model adaptation. Otherwise, D_{adap} is false, and in this case, $\hat{\tau}$ should not be used for model adaptation. Note that the value for θ_{adap} is set to a value greater than the value of θ_{auth} (the predefined authentication threshold in ADMU). In this way, for $\hat{\tau}$ to be suitable for model adaptation, $\hat{\tau}$ has to satisfy a stricter condition on top of the one set in ADMU. With this additional condition, the likelihood of using $\hat{\tau}_I$ for model adaptation can be decreased, thus reducing the likelihood of the "*creep in*" effect as much as possible.

5.4.2 Feature Spooling

Feature spooling is the process that follows feature screening. This process is responsible for reducing the computational overhead incurred by the two processes in model adaptation, i.e. feature reconstruction and feature classification.

Typically, feature reconstruction and feature classification are carried out immediately after feature screening. However, using this process flow may incur high overhead costs for the reason explained as follows. Previous studies

by [98, 222, 223] reported that, on average, users perform an authentication attempt to unlock their mobile devices 10 to 200 times per day. This statistic suggests that, if the above process flow is used for model adaptation, the feature reconstruction and feature classification processes will have to be carried out many times per day. Given the fact that these processes incur additional overhead costs, and if they are carried out many times per day, the negative impacts of such overhead costs on device performance and battery consumption will be large, reducing usability.

To reduce the impact of such overhead costs and to improve usability, we should reduce the frequency of carrying out feature reconstruction and feature classification processes. This can be done by adding a process called feature spooling. Feature spooling delays the execution of these two processes until a predefined number of $\hat{\tau}$ is accumulated, reducing the number of times required to carry out these two processes. The feature spooling operational flow is summarised in Figure 5.2 and described below.

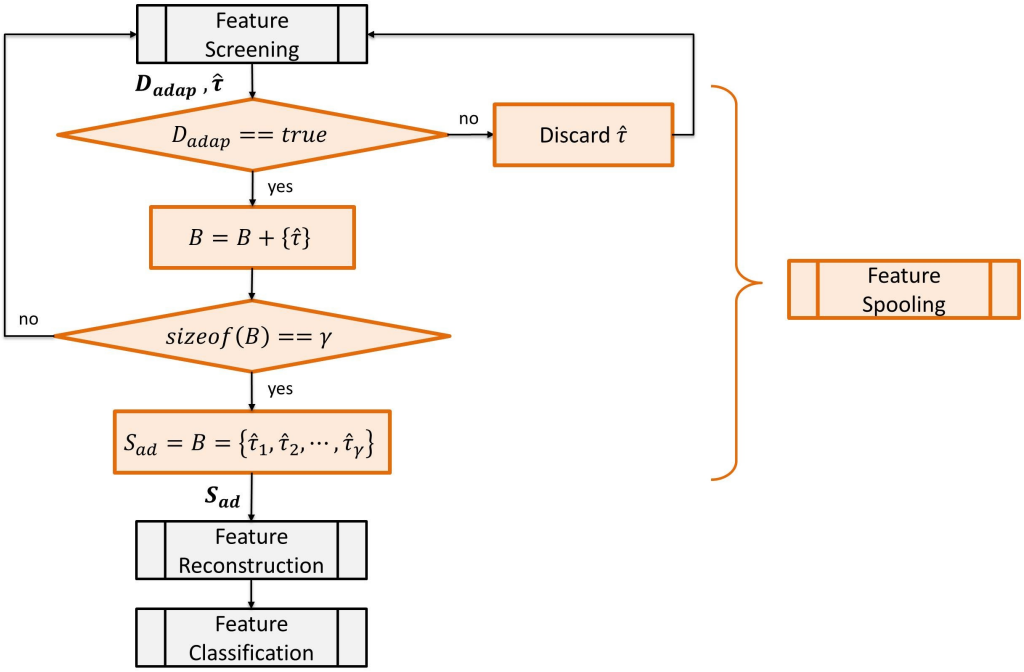

Figure 5.2: Feature spooling operational flow

In the descriptions, it is assumed that there are two inputs and one output for feature spooling. The inputs are D_{adap} and $\hat{\tau}$, both are from the feature screening process. The output is an adaptation set, S_{ad}.

Step 1: If D_{adap} is true, proceed to Step 2. Otherwise, $\hat{\tau}$ is discarded and repeat Step 1 with the next set of D_{adap} and $\hat{\tau}$.

Step 2: $\hat{\tau}$ is stored in a buffer, B (a temporary storage space in the device's memory). The samples are stored following the order in which they are acquired.

Step 3: If the size of B is equals to a predefined size limit, γ, proceed to Step 4. Otherwise, repeat Step 1 with the next set of D_{adap} and $\hat{\tau}$.

Step 4: The stored samples in B are moved to S_{ad} and released for feature reconstruction and then feature classification.

To further reduce the impact of the overhead costs incurred by feature reconstruction and feature classification processes, we could further delay the execution of these two processes (even after S_{ad} has been released by feature spooling) until the period when the execution of these processes have the least effect on a device. Examples of such period are: (i) when the device is in standby mode; (ii) when the processor of the device is idle; (iii) when the device is plugged in to a power source for recharging; and (iv) during a user-scheduled time.

5.4.3 Feature Reconstruction

Feature reconstruction uses S_{ad} and the training set stored in DSU to reconstruct an updated training set. In our design, three methods are used to carry out feature reconstruction: (i) no adaptation (NA), (ii) accumulative adaptation (AA), and (iii) progressive adaptation (PA). The methods are summarised in Table 5.1 and described in the following sections.

Methods	**Equations**
NA	$S_{tr}^{t+1} = S_{tr}^{1}$
AA	$S_{tr}^{t+1} = S_{tr}^{t} + S_{ad}$
PA	$S_{tr}^{t+1} = S_{tr}^{t} - \{\tau_1, \tau_2, ..., \tau_{n-u}\} + S_{ad}$

Table 5.1: A summary of the three feature reconstruction methods

In the table and descriptions, we have used the following assumptions. Let $S_{tr}^{t} = \{\tau_1, \tau_2, \ldots, \tau_n\}$ represents a training set stored in DSU at a given time t,

where n refers to the size of, or the number of samples in, S_{tr}^t. The samples in S_{tr}^t are stored in chronological order. For example, τ_1 is the earliest sample and τ_n is the most recent sample in the set. $S_{ad} = \{\hat{\tau}_1, \hat{\tau}_2, \ldots, \hat{\tau}_\gamma\}$ represents an adaptation set, where γ refers to the size of S_{ad}. The samples in S_{ad} are stored in the same way as the samples in S_{tr}^t. S_{tr}^{t+1} represents an updated training set after the execution of one feature reconstruction iteration. Finally, u refers to the number of samples in S_{tr}^t that will be reused for feature reconstruction.

5.4.3.1 Method 1: No Adaptation (NA)

With this method, the stored training set and the authentication model are not updated, as would be the case as if there is no model adaptation. It represents the case of the ToDiTA system, using as a baseline for comparison with the AA and PA methods. Figure 5.3 shows the concept of the NA method graphically. The numbers in the boxes indicate the order in which the samples are acquired. As can be seen from the figure, because S_{ad} is not used to update S_{tr}^t, the samples in S_{tr}^t remain unchanged for all values of t, and so are the values of n.

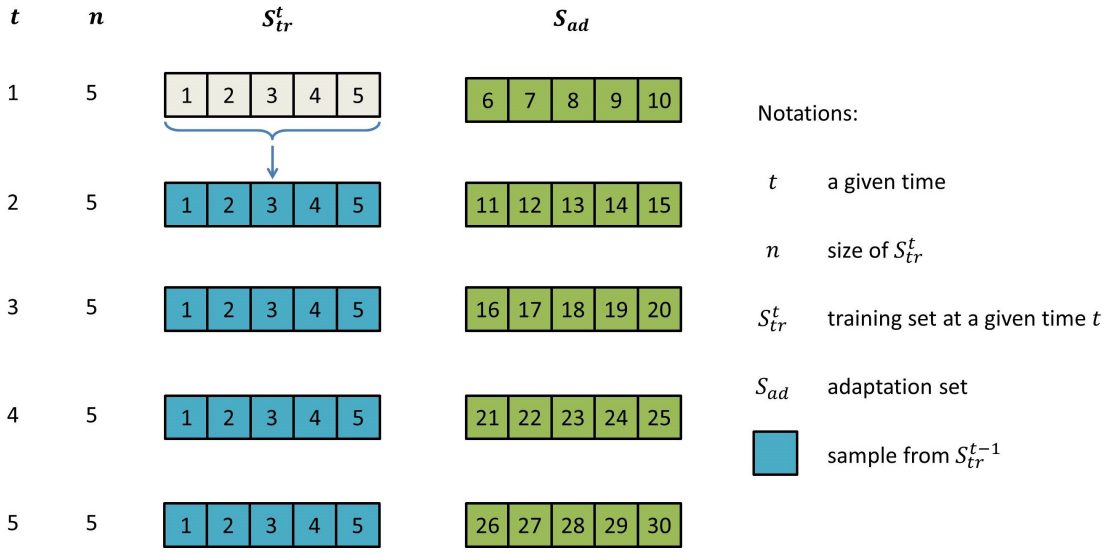

Figure 5.3: Feature reconstruction using the NA method

5.4.3.2 Method 2: Accumulative Adaptation (AA)

The AA method uses a straightforward approach to update the stored training set. With this method, at each feature reconstruction iteration, the samples in

S_{ad} are appended to the samples in S_{tr}^{t} to construct S_{tr}^{t+1}. After each iteration, the size of S_{tr}^{t+1} increases from n to $(n+\gamma)$, as all the samples in S_{tr}^{t} are reused to construct S_{tr}^{t+1}. Figure 5.4 shows the concept of the AA method graphically. From the figure, it can be seen that the value of n indefinitely increases by a constant amount, resulting in a large training set size over time. The use of a very large training set may lead to three problems. Firstly, a higher computational complexity thus a higher cost in retraining an updated authentication model. Secondly, a higher risk of overfitting the updated model, leading to a lower accuracy performance of the model. Thirdly, a larger storage requirement for storing the samples in the training set.

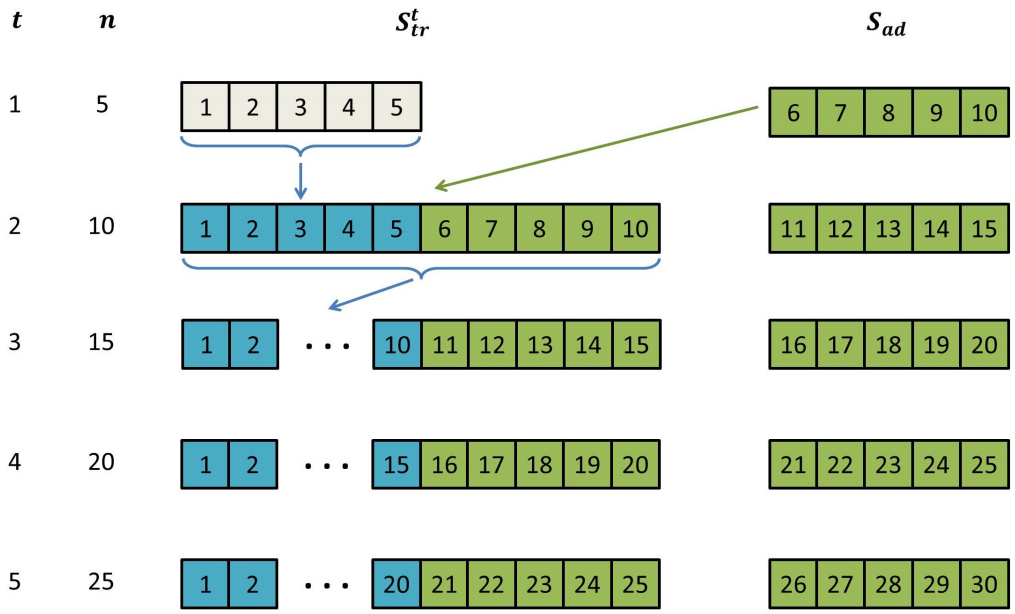

Figure 5.4: Feature reconstruction using the AA method

To prevent the problems mentioned above, we proposed a third method called the progressive adaptation (PA) method. The method is described in the following section.

5.4.3.3 Method 3: Progressive Adaptation (PA)

The PA method uses a more complex approach to update the stored training set. Unlike the AA method, the PA method reuses only a predefined number (instead of all) of the samples in S_{tr}^{t} to construct S_{tr}^{t+1}. Figure 5.5 shows the concept of the PA method graphically.

With this method, each feature reconstruction iteration consists of three steps as follows.

Step 1: The number of samples to be reused, u, is determined by using the equation below:

$$u = \lfloor n \times \omega \rceil \tag{5.2}$$

where n refers to the number of samples in S_{tr}^{t} and ω refers to a reuse ratio. The value of ω is set to a decimal number between 0 and 1. The value of u is rounded to the nearest whole number so that it properly represents the number of samples in a training set.

Step 2: The first $n - u$ samples in S_{tr}^{t} are removed from the set. The remaining samples in the set are the u most recently acquired samples, and these samples are used in Step 3.

Step 3: The samples in S_{ad} are appended to the end of S_{tr}^{t} to construct S_{tr}^{t+1}.

After one feature reconstruction iteration, the size of S_{tr}^{t+1} increases from n to $(u + \gamma)$, and after subsequent iterations, the size remains constant. As shown in Figure 5.5, the value of n increases at the beginning of time and gradually plateaus over time. This is in stark contrast to the case of the AA method, where the size of S_{tr}^{t+1} increases indefinitely.

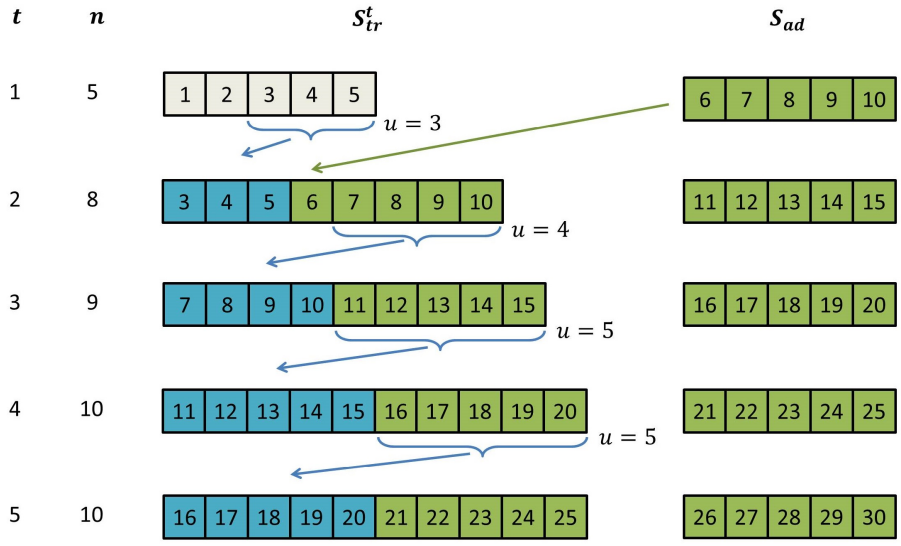

Figure 5.5: Feature reconstruction using the PA method

The PA method has two benefits over the AA method. Firstly, after t reaches a threshold value, the value of n stops increasing and remain constant. For example, referring to Figure 5.5, this threshold value is 4, and the value of n remain constant at 10 when $t \geq 4$. Secondly, the training set size is smaller at all values of t. Having these two benefits prevent the problems associated with the AA method as discussed in the section above.

5.4.4 Feature Classification

Once the updated training set is obtained, they are used by the feature classification to build an updated authentication model. The feature classification process used in MAU is similar to that in MTU in Section 4.5.3. As described in that section, feature classification involves two tasks, the selection of the classifier and the implementation of the selected classifier. In the following, we highlight the differences in the feature classification process used in MAU as compared to the one used in MTU with regards to these two tasks.

For the first task, we have selected the SVDD classifier for feature classification. The selection of the classifier is supported by the observations drawn from the evaluation results reported in Section 4.7.3.2 as follows. Firstly, the SVDD classifier (a type of OCC classifier) consumes less model training and testing time than TCC classifiers. Secondly, the accuracy performance of the model built by the SVDD classifier is the highest among the evaluated OCC classifiers.

With regards to the second task, the SVDD classifier used in our experiments is implemented using the Matlab programming platform (version 9.0.0.370719) and the dd_tools open source toolbox [210]. Similar to the case for MTU, the implementation of the classifier involves two phases, training and testing. The testing phase is identical to that in MTU, so we do not discuss it further. In the training phase, the SVDD classifier width and regularisation parameters of the RBF kernel are empirically set. In our experiments, we have used three datasets (to be described in Section 5.5.1). Typically, different datasets have different samples. When the samples are different, the boundaries that are required to contain the samples are also different. This means that the RBF kernel parameters used to fit the boundaries are also different. For this reason, in our experiments, we set the RBF kernel parameters values for each dataset separately. Table 5.2 gives the RBF kernel parameters values for each of the datasets.

Datasets	Width Parameters	Regularisation Parameters
RHU	300	0
MBK-E	10^7	0
MBK-S	75	0
MBK-LS	570	0

Table 5.2: The RBF kernel parameters values for the datasets used in our experiments

5.5 Evaluation Methodology

This section describes how the accuracy performance of E-ToDiTA is evaluated. It covers three issues, the evaluation datasets, metrics and method used. In this section, we only focus on the evaluation datasets and method used, as the evaluation metrics used are identical to those used in ToDiTA (as described in Section 4.6).

5.5.1 Datasets

This section describes the datasets used to evaluate the accuracy performance of E-ToDiTA. It first gives the requirements of the datasets and then describes the datasets in detail.

To evaluate the accuracy performance of E-ToDiTA, we need to use touch dynamics biometrics datasets that fulfil a specific set of requirements [224]. The requirements are as follows:

- The samples should be acquired from different data acquisition sessions.
- Each session should be separated by some interval.
- The sessions should span over a period of time.
- The sessions should be conducted in an uncontrolled environment.
- The number of samples for each subject should be sufficiently large.

- The samples should be stored in chronological order (i.e. following the order in which they are acquired).

The dataset we used to evaluate ToDiTA did not fulfil all the above requirements, so it could not be used to effectively evaluate E-ToDiTA. Therefore, we have to resort to find public datasets that fulfil the requirements. At the time of this writing, we are only able to find four public datasets that fulfil the requirements, and, at the same time, most relevant to our research context. These datasets are summarised in Table 5.3 and described in the following sections.

5.5.1.1 The RHU Dataset

The RHU dataset was published by [140]. The data acquisition process conducted to acquire this dataset involved 51 subjects. The input string used was a fixed character-based passcode, "rhu.university". The acquisition device used was a Nokia Lumia 920 touchscreen phone. Each subject attended three data acquisition sessions, and each session was separated by an average interval of five days. In each session, each subject entered five input samples of the passcode, resulting in a total of 15 samples acquired over the three sessions (per subject). In terms of typo error handling, any input mistake made by a subject was discarded, and the subject was prompted to repeat that particular input sample.

After analysing the dataset, we have discovered two inconsistencies: (i) four of the subjects have less than 15 samples, and (ii) some subjects have more than five samples acquired for the third session. To prevent these inconsistencies from affecting the evaluation results of our experiments, we have made two modifications to the dataset. To address inconsistency (i), we have excluded the samples of those four subjects concerned and used only the samples from the remaining 47 subjects. To address inconsistency (ii), for those affected subjects, we have discarded the additional samples and retained only the first five samples acquired from the third session. After discarding the additional samples, all subjects have the same number of samples for each of the three sessions (i.e. five samples for each session).

We have extracted four types of timing features from the samples in the dataset, and they are DT, FT1, FT2 and FT3. The methods used to extract these features are the same as those described in Section 4.5.2. Similar to the case for ToDiTA, we used the extracted features to form a PFS set and performed

Properties	**RHU**	**MBK-E**	**MBK-S**	**MBK-LS**
Subject Size	47	54		
Input String	rhu.university	kicsikutyatarka	.tie5Roanl	Kktsf2!2014
Acquisition Sessions	3	3		
Inter-Session Interval (days)	5	7		
Samples per Session	5	20		
Total Samples per Subject	15	60		
Feature Data Types	Timing	Timing, Spatial, Motion		
Typo Error Handling Method	Error input sample discarded, subject reenter the particular sample			
Device Screen Size (inch)	4.5	7		
Acquisition Device Model	Nokia Lumia 920	Nexus 7		
Age Range	7-65	19-26 (avg 20.61)		
Gender	51% male; 49% female	91% male; 9% female		
Touchscreen Device Usage Frequency	-	15% Rare; 31% Average; 54%Often		
Hand Preference	-	7% Left; 93% Right		

Table 5.3: Properties of the touch dynamics biometrics datasets used in our experiments

a feature selection process for the PFS set to obtain an OFS set. Table 5.4 shows the OFS set for the dataset, where the feature selection size used is set to 20.

Dataset	OFS Features
RHU	DT(am,qm,hm,gm,md,fq,tq), FT1(1,8,11,am,hm,tq), FT2(1,5,8,10,12,mn,am,fq), FT3(1,2,5,md)

Table 5.4: OFS set for the RHU dataset (with feature IDs given in brackets)

5.5.1.2 The MBK Datasets

There are three variants of MBK datasets: MBK-E, MBK-S and MBK-LS. They were published by [159]. The three datasets shared many common properties and differed mainly in terms of their input string used. In the following, we first describe the different input strings used by each of the datasets and then describe their common properties.

The input strings used for the MBK datasets were fixed character-based passcodes. The passcodes were designed to represent different passcode difficulty levels. There were three difficulty levels: easy, strong and logically strong. We used the first letter of each level as suffixes to represent the datasets, i.e. MBK-E for easy, MBK-S for strong and MBK-LS for logically strong. The passcode used for MBK-E was "kicsikutyatarka", representing an easy passcode which consists of only lowercase characters. The passcode used for MBK-S was ".tie5Roanl", representing a strong passcode which consists of a combination of digits, upper, lower and special case characters. The passcode for MBK-LS was "Kktsf2!2014", representing a logically strong passcode which consists of both some logic behind the selection of the passcode as well as the properties of a strong passcode. For example, the last four digits of the passcode refer to the year when the data acquisition has taken place.

The common properties among the MBK-E, MBK-S and MBK-LS datasets are as follows. The data acquisition process conducted to acquire these datasets involved 54 subjects. The acquisition device used was a Nexus 7 touchscreen phablet. Each subject attended three data acquisition sessions, and each session was separated by an interval of seven days. In each session, each subject entered at least 20 samples of the passcode, resulting in a total of at least 60 samples

acquired over the three sessions (per subject). The typo error handling method used was the same as the one used by the RHU dataset.

There were also some inconsistencies discovered in the MBK datasets. Some subjects have more samples per session than some of the other subjects. Since each subject has at least 20 samples per session, we have unified the number of samples in each session to 20. This is done by discarding any additional samples after the first 20 samples for each session. In other words, we used only the first 20 samples for each session.

We have extracted 11 types of features from the samples in the MBK datasets. The methods used to extract 5 out of the 11 types of features are the same as those described in Section 4.5.2.1. So here, we only described the remaining six types of features, and they are: (i) pressure intensity (PI), sum of distance (DC), (ii) input speed (IS), and (iii) mean accelerations along the x-, y- and z-axis (ACX, ACY and ACZ). Table 5.5 gives a list of the features along with their corresponding descriptions and the mathematical methods used to extract the features.

Similar to the case for the RHU dataset, for each MBK dataset, we used the extracted features to form a PFS set and performed a feature selection process for the PFS set to obtain an OFS set. Table 5.6 shows the OFS sets for the MBK datasets, where the feature selection sizes used are set to 20.

5.5.2 Method

The method used to evaluate the E-ToDiTA system is summarised in Algorithm 5.1 and is described below. In the description, we made three assumptions as follows. Firstly, the touch dynamics samples acquired from a set of subjects, B, are stored in a dataset, S. Secondly, each subject has a total of $K \times N$ samples, where K represents the number of data acquisition sessions and N represents the number of samples in each session. Thirdly, the samples are stored in chronological order.

Step 1: One of the subjects in B is assumed and assigned as a legitimate subject, b. The remaining subjects in B are regarded as illegitimate subjects.

Step 2: The samples in S are split into two subsets, S^+ and S^-. S^+ contains the samples of b, and S^- contains the samples of the illegitimate subjects.

Features	Descriptions	Equations
PI	The pressure intensity of the TAP of a key.	$PI_i = pi_i$
DC	The sum of the distances between the TAP positions of two successive keys, starting from the first key to the last key.	$DC = \sum_{i=1}^{m} dist((cx_i, cy_i), (cx_{i+1}, cy_{i+1}))$
IS	The speed between the TAP of the first key and the TAR of the last key.	$IS = \frac{DC}{IT}$
ACX	The mean acceleration along the x-axis of the TAP of all keys.	$ACX = \frac{\sum_{i=1}^{m} ax_i}{m}$
ACY	The mean acceleration along the y-axis of the TAP of all keys.	$ACY = \frac{\sum_{i=1}^{m} ay_i}{m}$
ACZ	The mean acceleration along the z-axis of the TAP of all keys.	$ACZ = \frac{\sum_{i=1}^{m} az_i}{m}$

cx, cy: the x- and y-coordinates recorded on the TAP action on a key

ax, ay, az: the accelerations along the x, y-, z-axis recorded on the TAP action on a key

Table 5.5: Descriptions and definitions of touch dynamics features extracted for the MBK datasets in addition to those described in Section 4.5.2.1

Datasets	OFS Features
MBK-E	DT(am,qm,hm,md,fq,tq), FT1(3,4,11,13, hm,gm), FT3(3,4,13,fq), PI(4,6,am,qm,hm,gm,md,sd,fq), PS(mx,am,qm,hm,tq), IS, ACX, ACY, ACZ
MBK-S	DT(am,qm,gm,md,fq,tq), FT1(2,3,9,11,hm), FT3(10,11,fq), PI(5,8,mx,am,qm,hm,gm,md,fq,tq), PS(mx,am,qm,tq), IS, ACX, ACY, ACZ
MBK-LS	DT(am,qm,hm,gm,md,fq,tq), FT2(4,10,11,hm), FT3(4,11), PI(1,7,mx,am,qm,hm,gm,md,fq,tq), PS(am,qm,hm,tq,se), IS, ACX, ACY, ACZ

Table 5.6: OFS sets for the MBK datasets (with feature IDs given in brackets)

Step 3: S^+ is split into K equal-sized subsets. Each subset contains N samples, and these samples are the samples acquired from the k^{th} data acquisition session.

Step 4: The number of training samples, r, is computed by multiplying a training sample ratio, R, by N. The value of r is rounded to the nearest whole number. In our evaluations, we empirically set $R = 0.7$.

Step 5: Here, the samples in the k^{th} subset, S_k^+, are split into two subsets, a legitimate training set, S_{tr}^+, and a legitimate testing set, S_{ts}^+. S_{tr}^+ contains the first r samples in S_k^+, and S_{ts}^+ contains the rest.

Step 6: A training set, T_{tr}^k, is established. Depending on whether the feature construction method, F, used is the NA, AA or PA method, T_{tr}^k is established by using Step 6a, 6b or 6c, respectively. The only exception is when k is 1 (referring to the first feature reconstruction iteration). In this special case, regardless of F used, T_{tr}^k is initialised to the samples in S_{tr}^+.

Step 6a: When F used is the NA method, T_{tr}^k is initialised to the samples in the first training set, T_{tr}^1.

Step 6b: When F used is the AA method, T_{tr}^k is initialised to the samples in the previous training set, T_{tr}^{k-1}, plus the samples in S_{tr}^+.

Step 6c: When F used is the PA method, T_{tr}^k is initialised to the last u samples in T_{tr}^{k-1}, plus the samples in S_{tr}^+. u is computed by multiplying a reuse ratio, ω, by the number of samples in T_{tr}^{k-1}. In our evaluations, we empirically set $\omega = 0$.

Step 7: T_{tr}^k is used to train C to build a model, M_b, for b.

Step 8: A testing set, T_{ts}^k, is established, where T_{ts}^k contains the samples in S_{ts}^+ and S^-.

Step 9: M_b is evaluated on T_{ts}^k, and the accuracy performance of M_b (for the k^{th} subset) is calculated and recorded as P_k.

Step 10: Step 5 to 9 is repeated for each of the remaining subsets in the K subsets.

Step 11: Step 1 to 10 is repeated, assigning each of the other subjects in B as b in turn.

Step 12: The overall accuracy performance of the model (over all subjects) for each of the subsets, P_k, in K subsets, is calculated by dividing P_k by B.

5.6 Evaluation Results and Discussions

This section describes the experiment carried out to evaluate the performance of the E-ToDiTA system and discusses the evaluation results obtained. The purpose of the experimental study is to check the effectiveness of the ideas implemented in the E-ToDiTA system. The experiment is carried out by comparing the performance of the E-ToDiTA system against that of the ToDiTA system in terms of accuracy performance gains and storage overheads.

In the experiment, the samples from three different data acquisition sessions were used. The samples from the first session represent the samples acquired from an enrolment phase (denote as S1). The samples from the remaining two sessions represent the samples acquired from subsequent verification phases (denote as S2 and S3, respectively). The experiment was conducted using the evaluation methodology described in Section 5.5. The experiment was repeated three times, each time using one of the three feature reconstruction methods (as described in Section 5.4.3) in turn. They are each denoted as ToDiTA, E-ToDiTA (AA) and E-ToDiTA (PA), respectively. Note that for the sake of clarity and simplicity, we have denoted the NA method as ToDiTA instead of E-ToDiTA (NA) since both systems are identical. These three experiments were further repeated for four times, each time applying the experiments to one of the four datasets (as described in Section 5.5.1) in turn.

The experimental results are presented in Table 5.7. As can be seen from the table, there are four sets of results, each set represents the results evaluated for one of the four datasets. In each set, there are three rows, and each row represents one of the three systems. Note that the EERs and sample sizes columns are further split into two columns, before adaptation and after adaptation. Before adaptation means that the samples from S1 are used to build the authentication models. After adaptation means that the samples from S2 and S3 are used to update the models.

From the table, it can be seen that the ToDiTA and E-ToDiTA systems perform the same before adaptation in all four cases of datasets. This is because E-ToDiTA performs in the same manner as ToDiTA when only the samples from

Algorithm 5.1: Evaluation method for E-ToDiTA

Input: Dataset S with B number of subjects, $\{d_1, d_2, \ldots, d_B\}$, number of data acquisition sessions K with N samples per session, training sample ratio R, classifier C, reuse ratio ω, feature reconstruction method F

Output: Accuracy performance of the model for K sessions $\{P_1, P_2, \ldots, P_K\}$

for $b = 1$ **to** B **do**
  $S^+ \leftarrow$ initialise the legitimate subject set $\{d_b\}$
  $S^- \leftarrow$ initialise the illegitimate subjects set $S - \{d_b\}$
  Split S^+ into K subsets with N samples each, $S_k^+ = \{s_{k,1}^+, s_{k,2}^+, \ldots, s_{k,N}^+\}$
  $r \leftarrow$ compute the number of training samples, $\lfloor N \times R \rceil$
  for $k = 1$ **to** K **do**
    $S_{tr}^+ \leftarrow$ initialise the legitimate training set, $\{s_{k,1}^+, s_{k,2}^+, \ldots, s_{k,r}^+\}$
    $S_{ts}^+ \leftarrow$ initialise the legitimate testing set, $S_k^+ - S_{tr}^+$
    if $k == 1$ **then**
      $T_{tr}^k \leftarrow$ initialise the training set, S_{tr}^+
    else
      if F *used is the NA method* **then**
        $T_{tr}^k \leftarrow$ initialise the training set, T_{tr}^1
      else if F *used is the AA method* **then**
        $T_{tr}^k \leftarrow$ initialise the training set, $T_{tr}^{k-1} + S_{tr}^+$
      else if F *used is the PA method* **then**
        $u \leftarrow$ compute the number of samples in T_{tr}^{k-1} to be reused, $\lfloor |T_{tr}^{k-1}| \times \omega \rceil$
        $T_{tr}^k \leftarrow$ initialise the training set, the last u samples in $T_{tr}^{k-1} + S_{tr}^+$
    end
    Train C on T_{tr}^k to build model M_b
    $T_{ts}^k \leftarrow$ initialise the testing set, $S_{ts}^+ + S^-$
    $P_k \leftarrow P_k +$ (test the accuracy performance of M_b on T_{ts}^k)
  end
end
for $k = 1$ **to** K **do**
  $P_k \leftarrow P_k / B$
end

Datasets	Methods	EERs (%)		Accuracy Performance Gains(%)	Sample Sizes	
		Before Adaptation	After Adaptation		Before Adaptation	After Adaptation
RHU	ToDiTA	8.0	10.6	-32.5	4	4
	E-ToDiTA (AA)		3.5	56.25		11
	E-ToDiTA (PA)		2.9	63.75		4
MBK-E	ToDiTA	20.6	30.4	-47.57	14	14
	E-ToDiTA (AA)		20.3	1.46		42
	E-ToDiTA (PA)		18.0	12.62		14
MBK-S	ToDiTA	25.8	28.8	-11.63	14	14
	E-ToDiTA (AA)		18.2	29.46		42
	E-ToDiTA (PA)		16.7	35.27		14
MBK-LS	ToDiTA	27.6	30.4	-10.14	14	14
	E-ToDiTA (AA)		25.1	9.06		42
	E-ToDiTA (PA)		21.4	22.46		14

Table 5.7: EER values and samples sizes before and after adaptation for ToDiTA and E-ToDiTA applied on four datasets

S1 are available (for use to train the authentication models). However, when the samples from S2 and S3 are available (to update the models), the E-ToDiTA systems outperform the ToDiTA system. This means that the ideas implemented in E-ToDiTA are effective in adapting to the subjects' touch dynamics patterns as their patterns changes over time. In the following, we provide further discussion on the results.

The left half part of Table 5.7 shows the effect of model adaptation on the accuracy performances of the authentication models. From this part of the table, we can see that, in the case of ToDiTA, the EER values of after adaptation are higher than those before adaptation in all four cases of datasets. This is because ToDiTA build the models with the samples from S1 only, and the built models remain unchanged (even when more recent samples (i.e. the samples from S2 and S3 are available), causing the models to be less accurate in representing the subjects' more recent touch dynamics patterns, thus resulting in the higher EER values. For both cases of E-ToDiTA (AA) and E-ToDiTA (PA), however, we see the opposite trend. The EER values for after adaptation are lower than those before adaptation in all four cases of datasets. The reason is as follows. Unlike the ToDiTA system where the models are not updated, the E-ToDiTA systems update the model when the samples from S2 and S3 are available. The updated models can better capture the subjects' most recent touch dynamics patterns, thus achieving lower EER values.

From this part of the table, we can also observe that, in the case of ToDiTA, the accuracy performance gains are in negative values for all four cases of datasets. This also means that the authentication models suffer accuracy performance losses after model adaptation. What is striking in these results is that the accuracy performance losses for both the RHU and MBK-E datasets are noticeably greater than those for the MBK-S and MBK-LS datasets. For example, for the RHU and MBK-E datasets, their losses are -32.5% and -47.57%, respectively, whereas, for the MBK-S and MBK-LS datasets, their losses are only -11.63% and -10.14%, respectively. These results can be explained as follows. The subjects' touch dynamics patterns (i.e. input patterns, styles and/or speed) can change over time, and they stabilise after some time [188]. When the patterns have stabilised, the models could no longer reflect the stabilise patterns, thus resulting in accuracy performance losses. The lesser the amount of time needed for the patterns to stabilise, the sooner the models incur accuracy performance losses. The amount

of time needed for the patterns to stabilise is affected by the input string difficulty level. The lower the difficulty level, the lesser the amount of time needed for the patterns to stabilise. In the case of the RHU and MBK-E datasets, the input strings (consist of only lowercase characters) have lower difficulty levels than those in the case of the MBK-L and MBK-LS datasets (consist of a combination of digits, upper, lower and special case characters). So, the patterns require less amount of time to stabilise, resulting in the models incur accuracy performance losses sooner. For the reasons given above, in the case of ToDiTA, the accuracy performance losses for the RHU and MBK-E datasets are greater than those of the MBK-S and MBK-LS datasets.

The right half part of Table 5.7 examines the effect of model adaptation on the storage overheads. From this part of the table, we can make three observations, and these observations are true for all four cases of datasets. The first observation is that, for the case of ToDiTA, the sample sizes stay the same before and after adaptation. This is due to the fact that the samples from S2 and S3 are not used to update the authentication model. The second observation is that the sample sizes of E-ToDiTA (AA) increased by approximately threefold after adaptation. This is because of the combination of two reasons as follows. Firstly, the sample sizes of S2 and S3 are the same as that of S1. Secondly, with the ToDiTA (AA) system, the samples from S2 and S3 are appended to the samples in DSU without discarding any existing samples in DSU, resulting in a larger sample size. The third observation is that, unlike the case of the ToDiTA (AA) system, the sample sizes of E-ToDiTA (PA) did not increase after adaptation. This is because we set the reuse ratio to 0, which means that the existing samples in DSU are discarded before the samples from the subsequent sessions are used for model adaptation. Discarding all or some of the existing samples before model adaptation prevents the sample size from growing larger indefinitely, and this also avoids overfitting the model. This may be the reason why the accuracy performances gains for E-ToDiTA (PA) are higher than those for E-ToDiTA (AA) after adaptation.

Based on the above results and discussions, particularly taking into consideration of the finding that, E-ToDiTA (PA) are more efficient than E-ToDiTA (AA), both in terms of accuracy performance gains and storage overheads, and that, in a mobile device context, usually computational resources (i.e. storage space, computational speed and battery lifespan) are limited, we recommend the use of E-ToDiTA (PA) as the best system in this application context.

5.7 Chapter Summary

This chapter has presented the design and evaluation of the E-ToDiTA system. E-ToDiTA is designed to overcome the weakness in the ToDiTA system by adapting itself to any changes in a user's touch dynamics pattern. The design of E-ToDiTA uses three ideas. The first idea is to use a feature screening process to examine whether the acquired new touch dynamics data are suitable for use to update the authentication model. The second idea is to use a feature spooling process to reduce the number of times required to carry out the model adaptation processes, reducing computational overhead and improving usability. The third idea is to use a progressive adaptation method which uses a fewer number of samples to update the model, reducing the storage overhead incurred in storing the samples and the risk of overfitting the model.

We have conducted experiments to evaluate the effectiveness of these ideas built in E-ToDiTA and compared the results from E-ToDiTA to ToDiTA. Our experimental results show that E-ToDiTA outperforms the ToDiTA system in the condition when a user's touch dynamics pattern changes.

Chapter 6

Conclusions and Future Work

The focus of this thesis is on investigating how to best exploit the use of users' touch dynamics biometrics to strengthen authentication on mobile devices. This chapter summarises the work presented in this thesis, highlighting the contributions and discoveries from this research, and gives recommendations for future work.

6.1 Thesis Conclusions

Background Research
At the start of this research, the related literature on the characteristics of authentication on mobile environment and the authentication factors used in such environment were reviewed to identify the security threats associated with the use of the authentication factors. Based on the identified threats, a set of requirements that are used to guide the selection of authentication factors for the design of an authentication system that counters these threats were specified. Then, a comparative analysis of the authentication factors against the specified requirements was conducted. Upon completion of the analysis, it becomes apparent that touch dynamics biometrics is the preferable authentication factor for the design of the system.

Following the background study, the related work in the topic area of touch dynamics biometrics authentication were thoroughly investigated and critically analysed to identify their respective strengths and weaknesses, so that in our system design we could maintain the strengths while overcoming the weaknesses. The critical analysis has led to the conclusion that existing work mainly focused

on improving accuracy performance of the system, and the characteristics of mobile devices and the way they are typically used, which have implications on usability in addition to effectiveness, could have been better considered in the design of such system. The insights gained from the analysis have led to the design of two systems: (i) a touch dynamics based two-factor authentication (ToDiTA) system, which supports user authentication on mobile devices in a secure and usable manner, and (ii) an enhanced version of the ToDiTA system (E-ToDiTA), which has the capability to adapt itself to any changes in a user's touch dynamics pattern.

ToDiTA: A Secure and Usable System for Strengthening User Authentication on Mobile Devices

The ToDiTA system is designed to strengthen user authentication on mobile devices with as high accuracy, and as low cost, as possible. This is achieved by (i) integrating touch dynamics biometrics into a PIN-based authentication method that has a wide social acceptance; (ii) using descriptive statistical methods to extract additional features from the already-acquired features instead of using features that have to be captured by using additional device sensors; (iii) reducing the number of features by selecting and using the most important set of features; and (iv) reducing the touch dynamics data required to train the model by using one-class classification approach. To demonstrate the effectiveness and efficiency of the ToDiTA system, we have evaluated the system using simulation method. The implementation of the simulation-based evaluation was carried out using MATLAB programming platform. In the evaluation, we have investigated the effects of various parameter settings on the performances of the designed system. In addition, we have used a more comprehensive dataset to evaluate the ToDiTA system so that the conclusion drawn from the evaluation results are more conclusive. The evaluation of ToDiTA showed that by integrating the touch dynamics authentication method into the PIN-based authentication method, the ability to counter impersonation attacks is greatly enhanced. For example, if a PIN is compromised, the success rate of an impersonation attempt is drastically reduced from 100% (if only a 4-digit PIN is used) to 9.9% (if both the PIN and the touch dynamics are used). These results also indicate that the idea of using touch dynamics biometrics to support user authentication on a mobile device or application context is effective provided that users' touch dynamics patterns do

not change over time. However, in many real-life cases, users' touch dynamics patterns do change over time. To make ToDiTA effective under these different cases, we have designed an enhanced version of the ToDiTA system, producing the E-ToDiTA system.

E-ToDiTA: An Enhanced Version of the ToDiTA System
E-ToDiTA is different from ToDiTA in that it updates the authentication model with new touch dynamics data as soon as the new data becomes available. A number of measures or ideas have been used to make E-ToDiTA more effective and efficient. Firstly, it uses a feature screening process to examine whether the acquired new touch dynamics data are suitable for use to update the authentication model, reducing the likelihood that the updated model may incorrectly deviate from the owner's real touch dynamic pattern. Secondly, it uses a feature spooling process to reduce the number of times required to carry out the model adaptation processes, reducing computational overhead and improving usability. Thirdly, it uses a progressive adaptation method which uses a fewer number of samples to update the model, reducing the storage capacity required to store the samples and the risk of overfitting the model. The performance of the E-ToDiTA system has been evaluated and compared against that of ToDiTA. The evaluation results show that E-ToDiTA is more effective without introducing excessive overhead to a mobile device. E-ToDiTA improves the accuracy performance by an average of 33.53% in comparison with ToDiTA, but with virtually zero increase in overhead cost.

6.2 Future Work

The following gives recommendations for future research.

Accuracy Performance Enhancement
The accuracy performance achieved by the E-ToDiMA system can be further improved. There are a number of ways of achieving this. For example, additional features may be used, such as those extracted by using contextual information (e.g. login time, frequency and location). Representing features in a different form may also worth further investigation. For example, by plotting the feature values in a graph and then use the graph in the form of an image as the features

to build an authentication model. Alternatively, instead of searching for features manually, we could automate this process by using state-of-the-art deep learning techniques. These techniques have been widely used in the field of computer vision to automatically extract representative features from image data [225–227]. Perhaps they can also be used to automatically extract representative features from touch dynamics data to build an accurate model.

Further Performance Evaluation

The E-ToDiMA system can be further evaluated to further validate the scalability and feasibility of the system. This can be done in the following ways: (i) evaluate the system against more complex threat scenarios (e.g. when the impersonator knows both the PIN and touch dynamics pattern of the owner); (ii) investigate the energy consumption and computational overhead introduced by the system on devices equipped with different CPU speed, battery and storage capacity; (iii) extending the experiment to cover more use cases (e.g. to include textual passcodes as input string); (iv) evaluate the system against more realistic scenarios (e.g. when the input string is chosen by the subjects); (v) conduct experiments on an even larger subject size, and the subjects should ideally be recruited from people across different geographical locations; and (vi) acquire a more comprehensive dataset (e.g. data should be acquired over more than ten sessions spread over a longer period and acquired via device with different screen sizes) to evaluate the system.

Supporting Dynamic Identification Mode

In the research presented in this thesis, it is assumed that a mobile device is used solely by the owner and not shared among multiple users. Therefore, the E-ToDiTA system is designed for a verification mode (one-to-one matching) instead of an identification mode (one-to-many matching). However, in some application scenarios, e.g. a mobile device sharing scenario, there is a need to establish an authentication credential (e.g. an ID, a PIN and touch dynamics data) for an unknown guest. In such a scenario, the identification mode could be a better option.

An identification mode can be implemented on a mobile device in such a way that when a guest is observed, a privacy protection mechanism will be activated. The mechanism prevents unauthorised access or modification to the owner's data,

files and system settings. After the guest leaves and the device returns to the owner's possession, the mechanism will be deactivated, returning full access of the device to the owner. If an impersonator is detected, the device will be locked, and the owner will be alerted through email notification.

Implementing an effective and robust authentication system that supports the identification as well as verification mode on a mobile device is not a straightforward task. Unlike the other biometrics identification systems (e.g. forensic investigation or intrusion detection systems) that involve searching for a matched identity across a large number of subjects/classes, the identification task in a mobile device is usually limited to three classes, i.e. the owner, guest and impersonator. Despite the much smaller number of classes involved, which may seems to have reduced the complexity of the identification task, two issues need to be resolved: (i) the touch dynamics data for the owner need to be incrementally acquired, and (ii) the touch dynamics data for the guest and impersonator are often very limited or even unavailable, so data classification operation carried out to identify the owner/guest/impersonator is rather challenging and needs future research.

Bibliography

[1] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian, "Mobile cloud computing: A survey, state of art and future directions," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 133–143, Apr. 2014.

[2] "Blackboard app for students." `http://www.blackboard.com/mobile-learning/blackboard-app.html`. [Online; accessed 06-August-2018].

[3] "Coursera mobile app." `https://www.coursera.org/about/mobile`. [Online; accessed 06-August-2018].

[4] "Udacity for mobile." `https://www.udacity.com/mobile`. [Online; accessed 06-August-2018].

[5] S. Kemp, "Number of social media users passes 3 billion with no signs of slowing." `https://thenextweb.com/contributors/2017/08/07/number-social-media-users-passes-3-billion-no-signs-slowing/`, 2017. [Online; accessed 06-August-2018].

[6] Y. Wang, I.-R. Chen, and D.-C. Wang, "A survey of mobile cloud computing applications: Perspectives and challenges," *Wirel. Pers. Commun.*, vol. 80, no. 4, pp. 1607–1623, Feb. 2015.

[7] T. Wijman, "Mobile revenues account for more than 50% of the global games market as it reaches $137.9 billion in 2018." `https://newzoo.com/insights/articles/global-games-market-reaches-137-9-billion-in-2018-mobile-games-take-half/`, 2018. [Online; accessed 05-August-2018].

[8] "Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021 white paper." `https://www.cisco.com/c/en/us/`

solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html. [Online; accessed 06-August-2018].

[9] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," in *Proceedings of the 4th USENIX Conference on Offensive Technologies*, WOOT'10, pp. 1–7, 2010.

[10] N. H. Zakaria, D. Griffiths, S. Brostoff, and J. Yan, "Shoulder surfing defence for recall-based graphical passwords," in *Proceedings of the Seventh ACM Symposium on Usable Privacy and Security*, SOUPS '11, pp. 6:1–6:12, 2011.

[11] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "ACCessory: Password inference using accelerometers on smartphones," in *Proceedings of the Twelfth ACM Workshop on Mobile Computing Systems & Applications*, HotMobile '12, pp. 9:1–9:6, 2012.

[12] A. J. Aviv, B. Sapp, M. Blaze, and J. M. Smith, "Practicality of accelerometer side channels on smartphones," in *Proceedings of the 28th ACM Annual Computer Security Applications Conference*, ACSAC '12, pp. 41–50, 2012.

[13] Z. Xu, K. Bai, and S. Zhu, "TapLogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors," in *Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WISEC '12, pp. 113–124, 2012.

[14] H. Khan, A. Atwater, and U. Hengartner, "Itus: An implicit authentication framework for Android," in *Proceedings of the 20th ACM Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pp. 507–518, 2014.

[15] Hanul Sieger, Niklas Kirschnick, and Sebastian Möller, "Poster: User preferences for biometric authentication methods and graded security on mobile phones." https://cups.cs.cmu.edu/soups/2010/posters/16.pdf, 2010.

[16] A. De Luca, A. Hang, E. von Zezschwitz, and H. Hussmann, "I feel like I'm taking selfies all day!: Towards understanding biometric authentication on smartphones," in *Proceedings of the 33rd ACM Annual Conference on*

Human Factors in Computing Systems, CHI '15, (New York, NY, USA), pp. 1411–1414, 2015.

[17] Y. H. Park, D. N. Tien, H. C. Lee, K. R. Park, E. C. Lee, S. M. Kim, and H. C. Kim, "A multimodal biometric recognition of touched fingerprint and finger-vein," in *Proceedings of the International Conference on Multimedia and Signal Processing*, vol. 1, pp. 247–250, May 2011.

[18] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: security and privacy concerns," *IEEE Security Privacy*, vol. 99, no. 2, pp. 33–42, Mar. 2003.

[19] 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, Jan 2004.

[20] A. Jain, L. Hong, and S. Pankanti, "Biometric identification," *Communications of the ACM*, vol. 43, no. 2, pp. 90–98, Feb. 2000.

[21] P. Campisi, E. Maiorana, M. Lo Bosco, and A. Neri, "User authentication using keystroke dynamics for cellular phones," *IET Signal Processing*, vol. 3, no. 4, pp. 333–341, 2009.

[22] T. Feng, X. Zhao, B. Carbunar, and W. Shi, "Continuous mobile authentication using virtual key typing biometrics," in *Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, TrustCom'13, pp. 1547–1552, July 2013.

[23] N. Scheffer, L. Ferrer, A. Lawson, Y. Lei, and M. McLaren, "Recent developments in voice biometrics: Robustness and high accuracy," in *Proceedings of the IEEE International Conference on Technologies for Homeland Security*, HST'13, pp. 447–452, Nov. 2013.

[24] S. G. Kong, J. Heo, B. R. Abidi, J. Paik, and M. A. Abidi, "Recent advances in visual and infrared face recognition – a review," *Computer Vision and Image Understanding*, vol. 97, no. 1, pp. 103–135, Jan. 2005.

[25] X. Chen, P. J. Flynn, and K. W. Bowyer, "IR and visible light face recognition," *Computer Vision and Image Understanding*, vol. 99, no. 3, pp. 332 – 358, Sept. 2005.

[26] C. Shen, Y. Zhang, X. Guan, and R. Maxion, "Performance analysis of touch-interaction behavior for active smartphone authentication," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 3, pp. 498–513, Mar. 2016.

[27] S. Cimato, R. Sassi, and F. Scotti, "Biometrics and privacy," *Recent Patents on Computer Science*, vol. 1, no. 2, pp. 98–109, June 2008.

[28] G. Perrucci, F. Fitzek, G. Sasso, W. Kellerer, and J. Widmer, "On the impact of 2g and 3g network usage for mobile phones' battery life," in *Proceedings of the European Wireless Conference*, EW'09, pp. 255–259, May 2009.

[29] P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen, "A survey on touch dynamics authentication in mobile devices," *Computers & Security*, vol. 59, pp. 210–235, June 2016.

[30] P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen, "TDAS: A touch dynamics based multi-factor authentication solution for mobile devices," *International Journal of Pervasive Computing and Communications*, vol. 12, no. 1, pp. 127–153, Apr. 2016.

[31] P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen, "Recognizing your touch: Towards strengthening mobile device authentication via touch dynamics integration," in *Proceedings of the 13th ACM International Conference on Advances in Mobile Computing and Multimedia*, MoMM '15, pp. 108–116, 2015.

[32] P. S. Teh, N. Zhang, A. B. J. Teoh, K. Chen, and Q. Shi, "Strengthening user authentication on touchscreen mobile devices with user's touch dynamics pattern," *International Journal of Human-Computer Studies*.

[33] Mobile Working Group, "Security guidance for critical areas of mobile computing," tech. rep., Cloud Security Alliance, Nov. 2012.

[34] S. Uellenbeck, M. Dürmuth, C. Wolf, and T. Holz, "Quantifying the security of graphical passwords: The case of Android unlock patterns," in *Proceedings of the ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pp. 161–172, 2013.

[35] S. Almuairfi, P. Veeraraghavan, and N. Chilamkurti, "A novel image-based implicit password authentication system (IPAS) for mobile and non-mobile devices," *Mathematical and Computer Modelling*, vol. 58, no. 12, pp. 108–116, July 2013.

[36] M. Martinez-Diaz, J. Fierrez, C. Martin-Diaz, and J. Ortega-Garcia, "DooDB: A graphical password database containing doodles and pseudo-signatures," in *Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition*, ICFHR '10, pp. 339–344, 2010.

[37] A. C. Tao Hai, "Pass-Go: A proposal to improve the usability of graphical passwords," *International Journal of Network Security*, vol. 7, no. 2, pp. 273–292, 2008.

[38] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon, "PassPoints: Design and longitudinal evaluation of a graphical password system," *Int. J. Hum.-Comput. Stud.*, vol. 63, no. 1-2, pp. 102–127, July 2005.

[39] T. Takada and H. Koike, "Awase-E: Image-based authentication for mobile phones using user's favorite images," in *Human-Computer Interaction with Mobile Devices and Services* (L. Chittaro, ed.), no. 2795 in Lecture Notes in Computer Science, pp. 347–351, Springer Berlin Heidelberg, Jan. 2003.

[40] Z. Li, Q. Sun, Y. Lian, and D. D. Giusto, "A secure image-based authentication scheme for mobile devices," in *Advances in Intelligent Computing* (D.-S. Huang, X.-P. Zhang, and G.-B. Huang, eds.), no. 3645 in Lecture Notes in Computer Science, pp. 751–760, Springer Berlin Heidelberg, Jan. 2005.

[41] K. Kawagoe, S. Sakaguchi, Y. Sakon, and H.-H. Huang, "Tag association based graphical password using image feature matching," in *Database Systems for Advanced Applications* (S.-g. Lee, Z. Peng, X. Zhou, Y.-S. Moon, R. Unland, and J. Yoo, eds.), no. 7239 in Lecture Notes in Computer Science, pp. 282–286, Springer Berlin Heidelberg, Jan. 2012.

[42] Z. Wang, J. Jing, and L. Li, "Time evolving graphical password for securing mobile devices," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, ASIA CCS '13, pp. 347–352, 2013.

[43] X. Suo, Y. Zhu, and G. S. Owen, "Graphical passwords: a survey," in *Proceedings of the 21st Annual Computer Security Applications Conference*, ACSAC '05, pp. 462–472, Dec 2005.

[44] A. Sethi, O. Manzoor, and T. Sethi, "User authentication on mobile devices," tech. rep., Cigital, 2012.

[45] F. Aloul, S. Zahidi, and W. El-Hajj, "Two factor authentication using mobile phones," in *Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications*, AICCSA'09, pp. 641–644, May 2009.

[46] H. Saevanee, N. L. Clarke, and S. M. Furnell, "Multi-modal behavioural biometric authentication for mobile devices," in *Information Security and Privacy Research* (D. Gritzalis, S. Furnell, and M. Theoharidou, eds.), no. 376 in IFIP Advances in Information and Communication Technology, pp. 465–474, Springer Berlin Heidelberg, 2012.

[47] Mahnoush Babaeizadeh, Majid Bakhtiari, and Mohd Aizaini Maarof, "Authentication method through keystrokes measurement of mobile users in cloud environment," *Int. J. Advance Soft Compu. Appl*, vol. 6, no. 3, pp. 94–112, Nov. 2014.

[48] D. DeFigueiredo, "The case for mobile two-factor authentication," *IEEE Security Privacy*, vol. 9, no. 5, pp. 81–85, Sept. 2011.

[49] A. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 846–859, May 2000.

[50] "Use Touch ID on iPhone and iPad." `https://support.apple.com/en-gb/HT201371`. [Online; accessed 28-August-2018].

[51] M. Ayad, M. Taher, and A. Salem, "Real-time mobile cloud computing: A case study in face recognition," in *Proceedings of the 28th International Conference on Advanced Information Networking and Applications Workshops*, WAINA'14, pp. 73–78, May 2014.

[52] P. Tresadern, T. Cootes, N. Poh, P. Matejka, A. Hadid, C. Levy, C. McCool, and S. Marcel, "Mobile biometrics: Combined face and voice verification for

a mobile platform," *IEEE Pervasive Computing*, vol. 12, no. 1, pp. 79–87, Jan. 2013.

[53] M. Gargi, J. J. S. Rani, M. Ramiah, N. T. N. Babu, A. A. Fathima, and V. Vaidehi, "Mobile authentication using iris biometrics," in *Networked Digital Technologies* (R. Benlamri, ed.), no. 294 in Communications in Computer and Information Science, pp. 332–341, Springer Berlin Heidelberg, Jan. 2012.

[54] J.-S. Kang, "Mobile iris recognition systems: An emerging biometric technology," *Procedia Computer Science*, vol. 1, no. 1, pp. 475–484, May 2010.

[55] D. Currie, "Shedding some light on voice authentication," Tech. Rep. GSEC- V1.4b, SANS Institute, 2003.

[56] H. Crawford, K. Renaud, and T. Storer, "A framework for continuous, transparent mobile device authentication," *Computers & Security*, vol. 39, Part B, pp. 127–136, Nov. 2013.

[57] M. Derawi, C. Nickel, P. Bours, and C. Busch, "Unobtrusive user-authentication on mobile phones using biometric gait recognition," in *Proceedings of the Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, IIH-MSP'10, pp. 306–311, Oct. 2010.

[58] J. Guna, E. Stojmenova, A. Lugmayr, I. Humar, and M. Pogačnik, "User identification approach based on simple gestures," *Multimedia Tools and Applications*, vol. 71, no. 1, pp. 179–194, Aug. 2013.

[59] J. Guerra-Casanova, C. Sánchez-Ávila, G. Bailador, and A. d. S. Sierra, "Authentication in mobile devices through hand gesture recognition," *International Journal of Information Security*, vol. 11, no. 2, pp. 65–83, Jan. 2012.

[60] S. Lee, K. Song, and J. Choi, "Access to an automated security system using gesture-based passwords," in *Proceedings of the 15th International Conference on Network-Based Information Systems*, NBiS'12, pp. 760–765, Sept. 2012.

[61] J. Guerra-Casanova, C. Sánchez-Ávila, V. Jara-Vera, A. de Santos-Sierra, and G. Bailador, "Architectures to Implement In-Air Signature Mobile Authentication to Increase the Security of E-Commerce Applications and Opinion of End Users," in *Digital Enterprise and Information Systems* (E. Ariwa and E. El-Qawasmeh, eds.), no. 194 in Communications in Computer and Information Science, pp. 478–492, Springer Berlin Heidelberg, 2011.

[62] Y. Niu and H. Chen, "Gesture authentication with touch input for mobile devices," in *Security and Privacy in Mobile Information and Communication Systems* (R. Prasad, K. Farkas, A. U. Schmidt, A. Lioy, G. Russello, and F. L. Luccio, eds.), no. 94 in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 13–24, Springer Berlin Heidelberg, 2012.

[63] K. Abhishek, S. Roshan, A. Kumar, and R. Ranjan, "A comprehensive study on two-factor authentication with one time passwords," in *Computer Networks & Communications (NetCom)* (N. Chaki, N. Meghanathan, and D. Nagamalai, eds.), no. 131 in Lecture Notes in Electrical Engineering, pp. 405–415, Springer New York, Jan. 2013.

[64] "Stronger security for your Google account." `https://www.google.com/landing/2step/`, 2018. [Online; accessed 22-June-2018].

[65] Song, Andrew, "Introducing login approvals." `https://www.facebook.com/notes/facebook-engineering/introducing-login-approvals/10150172618258920`, 2011. [Online; accessed 09-July-2015].

[66] "How to use login verification." `https://help.twitter.com/en/managing-your-account/two-factor-authentication`. [Online; accessed 22-June-2018].

[67] "How to enable two-step verification." `https://www.dropbox.com/help/security/enable-two-step-verification`. [Online; accessed 22-June-2018].

[68] R. Schlöglhofer and J. Sametinger, "Secure and usable authentication on

mobile devices," in *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, MoMM'12, pp. 257–262, ACM, 2012.

[69] William E. Burr, Donna F. Dodson, Elaine M. Newton, Ray A. Perlner, W. Timothy Polk, Sarbari Gupta, Emad A. Nabbus, and William E. Burr, "Electronic authentication guideline," tech. rep., National Institute of Standards and Technology (NIST), Aug. 2013.

[70] J. Bonneau, C. Herley, P. C. v. Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *Proceedings of the IEEE Symposium on Security and Privacy*, SP '12, pp. 553–567, 2012.

[71] "Four barriers to adopting strong authentication," tech. rep., Nok Nok Labs, July 2014.

[72] A. K. Das, "Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards," *IET Information Security*, vol. 5, no. 3, pp. 145–151, Sept. 2011.

[73] S. Liu and M. Silverman, "A practical guide to biometric security technology," *IT Professional*, vol. 3, no. 1, pp. 27–32, Jan. 2001.

[74] W. L. Bryan and N. Harter, "Studies in the physiology and psychology of the telegraphic language," *Psychological Review*, vol. 4, no. 1, pp. 27–53, 1897.

[75] R. S. Gaines, W. Lisowski, S. J. Press, and N. Shapiro, "Authentication by keystroke timing: Some preliminary results," Tech. Rep. R-2526-NSF, Rand Corporation, 1980.

[76] S. Bleha, C. Slivinsky, and B. Hussien, "Computer-access security systems using keystroke dynamics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 12, pp. 1217–1222, 1990.

[77] M. S. Obaidat, "A verification methodology for computer systems users," in *Proceedings of the ACM Symposium on Applied Computing*, SAC'95, pp. 258–262, 1995.

[78] S. Cho, C. Han, D. H. Han, and H.-I. Kim, "Web-based keystroke dynamics identity verification using neural network," *Journal of Organizational Computing and Electronic Commerce*, vol. 10, no. 4, pp. 295–307, 2000.

[79] J. C. Stewart, J. V. Monaco, S.-H. Cha, and C. C. Tappert, "An investigation of keystroke and stylometry traits for authenticating online test takers," in *Proceedings of the International Joint Conference on Biometrics*, IJCB'11, pp. 1–7, 2011.

[80] H. Crawford, "Keystroke dynamics: Characteristics and opportunities," in *Proceedings of the Eighth International Conference on Privacy Security and Trust*, PST '10, pp. 205–212, 2010.

[81] M. Karnan, M. Akila, and N. Krishnaraj, "Biometric personal authentication using keystroke dynamics: A review," *Applied Soft Computing*, vol. 11, no. 2, pp. 1565–1573, 2011.

[82] P. S. Teh, A. B. J. Teoh, and S. Yue, "A survey of keystroke dynamics biometrics," *The Scientific World Journal*, vol. 2013, p. e408280, Nov. 2013.

[83] N. L. Clarke and S. M. Furnell, "Authenticating mobile phone users using keystroke analysis," *International Journal of Information Security*, vol. 6, no. 1, pp. 1–14, 2007.

[84] P. Saravanan, S. Clarke, D. H. P. Chau, and H. Zha, "LatentGesture: Active user authentication through background touch analysis," in *Proceedings of the Second International Symposium of Chinese CHI*, Chinese CHI '14, pp. 110–113, 2014.

[85] D. Buschek, A. De Luca, and F. Alt, "Improving accuracy, applicability and usability of keystroke biometrics on mobile touchscreen devices," in *Proceedings of the 33rd ACM Annual Conference on Human Factors in Computing Systems*, CHI '15, pp. 1393–1402, 2015.

[86] L. Zhou, Y. Kang, D. Zhang, and J. Lai, "Harmonized authentication based on ThumbStroke dynamics on touch screen mobile phones," *Decision Support Systems*, vol. 92, pp. 14–24, Dec. 2016.

[87] A. Buriro, S. Gupta, and B. Crispo, "Evaluation of motion-based touch-typing biometrics for online banking," in *Proceedings of the International*

Conference of the Biometrics Special Interest Group, BIOSIG '17, pp. 1–5, Sept. 2017.

[88] T. Connie, A. Teoh, M. Goh, and D. Ngo, "PalmHashing: a novel approach for cancelable biometrics," *Information Processing Letters*, vol. 93, no. 1, pp. 1 – 5, Jan. 2005.

[89] A. Arif, M. Pahud, K. Hinckley, and W. Buxton, "A tap and gesture hybrid method for authenticating smartphone users," in *Proceedings of the 15th ACM International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pp. 486–491, 2013.

[90] W. Meng, D. Wong, S. Furnell, and J. Zhou, "Surveying the development of biometric user authentication on mobile phones," *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, pp. 1268–1293, 2015.

[91] T.-Y. Chang, C.-J. Tsai, and J.-H. Lin, "A graphical-based password keystroke dynamic authentication system for touch screen handheld mobile devices," *Journal of Systems and Software*, vol. 85, no. 5, pp. 1157–1165, May 2012.

[92] K. Chen, "Towards better making a decision in speaker verification," *Pattern Recognition*, vol. 36, no. 2, pp. 329–346, Feb. 2003.

[93] L. Cai and H. Chen, "TouchLogger: Inferring keystrokes on touch screen from smartphone motion," in *Proceedings of the 6th USENIX Conference on Hot Topics in Security*, HotSec'11, pp. 9–9, 2011.

[94] L. Findlater, J. O. Wobbrock, and D. Wigdor, "Typing on flat glass: Examining ten-finger expert typing patterns on touch surfaces," in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 2453–2462, 2011.

[95] D. Damopoulos, G. Kambourakis, and S. Gritzalis, "From keyloggers to touchloggers: Take the rough with the smooth," *Computers & Security*, vol. 32, no. Supplement C, pp. 102–114, Feb. 2013.

[96] M. Ehatisham-ul Haq, M. A. Azam, J. Loo, K. Shuang, S. Islam, U. Naeem, and Y. Amin, "Authentication of smartphone users based on activity recognition and mobile sensing," *Sensors*, vol. 17, no. 9, p. 2043, Sept. 2017.

[97] H. Çeker and S. Upadhyaya, "Adaptive techniques for intra-user variability in keystroke dynamics," in *Proceedings of the 8th IEEE International Conference on Biometrics Theory, Applications and Systems*, BTAS'16, pp. 1–6, Sept. 2016.

[98] M. Harbach, E. v. Zezschwitz, A. Fichtner, A. D. Luca, and M. Smith, "It's a hard lock life: A field study of smartphone (un)locking behavior and risk perception," in *Proceedings of the 10th USENIX Symposium On Usable Privacy and Security*, SOUPS'14, pp. 213–230, 2014.

[99] S. Egelman, S. Jain, R. S. Portnoff, K. Liao, S. Consolvo, and D. Wagner, "Are you ready to lock?," in *ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pp. 750–761, 2014.

[100] M. Harbach, A. De Luca, N. Malkin, and S. Egelman, "Keep on lockin' in the free world: A multi-national comparison of smartphone locking," in *Proceedings of the 2016 ACM CHI Conference on Human Factors in Computing Systems*, CHI '16, pp. 4823–4827, 2016.

[101] N. Malkin, M. Harbach, A. De Luca, and S. Egelman, "The anatomy of smartphone unlocking: Why and how Android users around the world lock their phones," *GetMobile: Mobile Comp. and Comm.*, vol. 20, no. 3, pp. 42–46, Jan. 2017.

[102] H. Crawford and E. Ahmadzadeh, "Authentication on the go: Assessing the effect of movement on mobile device keystroke dynamics," in *Proceedings of the Thirteenth USENIX Symposium on Usable Privacy and Security*, SOUPS'17, pp. 163–173, 2017.

[103] T. Samura, M. Izumi, and H. Nishimura, "Flick input authentication in Japanese free text entry on smartphones," in *Proceedings of the 53rd Annual Conference of the Society of Instrument and Control Engineers of Japan*, SICE '14, pp. 1348–1353, Sept. 2014.

[104] U. A. Johansen, *Keystroke dynamics on a device with touch screen.* PhD thesis, Gjøvik University College, 2012.

[105] H. Seo, E. Kim, and H. Kang, "A novel biometric identification based on a user's input pattern analysis for intelligent mobile devices," *International Journal of Advanced Robotic Systems*, vol. 9, no. 2, p. 46, 2012.

[106] A. Serwadda, V. Phoha, and Z. Wang, "Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms," in *Proceeding of the Sixth IEEE International Conference on Biometrics: Theory, Applications and Systems*, BTAS'13, pp. 1–8, Sept. 2013.

[107] W. Meng, D. S. Wong, and L.-F. Kwok, "The effect of adaptive mechanism on behavioural biometric based mobile phone authentication," *Information Management & Computer Security*, vol. 22, no. 2, pp. 155–166, 2014.

[108] C. Shen, Y. Zhang, Z. Cai, T. Yu, and X. Guan, "Touch-interaction behavior for continuous user authentication on smartphones," in *Proceedings of the International Conference on Biometrics*, ICB'15, pp. 157–162, May 2015.

[109] M. Trojahn, F. Arndt, and F. Ortmeier, "Authentication with time features for keystroke dynamics on touchscreens," in *Communications and Multimedia Security* (B. D. Decker, J. Dittmann, C. Kraetzer, and C. Vielhauer, eds.), no. 8099 in Lecture Notes in Computer Science, pp. 197–199, Springer Berlin Heidelberg, 2013.

[110] Z. Cai, C. Shen, M. Wang, Y. Song, and J. Wang, "Mobile authentication through touch-behavior features," in *Biometric Recognition* (Z. Sun, S. Shan, G. Yang, J. Zhou, Y. Wang, and Y. Yin, eds.), no. 8232 in Lecture Notes in Computer Science, pp. 386–393, Springer International Publishing, 2013.

[111] N. Alotaibi, E. P. Bruno, M. Coakley, A. Gazarov, V. Monaco, S. Winard, F. Witkowski, A. Copeland, P. Nebauer, C. Keene, and J. Williams, "Text input biometric system design for handheld devices," in *Proceedings of Student-Faculty Research Day*, pp. B7.1–B7.8, May 2014.

[112] M. Rybnicek, C. Lang-Muhr, and D. Haslinger, "A roadmap to continuous biometric authentication on mobile devices," in *Proceedings of the International Conference Wireless Communications and Mobile Computing*, IWCMC'14, pp. 122–127, Aug. 2014.

[113] Y. Meng, D. S. Wong, R. Schlegel, and L.-f. Kwok, "Touch gestures based biometric authentication scheme for touchscreen mobile phones," in *Information Security and Cryptology* (M. Kutyowski and M. Yung, eds.),

no. 7763 in Lecture Notes in Computer Science, pp. 331–350, Springer Berlin Heidelberg, 2013.

[114] B. Draffin, J. Zhu, and J. Zhang, "KeySens: Passive user authentication through micro-behavior modeling of soft keyboard interaction," in *Mobile Computing, Applications, and Services* (G. Memmi and U. Blanke, eds.), no. 130 in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 184–201, Springer International Publishing, 2014.

[115] L. Sun, Y. Wang, B. Cao, P. S. Yu, W. Srisa-an, and A. D. Leow, "Sequential keystroke behavioral biometrics for mobile user identification via multi-view deep learning," in *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pp. 228–240, Springer, Cham, Sept. 2017.

[116] B. Taylor, "5 ways the smartphone is conquering the tablet." `http://www.pcworld.com/article/2889275/5-ways-the-smartphone-is-conquering-the-tablet.html`, 2015. [Online; accessed 07-April-2015].

[117] J. Gurary, Y. Zhu, N. Alnahash, and H. Fu, "Implicit authentication for mobile devices using typing behavior," in *Human Aspects of Information Security, Privacy, and Trust*, Lecture Notes in Computer Science, pp. 25–36, Springer International Publishing, July 2016.

[118] G. Kambourakis, D. Damopoulos, D. Papamartzivanos, and E. Pavlidakis, "Introducing touchstroke: Keystroke-based authentication system for smartphones," *Security and Communication Networks*, vol. 9, no. 6, pp. 542–554, 2016.

[119] B. Mayo, "Xcode 7 allows anyone to download, build and sideload iOS apps for free." `https://9to5mac.com/2015/06/10/xcode-7-allows-anyone-to-download-build-and-sideload-ios-apps-for-free/`, 2015. [Online; accessed 13-March-2018].

[120] "Android phone manufacturer market share - AppBrain." `https://www.appbrain.com/stats/top-manufacturers`. [Online; accessed 19-March-2018].

[121] "IDC: Smartphone OS market share." `https://www.idc.com/promo/smartphone-market-share/os`. [Online; accessed 19-March-2018].

[122] "Purchase and activation." `https://developer.apple.com/support/purchase-activation/`. [Online; accessed 13-March-2018].

[123] "Register for a Google play developer account." `https://support.google.com/googleplay/android-developer/answer/6112435?hl=en-GB`. [Online; accessed 13-March-2018].

[124] "Register as an app developer." `https://developer.microsoft.com/en-us/store/register`. [Online; accessed 03-March-2018].

[125] R. A. Maxion and K. S. Killourhy, "Keystroke biometrics with number-pad input," in *Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks*, DSN'10, pp. 201–210, 2010.

[126] H. Xu, Y. Zhou, and M. R. Lyu, "Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones," in *Proceedings of the USENIX Symposium On Usable Privacy and Security*, SOUP'14, pp. 187–198, 2014.

[127] N. Bartlow and B. Cukic, "Evaluating the reliability of credential hardening through keystroke dynamics," in *Proceedings of the 17th International Symposium on Software Reliability Engineering*, ISSRE'06, pp. 117 –126, Nov. 2006.

[128] H. Jagadeesan and M. S. Hsiao, "A novel approach to design of user re-authentication systems," in *Proceedings of the 3rd IEEE international conference on Biometrics: Theory, applications and systems*, BTAS'09, pp. 379–384, 2009.

[129] J. Angulo and E. Wstlund, "Exploring Touch-Screen Biometrics for User Identification on Smart Phones," in *Privacy and Identity Management for Life* (J. Camenisch, B. Crispo, S. Fischer-Hbner, R. Leenes, and G. Russello, eds.), no. 375 in IFIP Advances in Information and Communication Technology, pp. 130–143, Springer Berlin Heidelberg, 2012.

[130] K. R. Rao, V. P. K. Anne, U. S. Chand, V. Alakananda, and K. N. Rachana, "Inclination and pressure based authentication for touch devices," in *ICT*

and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India- Vol I (S. C. Satapathy, P. S. Avadhani, S. K. Udgata, and S. Lakshminarayana, eds.), no. 248 in Advances in Intelligent Systems and Computing, pp. 781–788, Springer International Publishing, 2014.

[131] A. Salem, D. Zaidan, A. Swidan, and R. Saifan, "Analysis of strong password using keystroke dynamics authentication in touch screen devices," in *Proceedings of the Cybersecurity and Cyberforensics Conference*, CCC'16, pp. 15–21, Aug. 2016.

[132] M. Trojahn, F. Arndt, and F. Ortmeier, "Authentication with keystroke dynamics on touchscreen keypads - effect of different n-graph combinations," in *Proceedings of the The Third International Conference on Mobile Services, Resources, and Users*, MOBILITY'13, pp. 114–119, Nov. 2013.

[133] H. Gascon, S. Uellenbeck, C. Wolf, and K. Rieck, "Continuous authentication on mobile devices by analysis of typing motion behavior," in *Lecture Notes in Informatics*, vol. 2014, pp. 1–12, 2014.

[134] K. Krombholz, T. Hupperich, and T. Holz, "Use the force: Evaluating force-sensitive authentication for mobile devices," in *Proceedings of the Twelfth USENIX Symposium on Usable Privacy and Security*, SOUPS'16, pp. 207–219, 2016.

[135] S. M. Kolly, R. Wattenhofer, and S. Welten, "A personal touch: Recognizing users based on touch screen behavior," in *Proceedings of the Third ACM International Workshop on Sensing Applications on Mobile Phones*, PhoneSense '12, pp. 1:1–1:5, 2012.

[136] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pp. 453–456, 2008.

[137] M. Antal and L. Z. Szab, "Keystroke dynamics on android platform," in *Proceedings of the 8th International Conference Interdisciplinarity in Engineering*, INTER-ENG'14, pp. 131–136, 2014.

[138] C. Giuffrida, K. Majdanik, M. Conti, and H. Bos, "I sensed it was you: Authenticating mobile users with sensor-enhanced keystroke dynamics," in *Detection of Intrusions and Malware, and Vulnerability Assessment* (S. Dietrich, ed.), no. 8550 in Lecture Notes in Computer Science, pp. 92–111, Springer International Publishing, July 2014.

[139] V.-D. Stanciu, R. Spolaor, M. Conti, and C. Giuffrida, "On the effectiveness of sensor-enhanced keystroke dynamics against statistical attacks," in *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, CODASPY '16, pp. 105–112, 2016.

[140] M. El-Abed, M. Dafer, and R. El Khayat, "RHU keystroke: A mobile-based benchmark for keystroke dynamics systems," in *Proceedings of the International Carnahan Conference on Security Technology*, ICCST'14, pp. 1–4, Oct. 2014.

[141] M. J. Coakley, J. V. Monaco, and C. C. Tappert, "Numeric-passcode keystroke biometric studies on smartphones," in *Proceedings of Student-Faculty Research Day*, (Pace University), pp. B4.1–B4.6, May 2015.

[142] C. Praher and M. Sonntag, "Applicability of keystroke dynamics as a biometric security feature for mobile touchscreen devices with virtualised keyboards," *International Journal of Information and Computer Security*, vol. 8, no. 1, pp. 72–91, Jan. 2016.

[143] Y. Meng, D. S. Wong, and L.-F. Kwok, "Design of touch dynamics based user authentication with an adaptive mechanism on mobile phones," in *Proceedings of the 29th ACM Annual Symposium on Applied Computing*, SAC '14, pp. 1680–1687, 2014.

[144] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon, "Biometric-rich gestures: A novel approach to authentication on multi-touch devices," in *Proceedings of the ACM Annual Conference on Human Factors in Computing Systems*, CHI '12, pp. 977–986, 2012.

[145] X. Huang, G. Lund, and A. Sapeluk, "Development of a typing behaviour recognition mechanism on Android," in *Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, TrustCom'12, pp. 1342–1347, June 2012.

[146] Z. Hao and Q. Li, "Towards user re-authentication on mobile devices via on-screen keyboard," in *Proceedings of the Fourth IEEE Workshop on Hot Topics in Web Systems and Technologies*, HotWeb'16, pp. 78–83, Oct. 2016.

[147] N. Zheng, K. Bai, H. Huang, and H. Wang, "You are how you touch: User verification on smartphones via tapping behaviors," in *Proceedings of the 22nd IEEE International Conference on Network Protocols*, ICNP'14, pp. 221–232, Oct. 2014.

[148] S.-H. Lee, J.-H. Roh, S. Kim, and S.-H. Jin, "A study on feature of keystroke dynamics for improving accuracy in mobile environment," in *Information Security Applications*, pp. 366–375, Springer International Publishing, Aug. 2016.

[149] L. Jain, J. V. Monaco, M. J. Coakley, and C. C. Tappert, "Passcode keystroke biometric performance on smartphone touchscreens is superior to that on hardware keyboards," *International Journal of Research in Computer Applications & Information Technology*, vol. 2, no. 4, pp. 29–33, Aug. 2014.

[150] M. J. Coakley, J. V. Monaco, and C. C. Tappert, "Keystroke biometric studies with short numeric input on smartphones," in *Proceedings of the 8th IEEE International Conference on Biometrics Theory, Applications and Systems*, BTAS'13, pp. 1–6, Sept. 2016.

[151] M. Wolff, "Behavioral biometric identification on mobile devices," in *Foundations of Augmented Cognition* (D. D. Schmorrow and C. M. Fidopiastis, eds.), no. 8027 in Lecture Notes in Computer Science, pp. 783–791, Springer Berlin Heidelberg, 2013.

[152] S. Sen and K. Muralidharan, "Putting 'pressure' on mobile authentication," in *Proceedings of the Seventh International Conference on Mobile Computing and Ubiquitous Networking*, ICMU'14, pp. 56–61, Jan. 2014.

[153] C.-J. Tasia, T.-Y. Chang, P.-C. Cheng, and J.-H. Lin, "Two novel biometric features in keystroke dynamics authentication systems for touch screen devices," *Security and Communication Networks*, vol. 7, no. 4, pp. 750–758, Apr. 2014.

[154] M. Martinez-Diaz, J. Fierrez, and J. Galbally, "The DooDB graphical password database: Data analysis and benchmark results," *IEEE Access*, vol. 1, pp. 596–605, 2013.

[155] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann, "Touch me once and I know it's you!: Implicit authentication based on touch screen patterns," in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 987–996, 2012.

[156] S. Dhage, P. Kundra, A. Kanchan, and P. Kap, "Mobile authentication using keystroke dynamics," in *Proceedings of the International Conference on Communication, Information Computing Technology*, ICCICT'15, pp. 1–5, Jan. 2015.

[157] M. Inoue and T. Ogawa, "One tap owner authentication on smartphones," in *Proceedings of the 15th ACM International Conference on Advances in Mobile Computing & Multimedia*, MoMM'17, pp. 22–28, 2017.

[158] X. L. Pham, T. H. Nguyen, and G. D. Chen, "Factors that impact quiz score: A study with participants in a mobile learning app," in *Proceedings of the 17th IEEE International Conference on Advanced Learning Technologies*, ICALT'17, pp. 103–105, July 2017.

[159] M. Antal and L. Nemes, "The MOBIKEY keystroke dynamics password database: Benchmark results," in *Software Engineering Perspectives and Application in Intelligent Systems* (R. Silhavy, R. Senkerik, Z. K. Oplatkova, P. Silhavy, and Z. Prokopova, eds.), no. 465 in Advances in Intelligent Systems and Computing, pp. 35–46, Springer International Publishing, 2016.

[160] K. W. Nixon, Y. Chen, Z.-H. Mao, and K. Li, "User classification and authentication for mobile device based on gesture recognition," in *Network Science and Cybersecurity* (R. E. Pino, ed.), no. 55 in Advances in Information Security, pp. 125–135, Springer New York, 2014.

[161] A. Buriro, B. Crispo, F. D. Frari, and K. Wrona, "Touchstroke: Smartphone user authentication based on touch-typing biometrics," in *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops* (V. Murino, E. Puppo, D. Sona, M. Cristani, and C. Sansone, eds.), no. 9281 in Lecture

Notes in Computer Science, pp. 27–34, Springer International Publishing, Sept. 2015.

[162] N. Jeanjaitrong and P. Bhattarakosol, "Feasibility study on authentication based keystroke dynamic over touch-screen devices," in *Proceedings of the 13th International Symposium on Communications and Information Technologies*, ISCIT'13, pp. 238–242, Sept. 2013.

[163] C. Shen, T. Yu, S. Yuan, Y. Li, and X. Guan, "Performance analysis of motion-sensor behavior for user authentication on smartphones," *Sensors*, vol. 16, no. 3, p. 345, Mar. 2016.

[164] M. Trojahn and F. Ortmeier, "Toward mobile authentication with keystroke dynamics on mobile phones and tablets," in *Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops*, WAINA'13, pp. 697–702, Mar. 2013.

[165] T. Hori, Y. Kita, K. Toyoda, N. Okazaki, and M. Park, "Empirical evaluation of rhythm-based authentication method for mobile devices," in *Advances in Network-Based Information Systems*, Lecture Notes on Data Engineering and Communications Technologies, pp. 529–538, Springer International Publishing, Aug. 2017.

[166] I. d. Mendizabal-Vazquez, D. de Santos-Sierra, J. Guerra-Casanova, and C. Sanchez-Avila, "Supervised classification methods applied to keystroke dynamics through mobile devices," in *Proceedings of the International Carnahan Conference on Security Technology*, ICCST'14, pp. 1–6, Oct. 2014.

[167] D. Soni, M. Hanmandlu, and H. C. Saini, "A machine learning approach for user authentication using touchstroke dynamics," in *Proceedings of First International Conference on Smart System, Innovations and Computing*, Smart Innovation, Systems and Technologies, pp. 391–410, Springer Singapore, 2018.

[168] G. Ho, "TapDynamics: Strengthening user authentication on mobile phones with keystroke dynamics," tech. rep., Stanford University, 2013.

[169] S. Alghamdi and L. Elrefaei, "Dynamic user verification using touch keystroke based on medians vector proximity," in *Proceedings of the 7th*

International Conference on Computational Intelligence, Communication Systems and Networks, CICSyN'15, pp. 121–126, June 2015.

[170] N. M. Al-Obaidi and M. M. Al-Jarrah, "Statistical keystroke dynamics system on mobile devices for experimental data collection and user authentication," in *Proceedings of the 9th International Conference on Developments in eSystems Engineering*, DeSE'16, pp. 123–129, Aug. 2016.

[171] W. Shi, J. Yang, Y. Jiang, F. Yang, and Y. Xiong, "SenGuard: Passive user identification on smartphones using multiple sensors," in *Proceedings of the IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications*, WiMOB'11, pp. 141–148, Oct. 2011.

[172] T. Shimshon, R. Moskovitch, L. Rokach, and Y. Elovici, "Clustering digraphs for continuously verifying users according to their typing patterns," in *Proceedings of the 26th IEEE Convention of Electrical and Electronics Engineers in Israel*, IEEEI'10, pp. 445–449, 2010.

[173] William F. Bond and Ahmed Awad E.A, "Touch-based static authentication using a virtual grid," in *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '15, pp. 129–134, 2015.

[174] J. Wu and Z. Chen, "An implicit identity authentication system considering changes of gesture based on keystroke behaviors," *International Journal of Distributed Sensor Networks*, vol. 2015, p. e470274, June 2015.

[175] T.-Y. Chang, C.-J. Tsai, W.-J. Tsai, C.-C. Peng, and H.-S. Wu, "A changeable personal identification number-based keystroke dynamics authentication system on smart phones," *Security and Communication Networks*, vol. 9, no. 15, pp. 2674–2685, May 2015.

[176] M. Antal and L. Z. Szabó, "An evaluation of one-class and two-class classification algorithms for keystroke dynamics authentication on mobile devices," in *Proceedings of the 20th IEEE International Conference on Control Systems and Computer Science*, CSCS'15, pp. 343–350, 2015.

[177] A. J. Aviv, J. T. Davin, F. Wolf, and R. Kuber, "Towards baselines for shoulder surfing on mobile authentication," in *Proceedings of the 33rd ACM*

Annual Computer Security Applications Conference, ACSAC '17, pp. 486–498, 2017.

[178] J. h. Roh, S. H. Lee, and S. Kim, "Keystroke dynamics for authentication in smartphone," in *Proceedings of the International Conference on Information and Communication Technology Convergence*, ICTC'16, pp. 1155–1159, Oct. 2016.

[179] R. Amin, T. Gaber, and G. ElTaweel, "Implicit authentication system for smartphones users based on touch data," in *Intelligent Data Analysis and Applications* (A. Abraham, X. H. Jiang, V. Snel, and J.-S. Pan, eds.), no. 370 in Advances in Intelligent Systems and Computing, pp. 251–262, Springer International Publishing, 2015.

[180] Y. Li, J. Yang, M. Xie, D. Carlson, H. G. Jang, and J. Bian, "Comparison of PIN- and pattern-based behavioral biometric authentication on mobile devices," in *Proceedings of the IEEE Military Communications Conference*, MILCOM'15, pp. 1317–1322, Oct. 2015.

[181] M. El-Abed, R. Giot, B. Hemery, and C. Rosenberger, "Evaluation of biometric systems: A study of users' acceptance and satisfaction," *International Journal of Biometrics*, vol. 4, no. 3, pp. 265–290, Jan. 2012.

[182] F. Schaub, R. Deyhle, and M. Weber, "Password entry usability and shoulder surfing susceptibility on different smartphone platforms," in *Proceedings of the 11th ACM International Conference on Mobile and Ubiquitous Multimedia*, MUM '12, pp. 13:1–13:10, 2012.

[183] E. von Zezschwitz, A. De Luca, and H. Hussmann, "Honey, I shrunk the keys: Influences of mobile devices on password composition and authentication performance," in *Proceedings of the 8th ACM Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, NordiCHI '14, pp. 461–470, 2014.

[184] N. Ben-Asher, N. Kirschnick, H. Sieger, J. Meyer, A. Ben-Oved, and S. Möller, "On the need for different security methods on mobile phones," in *Proceedings of the 13th ACM International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pp. 465–473, 2011.

[185] E. von Zezschwitz, P. Dunphy, and A. De Luca, "Patterns in the wild: A field study of the usability of pattern and pin-based authentication on mobile devices," in *Proceedings of the 15th ACM International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pp. 261–270, 2013.

[186] T. K. Ho, "Nearest neighbors in random subspaces," in *Advances in Pattern Recognition* (A. Amin, D. Dori, P. Pudil, and H. Freeman, eds.), no. 1451 in Lecture Notes in Computer Science, pp. 640–648, Springer Berlin Heidelberg, Aug. 1998.

[187] J. Wang, J. Tang, G. Xue, and D. Yang, "Towards energy-efficient task scheduling on smartphones in mobile crowd sensing systems," *Computer Networks*, vol. 115, pp. 100–109, Mar. 2017.

[188] B. Ngugi, M. Tremaine, and P. Tarasewich, "Biometric keypads: Improving accuracy through optimal PIN selection.," *Decision Support Systems*, vol. 50, no. 4, pp. 769–776, 2011.

[189] Kenneth N. Ross, *Sample design for educational survey research*. Evaluation in Education, Pergamon Press, 1978.

[190] B. Taylor, "Why smartphone screens are getting bigger: Specs reveal a surprising story." `http://www.pcworld.com/article/2455169/why-smartphone-screens-are-getting-bigger-specs-reveal-a-surprising-story.html`, 2014. [Online; accessed 16-February-2016].

[191] J. A. Robinson, V. M. Liang, J. A. M. Chambers, and C. L. MacKenzie, "Computer user verification using login string keystroke dynamics," *Ieee Transactions on Systems Man and Cybernetics Part a-Systems and Humans*, vol. 28, no. 2, pp. 236–241, 1998.

[192] E. Maiorana, P. Campisi, N. González-Carballo, and A. Neri, "Keystroke dynamics authentication for mobile phones," in *Proceedings of the ACM Symposium on Applied Computing*, SAC '11, pp. 21–26, 2011.

[193] Android Developers, "nanoTime - System." `https://developer.android.com/reference/java/lang/System.html#nanoTime()`, 2017. [Online; accessed 10-December-2017].

[194] Android Developers, "getSize - MotionEvent." `https://developer.android.com/reference/android/view/MotionEvent.html#getSize(int)`, 2017. [Online; accessed 10-December-2017].

[195] Android Developers, "getPressure - MotionEvent." `https://developer.android.com/reference/android/view/MotionEvent.html#getPressure(int)`, 2017. [Online; accessed 10-December-2017].

[196] B. Ngugi, B. K. Kahn, and M. Tremaine, "Typing biometrics: Impact of human learning on performance quality," *Journal of Data and Information Quality*, vol. 2, no. 2, pp. 11:1–11:21, 2011.

[197] Prem S. Mann, *Introductory Statistics*. Wiley, 9th edition ed., Feb. 2016.

[198] P. Juszczak, D. Tax, and B. Duin, "Feature scaling in support vector data description," in *Proc. ASCI*, pp. 95–102, Citeseer, 2002.

[199] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, no. 12, pp. 2270–2285, Dec. 2005.

[200] E. P. Xing, M. I. Jordan, and R. M. Karp, "Feature selection for high-dimensional genomic microarray data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 601–608, 2001.

[201] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[202] T. M. Cover and J. A. Thomas, *Elements of Information Theory 2nd Edition*. Hoboken, N.J: Wiley-Interscience, 2 edition ed., July 2006.

[203] G. Brown, A. Pocock, M.-J. Zhao, and M. Lujn, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 27–66, Jan. 2012.

[204] MathWorks, "Histogram bin counts - MATLAB histcounts." `https://uk.mathworks.com/help/matlab/ref/histcounts.html?requestedDomain=www.mathworks.com`, 2016. [Online; accessed 11-December-2016].

[205] C. Bellinger, S. Sharma, and N. Japkowicz, "One-class versus binary classification: Which and when?," in *Proceedings of the 11th International Conference on Machine Learning and Applications*, ICMLA'12, pp. 102–106, Dec. 2012.

[206] D. M. J. Tax, *One-class Classification*. Ph.D. thesis, Delft University of Technology, 2001.

[207] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, Jan. 2004.

[208] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.

[209] R. Simon, "Resampling strategies for model assessment and selection," in *Fundamentals of Data Mining in Genomics and Proteomics* (W. Dubitzky, M. Granzow, and D. Berrar, eds.), pp. 173–186, Springer US, 2007.

[210] D. Tax, "Ddtools, the data description toolbox for matlab, version 2.1.2." `https://www.tudelft.nl/ewi/over-de-faculteit/afdelingen/intelligent-systems/pattern-recognition-bioinformatics/pattern-recognition-laboratory/data-and-software/dd-tools/`, Jun 2015.

[211] R. Duin and E. Pekalska, "Prtools, a matlab toolbox for pattern recognition, version 5.3.1." `http://prtools.org`, Jun 2015.

[212] A. K. Jain, A. Ross, and S. Pankanti, "Biometrics: A tool for information security," *Trans. Info. For. Sec.*, vol. 1, no. 2, pp. 125–143, Nov. 2006.

[213] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, Jan. 2013.

[214] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," tech. rep., National Taiwan University, 2003.

[215] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," in *Encyclopedia of Database Systems* (L. LIU and M. T. ZSU, eds.), pp. 532–538, Springer US, 2009.

[216] X. Bao, U. Lee, I. Rimac, and R. R. Choudhury, "Dataspotting: Offloading cellular traffic via managed device-to-device data transfer at data spots," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 14, no. 3, pp. 37–39, Dec. 2010.

[217] E. Miluzzo, R. Cceres, and Y.-F. Chen, "Vision: Mclouds - computing on clouds of mobile devices," in *Proceedings of the Third ACM Workshop on Mobile Cloud Computing and Services*, MCS '12, pp. 9–14, 2012.

[218] J. H. Huh, H. Kim, R. B. Bobba, M. N. Bashir, and K. Beznosov, "On the memorability of system-generated PINs: Can chunking help?," in *Proceedings of the Eleventh USENIX Symposium On Usable Privacy and Security*, SOUPS'15, pp. 197–209, 2015.

[219] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[220] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern Recognition*, vol. 43, no. 1, pp. 5–13, Jan. 2010.

[221] A. Rattani, B. Freni, G. L. Marcialis, and F. Roli, "Template update methods in adaptive biometric systems: A critical review," in *Advances in Biometrics*, pp. 847–856, Springer, Berlin, Heidelberg, June 2009.

[222] K. N. Truong, T. Shihipar, and D. J. Wigdor, "Slide to X: Unlocking the potential of smartphone unlocking," in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pp. 3635–3644, 2014.

[223] B. Bajarin, "Apple's penchant for consumer security." `https://techpinions.com/apples-penchant-for-consumer-security/45122`, Apr. 2016. [Online; accessed 24-October-2017].

[224] A. Rattani, G. L. Marcialis, F. Roli, and F. Roli, "A multi-modal dataset, protocol and tools for adaptive biometric systems: A benchmarking study," *International Journal of Biometrics*, vol. 5, no. 3/4, 2013.

[225] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 253–256, May 2010.

[226] Y. Xu, T. Mo, Q. Feng, P. Zhong, M. Lai, and E. I. C. Chang, "Deep learning of feature representation with multiple instance learning for medical image analysis," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal*, ICASSP'14, pp. 1626–1630, May 2014.

[227] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 1988–1996, 2014.