# Continuous Touchscreen Biometrics: Authentication and Privacy Concerns




Martin Kostadinov Georgiev

Kellogg College

University of Oxford

A thesis submitted for the degree of

*Doctor of Philosophy*

Michaelmas 2022

# Acknowledgements


## Personal

First and foremost, I would like to thank my family, my supportive parents, Daniela and Kostadin, and my inspirational brother Ivan. You have been amazingly encouraging and allowed me to pursue my dreams. Thank you for everything you have done for me!

I want to thank all the friends I made during the past few years at Oxford. You made this challenging endeavour enjoyable, and I learned a lot from you. Thank you to Sebastian for brightening up the mundane moments in the office, the freshly baked bread and the wonderful travel memories. Anirudh, for the ridiculous projects we undertook and for pushing me to always be my best. Freddie, for the late nights at the gym pondering our next steps in the DPhil. Klaudia, for sharing delicious green tea while chatting about life during these demanding weekends at the office. George for being a great company when travelling and attending events at Oxford. Jack, for the good nights out and the exciting life stories.

I am thankful to my friends in Bulgaria — Ivaylo, Konstantin and Dimitar for being some of my oldest friends and staying together during every step of my educational journey — from primary school to a DPhil. Georgi, for the enjoyable hikes and informative fitness conversations. Petar, Ivan and Kristyan for being the people I always look forward to seeing and sharing ideas with back in Sofia.

I would like to extend my thanks to all my other friends scattered across the world. Katie, Hector, Carina, Manpreet, Robin, Vy, Andras, Federico, Parnavi, Nikita, Giovanni, Dabal, Summit, Isabella, Danail, Jon, Felix and everyone else I may forget. We may not always be in touch, but you have greatly enriched my life throughout these years.

I would like to express my sincere gratitude to Professor Andrew Simpson and Professor Kevin Butler for their time and thoughtful feedback in examining this thesis, which has significantly improved the quality of the work. I also want to extend my thanks to Professor Andrew Martin and Professor Michael Goldsmith for their invaluable feedback during the milestone examinations at Oxford, which helped shape the direction of this thesis.

Last but not least, I am grateful to my supervisor, Ivan Martinovic, for his support and guidance throughout this journey.

## Institutional

I would like to express my sincere gratitude to the Engineering and Physical Sciences Research Council (ESPRC) for their general financial and administrative support.

Thanks to the Centre for Doctoral Training (CDT) for organising this exciting program and helping students pursue excellence. Janet, I cannot thank you enough for your assistance during the final stages of this journey. Your support in both academic and non-academic settings has been invaluable, and I am so glad that we have now become friends.

Thank you to my home away from home, Kellogg College, for the delicious meals, fancy formals and for supporting me in innumerable ways.

# Abstract

In the age of instant communication, smartphones have become an integral part of our daily lives, with a significant portion of the population using them for a variety of tasks such as messaging, banking, and even recording sensitive health information. However, the increasing reliance on smartphones has also made them a prime target for cybercriminals, who can use various tactics to gain access to our sensitive data. In light of this, it is crucial that individuals and organisations prioritise the security of their smartphones to protect against the abundance of threats around us. While there are dozens of methods to verify the identity of users before granting them access to a device, many of them lack effectiveness in terms of usability and potential vulnerabilities.

In this thesis, we aim to advance the field of touchscreen biometrics which promises to alleviate some of the recurring issues. This area of research deals with the use of touch interactions, such as gestures and finger movements, as a means of identifying or authenticating individuals. First, we provide a detailed explanation of the common procedure for evaluating touch-based authentication systems and examine the potential pitfalls and concerns that can arise during this process. The impact of the pitfalls is evaluated and quantified on a newly collected large-scale dataset. We also discuss the prevalence of these issues in the related literature and provide recommendations for best practices when developing continuous touch-based authentication systems. Then we provide a comprehensive overview of the techniques that are commonly used for modelling touch-based authentication, including the various features, classifiers, and aggregation methods that are employed in this field. We compare the approaches under controlled, fair conditions in order to determine the top-performing techniques. Based on our findings, we introduce methods that outperform the current state-of-the-art.

Finally, as a conclusion to our advancements in the development of touchscreen authentication technology, we explore any negative effects our work may cause to an ordinary user of mobile websites and applications. In particular, we look into any threats that can affect the privacy of the user, such as tracking them and revealing their personal information based on their behaviour on smartphones.

# Contents

**List of Figures** **ix**

**List of Tables** **xi**

**List of Abbreviations** **xiii**

**1 Introduction** **1**
1.1 Motivation . . . 1
1.2 Ethical Considerations . . . 5
1.3 Research Contributions . . . 6
1.4 Thesis Outline . . . 8

**2 Background** **11**
2.1 Authentication . . . 12
2.1.1 Mobile Authentication . . . 15
2.1.2 Continuous Touch-Based Authentication . . . 19
2.2 Tracking and Personal Information Leakage . . . 23
2.2.1 Fingerprinting . . . 24
2.2.2 Personal Information Leakage . . . 28
2.3 Threat model . . . 31
2.3.1 Authentication Threat Model . . . 31
2.3.2 Privacy Threat Model . . . 33

**3 Fair Evaluation of Touch-Based Authentication Systems** **35**
3.1 Introduction . . . 36
3.2 Common Evaluation Pitfalls . . . 38
3.3 Prevalence of Evaluation Pitfalls . . . 42
3.4 Large-Scale Data Collection of Touch Interactions . . . 45
3.4.1 Remote collection . . . 47
3.4.2 Lab collection . . . 54
3.4.3 Dataset comparison . . . 58
3.5 Continuous Touch-Based Authentication Modelling Pipeline . . . 60
3.6 Analysis of Pitfalls . . . 63

3.7 Best Practices for Evaluating Touch-Based Authentication Systems 83
3.8 Conclusion . . . 87

**4 Techniques for Touch-Based Authentication Modeling 89**
4.1 Introduction . . . 90
4.2 Techniques for Continuous Touch-Based Authentication . . . 92
4.2.1 Methods . . . 92
4.2.2 Findings . . . 96
4.2.3 Datasets . . . 100
4.3 Performance Evaluation . . . 101
4.3.1 Comparison . . . 104
4.3.2 Results . . . 107
4.4 Discussion . . . 110
4.4.1 Limitations . . . 112
4.5 Conclusion . . . 113

**5 Privacy Concerns in Touch-Based Systems 115**
5.1 Introduction . . . 116
5.2 Related Work . . . 117
5.3 Dataset and Features . . . 119
5.4 Fingerprinting . . . 120
5.4.1 Evaluation Approaches . . . 121
5.4.2 Formalising our Approach . . . 122
5.4.3 Method . . . 124
5.4.4 Results . . . 126
5.5 Personal Information Leakage . . . 130
5.5.1 Method . . . 131
5.5.2 Results . . . 135
5.6 Discussion . . . 137
5.6.1 Countermeasures . . . 139
5.6.2 Limitations . . . 141
5.7 Conclusion . . . 143

**6 Conclusion 145**
6.1 Summary and Key Findings . . . 145
6.2 Directions for Future Work . . . 147
6.3 Final Remarks . . . 149

**References 151**

# List of Figures


2.1 Overview of authentication categories . . . . . 12
2.2 Stages of a biometric authentication system . . . . . 14
2.3 Overview of mobile authentication approaches. . . . . 16
2.4 List of personal information leakage attributes. . . . . 28

3.1 Evaluation pitfalls in continuous touch-based authentication systems 37
3.2 Examples of training data selection approaches . . . . . 39
3.3 Visualisation of attacker modelling approaches . . . . . 39
3.4 Participation retention in the remote data collection . . . . . 48
3.5 Remote data collection application screenshots . . . . . 52
3.6 In-person data collection application screenshots . . . . . 57
3.7 Age of participants in data collection . . . . . 59
3.8 Per-user distribution of touch-based authentication performance . . 65
3.9 ROC curves for common evaluation pitfalls . . . . . 66
3.10 ROC curve for the cumulative effect of all pitfalls . . . . . 67
3.11 Differences between extrapolated and empirical EER . . . . . 67
3.12 Comparison of performance between sessions collected at an early or late stages of the data collection experiment . . . . . 67
3.13 Comparison of performance at a varying number of sessions . . . . 67
3.14 Relationship between per-user EER and number of swipes available 67
3.15 ROC curves of individual phone models compared to mixing them . 71
3.16 Confusion matrix of phone model prediction for iOS devices . . . . 73
3.17 Performance difference between IncludeAtk and ExcludeAtk attacker modelling approaches . . . . . 75
3.18 Absolute difference between IncludeAtk and ExcludeAtk attacker modelling approaches . . . . . 75
3.19 ROC curves for including and excluding attacker data into training 77
3.20 Performance of aggregation model at different window sizes . . . . . 79
3.21 FAR of an aggregation model when including attacker swipes . . . . 79

4.1 Prevalence of classifiers, aggregation methods, and performance metrics in continuous touch-based authentication studies . . . . . 98

4.2 Performance of different feature sets, classifiers and aggregation methods on three datasets . . . . . 109

5.1 Performance of fingeprinting simulation when varying session size . 128
5.2 Performance of discrimination fingerprinting simulation when varying the new user session decision threshold . . . . . 129
5.3 Performance of combined fingerprinting simulation when varying the new user session decision threshold . . . . . 129
5.4 Personal information leakage multi-stroke model performance at varying number of strokes . . . . . 139

# List of Tables


3.1 Data collection and analysis choices in touch-based studies . . . . . 43
3.2 Data format format for storing touch interactions. . . . . 47
3.3 Specification sheet details for phone models used in our experiments 49
3.4 Summary and comparison of the two datasets collected in our study 58
3.5 Model performance for each stroke direction . . . . . 65
3.6 Performance difference between using individual phone models or mixing multiple models . . . . . 72
3.7 Performance for common training data selection approaches . . . . 76
3.8 Impact of common evaluation pitfalls on different classifiers . . . . . 81

4.1 Techniques in continuous touch-based authentication studies . . . . 95
4.2 Publicly available touch-based datasets . . . . . 102
4.3 Reproducible feature sets used in performance comparison . . . . . 104
4.4 List of stroke-based features found in related work . . . . . 106
4.5 Performance of classifiers applied to different feature sets . . . . . 107
4.6 Performance of aggregation methods . . . . . 108
4.7 Performance difference between our techniques and the next best and median methods in related work . . . . . 112

5.1 Performance of touch-based user fingerprinting . . . . . 127
5.2 Personal information attributes in our dataset . . . . . 131
5.3 Personal information leakage prediction performance . . . . . 135
5.4 Performance difference between realistic and unrealistic personal information leakage prediction models . . . . . 138

# List of Abbreviations

**2FA** . . . . . Two-Factor Authentication

**AB** . . . . . Ada Boost

**ACC** . . . . . Accuracy

**ANGA** . . . . Average Number of Genuine Actions

**ANIA** . . . . . Average Number of Impostor Actions

**ANOVA** . . . . Analysis of Variance

**API** . . . . . Application Programming Interface

**AR** . . . . . Augmented Reality

**AUC** . . . . . Area Under Curve

**BN** . . . . . Bayesian Network

**CDF** . . . . . Cumulative Distribution Function

**CPANN** . . . . Counter Propagation Artificial Neural Network

**CUREC** . . . . Central University Research Ethics Committee

**DT** . . . . . Decision Tree

**EE** . . . . . Elliptic Envelop

**EEG** . . . . . Electroencephalogram

**ENS** . . . . . Ensemble

**FAR** . . . . . False Acceptance Rate

**FPR** . . . . . False Positive Rate

**FRR** . . . . . False Rejection Rate

**GB** . . . . . Gradient Boosting

**GPS** . . . . . Global Positioning System

**GUI** . . . . . Graphical User Interface

**HIT** . . . . . Human Intelligence Tasks

**HMM** . . . . . Hidden Markov Models

**HTER** . . . . . Half Total Error Rate

**I/O** . . . . . Input / Output

**IB** . . . . . Image-Based

**IF** . . . . . Isolation Forest

**KDTGR** . . . Kernel Dictionary-based Touch Gesture Recognition

**kNN** . . . . . k Nearest Neighbors

**KSRC** . . . . . Kernel Sparse Representationbased Classification

**LOF** . . . . . Local Outlier Factor

**LR** . . . . . Logistic Regression

**MTurk** . . . . Amazon Mechanical Turk

**NB** . . . . . Naive Bayes

**NN** . . . . . Neural Network

**OC-SVM** . . . One-Class Support Vector Machine

**OS** . . . . . Operating System

**PIN** . . . . . Personal Identification Number

**PPG** . . . . . Photoplethysmogram

**PSORBFN** . . Particle Swarm Optimisation Radial Basis Function Network

**RBF** . . . . . Radial Basis Function

**RC** . . . . . Random Committee

**RF** . . . . . Random Forest

**ROC** . . . . . Receiver Operating Characteristic

**SM** . . . . . Scaled Manhattan

**SSB** . . . . . Session-Based

**STB** . . . . . Stroke-Based

**StrOUD** . . . . Strangeness based OUtlier Detection

**SVM** . . . . . Support Vector Machine

**TPR** . . . . . True Positive Rate

**VR** . . . . . Virtual Reality

*The journey is part of the experience — an expression of the seriousness of one's intent.*

— Anthony Bourdain

# 1 Introduction

## Contents


**1.1 Motivation** . . . . . 1
**1.2 Ethical Considerations** . . . . . 5
**1.3 Research Contributions** . . . . . 6
**1.4 Thesis Outline** . . . . . 8


In this chapter, we give the motivation for this work, including a brief introduction to the subject of this thesis and how the ideas surrounding it developed. Following that, we present the research questions examined in the thesis and the related ethical considerations emerging from answering them. Then we present the research publications resulting from the findings in this thesis. Finally, we give an outline and structure of the remainder of this work.

## 1.1 Motivation

Smartphones have become an integral part of our daily lives. Mobile devices have grown to deliver services far beyond that which can be considered trivial and mundane and now are used for activities that involve the processing and storage of sensitive and private information. Both mobile websites and dedicated applications are used for a variety of tasks, including entertainment, banking,

and shopping. The prevailing method for interacting with smartphones is by using their touchscreen displays. We refer to the human behaviour of interaction with touch-capacitive displays as *touchscreen biometrics*. They are also referred to as touch-based biometrics and touch dynamics. Touch interactions, such as swiping and scrolling on mobile devices, are necessary to navigate websites and applications, and as such, data related to our touch behaviour can be collected trivially in both settings. Touch-based biometrics have been actively studied for their potential to enhance the security of users. One such application of the technology is continuous authentication [1, 2].

Continuous touch-based mobile authentication systems passively ensure user identity throughout interactive sessions with the device. This is done by comparing new patterns of interaction with the legitimate ones of the enrolled user. When a significant mismatch is detected, the system can block the malicious user and notify the owner of the device. It is worth noting that the system runs in the background, and its operation is invisible to the user. Hence an ideal application of the technology does not impair the usability of the device while aiding with its security. It offers significant advantages over traditional methods (passwords, pins, fingerprints, and face recognition) by relying on dynamic and inherent properties of users rather than static attributes and memorised sequences.

Continuous touch-based authentication can be useful when a PIN has been stolen, or an unauthorised user gets access to an unlocked phone. Then, when the illegitimate user starts using the phone, they will be stopped by the continuous authentication system as the pattern of usage deviates from the owner of the smartphone. Another use of this technology is to detect suspicious behaviour and request additional checks when operating sensitive applications such as the ones dealing with finance and banking.

The features of touchscreen biometric systems are derived from the coordinates and pressure points at contact while interacting with the screen. Some continuous touch-based authentication systems augment these strokes with additional data,

such as sensor information from the accelerometer and gyroscope. However, the focus of this thesis is solely on stroke-based systems.

Despite all the benefits, this frictionless and highly secure system is not widely deployed in our devices, even after almost a decade of research into this field and positive sentiment from the community for such technologies [3].

The aim of this thesis is to advance the field of touch-based authentication by enabling the development of next-generation research and practical applications of the technology. To this end, the thesis focuses on three main areas. Firstly, establishing best practices for data collection and system modelling to ensure the robust evaluation and comparability of touch-based authentication systems. Secondly, consolidating and benchmarking prevalent touch-based authentication techniques while introducing novel methods that outperform the state-of-the-art. The goal is to aid researchers and practitioners in understanding and developing high-performing touch-based authentication systems. Lastly, addressing any privacy concerns that may arise from the development and integration of the technology.

Related work has shown that evaluation in biometric systems is sometimes done incorrectly, leading to overestimation of performance. As a starting point of our investigation, we aimed to explore the evaluation practices in the touch-based authentication research area and whether there are inconsistencies with the results reported in the field. When examining the touch-based authentication field, it becomes evident that the research is fragmented and difficult to navigate, especially for people unfamiliar with the topic. Our aim is to consolidate the field and make it easier for researchers and practitioners to easily understand and build upon what has been developed. Touchscreen biometrics achieve great authentication performance, however, they have hardly been studied in other contexts. This prompted us to investigate the potential for malicious applications of the technology which can infringe on user privacy.

In the rest of this section, we provide an overview of the research goals and questions pursued in this thesis. The first research questions we investigated aim to understand how the evaluation of touch-based authentication systems might play a

role in the low adoption rate due to overestimation of performance. The goal is to identify common issues, establish a realistic way for evaluating touch-based systems and recommend best practices for data collection and evaluation to the community.

- RQ1.1: *How do we collect touch-based interaction data at scale, and what are the potential differences with traditional lab methods?*
- RQ1.2: *What are the common evaluation pitfalls made in touch-based authentication models, and how do they affect the performance of the system?*
- RQ1.3: *What are the best practices for evaluating the performance of touch-based authentication systems?*

The second set of research questions aims to understand how to develop state-of-the-art continuous touch-based authentication models and accelerate the progress of the field. This is done by conducting a large-scale literature review of the prevalence of methods used for feature extraction, classification, and aggregation. Then we perform extensive experimentation to compare the performance of various techniques across multiple datasets. Building on what is learned from the comparison, we introduce novel approaches which outperform the current state-of-the-art in the field.

- RQ2.1: *What are the most prevalent methods for feature extraction, classification, aggregation and measuring performance in touch-based authentication?*
- RQ2.2: *Which are the best-performing techniques for continuous touch-based authentication, and how can we fairly establish them?*
- RQ2.3: *How can we improve upon the state-of-the-art in continuous touch-based authentication?*

The third set of research questions looks into the broader impact of advancing continuous touchscreen biometric systems. There are many positives for the security and usability of users stemming from the use of the authentication system. However, the consequences of using touch-based biometric systems in other contexts have

been largely neglected. Our aim is to investigate the privacy implications of the technology, including to what extent users can be tracked across websites, and to examine whether touch interactions reveal the gender, age, and other demographic characteristics of a user.

- RQ3.1: *What privacy implication do touch-based systems possess in terms of tracking user behaviour?*
- RQ3.2: *Can services reveal the personal information of users based on touchscreen interactions?*

To summarise, the research questions of this thesis investigate the evaluation approaches in continuous touch-based authentication systems and clarify the ways to straightforwardly develop well-performing systems while considering the privacy threats resulting from their use.

## 1.2 Ethical Considerations

While we believe the ethical impact of our work is limited, we address the potential issues and considerations in this section.

We collected a large dataset as part of the fair evaluation work in Chapter 3. This data was also used in a variety of contexts throughout the rest of the thesis. The data collection experiments involved human participation, which can result in a multitude of ethical considerations. Biometric data is particularly sensitive as it has the potential to be used to impersonate users in systems that use this modality. Furthermore, the data might be used to reveal the identity of the subjects in our experiments. Participants also revealed personal information, such as their age and gender, which was used in our privacy investigation as part of Chapter 5. We addressed these issues by anonymising the data and storing it securely such that we minimise the identifying information of the participants. These experiments were approved by Oxford's Central University Research Ethics Committee (CUREC) process with code SSD/CUREC1A CSC 1A 19 013.

We do not foresee any direct ethical issues arising from the advancements of the continuous touch-based authentication technology as part of this thesis. However, we acknowledge that we have not seen a large-scale deployment of the system, hence some unexpected complications may arise. Furthermore, with the advancement of the technology, we can see the formation of privacy concerns which are explored in Chapter 5. We believe that not revealing these threats publicly is an inadequate idea because it is not a real solution for enhancing the security of users. It would only result in worse security and privacy for everyone, as malicious actors could develop and exploit the technology regardless of our publication. However, disclosing these threats could help inform both the general public and specialists, such that concrete steps are taken to prevent their misuse.

## 1.3 Research Contributions

The following section provides a description of the peer-reviewed publications that form the foundation of this thesis. These publications represent the results of collaborative research with other individuals, but only those studies in which the author of this thesis served as the first author and primary contributor are included. A statement of authorship is also provided in this section to clarify the contributions of each individual involved in the research.

- The basis of Chapter 3 is a paper investigating the evaluation pitfalls in touch-based authentication systems. The study was published and presented at *The 17th ACM ASIA Conference on Computer and Communications Security (ACM ASIACCS 2022)* [4]. An extension of this work, including a new, in-person dataset and further evaluation experiments, is available as a pre-print on arXiv [5].

- A study exploring and improving upon the techniques used in touch-based authentication systems underpins Chapter 4. It was published and presented at *The 17th International Conference on Information Security Practice and*

*Experience (ISPEC 2022)* [6]. The study received the best paper award at the conference.

- Chapter 5 investigates the privacy concerns related to the use of touch-based biometric technology. The chapter is based on a paper that was published and presented at *The 21st ACM Workshop on Privacy in the Electronic Society (WPES 2022)* collocated with *The Conference on Computer and Communications Security (CCS 2022)* [7].

**Authorship**

Apart from the few exceptions listed below, the research described in this thesis has been conducted solely by myself under the guidance and expertise of my supervisor Ivan Martinovic who was involved with planning each of the studies included in the final dissertation and giving direction until publication. I would like to point out the following exceptions and thank the co-authors for their helpful contributions:

- Henry Turner, Guilio Lovisotto, and Simon Eberz were all involved in a research study, which forms the basis of Chapter 3. In particular, they helped with obtaining ethics approval and developing the server and payment architecture for data collection. In addition, they contributed to the creation of some of the figures that were used to illustrate the findings of the study. They also assisted with refining the writing of the final manuscript.

- Simon Eberz has provided ongoing support and assistance for all of the research published in the thesis. His contributions have been essential in the development and refinement of the initial ideas, and he has played a key role in ensuring that the completed papers are of high quality. His expertise and guidance have been invaluable in helping to shape the research and bring it to its final form.

## 1.4 Thesis Outline

The remainder of the thesis is structured as follows:

- **Chapter 2** introduces the two concepts this thesis is built upon by providing the background needed to understand them and the related research work to date. Firstly, we explore the field of authentication on desktop and mobile devices in order to put continuous touch-based authentication in context. Then we provide further details and related work for touchscreen systems in particular. In the second part of this chapter, we focus on concepts that are useful in understanding the privacy issues related to using touch-based biometric technology. This includes fingerprinting and personal information leakage in desktop and mobile environments, as well as their implications, as portrayed in the related work.

- **Chapter 3** presents a method for large-scale data collection and evaluation of continuous touch-based authentication systems. We introduce the dataset collected using our approach. The data gathered in this chapter is utilised throughout the rest of the thesis. The common procedure for evaluating touch-based authentication systems is described in detail. We explore the common evaluation pitfalls in touch dynamic systems and highlight their prevalence in the related literature. Other potential concerns within the evaluation of such systems are also investigated. Lastly, we give our recommendation for best practices when developing continuous touch-based authentication systems.

- **Chapter 4** describes and categorises the techniques used for modelling touch-based authentication. These include the features, classifiers and aggregation methods used in the field. We show the prevalence of these approaches in the related work. Then, the techniques are benchmarked under fair conditions, and the best-performing ones are highlighted for potential use in research and industry. We build upon the findings and introduce a novel feature set and

classifier that outperform the current state of the art. At the end, we verify the results across three datasets.

- **Chapter 5** establishes the privacy issues related to touchscreen biometrics technology. This chapter is divided into two sections, each describing a particular privacy threat. Firstly, we investigate the potential for fingerprinting users on mobile applications and websites by using the way they interact with the touchscreen of their devices. Then we look into the possibility of using the same technology to reveal personal information, such as age, gender, height, weight and country of origin. Finally, we briefly look into the potential countermeasures to these privacy threats.

- **Chapter 6** is the conclusion of this thesis. We provide a summary of previous chapters and the key conclusions of the thesis. The chapter presents recommendations for applying the technology and directions for future work in continuous touch-based authentication and the privacy implications of the technology.

*Alea iacta est.*

— Julius Caesar

# 2 Background

## Contents


**2.1 Authentication . . . . . 12**
2.1.1 Mobile Authentication . . . . . 15
2.1.2 Continuous Touch-Based Authentication . . . . . 19
**2.2 Tracking and Personal Information Leakage . . . . . 23**
2.2.1 Fingerprinting . . . . . 24
2.2.2 Personal Information Leakage . . . . . 28
**2.3 Threat model . . . . . 31**
2.3.1 Authentication Threat Model . . . . . 31
2.3.2 Privacy Threat Model . . . . . 33


In this chapter, we begin by providing a broad overview of the various authentication systems that exist and the different approaches that are typically taken into consideration when safeguarding personal information on digital devices. After this general introduction, we delve deeper into the specific topic of mobile authentication while pointing out the drawbacks of using traditional approaches. Furthermore, we examine the various benefits that can be gained by implementing continuous authentication, with a particular emphasis on systems that are based on touchscreen interactions. We provide an overview of the different data collection methods, classification approaches, and performance metrics that have been used in the design and implementation of these types of systems. Additionally, we provide

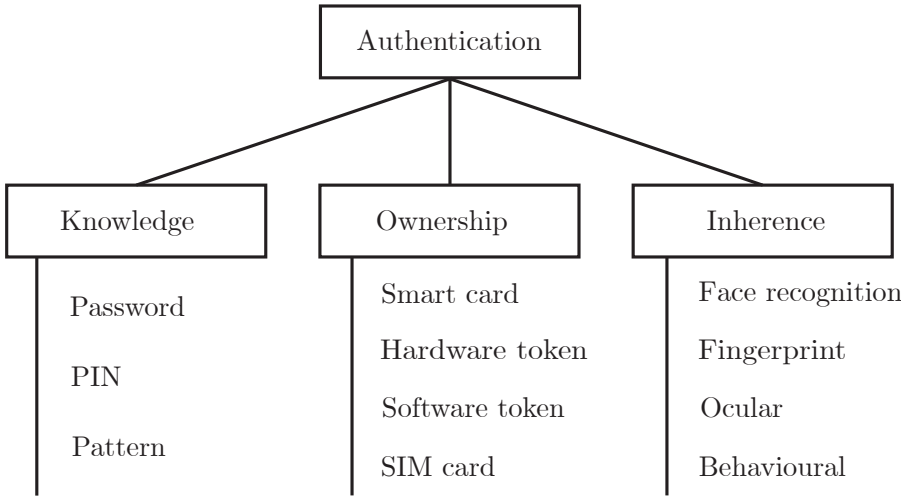



**Figure 2.1:** Overview of authentication categories with examples for each type.

an overview of the literature on different types of attacks that can be launched against continuous touch-based biometric systems, as well as the countermeasures that have been proposed to mitigate these risks. We introduce privacy concepts that are related to continuous touch-based biometric systems. In particular, we examine the literature on user tracking, both on desktop and mobile environments, and describe the methods for personal information leakage with a focus on smartphones.

## 2.1 Authentication

As technology advances and more of our personal information becomes accessible through digital devices, it has become increasingly important to have systems in place to verify the identity of users and protect against unauthorised access. There are various methods for authenticating users, which can be grouped into three main categories based on the type of information or characteristics used for verification: knowledge-based authentication, ownership-based authentication, and inherence-based authentication. Figure 2.1 provides examples of each of these approaches. These methods are used to ensure that a user is who they claim to be and has the appropriate permissions to access a particular system or device.

**Knowledge.** Knowledge-based authentication systems require users to remember

something that only they themselves know. For instance, this could be a PIN (a short number) or a password (a string of any characters). While being the most widespread method of authentication, it has a multitude of downsides. Due to the need to remember passwords, users tend to choose less secure ones and reuse them often [8]. This practice also leads to easily guessable passwords as demonstrated by Melicher et al. [9] and Dürmuth et al. [10]. Furthermore, passwords are susceptible to simple "shoulder surfing" attacks where the adversary observes or records the input interface in order to reproduce what has been written [11]. Password managers are software programs which remember all credentials of a particular user and can only be accessed by providing a single strong master password. They encourage users to generate long, random and unique passwords for each new account. Although this produces a single point of failure, as long as the master password is well-protected, security can be increased. Lysterin et al. quantified this and found that users of password managers had both stronger and more unique passwords [12].

**Ownership.** Instead of depending on remembering something, ownership-based authentication systems rely on the possession of something to prove user identity. This could be in the form of a hardware device such as a smart card or token containing a certificate which authenticates the user. Due to the ease of losing or stealing such devices, they are typically paired with a second layer of authentication, such as PIN codes. One of the main uses of ownership-based authentication systems, however, is as part of a two-factor authentication (2FA) system where users are required to know a password but also have access to a software token, hardware token [13] or even a mobile phone SIM card [14]. Systems which take advantage of three or more factors have also been proposed [15]. Although 2FA systems (and the ones using more modalities) can certainly increase user security, they are not necessary an unassailable solution. Man-in-the-middle and malware attacks, for instance, can still bypass the added security of the method [16].

**Inherence (Biometric).** Inherence (also known as biometric) based systems form a diverse and complex authentication category [17]. They rely on the "something

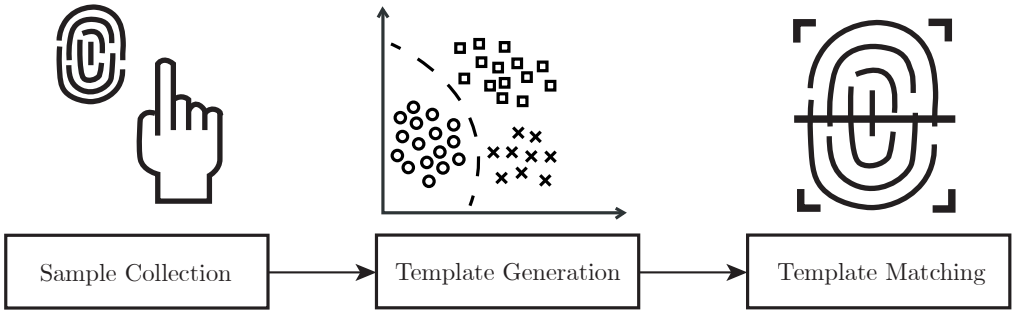



**Figure 2.2:** Stages of a biometric authentication system. The lifecycle includes methods for enrolment into the system, generation of generic user templates and techniques for matching them with high accuracy.

you are" paradigm and employ the unique physical and behavioural traits of users. These characteristics do not have to be remembered and are intrinsic to the device owner, which makes them especially user-friendly. Inherence authentication systems are useful in a variety of settings as a replacement for traditional password-based authentication. For instance, most modern smartphones have an onboard camera installed, making it possible to utilise it for face-recognition systems [18]. Furthermore, since phones are always operated by hand, fingerprint-based systems have been deployed to ease the process of authentication [19]. Voice is another biometric which is prevalent in the context of smart speakers and smartphones [20]. This allows for commands to only be activated if spoken by a legitimate user who is authenticated through their voice. Highly secure environments might require more advanced authentication approaches. Iris-based biometrics take advantage of the unique patterns of our irises. These systems require high-quality imaging, which is not suitable for most applications. However, the features extracted are unique and stable, and such methods achieve strong authentication performance [21]. More unconventional biometric systems have also been proposed. Chuang et al. [22] proposed an authentication system based on brainwaves (EEG), and Kaczmarek et al. [23] introduced a biometric system based on the way people sit on their office chairs.

A typical biometric system consists of multiple stages. First of all, users have to enrol into the system by providing a predefined amount of data. For instance,

they might be required to place their fingerprint on a sensor multiple times. Then the system uses this data to create a template for this user, which is different from other people. This can be done by extracting features from the data and training a machine-learning model. Finally, on authentication, a newly provided sample is compared to the template, and the system decides to accept or reject the user. The process is illustrated in Figure 2.2.

Biometric systems have a few drawbacks. The system can be prone to making mistakes, unlike other methods of authentication. Comparing passwords requires a perfect match but comparing fingerprint templates is not perfect and can lead to blocking the access of a legitimate user or letting a malicious one get through. Moreover, the template created cannot be easily revoked, and a malicious actor with a copy of the template could reveal the identity of the original user. Cancellable biometrics are methods proposed to combat this issue [24, 25]. The general idea is to introduce fixed, per-user perturbations when creating a template in order to avoid invertibility and linkability and ensure access can be revoked.

Some of the biometric methods provide identification based on the way humans perform specific actions. These are known as behavioural biometrics. Features can be extracted from the way users walk or run (gait) [26], the tone and intonation of their voice [27], their behaviour on web browsers [28] and how they interact with input methods such as keyboards [29], mice [30] and touchscreens [1]. More recently, Pfeuffer et al. have demonstrated the potential of behavioural biometrics in Virtual Reality (VR) settings by proposing a system that authenticates users based on the patterns in their movements [31].

### 2.1.1 Mobile Authentication

The adoption and use of mobile devices have increased dramatically over the past few years, leading to vast amounts of sensitive data being easily accessible through a single endpoint. Furthermore, many smartphones are used to access and control smart home features, financial transactions and even medical data through the use of other connected devices. Protecting these smart machines

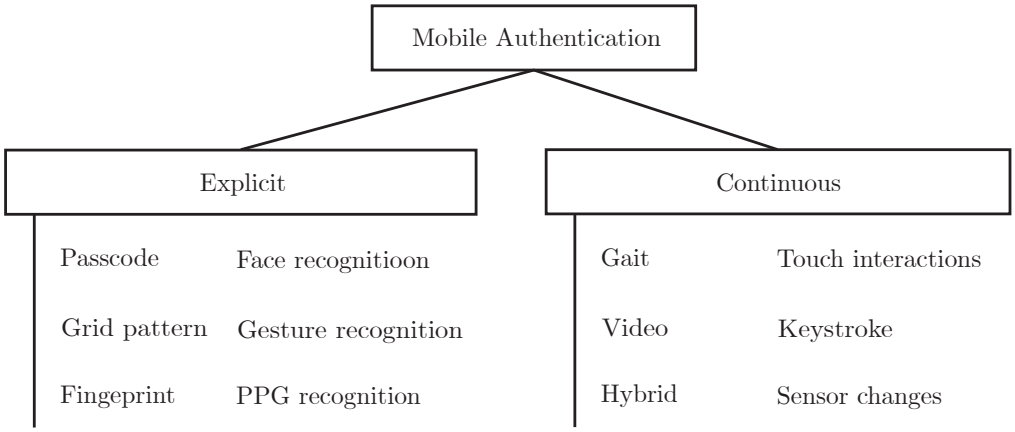



**Figure 2.3:** Overview of mobile authentication approaches with examples for each type.

requires an entirely different approach from desktop computers. Due to their small form factor, input methods and the frequency of locking and unlocking, well-known authentication methods such as passwords are rarely used as the default mechanism for mobile phones. Figure 2.3 presents the methods used in mobile authentication. In this section we explore both explicit and implicit (continuous) mobile authentication methods.

### Explicit Authentication

Due to these limitations, the security of smartphones has traditionally relied on explicit (login-based) authentication methods such as short PINs. Grid patterns, where users remember how to connect points sequentially on a square grid, have also been used with the introduction of modern touchscreen devices. The ineffectiveness of these approaches has been demonstrated in numerous studies. They are tedious and inconvenient, leading many users to avoid using them completely [32, 33]. Furthermore, they are prone to shoulder surfing attacks [11] where an adversary simply observes and remembers the short input interaction. Smudge attacks, where the entry method is deduced from residue on the touchscreen, are also easily deployed [34, 35]. Market et al. [36] even suggest that PIN guessing, without other revealing information, can be surprisingly successful in gaining unauthorised access to mobile devices. Other methods for explicit authentication have also been proposed. Cheon et al. [37] demonstrate a free-form gesture authentication

system which requires users to create a unique drawing pattern and recall it when authenticating. We know that we can use photoplethysmogram (PPG) features to authenticate users by examining the blood volume changes in the microvascular bed of their tissues [38]. Lovisotto et al. [39] demonstrate that it is possible to utilise this method by using the camera of modern smartphones. In general, PPG and gesture systems for explicit authentication have some advantages over other methods as they typically possess a larger search space, require decreased visual attention and are less prone to shoulder surfing attacks. Such methods of authentication, however, have not been deployed in large commercial systems. Despite their complexity, they still suffer from some of the issues affecting PINs and patterns.

Each unlocking interaction in the methods described above does not take a particularly long time to complete. However, the total adds up to a significant amount considering the need for performing the action frequently on mobile devices. The convenience and speed provided by the following biometric systems are the reason for their mass adoption in modern devices [40]. The most widespread methods are based on fingerprint [19] and facial [18] recognition. These methods, however, come with the possibility of spoofing attacks. For instance, fingerprints can be collected from everyday objects or the phone itself. A fake fingerprint can then be created out of malleable materials in order to fool the system. Galbally et al. [41] demonstrate the feasibility of this attack in practice. Facial recognition systems can also be spoofed by hardware and software-based attacks [42]. The mandatory use of gloves and masks during a health crisis also greatly impairs the usability of these two biometrics.

#### Continuous authentication

In general, login-based approaches like the ones described above have a few noticeable flaws. Some are time-consuming, others easy to forget, and all of them seem prone to security attacks. Moreover, once the device is authenticated, it has to be locked again, or it might be compromised by an adversary. Continuous (sometimes referred to as implicit, zero-effort or transparent) authentication provides a solution to these

problems. It is difficult to impersonate or mimic continuous actions. Furthermore, it improves upon the user experience of a system as it does not require time-consuming actions or a need to remember unique information. There has been a lot of research on the topic, and while there has been progress in the area, there are many challenges remaining [43].

Some of the early research focused on maintaining a trust score on the mobile device based on behaviour profiling [44–46]. These systems take advantage of behavioural actions which are already commonly performed by the user, such as calls, messages, app activity, screen status and I/O (input/output) interactions. Other research focuses on gait-based continuous authentication [26, 47, 48]. In the case of mobile authentication, this is done by using the built-in accelerometer and gyroscope sensors. However, this technique for continuous authentication is limited to users being in motion, which reduces its usefulness. Video-based authentication, where the phone is continuously recording and authenticating the user, has also been proposed [49, 50]. This differs from face-recognition authentication, where the action is performed only once. While this approach can be very successful, it comes at a cost to the user experience. The camera is required to point at the face at all times, and its continuous use might bring many privacy concerns for the owner and their surroundings. Spoofing attacks, similar to the ones in facial recognition systems, are also possible. Hybrid (fusion) systems have also been proposed [51, 52]. They take into account multiple modalities and information, such as video, sound, browsing history and sensors, in order to decide when login-based authentication is needed.

Continuous authentication on desktop computers using keystrokes is a heavily studied area of research [29, 53]. This technique uses the latency between typing characters, digraphs and n-graphs and sometimes the key hold and pressure data to create a unique template for each user. Approaches in this area mostly vary in their feature extraction and machine learning methodology. This idea easily translates to the mobile domain. It has been studied in the context of mobile authentication with keystrokes on virtual keyboards [54, 55]. Smartphone devices allow for the use of extra features such as directional information, touch pressure and touch

size. Furthermore, some of the work uses keystroke dynamics in combination with mobile sensor data from the accelerometer and gyroscope [56, 57]. An issue with this authentication approach is that attackers can still perform malicious actions without the need to use the virtual keyboard.

### 2.1.2 Continuous Touch-Based Authentication

One well-researched area in continuous mobile authentication focuses on the uniqueness of touchscreen interactions performed by users. This approach collects information on how users navigate the interface of a phone by recording all swipes, scrolls and gestures together with their pressure, size and duration. The method is comparable to mouse-movements-based authentication on desktop computers where motion and clicks are recorded [30]. The difference between the two methods is that with smartphones, users can lift and place their finger on the screen, which creates separate actions, unlike mouse movements. Moreover, tapping [58] and multi-touch gestures [59] such as pinch-to-zoom are also possible. The focus of this thesis is on continuous touch-based biometric systems and, more specifically, on the use of horizontal and vertical displacements on touch-capacitive displays done using a single finger which are broadly called strokes.

The lifecycle of a continuous touch-based authentication system consists of a few steps similar to the ones used for generic biometric approaches. The data collection step could be the experimental setup for a study, or in the case of a deployed system, it could be the enrolment phase where individual templates of behaviour are created. The feature extraction step in continuous touch-based authentication aims at obtaining unique information from touchscreen interactive sessions with the smartphone, which can be used to differentiate between users of the system. The classifier step relies on models to make a decision about the legitimacy of a particular stroke based on enrolment patterns. These are typically machine learning algorithms that are trained on the features extracted in the previous steps. Furthermore, a single stroke may not provide enough distinguishing information for an acceptable authentication performance. For this purpose, some systems perform aggregation

of successive strokes to improve system performance. In the final step, a variety of metrics could be used to capture and report the success of the biometric system.

### Origin of Touch-Based Authentication

Continuous touch-based authentication systems were first proposed in the early 2010s by applying well-established behavioural biometric ideas to the strongly emerging smartphone ecosystem. Feng et al. developed one of the earliest systems in touch-based authentication on smartphones [60]. They successfully utilised an external glove for precise data collection, but the approach is unrealistic for practical use. Soon after, systems solely based on the data provided by the phone were developed [1, 2, 61]. Zheng et al. have taken a slightly different approach to touch dynamic authentication by focusing on tapping behaviours during PIN entry [58]. Some studies also take into account multi-touch gestures such as pinching [54, 62]. Similarly to other continuous authentication methods, many hybrid approaches have also been proposed. Some recent research includes sensor data coming from the accelerometer and gyroscope [63, 64]. Deb et al. include 30 different modalities, including GPS and magnetometer [65]. Rahul et al. have even taken into account the power usage of the device [66].

### Data collection modalities

There are varying approaches for data collection in touch-based authentication. Frank et al. use text-reading to collect vertical scrolls, and a "spot the difference" game to gather horizontal swipes [1]. Similarly, Antal et al. use text reading and image gallery tasks [67]. Others include social media interactions [66], zooming on pictures [68] and questionnaires [69]. Buschek et al. evaluate the influence of GUI elements and hand postures on the performance of touch-dynamic systems [70]. In order to analyse the time stability of the biometric, some recent studies collect data over multiple sessions or days. Watanabe et al. specifically look into the long-term performance of touch-based authentication systems by collecting user data for over six months [71]. They demonstrate promising results for the time-stability of the biometric. While the data from experiments is typically collected in a restricted

environment during lab sessions, Feng et al. [68] recruited 100 users to use their data collection application over the course of 3 weeks to provide a more realistic environment when performing everyday tasks.

### Feature extraction and classification modalities

Most feature extraction methods in touch authentication systems focus on describing the geometrical attributes of swipes such as coordinates, duration, acceleration, deviation, and direction [1, 2]. One alternative is to use computer vision approaches to describe and differentiate the strokes. Zhao et al., for instance, use a method to convert the stroke information into an image that can be used for statistical feature model extraction [62].

On the other hand, there is a vast variability in the classification approaches in touch-based authentication. Some studies have focused on systematising and comparing knowledge within the field. Fierrez et al. [72] analyse and compare recent efforts in the field in terms of datasets, classifiers, and performance. Serwadda et al. compare the most common machine learning algorithms in the context of touch-based authentication [69]. These include Logistic Regression, Support Vector Machine (SVM), Random Forest, Naive Bayes, Neural Network, k-Nearest Neighbours (kNN), and others. The studies suggest that the Support Vector Machine and Random Forest perform the best for touch-based tasks. Fierrez et al. provide insights into model and design choice performance by benchmarking open-access datasets [73]. They find that landscape phone orientation and horizontal gestures prove to be more stable and discriminative.

### Performance and metrics

There are many different metrics that can be used to evaluate touch-based authentication systems. The metrics used can depend on the specific goals and objectives of the evaluation, as well as the characteristics and limitations of the system being evaluated. We provide a definition of the most commonly used metrics for touch-based authentication:

- False Acceptance Rate (FAR) - the rate at which unauthorised strokes are accepted as benign.
- False Rejection Rate (FRR) - the rate at which strokes from an authorised user are rejected as malicious.
- Receiver Operating Characteristics (ROC) curve - illustrates the performance of a model in terms of FAR and FRR when the threshold of acceptance (which is typically set at 0.5) is varied from 0 to 1.
- Equal Error Rate (EER) - the point at which the FAR and FRR are equal on the ROC curve.
- Accuracy - total accurate predictions divided by the total number of predictions.

While it has been argued that EER does not adequately describe systematic errors [74], it is generally accepted as a good measure of average system performance. Furthermore, [75] argues the importance of considering the ROC curve for performance as the EER metric could be misleading depending on TPR (True Positive Rate) and FPR (False Positive Rate) system requirements. In this thesis, we abstract away from the variety of experimental choices outlined in this section and investigate the fundamental effects of evaluation pitfalls on the EER and the ROC curve. Some studies may focus on the usability of the touch-based authentication system, which can result in other metrics, such as the time required to authenticate a user. While we report on such metrics throughout this thesis, usability is not the primary focus of our investigation.

The way in which data is collected and analysed can significantly impact the results that are reported in the field. Different authors may use different techniques for data collection and classification, which can result in a range of outcomes. Specifically, some studies have reported Equal Error Rates as low as 0% [1, 76], while others have reported EER as high as 22.1% [64]. It is important to consider the methods used when evaluating the accuracy of a biometric system, as the results

can vary significantly depending on these factors. This is one of the motivating factors for our fair comparison between models in Chapter 4.

### Attacks on touch-based systems

There is a limited amount of information available in the literature regarding attacks on touch dynamic systems. One potential reason for the lack of research on this topic is the inherent difficulty in successfully attacking this type of system. Furthermore, there is an overwhelming diversity of methods proposed in the relevant research papers, making it difficult to focus on a particular model as a target of attack without drawing criticism. However, it is also possible that there is no interest from the research community because of the limited practical implementations of touch-based authentication systems altogether.

To our knowledge, Serwadda et al. demonstrate the only successful attack on a touch-based system [77]. They create a standard swiping template from a group of users and use a LEGO robot in order to simulate desired actions. This results in decreased performance of the classifier EER by between 339% and 1004%. Gong et al. developed a system which is resistant to these types of attacks by applying a random secret on user interactions [78]. Eberz et al. show that many behavioural biometric studies do not evaluate their systems correctly [74]. We show that some of these effects are strongly applicable to continuous touch-based authentication and further explore evaluation issues in Chapter 3.

## 2.2 Tracking and Personal Information Leakage

The increasing speed at which technology is being integrated into every aspect of our daily lives has resulted in a significant threat to our personal privacy. It has become easier for individuals and organisations to collect, store, and share our personal information. This includes data about our online activities, location, and even our physical health and fitness. The tracking of user behaviour has become a common practice in today's digital landscape. The information collected can be used for a wide range of purposes, some of which are relatively harmless, while others can

be much more nefarious. For example, tracking user behaviour can be used to personalise advertising, improve the user experience, or detect and prevent fraud. On the other hand, it can also be used for more malicious purposes, such as surveillance, identity theft, or even blackmail. Regardless of the intent behind the tracking of user behaviour, it is important for individuals to be aware of the potential dangers and be given the opportunity to make informed decisions about their privacy.

Privacy threats and defences constitute an enormous research area which is studied systematically and holistically across a large spectrum of topics. In this section, we focus on the concepts of tracking users and revealing their personal information to provide the foundations for the work we examine in Chapter 5.

### 2.2.1 Fingerprinting

Fingerprinting is a type of user tracking that can be used to identify and trace people based on their unique configurations. Unlike tracking with cookies and sessions where users explicitly provide details to prove their identity, fingerprinting takes advantage of the hardware, software and other preferences to recognise the user. Examples of such unique identifiers include the operating system, user agent, timezone, display size and installed fonts. This makes it difficult for users to avoid tracking as the fingerprint persists even when we delete cookies or use private browsing mode. Furthermore, this makes it especially easy to share tracking information between services.

If a given fingerprinting algorithm is able to distinguish a significant number of users based on their device and browser characteristics, that fingerprint may effectively serve as a global identifier for those users. This unique data would be similar to a cookie that cannot be easily deleted. Even if a particular fingerprint does not serve as a unique identifier for all users, it may still be used in combination with other data to track users in certain contexts.

Fingerprinting poses a significant threat to the privacy of all users on the web. Understandably, the initial scientific work on the topic has been done in the context of desktop web interfaces. The field of desktop fingerprinting is mature and difficult

to summarise as it has drastically evolved throughout the years. We describe a few of the detailed surveys which have explored the development of the research area.

Lerner et al. provide a comprehensive analysis of the history and evolution of web-tracking techniques from 1996 to 2016 [79]. The paper discusses the various types of technologies that have been used over the years and how they have increased in sophistication and complexity. The authors also examine the privacy implications of these technologies and the ways in which they have been used to track and target users online. Similarly, Bujlow et al. categorise the types of tracking into five classes and provide a comprehensive overview of the various techniques used for tracking users on the web [80]. The authors also examine the privacy implications of web tracking and the ways in which fingerprinting can be used to target users with personalised online advertisements [81] and search results [82], but also with much more serious forms of content. They also suggest that there are other consequences, such as price discrimination [83, 84], government surveillance and identity theft which can be significantly more harmful. The paper also discusses various defences against tracking, including browser privacy settings, tracking blockers, and privacy-enhancing technologies. Finally, the study provides an outlook for the future of tracking and its implications for the privacy of users.

Laperdix et al. provide a detailed and in-depth look at the development of browser fingerprinting methods specifically [85]. The paper discusses the various techniques used for browser fingerprinting, including the use of browser and device features, plug-ins and extensions, and other, more advanced types of data. The study also investigates various defences against browser fingerprinting, including the use of privacy-enhancing technologies and the implementation of stricter privacy regulations. Other papers have also specifically looked into browser fingerprinting and the available defences [86, 87].

Acar et al. introduce methods for persistent tracking mechanisms on the web, including canvas fingerprinting, evercookies and the use of "cookie syncing" [88]. They focus on measuring the use of the techniques in the wild and show that these mechanisms are widely used, with the majority of websites employing at

least one form of tracking. Furthermore, they describe how rapidly new threats become integrated into websites and are used in the wild. Similarly, Englehardt et al. analysed the prevalence and behaviour of online tracking on a large sample of websites (1 million) [89]. The study found that a significant percentage of websites engage in some form of tracking, with third-party being the most common type. The authors also found that a small number of companies were responsible for the majority of the tracking that occurred and that, in most cases, it was done using a combination of different technologies.

The field is rapidly evolving, with old threats being mitigated and new ones continuously emerging. For instance, it is possible to effectively attack even the most sophisticated privacy-enhancing tools such as the Tor network [90, 91].

### Mobile Fingerprinting

More recently, with the rapid adoption of smartphones, new methods for fingerprinting have been proposed that focus entirely on mobile devices. While many of the existing methods for fingerprinting are available on mobile platforms, there are other avenues for privacy threats as well.

Hupperich et al. [92] conducted a comprehensive and large-scale study of 900 participants on fingerprinting of mobile devices using 45 features such as user-agent, operating system, screen height, and width. The authors show that they can detect new sessions or match them to their original users with high degrees of accuracy. The paper also introduces ways to evade detection with changing the browser and its settings yielding the best results. Similarly, Kurtz et al. [93] used personalised configurations composed of 29 features (e.g. device language, installed apps) as a way to identify and track mobile devices. The authors demonstrate the effectiveness of their approach through experiments on a dataset of 8,000 real-world devices on the iOS ecosystem. The system achieves a total accuracy of over 97% for matching fingerprints to devices. Fingerprinting using a combination of application and web data has also been proposed. Khanna et al. propose various techniques used at different layers of the networking protocol stack to fingerprint mobile users [94].

The built-in sensors of smartphones have also been proposed as another method to fingerprint users and their devices. One of the earliest works by Bojinov et al. [95] recorded speaker-microphone and accelerometer sensor data from 10,000 mobile devices. They showed that it is possible to uniquely identify a device among thousands with a low chance of collision. Several other studies have also used motion sensors such as accelerometers and gyroscopes for mobile fingerprinting in various conditions. Zhang et al. create a system that infers the factory calibration data from a device by careful analysis of the sensor output alone and use it as a unique fingerprint [96]. They achieve a high amount of entropy by using 100 samples of sensor data which takes less than a second to collect and process on both iOS and Android devices. Yang et al. use the motion sensors' movement when users type on the virtual keyboard to create a unique fingerprint [97]. They achieve an accuracy of more than 85% by using only ten letter keystrokes.

Das et al. also show that these types of attacks are possible [98]. However, they propose obfuscation techniques which strongly mitigate the threat. In another study, Das et al. consider the real-world application of the technology based on mobile websites and, in particular, the HTML5 DeviceMotion interface [99]. The authors find that fingerprinting works well in real-world settings and explore the countermeasures with a study showing that usability is not strongly impacted. Amini et al. also address some of the difficulties in using this technology in practice by devising new strategies based on deep neural networks [100].

Sensor and hardware-based fingerprinting can also be used to uniquely identify devices which can helpful in forensic investigations and detection of counterfeits. Baldini et al. [101] provide an overview of the methods used in this area of research. The authors show that it is possible to use hardware such as clock differences, radio frequencies and sensors for a variety of useful fingerprinting settings without the need for traditional cryptographic means. Other, more unconventional sensors, such as magnetometers, have also been shown to work for device fingerprinting mobile devices. Baldini et al. present a fingerprinting method which can differentiate mobile devices with up to 98% accuracy [102]. Matyunin et al. show that the

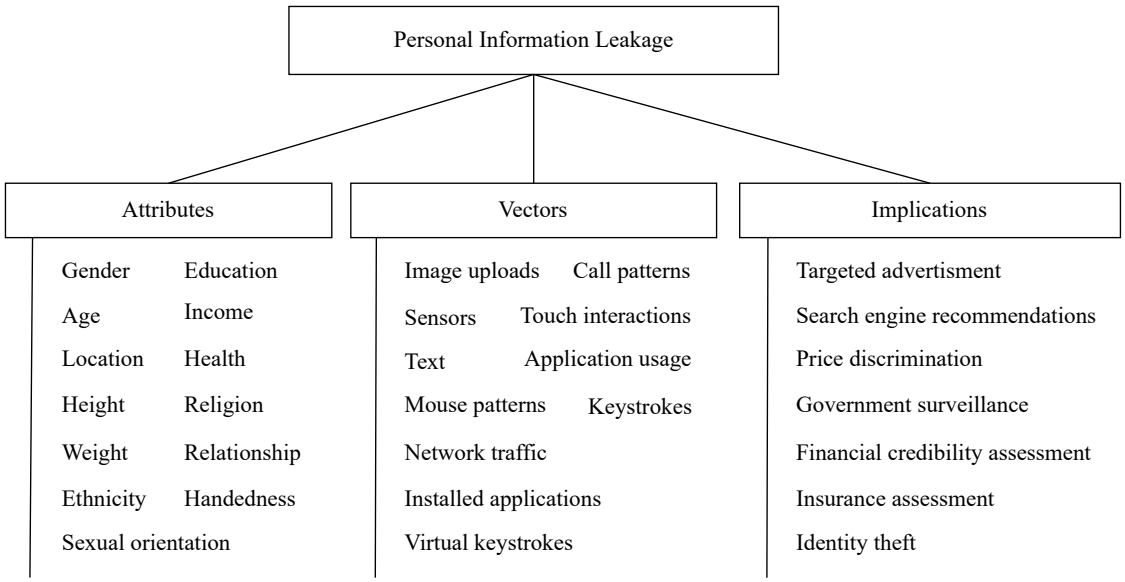



**Figure 2.4:** List of personal information leakage attributes, vectors of attack and privacy implications on mobile and desktop devices.

magnetometer can also fingerprint applications and websites running on a device, hence revealing the behaviour of a user on the phone [103].

### 2.2.2 Personal Information Leakage

Personally identifiable information such as age, gender and income of mobile users can be revealed by a variety of side-channel methods. Sometimes these attributes are referred to as "soft biometrics". Similarly to fingerprinting, the personal information of users can be used for commercial purposes such as target advertisement but also has the potential to be applied with much more harmful intentions. In this section, we will focus on how personal data can be revealed by the behaviour of users on mobile and desktop websites and applications. However, a more general and comprehensive survey of the types of personal information inferences from physical biometrics such as iris, hand, voice and gait has been produced by Dantcheva et al. [104].

Some soft biometrics can be predicted by using publicly available data generated on websites and applications. The techniques used include image processing for profile pictures and natural language processing for user-generated text. Cheng

et al. illustrate the potential for predicting gender based on the text used for email interactions [105]. They achieve an impressive accuracy of 82.2% using 545 features and an SVM classifier. Similar approaches have also proven successful with generic texts [106, 107]. Burger et al. show that it is possible to discriminate between genders by the short text patterns of users on Twitter [108]. Marquardt et al. demonstrate that predicting age and gender is also possible on blogs, hotel reviews and generic social media websites [109].

Eidinger et al. show that age and gender can be predicted from images [110]. They achieve up to 95% accuracy for age prediction (out of 8 age groups) based on facial images. Using the same pictures, the authors achieve up to 88% for the gender prediction (with a baseline of 50%). The same type of data can also be used to predict other attributes. Dantcheva et al. suggest that facial images contain information about height, weight, and body mass index that is comparable to that found in body images and videos [111]. They validate these results on a dataset of 1026 subjects.

The way users type on a keyboard (keystrokes or keystroke dynamics) has been used to make predictions about the personal information attributes of users. Irdus et al. predict the hand category (one or two-handed), gender, age (above or below 30) and dominant hand by the way users type passwords [112]. Similarly, Buriro et al. have also been successful in revealing age, gender, and operating hand with accuracies varying between 82% and 95% [113]. Uzun et al. show that children and adults (above or below 15 years) can be distinguished with high probability based on keystrokes [114]. The mouse patterns generated while interacting with web pages and desktop applications have also been used to predict the age and gender of users alongside their keystroke data [115].

Hinds et al. provide a large-scale systematic review of the types of demographic data that can be revealed by our digital interactions [116]. Apart from the widely studied age, gender and location attributes, the authors include an analysis of studies looking into predicting political affiliation, sexual orientation, ethnicity, relationship status, education level, income and religion. It is clear that most users do not intend

to share such privacy-invasive data. The use of demographic prediction technology does not allow for a safe, secure, and dignified use of digital resources.

### Mobile Devices

In many ways, the data used for predicting personal information on desktop devices is applicable, if not amplified, in the mobile phone domain. For instance, technology that reveals "soft biometrics" on social networks can, in most cases, be applied to text and images uploaded from smartphones. However, in addition to what we have described so far, there are some mobile-specific threats to personal information privacy, which we discuss in this section. The uniqueness of these threats is based on the difference of input methods in mobile devices and the fact that they are physically present with us for large parts of the day.

Malmi et al. [117] used the apps installed on a device to predict the gender, age, race, children count, marriage status, and income of mobile users. They achieved results between 60% and 80% accuracy in these categories. Other studies examine models which take advantage of the general patterns of usage in applications, browsers, WiFi and Bluetooth. Neal et al. performed gender classification using 1,000 features related to the frequency of events stemming from such data [118]. The authors achieved up to 91.8% accuracy. Using a larger dataset and more sophisticated approaches, the same authors achieved accuracies between 60% and 100% on a multitude of personal information attributes such as age, gender, education level, marital status, employment, primary language, car ownership and others [119]. Similarly, Mo et al. reveal the gender, occupation and marital status with accuracies as high as 96%, 83% and 86% respectively [120]. They use a comprehensive feature set which includes data about the applications used, calendar events, bluetooth, contacts and others.

Frias-Martinez et al. focused on gender classification using calling patterns history and achieved 80% accuracy [121]. In another study, calling patterns were used to predict more "soft biometrics" such as age, gender, employment status and education level with accuracies ranging between 60% and 80% accuracy [122].

Finally, it is also possible to make predictions about an individual "soft biometrics" by the gait movement provided by the sensors found in smartphones. A study by Van et al. utilised the built-in accelerometer and gyroscope sensors in smartphones to predict the age and gender of users. Through their research, they were able to achieve relatively high accuracy, with 76% for gender classification and a mean absolute error of 5.3 for age prediction. Despite these promising results, the practical application of this technology is limited in scope.

## 2.3 Threat model

In this section, we introduce the threat model for touch-based biometric systems. The goal is to provide a structured approach to identifying and assessing potential threats to such systems. We state the assumptions about the device, system and capabilities of potential attackers. We clarify the threat model for two cases applicable to our thesis: authentication and privacy attacks. The first case concerns attacks, where a malicious user attempts to bypass the authentication process to gain access to sensitive information or resources. The second case involves privacy attacks, where an attacker attempts to extract personal data from non-consenting users.

### 2.3.1 Authentication Threat Model

In the context of threat modelling for touch-based biometric systems, there are two primary authentication approaches: user-to-device and user-to-remote. The user-to-device authentication method involves authenticating a user to the device itself or to an application installed on it. On the other hand, the user-to-remote authentication model requires the user or application to authenticate to a third party, typically by sending data over the internet to a remote server. Although the underlying authentication modelling is essentially the same for both approaches, the threat modelling can differ significantly. It is important to consider these differences when assessing potential threats to a system or application in each scenario.

The objectives of an attacker targeting touch-based biometric systems can vary widely and may include stalking, financial theft, data breach, and other malicious

activities. Generally speaking, attackers seek to gain access to sensitive data stored on the device or perform actions that would benefit them.

In our threat model, we assume that the attacker is non-technical and is not utilising any additional malicious techniques in combination with simply using the device. Moreover, we assume that the attacker is not attempting to mimic the behaviour of the original user, but instead is using the device as they normally would. We assume that the device used for both user-to-device and user-to-remote authentication approaches is a smartphone. As such, it is equipped with a touchscreen that can provide 60Hz of touch coordinate and pressure data. Furthermore, we assume that the device is free of malware, which means that we do not consider additional potential threats arising from malware attacks.

**User-To-Device**

In this scenario, we assume the attacker has physical access to a device that is protected by a touch-based authentication system. For example, the attacker could have gained access to the device by stealing it or by finding a lost one. The objective of the touch-based authentication system is to prevent the attacker from misusing the device by identifying that their touch interaction patterns differ from those of the original user.

In our threat model, we make the assumption that the device is intended for use solely by its owner and is not shared among multiple users, such as family members or friends. Additionally, we assume that the device is either unprotected by another authentication system, or is already unlocked. For example, the user may have forgotten to lock their phone, or the attacker may have obtained the PIN of the original user, allowing them to unlock the device.

In this thesis, we generally assume that this is the authentication mode of the systems we examine. However, we specify the implications of the user-to-remote approach where relevant.

#### User-To-Remote

In this scenario, we assume that the attacker does not have physical access to the device but has access to an account of the target user. For example, the attacker may have obtained the user's bank account credentials and is attempting to initiate a malicious transaction from their own smartphone. Similar to the previous scenario, we assume that there is no tampering with the attacker's phone, the network, or the application servers.

In this scenario, the objective of the touch-based authentication system is to identify and label any suspicious transaction requests made by the attacker, even if they have obtained access to the target user's account. If the system detects that the transaction request is suspicious, the bank can require additional validation before authorising the transfer of funds.

### 2.3.2 Privacy Threat Model

In the privacy threat model, we consider the scenario where the attacker is an organisation that operates a malicious mobile website or application. We assume the attacker collects touch coordinate and pressure data from touchscreen-enabled devices, which they can capture at a frequency of 60Hz. This can be achieved through the use of JavaScript code on mobile websites or by accessing the API of the mobile operating system for applications.

In the case of a mobile website, we assume that the attacker can distinguish between a desktop computer and a smartphone device. This can be accomplished through a variety of techniques, including analysing the user agent string or examining the size of the screen.

Similar to the authentication threat model, in the privacy threat model, we assume that the user is using a smartphone with a screen that refreshes at a rate of 60Hz. We further assume that the user is non-technical and is not using any tools to block the data collection process. The goal of the attacker in this scenario is to extract personal information about the user, such as their gender, age, and other demographic data. This information can then be used for targeted

advertising, which can lead to increased profits for the attacker. However, it can also be used for more malicious purposes, such as identity theft or other fraudulent activities, as discussed in Chapter 5.

*Sometimes it's not enough to know what things mean,*
*sometimes you have to know what things don't mean.*

— Bob Dylan

# 3

# Fair Evaluation of Touch-Based Authentication Systems

## Contents


**3.1 Introduction . . . . . 36**
**3.2 Common Evaluation Pitfalls . . . . . 38**
**3.3 Prevalence of Evaluation Pitfalls . . . . . 42**
**3.4 Large-Scale Data Collection of Touch Interactions . . 45**
3.4.1 Remote collection . . . . . 47
3.4.2 Lab collection . . . . . 54
3.4.3 Dataset comparison . . . . . 58
**3.5 Continuous Touch-Based Authentication Modelling Pipeline . . . . . 60**
**3.6 Analysis of Pitfalls . . . . . 63**
**3.7 Best Practices for Evaluating Touch-Based Authentication Systems . . . . . 83**
**3.8 Conclusion . . . . . 87**


In this chapter, we begin our investigation by exploring common pitfalls affecting the evaluation of authentication systems based on touchscreen biometrics. We consider different factors that lead to misrepresented performance, are incompatible with stated system and threat models, or impede reproducibility and comparability with previous work. Specifically, we investigate the effects of (i) small sample sizes (both number of users and recording sessions), (ii) using different phone models in

training data, (iii) selecting non-contiguous training data, (iv) inserting attacker samples in training data and (v) stroke aggregation. We perform a systematic review of 30 touch dynamics papers showing that all of them overlook at least one of these pitfalls. To quantify each pitfall's effect, we designed a set of experiments and collected a new longitudinal dataset of touch interactions from 515 users over 31 days comprised of 1,194,451 unique strokes. Part of this data is collected in-lab with Android devices and the rest remotely with iOS devices, allowing us to make an in-depth comparison. We make this dataset and our code available online[1] Our results show significant percentage-point changes in reported mean EER for several pitfalls. We show that, in a common evaluation setting, the cumulative effects of these evaluation choices result in a substantial combined difference. We also largely observe these effects across the entire ROC curve. The pitfalls are evaluated on four distinct classifiers: SVM, Random Forest, Neural Network, and kNN. Furthermore, we explore additional considerations for fair evaluation when building touch-based authentication systems and quantify their impacts. Based on these insights, we propose a set of best practices that, if followed, will lead to more realistic and comparable reporting of results in the field.

## 3.1 Introduction

As described in Chapter 1, touch-based biometrics have been proposed as a way to improve the security of login-time authentication mechanisms and to enable continuous authentication while a device is being used. The field has been growing rapidly since the first studies on the topic were introduced in 2012, with at least 30 papers collecting unique swipe-and-scroll datasets published to date. Despite the growth in the field, no standard set of methods has been established to enable comparison between published work and transition to real-world deployment of the proposed systems. While authors largely report the EER as a metric of average system performance, there can be significant differences in the methods used to evaluate systems, particularly when using a static dataset. In order to fairly compare

[1] https://github.com/ssloxford/evaluation-pitfalls-touch

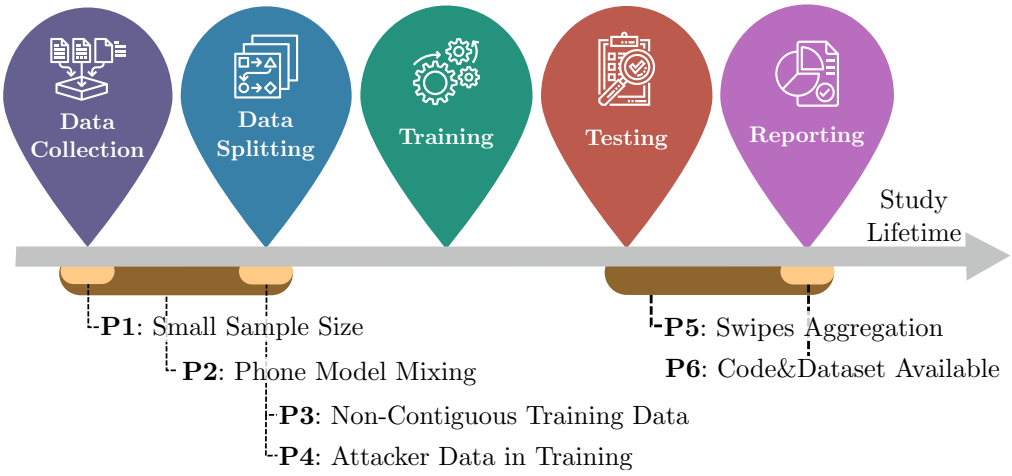



**Figure 3.1:** The six identified evaluation pitfalls in continuous touch-based authentication systems. Each pitfall occurs at a specific time over the course of a study lifetime.

the results of different studies and ensure that the conclusions are reliable and reproducible, it is important to understand and take into account the various methodological choices that can affect the reported performance of a system. This chapter aims to identify these methodological choices, investigate their prevalence in published research, and quantify their impact on the reported performance of systems. These steps are necessary in order to facilitate fair comparisons between papers, ensure the reproducibility of the conclusions, and provide results that are relevant and applicable to real-world systems and threat models.

As part of our analysis of the existing research in this area, we have identified six potential pitfalls that can occur in the design, data collection, or analysis of experiments, which can hinder comparability between studies or lead to unrealistic results. These pitfalls can significantly impact the validity and reliability of the conclusions that are drawn. Specifically, we investigate the effects of sample and model size, mixing different phone models in the analysis, using non-contiguous training data, including attacker data in training, using arbitrary aggregation windows, and the implications of code and data availability. In our analysis, we quantify the impact of each of the identified pitfalls on the equal error rate of a system. Our results show that these pitfalls can have a significant effect on the reported performance of a system, leading to conspicuous changes in the resulting EER. This

highlights the importance of considering these pitfalls when evaluating systems and taking steps to avoid them in order to obtain accurate and reliable results.

To assess the impact of the identified pitfalls on the performance of a touch dynamics system, we conducted the study using a new longitudinal, large-scale dataset of strokes. To collect this data, we recruited a sample of 515 users and asked them to interact with an application on a daily basis over a period of up to 31 days. We carefully designed our data collection methodology to ensure that we obtained a high-quality dataset that would be representative of real-world touch dynamics. In this chapter, we describe our data collection methodology in detail, including the specific methods and procedures that we used, in order to provide guidance and support for other researchers or practitioners who may be interested in collecting similar datasets in the future.

Based on the insights and findings from our study, we outline a set of best practices that can be used to avoid the identified pitfalls when evaluating the performance of a touch-based authentication system. These practices include recommendations for experimental design, as well as suggestions for ensuring the reproducibility and comparability of results in the field. We believe that these best practices will be valuable to researchers and practitioners working in this area, and hope that they will be widely adopted to improve the quality and usefulness of research in this field.

## 3.2 Common Evaluation Pitfalls

In this section, we present the evaluation pitfalls we have identified in touch-based authentication systems. To help illustrate the context in which these pitfalls can occur, we have provided a visual representation of a typical machine learning pipeline in this field in Figure 3.1. The figure shows the various stages of the machine learning process and illustrates the points at which the evaluation pitfalls that we have identified can be encountered.

**P1: Small sample size.** In this case, the term "sample size" can refer to both the number of people participating in a study and the number of data collection

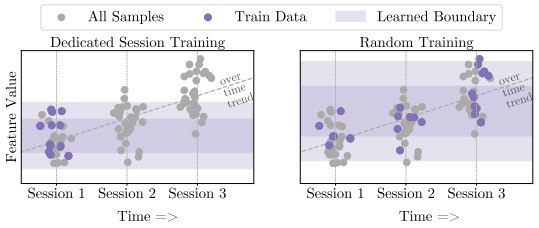



**Figure 3.2:** Examples of training data selection approaches. The "dedicated sessions" method samples data from self-contained sessions and does not posses information from future ones. The "random" method takes training samples from all session aiding in generalisation although it does not represent a realistic authentication scenario.

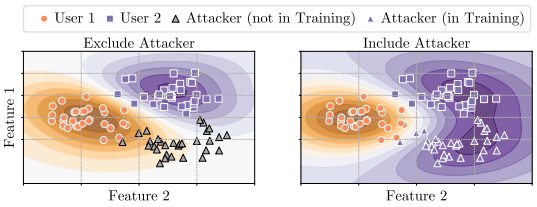



**Figure 3.3:** Visualisation of the difference between attacker modelling approaches. The "include attacker" model creates a better boundary between legitimate and invalid data but it does not represent a realistic authentication scenario as specific attacker data is rarely available at the time of model creation.

sessions recorded for each individual participant. In the context of evaluating touch authentication methods, due to various experimental limitations, it is common to work with a relatively small number of participants, with a median of ∼40 distinct individuals and two data collection sessions recorded per user. Nevertheless, the accuracy of the measured performance may benefit from a larger number of users.

When training a recognition model, it is important to consider the size of the sample used for negative training data, as this can have a significant impact on the performance of the model. In particular, using larger pools of users for negative training data can result in differences in the mean system performance. However, it is also necessary to consider the temporal aspect of data collection, as user behaviour may change over time and across different sessions. Collecting longitudinal data over an extended period of time can help to estimate the effect of these changes on model performance. These sample size effects are non-trivial to measure and hinder a robust generalisation of results found on smaller samples.

**P2: Phone model mixing.** There are many reasons why researchers in the field of touch authentication may choose to collect data from multiple distinct device models. For example, it may be more convenient to use a variety of devices, especially if the study is being conducted remotely. Additionally, some studies may be designed to

evaluate the performance of a touch authentication system on different hardware in order to demonstrate its versatility. It is worth noting that even if two phone models look similar on the surface, there can be significant differences in the resulting touchscreen data. This variations can be caused by a range of factors, including the physical shape of the phone, the resolution of the display, how the phone is held, the touchscreen sampling rate, and the range of values that can be measured by the pressure and area sensors. All of these factors can affect the accuracy of touch data collection and, as a result, the performance of the recognition system.

When evaluating the security of a touch authentication system, it is generally assumed that an attacker would use the same phone model as their victim, as this is how it would happen in an in-person attack. Mixing phone models during the testing phase violates this requirement, as attackers and victims use different device models. It is worth noting that this issue does not apply in the case of remote authentication, where the attacker can send data from any device model. In this scenario, the device model used by the attacker is not necessarily relevant, as long as the data collected is representative of the types of touch inputs that the system is designed to recognise.

**P3: Non-contiguous training data selection.** In practice, a biometric authentication system has an enrolment (the model training) phase, which precedes the use of the system (or its evaluation). However, when using the randomised training data selection method, strokes are randomly sampled from the whole user data as shown in Figure 3.2 (right). This does not resemble how a deployed system works, as it essentially evaluates the system by testing on samples from the past. This can result in incorrect performance estimates, as the system is not being tested in a manner that accurately reflects how it would function in practice. As a consequence, randomised training data selection leads to biased performance estimation.

**P4: Attacker data in training.** While there are several ways to design an authentication method, a common approach is to use a binary classifier that discriminates between legitimate and non-legitimate user samples. In this case, the negative samples (non-legitimate) are generally gathered from the available pool of users, and the same user pool is then used to test the system recognition rates.

However, most stated threat models rule out the possibility that the classifier was trained with negative training data belonging to an attacker: attacker samples should be *unknown*. Figure 3.3 illustrates this problem: including the attacker samples in the training data provides a significant benefit against attacks compared to what happens when the attacker is excluded from training. This property has been initially addressed in [74], where it is shown that it artificially reduces the zero-effort attack success rates. The inclusion of an attacker in training data is incompatible with a realistic threat model. It is important to clarify that the attacker data we use to delineate the negative class consists of legitimate strokes of other users. While active attacks are interesting to examine, we limit our analysis to zero-effort attackers.

**P5: Aggregation window size.** Intuitively, it can be understood that the use of multiple strokes when evaluating a particular model leads to an improvement in performance [1, 2, 62, 69, 123]. It is a legitimate practice to combine multiple strokes for the purpose of making an authentication decision, as it can help to prevent erratic behaviour and improve recognition. However, this approach also has two significant drawbacks. Firstly, it impedes straightforward comparison between different approaches when the aggregation window size is different. Secondly, in a realistic threat scenario, it provides the attacker with a non-negligible amount of time to carry out an attack since anomalous behaviour on the part of the attacker will not be detected until a certain number of strokes have been made, which depends on the size of the aggregation window.

**P6: Dataset and code availability.** It is not uncommon for researchers to be hesitant to share their datasets and code, especially if they perceive that doing so may compromise the novelty of their work or put them at a competitive disadvantage. The field of touch-based authentication is not an exception in this regard. This is a major impediment to reproducibility and progress in the field, as it does not allow other researchers to verify the findings of the study and build upon the work in new and innovative ways. Sharing datasets would enable researchers to reliably separate the effects of different models from those of the collected data. Furthermore, sharing the code used to obtain the results is especially important in light of the pitfalls

investigated in this chapter: oftentimes unstated assumptions are made which are not trivial to spot. Dataset and code sharing also helps to increase transparency and accountability in research, leading to more robust and reliable results.

## 3.3 Prevalence of Evaluation Pitfalls

The main theme of this chapter and the entire thesis is the investigation of mobile continuous authentication systems that are based on the way users swipe and scroll on their devices. While our research specifically focuses on the use of strokes, which are the most common form of touch input, there are other touch gestures that can also be used for authentication purposes. Examples include interactions such as "pinch to zoom" [124] and screen taps [125]. In this study, we will only be examining the use of swipes and scrolls, which are defined as horizontal and vertical movements on a touch-sensitive display made with a single finger.

In order to investigate the commonality of the issues outlined in the previous section, we conducted an analysis of the touch-authentication literature. Our analysis included a review of 30 studies published in the last ten years, each of which introduces a new and unique touch-based dataset. For our analysis, we only selected studies that involved experiments that incorporated natural swiping behaviour, such as navigating through common mobile interaction tasks. We excluded studies that only utilised mobile keystroke dynamics, sensors, tapping, and one-time gestures for authentication purposes.

Table 3.1 provides information about the prevalence of the evaluation shortcomings in the way the study was conducted or reported. The table indicates which details are often not shared by the authors of the studies included in the table. It is noteworthy that all of the studies listed in the table are subject to at least one of the pitfalls described in Section 3.2 of this chapter. The patterns that emerge from Table 3.1 regarding these evaluation pitfalls will are discussed in more detail throughout the chapter.

In our set of papers, there is a roughly even distribution of studies conducted in a lab environment versus those conducted remotely. Specifically, 15 of the

**Table 3.1:** Data collection and analysis choices in touch dynamics studies. ● denotes that the study fulfills the column recommendation (i.e., does not fall into the evaluation pitfall) and ○ denotes that it does not, ? means that the information was not shared or it is unclear from the paper, — means not applicable and ◐ in the last column means that the code or dataset is no longer available through the provided url (accessed on 4 January 2023)). The "Cont. (Period)" Sessions label indicates that the phone was given to the users for a period of time without specific instructions on how often to use it. The "Single Device Model" column marks whether the analysis separates data belonging to distinct phone models (even if the data collection included various phone models).

| | | P1 | | P2 | P3 | P4 | P5 | P6 |
|---|---|---|---|---|---|---|---|---|
| Study (Year) | Environment | Users | Sessions | Single Device Model | Contiguous Training Data | Exclude Attacker | Single Gesture Analysis Available (Aggregation Sizes) | Dataset / Code Availability |
| [1](2012) | Lab | 41 | 3 | ○ | ● | ○ | ● (1-20) | ● / ○ |
| [126](2012) | Lab | 40 | 1 | ● | ? | ○ | ● (1-9) | ○ / ○ |
| [2](2013) | Remote | 75 | Cont. (?) | ● | ● | ● | ○ (2-20) | ○ / ○ |
| [76](2013) | Remote | 100 | Cont. (?) | ? | ? | ○ | ● (1-30) | ○ / ○ |
| [69](2013) | Lab | 190 | 2 | ● | ● | ○ | ○ (10) | ◐ / ○ |
| [127](2014) | Remote | 32 | Cont. (5-10 weeks) | ○ | ● | ○ | ● (1) | ○ / ○ |
| [68](2014) | Remote | 23 | Cont. (3 weeks) | ○ | ○ | ● | ● (1-10) | ○ / ○ |
| [128](2014) | Lab | 20 | 1 | ● | ○ | ● | ● (1) | ○ / ○ |
| [62](2014) | Lab | 78 | 6 | ● | ? | ? | ● (1-7) | ○ / ○ |
| [124](2014) | Lab | 32 | 1 | ● | ● | ● | ● (1,3,5) | ◐ / ○ |
| [129](2015) | Lab | 50 | 1 | ● | ● | ○ | ● (1-19) | ○ / ○ |
| [130](2015) | Lab | 20 | 1 | ● | ? | ? | ● (1) | ○ / ○ |
| [67](2015) | Remote | 71 | 4 | ○ | ? | ○ | ● (1-20) | ● / ○ |
| [131](2015) | ? | 14 | 1 | ● | ? | ○ | ● (1-15) | ○ / ○ |
| [132](2015) | Remote | 22 | 30 | ● | ● | ○ | — | ○ / ○ |
| [133](2015) | Lab | 73 | 2 | ● | ● | ○ | ● (1) | ○ / ○ |
| [134](2016) | Lab | 24 | 3 | ● | ● | ○ | ● (1-20) | ○ / ○ |
| [135](2016) | Lab | 40 | 1 | ● | ○ | ○ | ● (1-5) | ○ / ○ |
| [136](2016) | Remote | 48 | Cont. (2 months) | ● | ● | ○ | ○ (2-16) | ● / ○ |
| [123](2016) | Remote | 28 | 7 | ○ | ● | ○ | ○ (4) | ○ / ○ |
| [137](2017) | ? | 40 | 1 | ○ | ? | ? | ● (1,5,11) | ○ / ○ |
| [71](2017) | Remote | 40 | Cont. (6 months) | ● | ● | ○ | ● (1) | ○ / ○ |
| [138](2017) | Lab | 20 | 8 | ● | ○ | ● | ● (1) | ○ / ○ |
| [139](2017) | Lab | 20 | 1 | ● | ● | ○ | ● (1-5) | ○ / ○ |
| [140](2018) | Remote | 48 | 20 | ● | ● | ○ | ● (1) | ○ / ○ |
| [141](2019) | Lab | 31 | 8 | ● | ? | ○ | ○ (5) | ● / ○ |
| [142](2019) | Remote | 2218 | 1 - 7619 | ○ | — | — | — | ● / — |
| [143](2019) | Remote | 45 | Cont. (2 weeks) | ● | ? | ○ | ● (1) | ○ / ○ |
| [144](2020) | Lab | 30 | 1 | ● | ○ | ○ | ● (1) | ○ / ○ |
| [145](2021) | Remote | 600 | 5 | ○ | — | — | — | ● / — |
| Ours | Remote | 470 | 31 | Both | Both | Both | ● (1-20) | ● / ● |

papers report conducting their research in a lab setting, while 13 of the papers describe their study as being conducted remotely. The collection environment for the remaining two studies is unclear.

We find that the median number of participants in the studies included in our set is 40 and that participants complete a median of 2 sessions. These relatively low numbers are a point of concern, and we analyse the potential impact of this factor (P1) in Section 3.6 of this chapter. In addition, we found that seven of the studies provided participants with devices to use for a specific period of time but did not provide specific instructions on how often these devices should be utilised. As a result, the precise number of sessions for these studies is unknown.

In our analysis, we found that approximately 28% of the studies mixed different device models in their data collection process without explicitly discussing separating them or considering the effects in their evaluation, hence falling into pitfall P2.

Likewise, approximately 30% of the studies fail to adequately explain the process by which they select their training and testing data. Additionally, a further 18% of the studies use a randomised approach for selecting their data, which may not be the most appropriate approach, as discussed in P3. For those that do not explain their selection process, the code is also not shared, making it impossible to know how the selection was performed.

In terms of attacker modelling, an overwhelming majority (80%) of the studies investigated use an unrealistic attacker modelling approach and include attacker data into the training set, falling victim to P4. A much smaller number of studies succumb to P5, with 17% reporting their results only on the analysis of an aggregation group of more than one stroke. This hinders the comparability across studies.

P6 also captures many works, with only 8 studies (27%) sharing their datasets upon publication, two of which no longer have functional web pages. Furthermore, none of the studies we examined shares a complete codebase of their work. One study [1] does share the feature extraction code files but not the rest of the analysis.

There have been a number of recent studies that have been able to gather a significant amount of data by making collection apps available on public app stores

such as the "Google Play Store" and "Apple App Store" [142, 145]. While this is a positive development in terms of increasing the size of datasets, it also presents a number of challenges and considerations that need to be taken into account. For instance, in the case of [142], there is data from 2218 users collected on 2418 different devices, and in [145], there is data from 600 users on 278 distinct devices. These numbers suggest that there is likely a significant variation in the types of unique device models being used, particularly when taking into account the fact that the Android ecosystem is known for its high level of fragmentation. Additionally, it is possible that multiple people may be using the same account to perform the tasks, such as a parent allowing a child to play the game on their phone.

## 3.4 Large-Scale Data Collection of Touch Interactions

The primary goal of our data collection experiment was to carefully examine the effects of the various pitfalls described in Section 3.2. To achieve this, we designed our experiment to be as comprehensive and robust as possible. This required us to make certain decisions and choices that resulted in our dataset being different from previous ones in several notable ways. These differences allowed us to more accurately measure the impacts of the pitfalls under investigation and to draw more informed conclusions from our results.

Specifically, we gathered data remotely from a carefully selected and constrained set of devices, ensuring that the hardware and software environments were as consistent as possible. Additionally, we recruited a large number of participants (470 individuals, well above the median of 40) and collected data from them over multiple sessions (up to 31 sessions per participant, compared to the median of 2). In addition to that, we also conducted in-person data collection sessions with 45 participants, during which each individual completed the same tasks on three different Android devices. This supplementary data provided us with the opportunity to compare the results from this method to those obtained via the remote collection method. This comparison allowed us to assess the potential biases

or variations that may exist between the two approaches, as well as make stronger conclusions about P2: Phone model mixing.

As mentioned in Chapter 1, we received ethical approval to conduct these experiments from our institution. The approval code is SSD/CUREC1A_CSC_1A_19_013. In the remainder of this section, we discuss the designs of the key parts of our data collection experiment.

The data relating to touch interactions is typically collected and organised in a specific way. Generally, a series of points is recorded while the finger is in contact with the touchscreen, and these points are captured at the refresh rate of the display of the device. This refresh rate refers to the number of times per second that the display is updated, and it is usually set at 60Hz (or 60 updates per second). Some newer devices have higher refresh rates, such as 90Hz or 120Hz, where the display is updated even more frequently. However, the ones in our experiments are operating at 60 Hz.

The recorded points consist of the X and Y coordinates at the contact point, the area covered by the finger, the touch pressure and a timestamp at the moment of recording. In some cases, additional data about the touch event may be recorded, such as the action being performed at the point of contact (e.g. putting a finger down on the screen, dragging a finger across the screen, lifting a finger off the screen) or the task that is currently performed. Example data storage of such values is given in Table 3.2, although the specific contents and structure of the table may vary on the needs of the system.

The accelerometer and gyroscope sensor data recording is stored in a similar manner. The refresh rate of these sensors is typically much higher than for touchscreens. For example, the iPhone maintains a minimum of 100Hz and can get up to 400Hz depending on the hardware and settings of an application. The main values that are typically recorded and stored for accelerometer and gyroscope data are the X, Y, and Z velocities, which indicate the movement or orientation of the device in different dimensions.

**Table 3.2:** Data format for storing touch interactions. The are and pressure values are in the range of 0 to 1, while the X and Y coordinates are bound by the screen resolution.

| Timestamp | X | Y | Pressure | Area | Action |
|---|---|---|---|---|---|
| 1334789740143 | 255 | 327 | 0.42 | 0.13333336 | FINGER_DOWN |
| 1334789740232 | 242 | 327 | 0.53 | 0.15555558 | MOVE |
| 1334789740262 | 228 | 328 | 0.64 | 0.1777778 | MOVE |
| 1334789740350 | 157 | 322 | 0.64 | 0.20000002 | MOVE |
| 1334789740402 | 101 | 320 | 0.64 | 0.22222224 | MOVE |
| 1334789740420 | 78 | 326 | 0.64 | 0.15555558 | MOVE |
| 1334789740463 | 54 | 337 | 0.18 | 0.04444445 | FINGER_UP |

### 3.4.1 Remote collection

One major benefit of remote collection is that it allows researchers to gather large amounts of data from a wide range of participants, regardless of their location. This is particularly useful when studying biometric systems where experiments need to be performed frequently. Another advantage of remote data collection is that it provides a way to continue the experiments in situations where it is impractical or impossible to conduct lab studies. For example, during the COVID-19 pandemic, many researchers were forced to shift to remote data collection methods due to lockdowns and travel restrictions, which made it difficult or impossible to conduct in-person research, leaving remote collection as the only viable option.

For this collection modality, we employed the use of Amazon Mechanical Turk (MTurk) — a widely-used crowdsourcing platform. MTurk is a platform that connects workers with businesses or researchers who need tasks completed that require human intelligence and judgement, known as Human Intelligence Tasks (HITs). Workers can choose to complete these tasks in exchange for payment. One advantage of using MTurk for data collection is that it has a diverse user base with workers from a variety of countries and backgrounds, which gives researchers access to a large pool of potential participants. Additionally, MTurk provides tools for

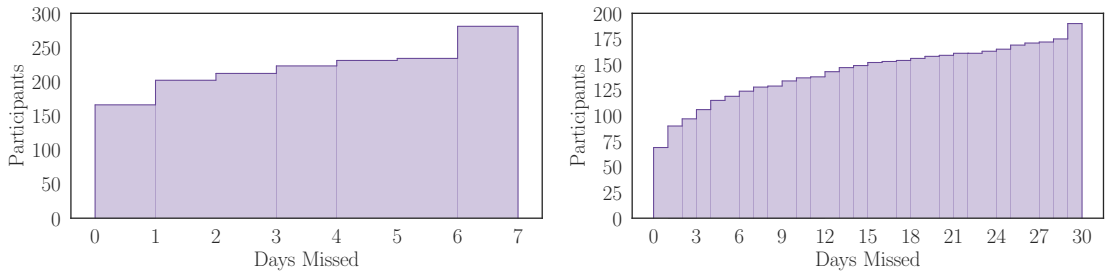



**Figure 3.4:** Cumulative distribution function (CDF) of participation retention in the remote data collection for seven-day (left) and 31-day (right) user batches.

researchers to target specific demographics, such as specific age ranges or genders, which can be useful for fair representation of participants.

For our data collection efforts on MTurk, we created a HIT that provided all of the necessary information and instructions for participants to complete the experiment. This HIT included details about the purpose of the study, the requirements for participating, and any other relevant information that participants needed to know.

To facilitate the collection of data from participants, we also provided a link to an app that participants were required to install on their devices. This app was distributed through TestFlight, which is an online service that allows developers to share beta versions of their iOS applications with a group of testers. TestFlight allows for easy over-the-air installation of apps on iOS devices and does not allow the general public to install the beta version of the application. As required by our institutional review board, the HIT also contained the participant information sheet, which provided participants with important information about the study, such as the purpose of the research, the risks and benefits of participating, and their rights as a participant.

Collecting data from participants over an extended period of time on MTurk could be a useful but intricate process. Turner et al. conducted a standalone study based on our data collection efforts [146].

### Study Duration

During the course of the study, individuals were invited to participate for either 7 days or 31 days. Each day, those who had opted to take part in the study would

**Table 3.3:** Specification sheet details for phone models used in our experiments.

| Model | Screen size | Resolution | Pixel density | Users | Accelerometer | Gyroscope |
|---|---|---|---|---|---|---|
| iPhone 6S | 4.7in | 1334x750 | 326 ppi | 70 | ● | ● |
| iPhone 6S Plus | 5.5in | 1920x1080 | 401 ppi | 19 | ● | ● |
| iPhone 7 | 4.7in | 1334x750 | 326 ppi | 73 | ● | ● |
| iPhone 7 Plus | 5.5in | 1920x1080 | 401 ppi | 50 | ● | ● |
| iPhone 8 | 4.7in | 1334x750 | 326 ppi | 68 | ● | ● |
| iPhone 8 Plus | 5.5in | 1920x1080 | 401 ppi | 55 | ● | ● |
| iPhone X | 5.8in | 2436x1125 | 458 ppi | 71 | ● | ● |
| iPhone XS | 5.8in | 2436x1125 | 458 ppi | 34 | ● | ● |
| iPhone XS Max | 6.5in | 2688x1242 | 458 ppi | 30 | ● | ● |
| OnePlus 5 | 5.5in | 1920x1080 | 401 ppi | 45 | ● | ● |
| BLU VIVO 6 | 5.5in | 1920x1080 | 401 ppi | 45 | ● | ● |
| MOTO G 3 | 5.0in | 1280x720 | 294 ppi | 45 | ● | ○ |

receive a notification from the app (provided they had enabled notifications) at 9:00am in the morning and again at 7:00pm in the evening, reminding them to complete the task for that day if they had not yet done so. Although all users were asked to complete their tasks on a consistent basis, this did not always occur. The cumulative distribution functions plots, which show the progress of the two groups participating in the remote experiment (those who participated for 7 days and those who participated for 31 days), are displayed in Figure 3.4. It can be seen that the majority of users who completed the initial few sessions of the experiment went on to complete most of the tasks throughout the duration of the study.

### Devices

We ultimately decided to use the iOS platform for our remote data collection efforts. One of the main reasons for this decision was the desire to maintain consistency in terms of hardware and software. By using a single operating system, we were able to eliminate any potential variability that could impact the accuracy and reproducibility of our results. On the other hand, the Android operating system,

while widely popular, presents a number of challenges when it comes to large-scale analysis. This is due to the fact that there is a much higher number of individual device models available, each with its own unique set of characteristics, including different screen sizes and sensors. Moreover, the majority of Android devices approximate their reported touch pressure values by considering the size of the touchpoint. The iPhone models we have chosen, on the other hand, support "3D touch", which is a true pressure sensor built into the screen of the devices. Due to these restrictions, we have narrowed down our remote collection efforts to the nine iPhone devices shown in Table 3.3.

The design decisions that were made resulted in a significant number of users utilising a limited number of models. This reduced the variability in our analysis and allowed us to make stronger conclusions about our results. However, we were still able to make comparisons in terms of the phone sizes, resolutions, and hardware variations among these models. To our knowledge, there is only one study [129] in the field which focuses on iOS devices for touch-based authentication. The dataset resulting from this study, however, is not publicly available. While we have placed specific restrictions on our data collection and experimentation, the dataset can be used for developing systems beyond the specifics of this thesis.

### Application

To facilitate our study, we developed an iOS application that collects touch and sensor data as users perform common smartphone interactions. We collected coordinate and pressure data for each user interaction with the screen at the maximum refresh rate of 60Hz. Furthermore, we also recorded the accelerometer and gyroscope data at a frequency of 100Hz.

Upon launching the application, users were presented with a consent form that they were required to complete before proceeding. This form sought to obtain approval for conducting the experiment and collecting demographic information similar to what might be collected in a laboratory study. After completing the consent form and providing the necessary information, users were required to

complete their first set of tasks once. This established a connection between MTurk and the application enabling the users to receive their first payment and ensuring that subsequent payments would be automatically generated each time they complete a task. By establishing this connection, we were able to streamline the payment process and ensure that our participants were fairly compensated for their time and efforts.

The application required users to complete two different tasks. The first task was a social media style task, which involved interacting with content in a way that is similar to how social media platforms work. The second task was an image gallery task, which involved viewing and interacting with a collection of images. The purpose and design of these tasks are explained in more detail in Section 3.4.1 of this chapter. In order to ensure that both tasks took an equal amount of time to complete and that we were able to collect a similar amount of data from each task, we carefully optimised the number of rounds repeated for each task. Both tasks were intended to be completed with the phone in a vertical position, and thus we did not allow a change in the layout when the device was rotated.

In order to encourage users to continue using the application and participating in the study, we included various elements on the home page that were designed to increase user retention. These elements included information about the user's completion streak, which indicated how many days in a row they had done the tasks, as well as information about their earning potential, which gave them an idea of how much money they could potentially earn by doing the task. By providing users with these types of incentives and rewards, we hoped to increase their motivation and engagement with the application, and encourage them to continue using it throughout the study.

We ensured that the order in which the two tasks were presented did not affect the results of the study. This was done by randomly determining the order before each session commenced, meaning that users could potentially complete the social media task before the image gallery task or vice versa. Regardless of the order in which the tasks were presented, we provided clear instructions to users before each task began, explaining exactly what they needed to do in order to complete it successfully.

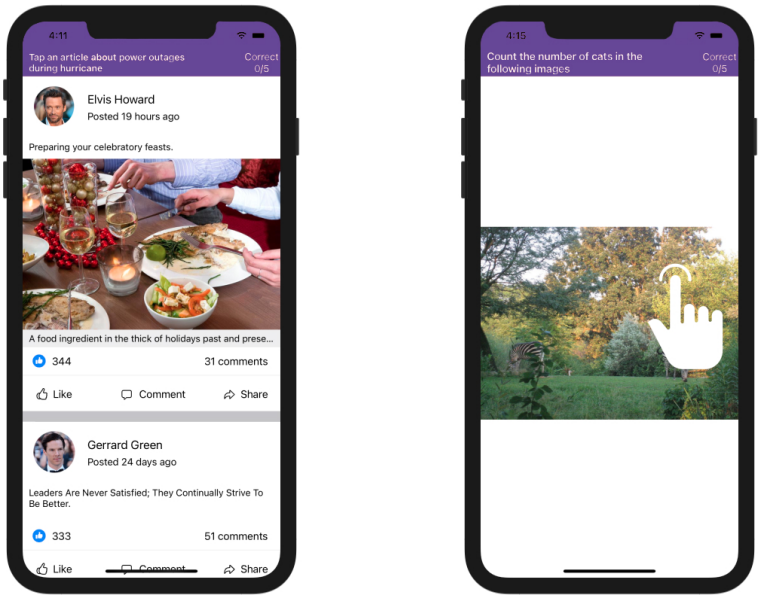



**Figure 3.5:** Screenshots from the remote data collection application on iOS devices. The vertical scrolling social media task is presented on the left and the horizontal swiping image gallery task on the right.

We required participants to complete five rounds of each task. During these rounds, we validated the correctness of their answers in order to ensure the legitimacy of the data and avoid abuse. If a user made a mistake on a particular round, they were prompted to repeat that round. Once both tasks were completed, the touch and sensor data that had been collected was transmitted and stored securely at a remote server.

### Task Design

As mentioned, we designed two tasks for users to perform: a social media task and an image gallery task.

The goal of the social media task is to gather touch data by simulating how users tend to use their phones on common vertical scrolling tasks. For instance, oftentimes mobile users browse social media feeds or scroll through a list of news articles. In this task, the objective was for users to search a list of articles and posts by scrolling through a feed in order to match one to a given description. For instance, our given description could be "Tap an article about gift ideas for graduate students and dads" and the actual article being "Tech Guru Shares Top Gift Ideas

for Dads and Grads" with a relevant image attached to it. These articles and their accompanying images were sourced from NewsUSA [147] and selected to be free of copyright restrictions. In addition, a clear and non-ambiguous description was provided for each article. There were a total of 600 article or post descriptions available in the system, each corresponding to a unique article or post. In each iteration, a single description-answer pair is chosen and mixed with a selection of decoy posts, creating a feed of 20 items. The goal was for users to locate the correct article pair among the mix of items presented in the feed.

The goal of the image gallery task is to gather touch data by simulating how users tend to use their phones on common horizontal swiping tasks such as browsing a list of photos or application screens. In this task, users were presented with a horizontal list of pictures in which only a single image was visible at any given time. As part of the image gallery task, users were asked to keep track of the number of times a particular object appeared as they swiped through the series of images. For example, the objects in question might depict different types of animals, such as dogs and cats, or different types of food, such as pizzas and fruits. All of the images used in the image gallery task were sourced from the "Common Objects in Context" (COCO) dataset [148], which is a large collection of images that have been labelled and annotated for use in computer vision research. There were a total of 200 unique images in the task, and each round of the challenge presented users with a gallery of 20 pictures to swipe through. The images were selected in such a way that between 2 and 6 of them would contain the target object that users were asked to count. Once users had finished swiping through the gallery and counting the objects, they were required to enter the total number of objects they had observed.

Example images from the two tasks described in this section are shown in Figure 3.5.

### Limitations

When conducting a remote data collection experiment, the absence of direct oversight by the experimenter can present a number of challenges. One concern is the

possibility of participants completing the study multiple times. This is highly problematic since the user data would appear twice in the dataset under different labels. However, we believe it would be quite difficult for them to do so successfully. In order to complete the study twice, a participant would need to have access to two MTurk accounts, two Apple accounts, and two physical devices. They would also need to be able to accept and complete the HIT twice before it expires, which would require a significant amount of time and effort. Therefore, while it is important to be aware of the potential for participants to try to cheat in this way, the logistical challenges involved make it unlikely to be a major issue in most cases.

Another concern is the possibility of participants enlisting the help of others to complete some of their sessions, which could also compromise the validity of the data. While it is harder to rule out this entirely, we have reminded participants not to do so at the start of each session. It is worth highlighting that the impact of this behaviour would likely be limited to individual sessions rather than affecting multiple users.

Additionally, it is important to note that the data may have been gathered under a variety of uncontrolled conditions that may vary significantly between different users and even different sessions for the same user. For example, a user may be using our application while sitting down, walking around, holding the phone in their hand, or placing it on a table. While these variations in usage conditions may negatively impact the overall performance of the system, it is important to consider them as they provide a more realistic representation of how the system will be used in real-world situations.

### 3.4.2 Lab collection

In order to thoroughly understand the potential limitations and differences of our remote data collection method, we decided to gather an additional dataset through in-person means. This allowed us to compare the results obtained through the two different methods and identify any issues that may have arisen during the remote data collection process that would not have been detectable otherwise.

There are several advantages to collecting data in person as opposed to remotely. One of the major benefits is that researchers are able to directly observe and interact with participants as they complete the tasks. This allows for a greater level of control over the data collection process and can help to ensure that the tasks are being performed correctly. In addition, the risk of encountering duplicate users or having participants pass their phones on to someone else to complete the tasks on their behalf is eliminated when data is collected in person. Another advantage of in-person data collection is that it allows for more accurate verification of the demographic information provided by subjects.

For this particular data collection variation, we asked participants to complete two tasks on three different Android devices, all in one sitting. In order to reduce the potential for bias, we took steps to randomise the order in which participants received the phones and the order in which they completed the tasks.

### Study Duration

The study design included three sessions per participant, each of which involved the completion of two tasks. These sessions were conducted on three distinct devices. The entire experiment was intended to take approximately 15 minutes to complete, and all data was collected in person over the course of a two-month time frame. Each participant completed all three sessions during a single session, with the tasks being performed consecutively on each of the phones in a random order.

### Devices

The data was collected on Android devices in contrast to the iOS devices used in the remote data collection. In total, we used three devices — the OnePlus 5, Blu Vivo 6 and Moto G3. The OnePlus and Blu smartphones have equal resolution and pixel density but also a slightly different form factor when held in hand. The Motorola phone has a lower resolution and pixel density than the other two. More detailed information regarding the devices used in the study can be located in Table 3.3. This table includes detailed specifications and characteristics of the devices, such as their screen size, resolution, pixel density and available sensors.

### Application

For the purpose of these experiments, we created an Android application that was similar in design to the one we had previously used for remote data collection. When users first opened the app, they were presented with a consent form that they were required to complete before proceeding. This consent form provided information about the nature of the experiments and obtained the users' agreement to participate. In addition to the consent form, our Android application also included a set of optional demographic questions that asked users for information such as their age, nationality, and level of experience using a smartphone. Before beginning each task, the participants were provided with both written and verbal explanations of what they were expected to do. They were not restricted in terms of how they held the phone or how they used it and were free to interact with the device in whatever way they felt most comfortable.

The application recorded detailed information about the user's touch interactions with the device's screen. The data was collected at a rate of up to 60 times per second, which is the maximum refresh rate of these particular devices. In addition to touch data, the application also collected sensor data from the device's accelerometer and gyroscope. However, it should be noted that the Moto G3 does not have a gyroscope, so no gyroscope data was collected for this particular device. The application required users to perform two different types of tasks. The first task was a social media style task that involved interacting with a simulated social media platform, similarly to the remote collection. The second task was a "spot the difference" game, in which the user was presented with two images and had to identify and tell the differences between them. Both of these tasks were designed to be completed in a vertical orientation (opposite of landscape). We carefully balanced the amount of time allotted for completion (2 minutes) and the total number of strokes that were required to complete each task.

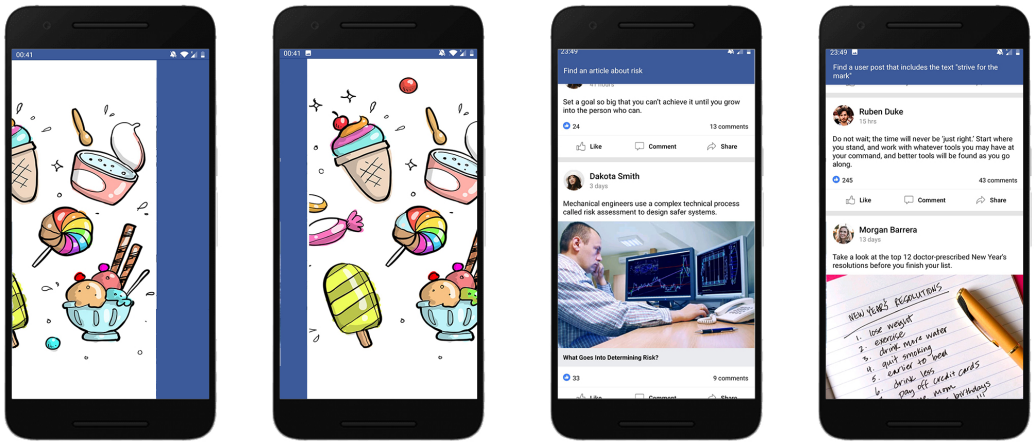



**Figure 3.6:** Screenshots from the in-person data collection tasks on Android devices. The horizontal "spot the difference" task is shown on the left and the vertical scrolling social media task is presented on the right.

### Task design

The two tasks we designed for this set of experiments were similar to the remote collection tasks. However, in this case, the tasks had only three variations, as opposed to the random feeds generated in the iOS variation. Here we had one variation for each phone, such that users do not get accustomed to the challenges when performing them consequently.

The social media task was nearly identical to the remote one with the same purpose of collecting vertical scrolling behaviour. Users were asked to scroll through a social media feed and find articles relating to a particular topic or posts which include a specific phrase. Unlike the remote collection, the feed order was always predetermined, as there were only three variations — one for each phone.

The spot the difference game was aimed at collecting horizontal swiping data. It involved an image comparison game which instructed participants to find differences between two pictures. The pictures were copyright-free, and the respective differences were digitally added to them. In order to ensure that the participants in the study were not able to cheat by quickly glancing at both images at the same time, the images were separated by a blue screen. This screen prevented the subjects from being able to see both pictures simultaneously, requiring them to lift their finger off the screen and swipe back and forth between the images in order to compare

**Table 3.4:** Summary and comparison of the two datasets collected in this study.

| Collection Method | Users | Strokes | Sessions | Mean User Sessions | Devices | Tasks | Stroke duration |
|---|---|---|---|---|---|---|---|
| Remote | 470 | 1,166,092 | 6,017 | 13 | 9 | 2 | 58ms |
| in-person | 45 | 28,355 | 135 | 3 | 3 | 2 | 241ms |

them. Additionally, in order to make the game more challenging and prevent players from simply memorising the location of the differences, three different sets of images were used between devices, each containing a unique set of differences. This task mimics common actions performed on mobile phones, such as browsing an image gallery or lists of applications.

Example images from both tasks can be found in Figure 3.6.

#### Limitations

The dataset that we have gathered is quite small in comparison to the remote dataset, both in terms of the number of users and the number of sessions. Additionally, we did not repeat the experiments over multiple sessions on different days. This means that our dataset may not be as comprehensive or reliable as the remote dataset. The difficulty in collecting in-person data at scale contributed to these limitations.

We collected data on Android devices, which allows us to make a comparison between ecosystems and show that touch-based authentication can be deployed in a variety of settings. However, it would have also been beneficial to collect the data in person using the same iOS devices as the remote collection. This would have allowed us to compare differences between in-person and remote data collection without the issue of introducing performance changes due to the mixing of different models. Despite this limitation, we were still able to gather valuable insights in other ways from the in-person data collection.

### 3.4.3 Dataset comparison

We summarise the differences between the lab-collected and remote-collected datasets in Table 3.4. In total, the remote dataset consists of 470 users amounting

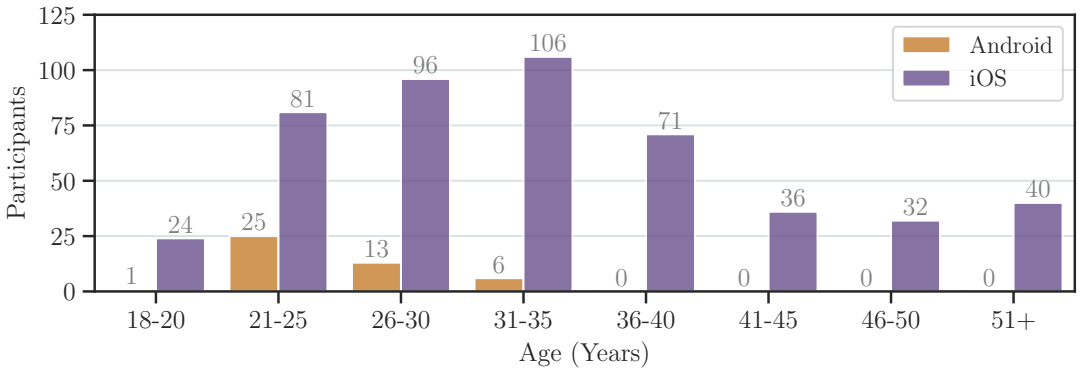



**Figure 3.7:** Age of participants in the experiment. Remote collection through Amazon Mechanical Turk allows for large scale collection and more diverse participants compared to traditional university lab studies.

to 6,017 unique sessions and 1,166,092 unique strokes. On average, each user participated in 13 sessions during the remote data collection process. The in-person dataset consisted of 45 users, 135 unique sessions and 28,355 strokes. All tasks in both the remote and lab settings took approximately 2 minutes to complete on average. The social media task specifically resulted in an average of 79 strokes in the remote setting and 101 strokes in the lab setting. The horizontal swiping tasks resulted in an average of 124 strokes in the remote setting and 108 strokes in the lab setting. The average duration of a stroke was found to be 58 milliseconds in the remote case and 241 milliseconds in the lab study.

The use of the MTurk platform for remote collection resulted in a dataset that was relatively balanced in terms of several demographic characteristics, including age, gender, handedness, and iPhone model. Specifically, the gender distribution of all users in the dataset was almost perfectly balanced, with 47% identifying as female (229 individuals), 51% identifying as male (252 individuals), and the remaining 1% identifying as neither male nor female (5 individuals). Only a small percentage of the participants in the study (14%, or 67 individuals) reported being left-handed, which is roughly comparable to the prevalence of left-handedness in the general population (which is estimated to be around 10%). The age distribution of the participants is depicted in Figure 3.7, which shows that the majority of participants were in the 31-35 year age range, but the dataset also includes individuals from

a wide range of age groups. These results suggest that our subject sample was representative of a diverse group of individuals.

## 3.5 Continuous Touch-Based Authentication Modelling Pipeline

In this section, we present our data and machine learning pipeline. We describe how we investigate the effect of the pitfalls P2, P3, and P4, as part of the specific steps they appear in. P1 and P5 are analysed directly by varying the sample size (part of any touch-based modelling) and the aggregation window size (the sample aggregation step is required), respectively.

**Division by phone model.** As outlined in Section 3.4.1, our larger, remote dataset consisted of 9 distinct phone models. These devices, although similar in terms of hardware and sensors, do have some variations in terms of their screen size, resolution, and overall shape. To ensure that the effect of P2 on our results is measured and controlled, we can create distinct subsets of data by separating out the data collected by each individual phone model. We label these subsets using the corresponding phone model name, such as XS MAX for example. We compare the performance on this phone model-specific subsets with the performance computed on the entire dataset containing data from all models, which we refer to as COMBINED.

**Preprocessing and feature extraction.** As the first step, we take all of the touch samples that have been collected during a specific task and group them into two categories: horizontal swipes, which occur when performing the image gallery task, and vertical scrolls, which occur when using a social media application. These touch samples contain information about the X/Y coordinates of the touch and the pressure that was applied. In all subsequent steps, the horizontal swipes and vertical scrolls are analysed separately and independently of one another.

To ensure that we are only analysing genuine swipes and scrolls and not including any unintentional strokes in our dataset, we apply a few filters to our data. Specifically, we remove any strokes that are shorter than three samples or

that do not deviate more than 5 pixels from the starting point. For each of the remaining swipes and scrolls, we then calculate a set of features based on [1]. To make sure that our analysis is consistent regardless of the device being used, we normalise all of the positional features (such as the X/Y coordinates of the touch) to the screen resolution which allows us to compare gestures across different devices and screen sizes in a meaningful way. Additionally, we distinguish between the direction (left-to-right/right-to-left or up/down) of both swipes and scrolls.

**Training data selection.** In order to control for the effect of P3, we consider four methods of dividing the target user's data into training and testing sets.

- RANDOM: we choose training samples for a user out of all the available samples at random, i.e., all sessions are merged, and testing uses the remaining samples. This process is repeated independently for each user.

- CONTIGUOUS: we combine all samples of a user, and we select the first portion (in chronological order) of samples for training. The remainder of the strokes from the user are used for testing.

- DEDICATEDSESSIONS: we select a subset of the user sessions for training and test on the remaining sessions. This ensures that each session is used for either training or testing and that training and testing samples are never drawn from the same session. We investigate selecting sessions both contiguously (in chronological order, with the first sessions used for training and later sessions used for testing) and randomly.

- INTRASESSION: we select a specific user session and use the first half of samples for training and the remainder for testing. Only samples from the chosen session are used for the positive data.

**Attacker modeling.** To evaluate the impact of P4, we compare two different scenarios. In the first scenario, we include samples from the attacker in the training data. In the second scenario, we exclude these samples from the training data.

For both of these scenarios, we train a binary classification model using the user's samples as the positive class and combined samples from multiple other users as the negative class. In the following, $U$ identifies the set comprising of all users, $N_i$ identifies the number of samples (strokes) belonging to user $i$, and $f_{train}$ and $f_{test}$ refer to the fraction of samples used for training and testing, respectively.

- EXCLUDEATK: For each user, we randomly divide the remaining users into two equally-sized sets $U_1$ and $U_2$. For training, we select positive class data from the available data from the user and negative class data from $U_1$. We ensure the two classes are balanced. For testing, we treat all users from $U_2$ as attackers and classify their samples along with the user's testing samples. This ensures that there is no overlap in the attackers used for training and testing. We use this approach over the leave-one-out method proposed in [74] to avoid overfitting when a separate threshold is chosen for each user-attacker pair. In the commonly used leave-one-out approach, a specific model is trained, and a threshold is chosen when calculating the EER for each individual attacker. However, in a practical scenario, it is not possible to choose that threshold for an arbitrary attacker and the threshold selection has to be done on a group of users before the model is deployed. Hence, modelling using the leave-one-out approach may lead to an overestimation of results and should be avoided.

- INCLUDEATK: We select a user and split the remaining user groups into $U_1$ and $U_2$. We first train and test the system using the chosen user and $U_1$. This involves training a model for each user $i$ where $N_i * f_{train}$ of the user's samples and $\frac{N_i * f_{train}}{|U_1|}$ of each attacker's samples are used for training and the rest for testing. This ensures that the negative and positive classes are balanced in the training data. The process is then separately repeated with $U_2$. While we could simply include all $|U| - 1$ users to form the negative class samples, we choose to repeat the $U_1, U_2$ split as it allows us to maintain comparability with the EXCLUDEATK approach. In fact, this way, the number of attackers is $(|U| - 1)/2$ in both methods.

**Scaling.** After separating the data into two sets, one for training and one for testing, and including or excluding the attacker data, we normalise each individual feature by computing the mean and standard deviation of the training data. The training and testing samples of both the user and the attackers are scaled by subtracting the mean and dividing by the standard deviation of this training data. This is done in order to scale the data and ensure that each feature is weighted equally during the training process.

**Classification.** After scaling the data, we use the training set to fit a classifier. This classifier is then applied to the samples in the testing set, which provides us with a probability for each sample belonging to a particular class. These probabilities are then used both for sample aggregation and threshold selection.

**Sample aggregation.** Aggregating multiple samples is an optional step in the touch-based authentication pipeline. However, it is still very common in the related work. Here, instead of treating samples independently, we group a set of consecutive samples together. Then we take their mean probability estimation and use that for threshold selection and final decision instead of individual probability estimation.

**Threshold selection.** Threshold selection involves choosing a probability value above which a sample is considered to belong to a particular class and below which it is considered not to belong to that class. This threshold value is chosen based on the desired level of accuracy and precision for the authentication system.

In this case, we take the probability estimation for the testing samples (both positive and negative samples) and compute the EER for each user. As described in Chapter 2, this is done by finding the threshold where the FAR and FRR are equal. The mean EER for a given configuration is the average EER across all users.

## 3.6 Analysis of Pitfalls

In order to accurately assess the impact of each pitfall on the overall evaluation performance, we carefully analyse the effect of each pitfall on an individual basis. Our system implementation is based on the findings and recommendations outlined

in one of the seminal papers in the field [1]. We report our results using the SVM classifier as it is the most widely used in the related work. However, we also conducted experiments using other popular classifiers, including Random Forest, Neural Network, and k-Nearest Neighbours, in order to gain a more comprehensive understanding of the relative strengths and weaknesses of each method. We will discuss the differences in performance between the various classifiers in more detail at the end of this section.

When investigating one pitfall, we control the remaining experimental choices estimating a baseline performance as follows: (i) CONTIGUOUS, (ii) EXCLUDEATK and no sample aggregation. We chose this specific configuration as a default in our experiments for the following reasons. For the training data selection, we chose the most common configurations in Table 3.1 - CONTIGUOUS. However, we chose to EXCLUDEATK as previous work on the topic has already suggested the negative effects of using the unrealistic INCLUDATK approach [74]. We do not use an aggregation of samples in our default configuration as it adds another dimension to the data and results, thus making comparison within experiments and previous work more complicated. Unless differently specified, we focus on the effect of pitfalls on the *mean EER*, i.e., for an experiment configuration, we train the system, then use the test set to estimate each user's EER (*per-user EER*) and report the average of those. We also report the mean ROC curve with 95% confidence intervals where appropriate.

In order to achieve our goal of thoroughly examining the fundamental effects of each evaluation pitfall, we have chosen to use the larger remotely collected dataset. Additionally, in order to minimise sources of variability, we have chosen to focus specifically on the most prevalent left stroke type. By limiting the scope of our experiment in this way, we can more accurately isolate and analyse the effects of each individual pitfall on the overall evaluation process.

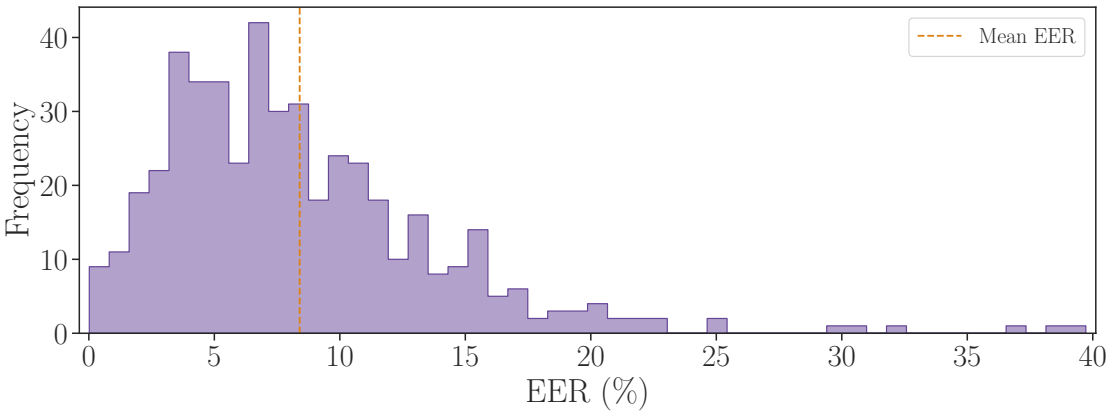



**Figure 3.8:** Per-user EER distribution using all users in our dataset ($n$=470). The performance results in a positively skewed distribution.

**Table 3.5:** Model performance for varying stroke directions.

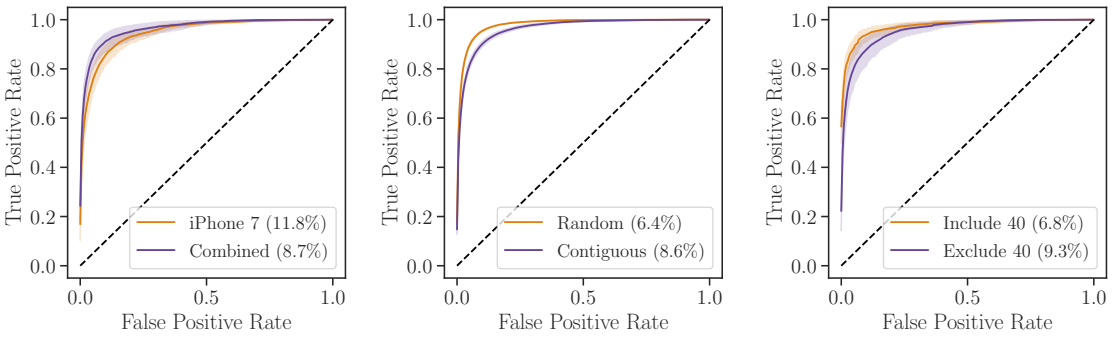
| Direction | Stroke Count | Mean EER (%) | Std. Dev. |
|---|---|---|---|
| Scroll Up | 376,236 | 10.1 | 7.2 |
| Scroll Down | 45,737 | 19.0 | 11.9 |
| Swipe Left | 718,036 | 8.4 | 5.6 |
| Swipe Right | 26,083 | 16.2 | 10.5 |

## General results

In this section, we describe a few preliminary results about our system before we investigate the effects of specific pitfalls.

We repeated our machine learning procedure until we measured the per-user EER for each of our participants ($n$=470). We report the per-user EER distribution in Figure 3.8. Naturally, some users perform quite poorly compared to the mean EER creating a positively skewed distribution. This information is rarely reported by other studies, although it is important to investigate the tail and understand why some users perform significantly worse than others. Furthermore, excluding the "outliers" from a performance evaluation can lead to a drastic artificial increase in performance and should be avoided.

We repeat our experiment for each swipe direction (Up, Down, Left, Right) and




**(a)** Phone model mixing **(b)** Data selection methods **(c)** Attacker data in training

**Figure 3.9:** ROC Curves for different evaluation pitfalls. All other parameters are fixed. EER (%) values reported in the legend.

report the result in Table 3.5 together with the amount of data available for each swipe direction. As shown in Table 3.5, down and right swipes are underrepresented as these interactions are performed rarely in our application, leading to much higher mean EERs of up to 19.0% and 16.2%, respectively. To reiterate, as our goal is to investigate the fundamental effects of evaluation pitfalls, we focus on the most populous left swipe type to limit sources of variability in the rest of this chapter.

The baseline system resulted in a mean EER of 8.4% and a standard deviation of $\pm 5.57$.

## P1: Small sample size

Here we investigate the non-trivial effects of user sample size and the effect of the amount of available data per user on the resulting mean EER.

### User sample size

Oftentimes, in related work, it is assumed that the EER of a given authentication method can be reliably estimated by sampling roughly 40 users (the median number of users in Table 3.1). To investigate the effects of this choice, we randomly sample $n < 470$ users from our dataset and compute the mean EER of the system fit on those $n$ and the standard deviation of each sample's per-user EER distribution. We focus on the standard deviation of the per-user EER distribution as it is a proxy

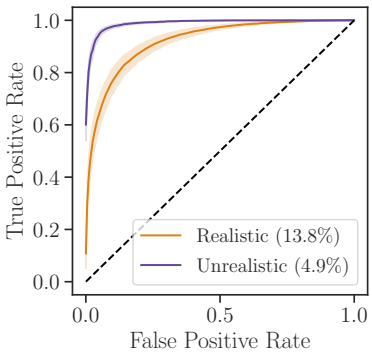



**Figure 3.10:** ROC Curves for the cumulative effect of all pitfalls (unrealistic) and fair evaluation (realistic). EER (%) values reported in the legend.

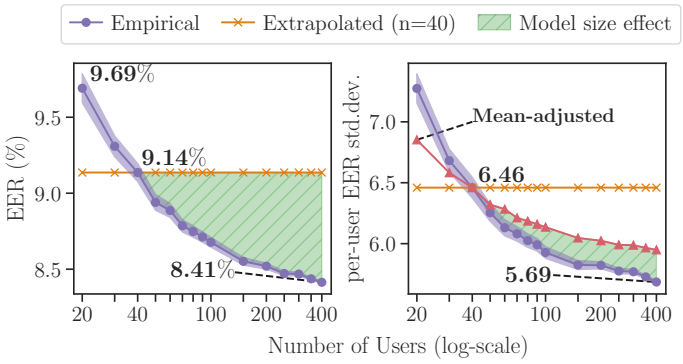



**Figure 3.11:** Differences between extrapolated EER using sample size of $n$=40 and empirical EER measured with various $n$. Left reports the changes in mean EER, right reports the standard deviation of the per-user EER distribution. Empirical data are computed using 1,000 random $n$-sized subsamples of our dataset, Extrapolated data are computed generalizing the findings for $n$=40.

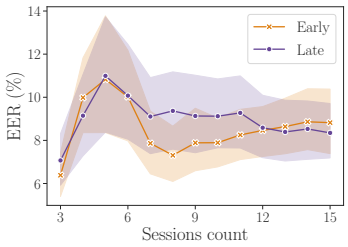



**Figure 3.12:** EERs of Early and Late session subsets for users with 31 completed sessions ($n$=68). No significant difference is found between the subsets, suggesting user task familiarisation does not affect behaviour.

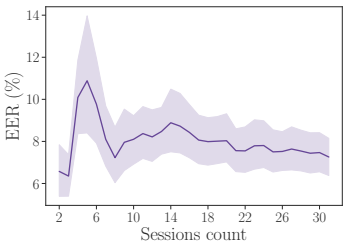



**Figure 3.13:** EERs when considering an increasing number of sessions for users with 31 completed sessions ($n$=68). The shaded areas report 95% confidence intervals.

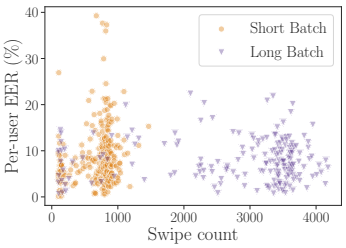



**Figure 3.14:** Relationship between per-user EER and number of swipes available for each user. Short and Long batch labels mark the studies users belong to (see Section 3.4).

to the evaluation of systematic errors and EER outliers: certain users with high per-user EER are responsible for a larger proportion of the resulting mean EER [74]. The sampling procedure is repeated 1,000 times for each $n$. We then use $n$=40 (median user sample size in Table 3.1) as a reference: we test whether the metrics obtained at $n$=40 reliably predict the behaviour for different $n$.

**Effect on mean EER.** The left-hand of Figure 3.11 reports the difference in

behaviour between the EER measured empirically for various $n$ and the EER extrapolated from the performance of the $n$=40 subset. The figure shows that increasing the number of users in the model has a non-negligible effect on the EER: while we obtain EER=9.14% for $n$=40, increasing the number of users has a large benefit, reaching EER=8.41% for $n$=400.

**Effect on per-user EER standard deviation.** The right-hand side of Figure 3.11 reports the difference in behaviour between the empirical per-user EER standard deviation for various $n$ and the standard deviation extrapolated from the performance of the $n$=40 subset. Given the effect described in the previous paragraph, to allow for meaningful comparison, we adjust the extrapolated standard deviation to account for the reduction in mean EER (which reduces the per-user EER standard deviation). We do so by adjusting the standard deviation extrapolated at each $n$ with the scaling ration between the empirical mean EER measured at $n$ and the one measured at 40;[2] this moves the two distributions to the same mean EER. Figure 3.11 (right) shows how for increasing $n$, there is a notable decrement in the per-user EER standard deviation, which is not solely explained by EER mean reduction presented above.

Overall, we find that increasing the user sample size greatly benefits the machine learning model (at least in our general method and SVM), thanks to the added variety of negative samples coming from larger pools of users. Larger sample sizes not only lead to lower and more accurate measurement of underlying EER but also have a regularising effect on the resulting per-user EER distribution, leading to fewer outliers. This also challenges previous findings regarding the usage of error distribution metrics [74] as user sample sizes also will have an effect on such EER distribution across users.

### Number of sessions and strokes

Increasing the amount of data collected per user may lead to differences in performance: (i) across several data collection sessions, users may get acclimatised to the task (leading to better stability of the collected strokes) and (ii) larger amount

[2]Given empirical per-user EER standard deviation and EER mean measured at $n$, $\sigma_n$ and $\mu_n$, we estimate $\hat{\sigma}_m$ using $n$=40 as $\hat{\sigma}_m = \frac{\mu_m}{\mu_{40}}\sigma_{40}$.

of data per user may generally benefit the performance of the machine learning model. In the following paragraphs, we test both factors separately. It is worth noting that the studies presented in Table 3.1 perform their data collection efforts over a small number of sessions with a median of only two sessions.

**Effect of user acclimatisation.** We use data from the 68 users who completed the full 31 sessions. Given a number of sessions $S$, we split the data into the earliest collected $s$ sessions (*Early*) and the latest collected $s$ sessions (*Late*). We postulated that if users gradually get used to the experimental settings (i.e., their behaviour exhibits reduced variation), then *Early* sessions will perform worse than *Late* sessions when the user has acclimatised after many repetitions. We apply our authentication pipeline on both early and late sets, doing several splits with $s$ ranging from 3 to 15. The results are summarised in Figure 3.12. Our findings suggest that there is no significant difference between the performance of *Early* and *Late* sessions. Therefore the data shows no evidence of task acclimatisation leading to changes in performance.

**Effect of amount of data per user.** We again use data from the 68 users who completed the full 31 sessions. Here, we consider the effect of increasing the amount of data per user and evaluate the system performance as the number of sessions grows. The results of this analysis are depicted in Figure 3.13, which shows the EER for various numbers of sessions. We found that there is no discernible pattern or trend that emerges as we vary the number of sessions.

In order to get a more comprehensive understanding of the data, we decided to extend our examination to include all users by considering the number of strokes per user rather than the number of sessions. We plotted this relationship and the resulting per-user EER in Figure 3.14. In this figure, the points are labelled as either *Short* or *Long* depending on whether the user was part of the short or long study batch, as described in Section 3.4. Upon further examination, it becomes apparent that there are three distinct groups that emerge. These groups include users who completed only a single session, users who were included in the short-term 7-day group, and users who were included in the long-term collection for 31 days.

It is interesting to note that high and low performance points can be observed in all three groups, and there appears to be a significant amount of variation in the short-term group, likely due to the fact that this group includes a larger number of users. It is worth noting that the number of strokes does not seem to have a consistent impact on performance, as points with both high and low performance can be found at various levels of swipe count. The figure suggests that the relationship between the number of strokes and performance is not straightforward and there does not appear to be a clear trend that emerges. We found that there is not a clear distinction or trend based on the number of strokes, reinforcing the previous results of Figure 3.13. Overall, both figures indeed suggest that the number of strokes is not a strong predictor of performance. While long-term studies are necessary to investigate the stability of the biometric, our results indicate that the availability of long-term data does not affect EER in a significant way.

## P2: Phone model mixing

In this section, we compare the system performance when working with data that belongs to individual phone models versus data that has been merged together from various phone models (referred to as COMBINED). In addition, we will be examining the accuracy with which we are able to predict the specific phone model that a particular set of strokes originated from.

**Effect of combining phone models.** As evidenced in Section 3.6, increasing $n$ leads to a reduction in EER (see Figure 3.11). To account for this, we compare each single-phone subset to a COMBINED subsample derived from all phone models. We ensure that the two subsets have an equal number of users.

In Table 3.6, we present the results of our comparison between the performance of the COMBINED dataset and the performance of individual phone models. The results indicate that the COMBINED approach leads to an overestimation of the model's overall performance. Specifically, we observed that the EER for each of the phone models was lower when using a single phone model rather than the combined dataset. In addition, we also conducted a $t$-test to formally evaluate the statistical

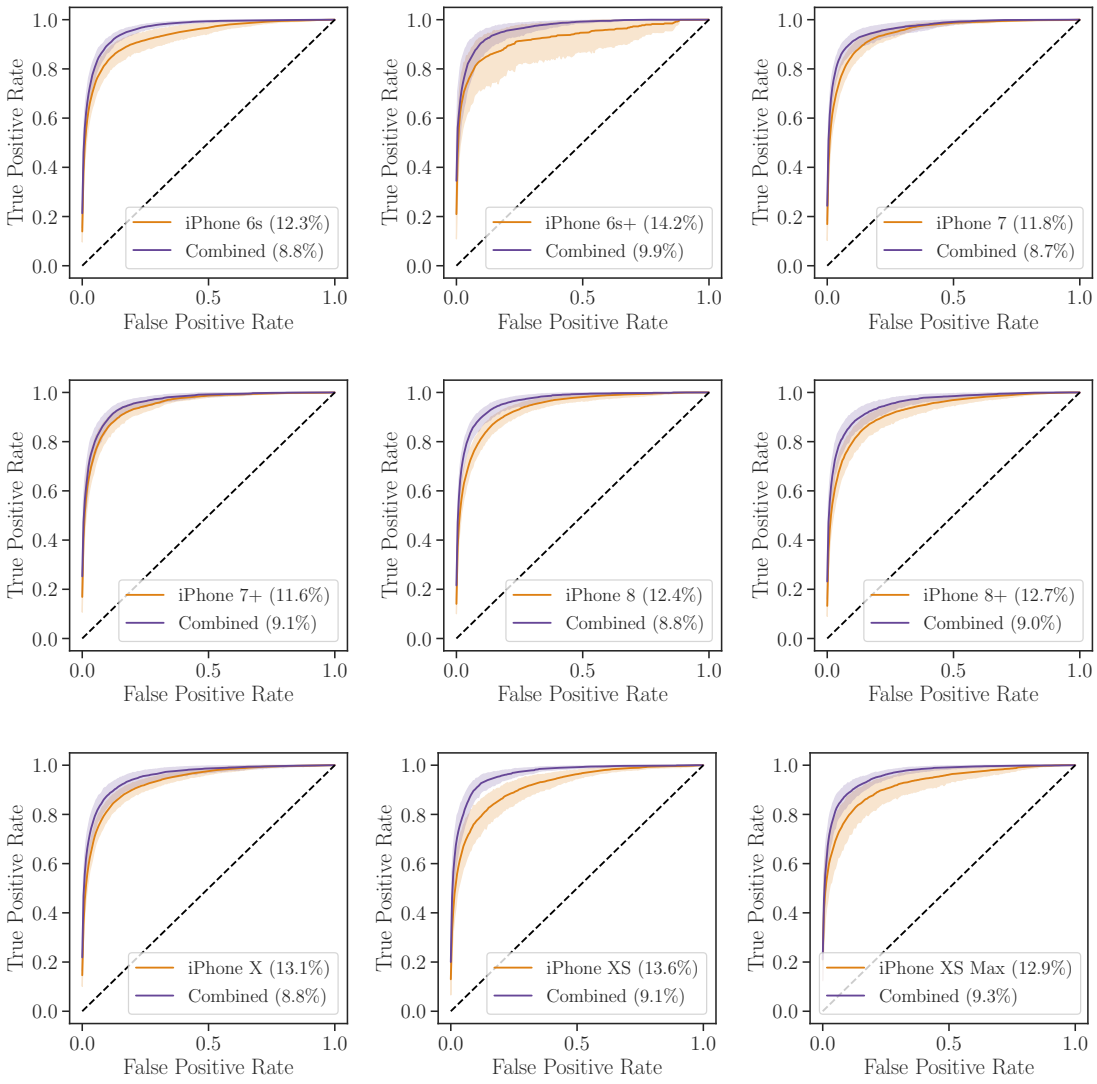



**Figure 3.15:** ROC Curves for individual phone models compared to COMBINED models which use the same number of users but merge multiple phone models.

significance of the differences in EER between the individual phone models and the COMBINED dataset. The results of this analysis showed that the difference in EER between a single phone model and the COMBINED dataset was statistically significant ($P < .05$) for all phone models except for 6S PLUS, 7 PLUS, and XS MAX. This further supports the conclusion that the use of the COMBINED dataset leads to an overestimation of the model's performance.

As shown in Figure 3.9a, the complete ROC curves for the iPhone 7 (which includes the most number of users in our dataset) model and its respective COMBINED

**Table 3.6:** Model performance when training and testing with the same phone model or when mixing phone models (COMBINED). COMBINED results in overestimation of performance even when subsampling to the number of users present in each specific phone model.

| Model | Users ($n$) | Mean EER (CI 95%) | COMBINED EER (CI 95%) | $p$-value |
|---|---|---|---|---|
| iPHONE 6S | 70 | 12.3% (±2.46) | 8.8% (±2.04) | **.032** |
| iPHONE 6S PLUS | 19 | 14.2% (±6.28) | 9.9% (±4.00) | .233 |
| iPHONE 7 | 73 | 11.8% (±1.60) | 8.7% (±1.17) | **.002** |
| iPHONE 7 PLUS | 50 | 11.6% (±2.19) | 9.1% (±1.81) | .082 |
| iPHONE 8 | 68 | 12.4% (±1.84) | 8.8% (±1.14) | **.001** |
| iPHONE 8 PLUS | 55 | 12.7% (±2.32) | 9.0% (±1.94) | **.014** |
| iPHONE X | 71 | 13.1% (±2.03) | 8.8% (±1.68) | **.002** |
| iPHONE XS | 34 | 13.6% (±3.01) | 9.1% (±2.01) | **.014** |
| iPHONE XS MAX | 30 | 12.9% (±4.01) | 9.3% (±2.66) | .135 |

model reveal that the overestimation of performance is present throughout the entire range of values, with the exception of extreme TPR and FPR. This pattern is also evident in the ROC curves for the other phone models, as shown in Figure 3.15.

**Phone model identifiability.** We create a phone model classifier whose aim is to identify the iPhone model from a group of given strokes. We merge all the available data and label each stroke with its originating phone model. Then it is divided into 80/20 train-test splits. The data is balanced such that each phone model had an equal number of strokes in the training split.

Our goal was to develop a classifier that is able to identify the specific model of an iPhone based on a set of stroke data. To do this, we first gather all of the available data and label each stroke with the corresponding iPhone model that it originated from. We then split the data into a training set and a testing set, with the training set comprising 80% of the data and the testing set comprising the remaining 20%. We ensured that the training data was balanced so that each iPhone model was represented equally. Furthermore, we made sure that users who were used in training were not considered in testing and vice versa (to avoid biasing the prediction with

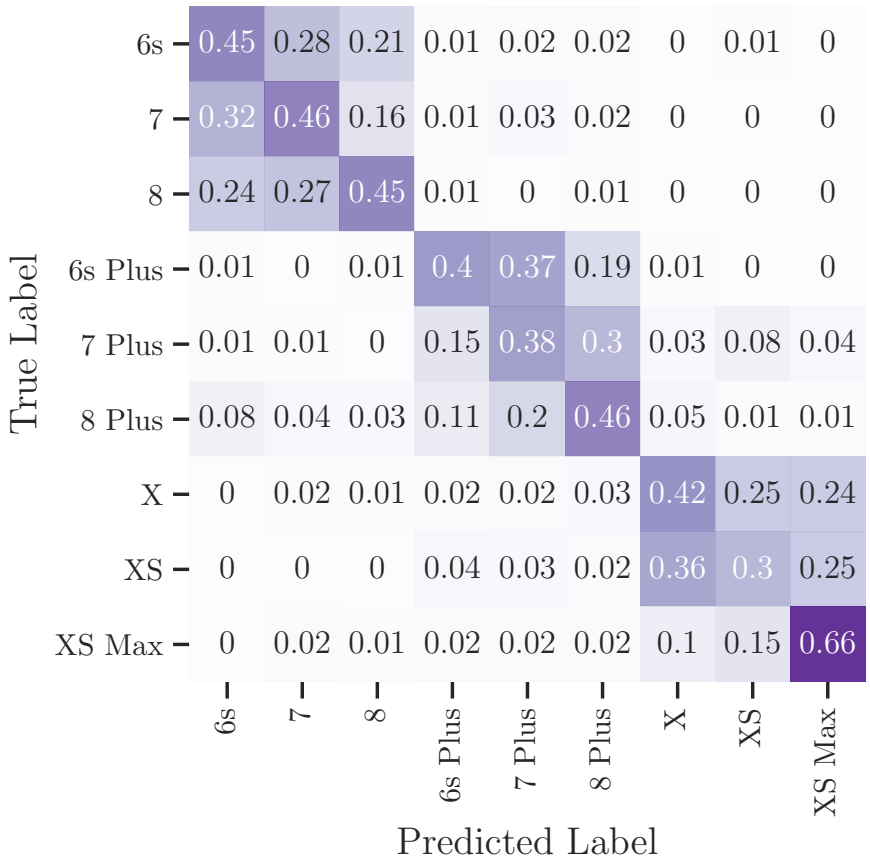



**Figure 3.16:** Confusion matrix of phone model prediction for the nine iPhone models in our remote data collection. The model prediction errors are concentrated in phones with similar dimensions and resolutions.

the users' identities). These steps help prevent any biases from being introduced during the training process and improve the overall reliability of the results.

We use an SVM classifier and conduct this experiment twice, with one iteration including data from all nine phone models and the second iteration only including data from the iPhone 6s, 7 and 8 models. The second group was selected because they have similar screen sizes, resolutions, and pixel densities.

The classifier achieved an overall accuracy of 44%. This is significantly better than a random baseline model, which would only achieve an accuracy of 11.1% (1/9). When we repeat the experiment using only data from the iPhone 6s, 7 and 8 models, we find that the accuracy of the classifier is 49%. This is still higher than the baseline accuracy of 33.3% that we would expect from a random model. However, the gap is lower due to the similarity of the device models at hand.

Figure 3.16 shows the confusion matrix of phone model predictions. The confusion matrix shows the number of correct and incorrect predictions for each class compared to the actual outcomes. The rows of the matrix correspond to the predicted classes, while the columns correspond to the actual classes. Each cell then gives the % of predictions which were classified as the predicted label but were actually part of the true label. The rows of the matrix, therefore, add up to 1, and diagonal elements (TP and TN) represent the number of correct predictions, while the off-diagonal elements (FP and FN) represent the number of incorrect predictions.

Our analysis shows that differences in the properties of the devices are reflected in the identification outcome, i.e., strokes belonging to similar phone models tend to be more similar. There are fundamental differences in the data which the machine learning models can detect, thus improving the accuracy. When classifying the test strokes, the model can easily label data as negative if it knows with a high probability that it is originating from a different phone model. This further suggests that mixing phones in touch-based authentication evaluation can lead to an overestimation of performance.

**Performance of identical user group on different devices.** We also compared how the same group of users doing the experiment would perform on two different devices. In this case, we use the in-person Android dataset and compare the performance of the OnePlus 5 and BLU Vivo 6 devices which consist of the same group of users. All other parameters are left as default. The experiments were repeated 100 times, and the average EER was reported. The results of the experiment showed that the performance on the OnePlus 5 device resulted in a 14.4% EER, while the BLU Vivo 6 device resulted in a 17.3% EER. Both experiments had a standard deviation of 0.5%. The difference between the two devices was found to be statistically significant, with a $p$-value of less than 0.05.

Although the OnePlus 5 and BLU Vivo 6 devices may seem similar based on their specifications, the performance of the two devices differed significantly. While it is possible that factors such as the way users performed each session could contribute to this difference, we believe that the majority of the difference is likely

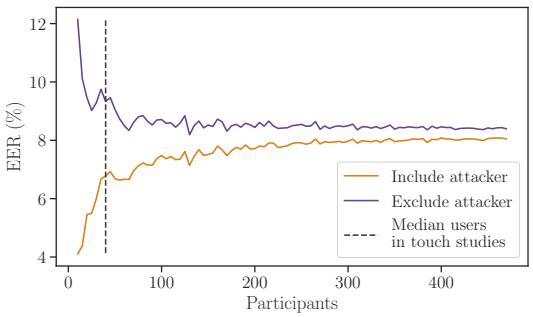



**Figure 3.17:** Resulting mean EER when using INCLUDEATK and EXCLUDEATK attacker modelling approaches. We report the mean of the EER across 10 random subsampling repetitions. A large EER difference is observed when considering a small number of users.

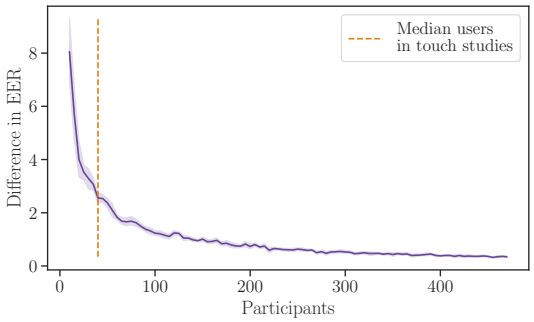



**Figure 3.18:** Absolute EER difference between INCLUDEATK and EXCLUDEATK attacker modelling approaches. For each number of users, the shaded areas report 95% confidence intervals on the mean difference from 10 random subsampling repetitions.

due to minor variations in the devices themselves. These could include differences in hardware or software that affect the overall performance.

The findings in this section indicate that it is undesirable to mix different phone models in data collection and analysis for touch-based authentication. Furthermore, it is irrelevant whether the mixed models have similar screen sizes, dimensions, or display pixel densities. The practice of mixing phone models can lead to an artificial increase of performance between 2.5% and 4.5% EER.

## P3: Non-contiguous training data selection

We compared the classification performance of our model under the conditions described in Section 3.5: (i) RANDOM, (ii) CONTIGUOUS, (iii) DEDICATEDSESSIONS and (iv) INTRASESSION. For a fair comparison, we only used data from the 409 users who have completed two or more sessions as this is a prerequisite for the DEDICATEDSESSIONS modality. We present our findings in Table 3.7.

As expected, the INTRASESSION method yielded the best performance (5.6% EER) as users have a more stable interaction pattern during a single session than through time. This is also supported by related work [1]. It is encouraging that the INTRASESSION model performed well in this specific category, but it is important to

**Table 3.7:** Model performance for common training data selection approaches. Random selection results in overestimated performance.

| Data Selection Method | Mean EER (%) | CI (95%) |
|---|---|---|
| RANDOM | 6.4 | ± 0.28 |
| CONTIGUOUS | 8.6 | ± 0.55 |
| DEDICATEDSESSIONS (Contiguous) | 10.1 | ± 0.70 |
| DEDICATEDSESSIONS (Random) | 10.2 | ± 0.68 |
| INTRASESSION | 5.6 | ± 0.25 |

note that when evaluating the effectiveness of touch-based authentication systems in real-world scenarios, it is necessary to consider the performance of the model over multiple sessions. While the model may have demonstrated good results in this particular instance, it is not necessarily an accurate representation of its performance in practice, as users typically engage in multiple sessions over an extended period of time. The INTRASESSION result should not be considered an accurate metric for touch-based authentication systems.

Mixing and randomising samples from all sessions (RANDOM approach) provided a similar effect with an EER of 6.4%. The model learns on information about users' interactions throughout all sessions, hence contradicting real-world implementation of the system as described in Section 3.2.

The CONTIGUOUS training approach is much more realistic and results in a performance of 8.6% EER. However, it also allows the model to learn from an overlapping session which is then used for testing. This ultimately yields better performance.

The DEDICATEDSESSIONS scenario is the most realistic one for a touch authentication system as it relies on self-contained training sessions. This is the most similar approach to how data could be managed in a deployed system.

We found that results between all of the methods vary considerably, and performance seems to be overestimated compared to the realistic DEDICATEDSESSIONS approach. There is a small, statistically insignificant difference between random and

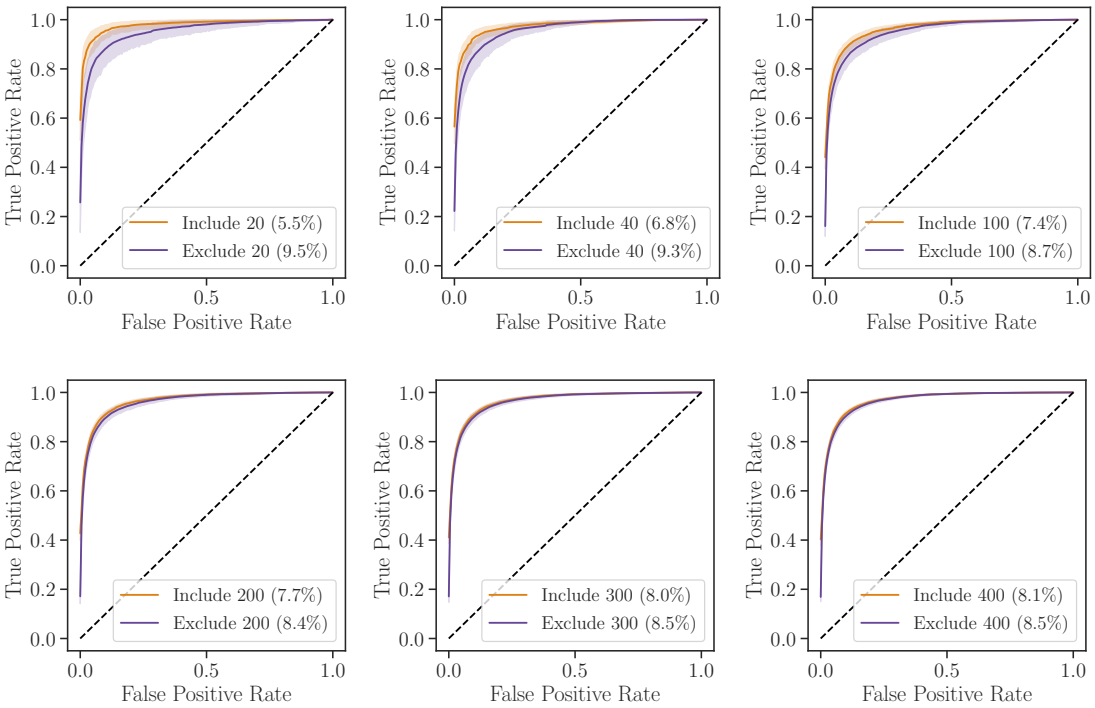



**Figure 3.19:** ROC Curves for including or excluding attacker data into the training set of a model at different sample sizes.

contiguous DEDICATEDSESSIONS selection procedures. In this section, therefore, we found that an unrealistic training data selection can lead to an increase in performance of 3.8% EER when using a RANDOM approach compared to the DEDICATEDSESSIONS approach. The complete ROC curve showing the difference between RANDOM and CONTIGUOUS performance is shown in Figure 3.9c. The ROC curve results are mostly consistent with the EER reported in Table 3.7.

## P4: Attacker data in training

We compared different attack modelling choices as described in Section 3.5: (i) EXCLUDEATK and (ii) INCLUDEATK. To do so, we randomly subsampled $n$ users from our dataset. We did that for a various number of users $n$, and for each one, we applied our pipeline and computed the resulting EER for the two approaches. This procedure was repeated 10 times. Figure 3.17 and Figure 3.18 illustrate the results.

Our experiments suggest that INCLUDEATK results in consistently lower mean EER when compared to EXCLUDEATK. This is illustrated in Figure 3.17. However,

Figure 3.18 shows how the EER difference between the two approaches decreases exponentially as the number of users ($n$) increases. This is expected because as the number of users decreases, the influence of attacker data on the classifier increases. For example, if there are only 11 users in the training set, attacker data constitutes 10% of the negative training data. However, if there are more than 101 users, attacker data makes up less than 1% of the negative training data. Furthermore, one might expect that a larger amount of data would lead to improved performance, but this is not the case in the IncludeAtk scenario. In this instance, adding more users to the model actually hinders its performance, likely due to the decreasing influence of the inclusion of the attacker data as the number of users grows.

Figure 3.9c shows the ROC curves of IncludeAtk and ExcludeAtk models for 40 users (the median number of users from Table 3.1). The ROC curves for some of the other $n$ we considered (20, 100, 200, 300 and 400) are shown in Figure 3.19. We found that our results are largely consistent throughout the length of the ROC curve.

As pointed out in Table 3.1, 80% of our reported studies falls into P4, meaning that these might not present performance metrics appropriate for the specified threat model. Figure 3.18 shows that when $n$=40, the EER difference between the two approaches is 2.55%. However, overall, depending on the user sample size considered, IncludeAtk can lead to an artificial performance gain of between 0.3% and 6.9%.

## P5: Aggregation window size

As described in Section 3.5, when reporting their results, many studies [1, 2, 62, 69, 123] consider the performance of a group of consecutive strokes instead of a single one as we have done so far in our analysis. Figure 3.20 shows the performance of our pipeline when we use an aggregation of consecutive strokes. We varied the number of strokes (aggregation window) between 1 and 20. The procedure was repeated 10 times, and shaded areas show a 95% confidence interval across the ten repetitions. As expected, increasing the aggregation window size leads to lower EERs: an EER of 8.2% obtained on single strokes drops more than a quarter (5.9%) when aggregating two strokes and drops to less than 3% at 12 strokes.

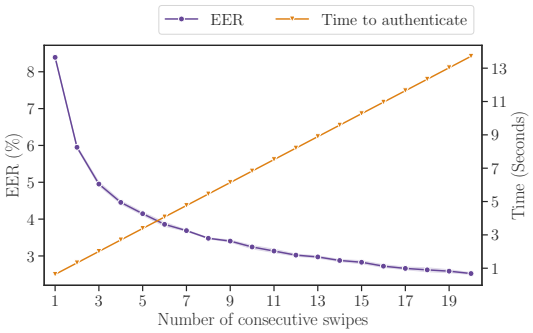



**Figure 3.20:** Performance of an aggregation model which selects the mean distance scores of a number of consecutive strokes before calculating EER. The shaded areas report 95% confidence intervals on the mean EER from 10 repetitions.

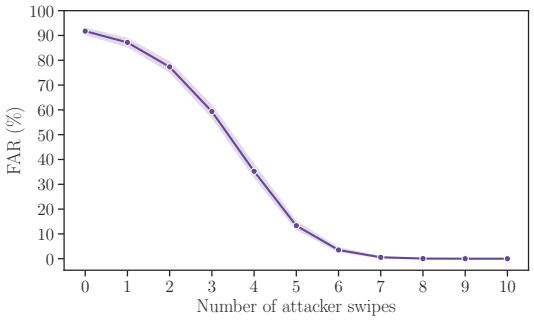



**Figure 3.21:** False Acceptance Rate of an aggregation model when the number of attacker swipes is varied. The aggregation window is 10 consecutive strokes. The shaded areas report 95% confidence intervals on the mean EER from 100 repetitions

It is important for research on touch-based authentication methods to be explicit about when and how they aggregate data, as this can significantly affect the system's performance. Additionally, it is worth considering that each touch-based interaction requires a certain amount of time to complete, which can potentially leave the system vulnerable to attacks. For instance, our dataset suggests that on the tasks considered, performing 20 strokes would take 14 seconds, during which the system would remain vulnerable. Therefore, it is crucial to find a balance between usability and security when implementing touch-based authentication methods.

## Cumulative effects of evaluation choices

In this section, we aim to measure the distinction between the results obtained from evaluating touch-based authentication systems in a way that realistically reflects actual usage scenarios, where the system is not subjected to any pitfalls (*realistic*), and the results obtained from evaluating the system in an unrealistic manner, where it is exposed to all potential pitfalls (*unrealistic*).

We repeated the following two procedures 100 times and reported the mean of all runs and the confidence interval at 95%. In the unrealistic methods experiment, we combined phone models (COMBINED), included the attacker into the training

data (INCLUDEATK), used the RANDOM data selection method, and each round randomly subsampled our dataset to the median of $n$=40 participants taken from Table 3.1 (to even out the effect of P1). This resulted in an EER of 4.9% and a confidence interval of ±0.09.

In the realistic method experiment, again we selected $n$=40 users from the most commonly used iPhone 7 phone model, and used EXCLUDEATK and the DEDICATEDSESSIONS training data selection. In each round, we randomly select which users are chosen as the attacker group. This approach resulted in a much worse EER of 13.8% with a confidence interval of ±0.14. Figure 3.10 illustrates the overestimation of performance throughout the ROC curves of these experiments.

The results of our analysis demonstrate that the use of flawed or inadequate evaluation methods can significantly influence the performance of touch-based authentication systems. In particular, we found that using flawed methods can artificially inflate the performance of the system by as much as 8.9% EER.

### Effects of classifiers on evaluation choices

In this subsection, we quantify the impact of pitfalls on performance on four (SVM, RF, kNN, NN) of the most widely used machine learning algorithms in the field.

We use the SVM, Random Forest, and kNN classifier implementation of the widely used machine learning library `scikit-learn`. The former two classifiers use the default parameters of the framework, and we chose n=18 for the kNN classifier based on preliminary experimentation, where it resulted in the best performance. Our Neural Network implementation uses the machine learning libraries Tensorflow and Keras. The feed-forward network consisted of 3 hidden layers of sizes 30, 30, and 15 with batch normalisation and a dropout layer (0.3) between them. The optimiser was Adam and the activation function was ReLU. Similarly, we chose the set parameters based on non-exhaustive preliminary experimentation.

The results of our experiments are presented in Table 3.8. All of the examined pitfalls introduce an overestimation of performance regardless of the classifier chosen. However, there are differences in individual performance across chosen classifiers.

**Table 3.8:** Impact of pitfalls on different classifiers. The table presents the percentage-point difference in EER between using realistic and unrealistic evaluation methods.

| Pitfall | SVM | Random Forest | Neural Network | k-NN |
|---|---|---|---|---|
| P1 400 users vs 40 users | 0.72 | 0.28 | 0.87 | 1.25 |
| P2 iPhone 7 vs Combined | 4.08 | 4.53 | 2.40 | 3.29 |
| P3 Contiguous vs Randomized | 2.27 | 2.62 | 2.06 | 2.35 |
| P4 Exclude vs Include | 2.55 | 2.69 | 3.41 | 3.96 |
| Cumulative Impact | 8.89 | 10.36 | 8.99 | 9.79 |

For instance, the kNN classifier relies heavily on individual strokes similar to the target one. Hence the impact of including the attacker data in training is much more pronounced. These results suggest that the pitfalls apply to a wide range of touch dynamics system implementations.

## Additional considerations

In this section, we will delve into a few additional considerations that should be taken into account when designing and evaluating touch-based authentication systems. To illustrate the impact of these decisions, we will present a series of experiments which quantify the effects of various design choices on the effectiveness of the authentication system.

Thus far in this chapter, and frequently in previous research on the subject, the threshold for acceptance into the touch-based authentication system is typically chosen based on the EER which is determined from the testing dataset. This approach to threshold selection assumes that we have ground truth knowledge of the testing data. However, in reality, when the touch-based authentication model is deployed in a live environment, we do not have access to this ground truth data. We are unable to know with certainty whether a given touch sample is genuine or an impostor, as we do not have access to the identity of the user who provided the touch. One way to more accurately evaluate the performance of the touch-based authentication model is to use the training data to select the threshold and then

apply this threshold to the testing data. We investigated the extent to which the difference between the two approaches would be noticeable in our results. We conducted experiments using the default configuration and found that when we selected the threshold using the testing data, we were able to achieve an EER of 8.7% with a standard deviation of 0.04. On the other hand, if we instead chose the threshold from the training data, we observed a performance of 10.2% with a standard deviation of 0.06. That could lead to an overestimation of performance by 1.5% in the more unrealistic scenario. Overall, despite the potential negative impact on performance this issue may have, we do not consider it to be a pitfall in the strictest sense. This is because it is possible that there may be other methods of threshold selection that could yield better results. However, it is still important to be aware of this issue and consider it when developing and deploying touch-based authentication systems, as their performance may be lower in real-world conditions.

When considering the effectiveness of different aggregation methods, it is often assumed that all of the strokes within the aggregation window will either be positive or negative. However, this may not always hold true in real-world situations. For example, if a malicious user begins using the device, the distribution of positive and negative strokes may be altered until the attacker has performed a number of strokes equal to the window size. Thus, the system might remain vulnerable for an extended period of time. For this purpose, we conducted a small experiment to show how the performance of an aggregation model varies depending on how many attacker strokes are included. We used a CONTIGUOUS and EXCLUDEATK configuration with a window size of 10. The number of malicious strokes ($n$) included was varied from 0 to 10. We reported on the FAR (i.e. what percentage of the malicious interaction windows are considered benign) at an EER threshold selected from the training data. We chose this metric to illustrate how the system lets a malicious user interact with the device when there are still many benign strokes in the window. The results are shown in Figure 3.21. This shows that the system is severely insecure during the first malicious interactions after normal operation.

Finally, we conducted a comparison to investigate whether there is any significant difference between collecting data remotely versus collecting data in a laboratory setting. In order to carry out this analysis, we implemented the following configuration: CONTIGUOUS data-selection method, single-phone model and no aggregation. We use a single phone model from each dataset to ensure fair comparison as the COMBINED configuration is not possible to implement with the lab-collected data (only a single session per phone has been done). Furthermore, for these experiments, we use the scrolling task, instead of the swiping task. This is because the social media task implementation is nearly identical in both Android and iOS applications, unlike the swiping tasks. The experiment was repeated 100 times on two phone models (iPhone 7 Plus and OnePlus 5), and the mean EER was reported. The experiment with the remote data resulted in a mean EER of 13.9% and a standard deviation of $\pm 0.5\%$. The lab-data experiments resulted in a slightly better performance of 10.5% $\pm 0.5$. The difference between the two models is statistically significant, with $p < 0.05$.

Our analysis has revealed that there are differences between the data collected remotely and the data collected in person. However, it is challenging to determine the exact cause of this difference. It is possible that the differences may be attributed to the lower quality of the data collected remotely, or it could be a result of the devices used for data collection (i.e. Android vs iOS). In order to gain a better understanding of the potential causes for these differences, it would be necessary to conduct a further investigation using the same set of devices for both remote and in-person data collection. This would allow us to more accurately determine whether the observed differences are due to the data collection methods or the devices used.

## 3.7 Best Practices for Evaluating Touch-Based Authentication Systems

In order to facilitate better comparison between future studies and achieve unbiased performance evaluation, we propose a standard set of practices to follow when evaluating touch-based authentication systems derived from our set of common evaluation pitfalls.

**P1: Small sample size.** While it is hard to advocate for a specific minimum number of users to be required by a study, we recommend researchers be aware of the effects of user sample sizes in pipelines similar to the one analysed in this chapter. Based on the findings in Section 3.6, we found that increasing sample size has two important effects: it reduces the resulting mean EER and smooths the variance of the per-user EER distribution. We propose two ways to determine the minimum number of participants needed for a research study. One way is to conduct a traditional statistical power analysis that takes into account the desired level of statistical power, statistical significance, and expected effect size. This method can be used to calculate the appropriate sample size for the study. An alternative approach to address this issue is by examining the empirical findings from our study. On the left-hand size Figure 3.11 we show the extrapolated results about the performance of a 40 participants model compared to the true performance of up to 400 participants. Our analysis indicates that at 400 subjects, the difference in EER is 0.73%. Such errors could be tolerable for some research studies as long as the potential discrepancies are disclosed. Similar statistical analysis can be repeated for any number of users. Regardless of the sample size selection approach, it is advisable that an analysis of the effect of sample size is included in new studies, and that results for a sample size of $n$=40 are also reported (when applicable). This best practice must be accounted for during the study design phase to ensure enough data is initially collected.

**P2: Phone model mixing.** A single phone model should be used to train and test a proposed system. While this might not always be the final use case (e.g., in other scenarios, one might want to test the generalisation performance of a device-specific classifier on a different device), this avoids the bias introduced by data collected on a specific phone model. Isolating data belonging to different phone models when training will produce more accurate performance measurements. Care must be taken in data collection to ensure there are enough samples for each phone model that will be studied.

**P3: Non-contiguous training data selection.** Randomised stroke selection should not be used to separate training and testing data. Test data must always have been collected at a time after the training data was collected, to mimic real-world usage, and to account for behaviour drift. For comparison between works, only an initial training phase (enrollment) should be included, as training updates increase the difficulty of comparing figures. Ideally, at least two sessions should be used to collect training and test data, as the bulk of real-world usage occurs with a time interval between enrollment and authentication.

**P4: Attacker data in training.** Studies should always exclude the attacker from the training set, as one shall never assume they have information about the attacker in a deployed system. In particular, care should be taken so that any attacker of a model is not included as a negative example when training the model. Excluding the attacker is particularly important in studies with a limited number of users, where the effect of such an attacker modelling approach greatly affects the resulting performance.

**P5: Aggregation window size.** Using aggregation of consecutive strokes is beneficial to performance, particularly when using the mean of the classifier outputs, as shown in Fig 3.20. However, researchers should report the performance of a single-stroke model in order to ensure comparability with other studies, as well as any reasonable numbers of strokes that similar papers have proposed. Furthermore, information about the flight time between strokes and their duration should also be shared, as these directly relate to the time the system is vulnerable to an attacker.

**P6: Dataset and code availability.** Historically, in this field, it has been rare for authors to share their data (see Table 3.1) and none of the studies examined in the related work share their analysis code. This leads to uncertainty when reproducing results. In fact, for some studies, it was unclear from the paper alone whether the study made certain choices regarding the experiments (e.g., we could not clearly define whether 30% of studies fell into P3). The code and datasets of touch authentication studies should be made freely available. This ensures

that results can be reproduced by others and reduces barriers to entry for those wishing to build upon existing work.

**Additional considerations.**

We established that there are further potential issues in evaluating touch-based systems that do not necessarily classify as a pitfall. Firstly, threshold selection has to be explained clearly and examined carefully while ideally selected using only training data. This avoids the unrealistic assumption that we have ground truth knowledge of the testing data, which is not the case in practice when deploying a system.

When using aggregation and an attacker starts interacting with the phone, the system is vulnerable for a certain amount of strokes. Hence a balance between lower EER (large window of strokes) and the potential for undetected malicious interactions is needed.

Finally, we found that our remote and in-lab data collection resulted in slightly different results. Therefore we recommend collecting data in a manner which is closest to the way the system will be used in practice. However, we note that this difference might be due to the devices used rather than the medium of collection, and further work is needed.

**Generality of results.**

Although this thesis focuses on touch-based authentication, we believe these best practices apply in similar ways to other types of biometric systems, such as facial recognition and keystroke authentication. In particular, non-contiguous training data selection (P3) and inclusion of attacker data in training (P4) are fundamentally flawed and should be avoided in all biometric system evaluations. However, the effect of mixing similar devices (P2) may vary across different modalities. Similarly, the sample size implications (P1) might differ in other systems from what we found in our experimentation. Nevertheless, these points should be examined with caution by the relevant literature.

Further work is required to examine to what extent these pitfalls are prevalent in the study of other biometric authentication systems.

## 3.8 Conclusion

In this chapter, we explored the impacts of evaluation choices on touch-based authentication methods. We investigated performance differences in approaches related both to data gathering and choices in the way classifiers are trained with a certain data split. For the purpose of this chapter, we collected a large open-source dataset for touch-based mobile authentication consisting of 470 users, which we made publicly available. To address RQ1.1, we investigated the feasibility of collecting data at scale through remote means using MTurk, as detailed in Section 3.4. Our findings revealed a noteworthy difference of 3.4% EER in collecting data remotely compared to in-person collection, which was statistically significant. However, further work is required to confirm the validity of these results by ensuring that the same users and devices are used in both modalities.

Regarding RQ1.2, we identified six common evaluation pitfalls. Four out of them have a direct influence on the EER of the model. We confirmed large variations in performance based on phone model mixing (up to 5.8% EER), training data selection (up to 3.8% EER), user sample size (up to 4% EER), and attacker modelling (up to 6.9% EER). Finally, combining all evaluation pitfalls results in an overestimation of performance by 8.9% EER. The results are largely similar regardless of the chosen classifier. We also note that, aside from some extreme threshold settings, these effects are observable throughout the ROC curve.

We proposed a set of recommended guidelines based on our findings to ensure precise reporting of results and comparability across studies. These guidelines are relevant to addressing RQ1.3, which aims to identify the optimal practices for evaluating the performance of touch-based authentication systems. By adhering to these guidelines, researchers can obtain reliable and consistent results, facilitating a meaningful comparison of the performance of various authentication systems across studies.

*We're either getting better or we're getting worse.*

— David Goggins

# 4

# Techniques for Touch-Based Authentication Modeling

## Contents


**4.1 Introduction . . . . . 90**
**4.2 Techniques for Continuous Touch-Based Authentication 92**
4.2.1 Methods . . . . . 92
4.2.2 Findings . . . . . 96
4.2.3 Datasets . . . . . 100
**4.3 Performance Evaluation . . . . . 101**
4.3.1 Comparison . . . . . 104
4.3.2 Results . . . . . 107
**4.4 Discussion . . . . . 110**
4.4.1 Limitations . . . . . 112
**4.5 Conclusion . . . . . 113**


In this chapter, we continue the advancement for the development of touch-based authentication systems for research and industry. We perform a systematic literature analysis of 30 studies on the techniques used for feature extraction, classification, and aggregation in continuous touch-based authentication systems, as well as the performance metrics reported by each study. Based on our findings, we formulate a series of experiments in order to comprehensively evaluate the relative efficacy of the most commonly utilised methods within this field. We take care to ensure that all conditions are precisely defined and controlled in order to eliminate any extraneous

variables that might impact the comparisons. In addition, we introduce two new techniques for continuous touch-based authentication: an expanded feature set (consisting of 149 unique features) and a multi-algorithm ensemble-based classifier. The comparison includes 13 feature sets, 11 classifiers, and 5 aggregation methods. In total, 204 model configurations are examined, and we show that our novel techniques outperform the current state-of-the-art in each category. The results are also validated across three different publicly available datasets. Finally, we discuss the findings of our investigation with the aim of making the field more understandable and accessible for researchers and practitioners alike.

## 4.1 Introduction

In Chapter 1, we discussed the current state of the field of touch-based authentication methods. Despite nearly a decade of research in the area and a positive sentiment from the general community about its potential, the technology is still not widely adopted or integrated into our daily devices. Despite the potential benefits that touch-based authentication could bring, it seems that there are a number of barriers that have prevented it from being embraced on a broader scale. However, it is clear that touch-based authentication has the potential to bring many benefits, such as more user-friendly experiences, better security and more seamless integration. Hence, it is important to continue researching and developing this technology to overcome the barriers that are preventing it from being more widely adopted.

One possible explanation for this lack of adoption can be attributed to the way studies are evaluated and the resulting overestimation of performance, as we investigated in Chapter 3. Our goal was to give recommendations for better evaluation practices in order to minimise the overestimation of performance and understand the true potential of the systems. However, the field of continuous touch-based authentication has also been rapidly developing over the last decade, creating a fragmented and difficult-to-navigate area for researchers and application developers alike. That has been impacted by the variety of methods investigated and a lack of methods to compare and evaluate models against a well-defined benchmark to

accurately assess what can be considered state-of-the-art. In order to make effective contributions and improve techniques for continuous touch-based authentication, it is imperative to clarify the landscape and provide methods to reason about model performance. In order to effectively address and clarify the research questions that we have established for the current chapter, we make use of a two-fold approach.

First, we undertook a thorough and comprehensive examination of the existing literature in the field of continuous touch-based authentication. The aim of this systematic literature review is to gain a broad and extensive understanding of the various methods that have been proposed and used in the field. Then, we proceeded to extract and analyse the various features, classifiers, aggregations, and metrics that have been used in each of the studies that we reviewed. Additionally, we categorised the feature extraction and aggregation methods that have been used in each study. This is an important step to organise and make sense of the large number of options available to researchers and developers.

In order to establish a thorough understanding of the current state-of-the-art models in our field and to identify areas for improvement, we evaluate a carefully selected range of models along common parameters. The analysis is repeated on three publicly available and widely-used datasets. Through this process, we aim to not only determine the current state-of-the-art models but also to gain insights into where future research efforts should be directed in order to advance the field further. In fact, we use our findings to propose a set of techniques that outperform the current state-of-the-art in some areas and quantify the difference with the current best-performing methods.

Our study sets itself apart from previous research in the field in several notable and distinct ways, making our contributions unique. In Chapter 2, a thorough examination of the evolution of continuous touch-based authentication is provided, including the history of its development, the current challenges that exist within the field, and the performance of various models. However, the related work described in the chapter does not adequately address the quantitative performance differences between feature extraction, classification, and aggregation techniques

when evaluated under controlled, fair conditions. As a result, it is difficult for researchers and practitioners to determine the best practices for this area of study. While some previous studies [69, 72] have compared a limited number of classification methods, our work stands out due to the extensive number of classifiers that we have evaluated and the wide range of feature extraction and aggregation techniques that we have examined. In addition, our research introduces new methods that demonstrate superior performance in the feature extraction and classification phases of the system's lifecycle. This sets the work in this chapter apart from other studies in this area.

To summarise, in this chapter, we present the results from a systematic review of 30 papers in the field of continuous touch-based authentication. This allowed us to extract 149 unique features and categorise the methods used in the field. We then compared 204 model configurations across three datasets, enabling us to identify the top-performing features, classifiers, and aggregation methods. Additionally, we introduced a novel set of accumulated features and an ensemble-based classification model, which showed superior performance compared to the current state-of-the-art methods.

## 4.2 Techniques for Continuous Touch-Based Authentication

In this section, we report upon a systematic literature review of papers proposing systems for continuous touch-based authentication. We quantify the prevalence of techniques for feature extraction, classification, and aggregation in continuous touch-based authentication systems, as well as the methods for measuring model performance. Furthermore, we group the approaches into semantically similar categories in order to consolidate the understanding of the field.

### 4.2.1 Methods

For this section, we first performed a systematic literature review, followed by an analysis of the features and classification and aggregation methods. The objective

of the literature review was to understand the methods used in each of the core components of the continuous touch-based authentication lifecycle so that we can next re-evaluate them in a common benchmark. For our systematic literature review, we relied on PRISMA [149] (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) to guide the search strategy to identify articles that proposed and evaluated continuous touch-based authentication models. PRISMA is a standardised reporting guideline for systematic reviews and meta-analyses. It provides a set of items that should be reported in a systematic review or meta-analysis, with the aim of improving the transparency and completeness of reporting and facilitating the assessment of the review's quality. The PRISMA methodology consists of a number of steps and incorporates a checklist of items that should be reported in the review. These are the research question, inclusion and exclusion criteria, search strategy, study selection process and synthesis of results. By following the PRISMA methodology and reporting these items, researchers can ensure that their systematic review or meta-analysis is transparent, comprehensive, and of high quality.

Our search was limited to English language and peer-reviewed published articles. We exclusively made use of the Google Scholar database and used the following search terms: ((*touch-based* OR *touchscreen*) AND (*authentication* OR *biometric**)) OR *touch dynamics* OR *touch biometrics* OR *touch authentication* OR *continuous touch*. Google Scholar serves as a database as well as a citation index and can be utilised by systematic review teams for both purposes. When conducting subject-based searches, as we do in this instance, Google Scholar is regarded as a database in the PRISMA 2020 flow diagram. We found Google Scholar to be sufficient for the purposes of this chapter as it includes the vast majority of established international peer-reviewed conferences and journals in the field.

The process for determining which articles were suitable for inclusion in our analysis was established through a set of specific eligibility criteria. In order to be considered for inclusion, the articles had to meet certain standards in regards to the research methods used. First and foremost, the primary focus of the study had to be on continuous touch-based authentication. This means that any articles that dealt

primarily with other types of authentication methods, such as mobile keystrokes or tapping, were excluded from the analysis. This was done in order to ensure that the studies being included were directly relevant to our thesis. Additionally, the studies had to make use of machine-learning-based models in their research. We did include articles that, in addition to touch-dynamics-based features, also incorporated other features in their analysis, such as ones based on accelerometer and gyroscope data. However, when it came time to execute our performance evaluation, we chose not to make use of these additional features. This decision was made because the data needed to extract these features is not always available in the public datasets that we consider in this thesis.

We then implemented an ancestry approach with the articles meeting the inclusion and exclusion criteria. Our keyword-based search identified a total of 685 articles that appeared to be potentially relevant. However, it was necessary to further evaluate each article to ensure that it met the inclusion criteria for our study. To do this, we implemented a screening process in which we inspected the title and description of each paper to ensure it is related to the topic at hand. We excluded any studies that were not published in peer-reviewed conferences and journals. This includes papers which were solely published in distribution services such as arXiv[1]. Additionally, any duplicate articles were removed during this step to avoid potential biases in our analysis. After completing this screening process, we were left with 103 articles.

Finally, we applied our complete eligibility criteria. This step included a more thorough analysis of the studies. We ensured the main focus of the studies is on touch-based authentication and that they utilise machine-learning models. After this process, we determined that 30 articles were suitable for inclusion in our final review. For each of the 30 articles that were included in the final review, we manually tabulated a number of details. This process included noting the specific features that were used in the study, the classification and aggregation methods that were employed, and the metrics that were used to evaluate performance.

[1]https://arxiv.org/

**Table 4.1:** Techniques in touch-based authentication studies. "STB" stands for stroke-based, "SSB" for session-based and "IB" for image-based. The following symbols are used in the table: **?** - unclear, ● - we can completely reproduce the features described in the paper and can compare it with other feature sets, ◐ - we can reproduce part of the features described in the study but cannot compare it with other feature sets, ○ - features are not described well enough to be reproduced.

| Study (Year) | Features (Count) | Feature Reprod. | Classifiers | Metrics | Aggregation |
|---|---|---|---|---|---|
| [150] (2012) | STB (53) | ○ | DT, RF, BN | FAR, FRR | Vote |
| [1] (2012) | STB (31) | ● | kNN, SVM | EER | Mean |
| [2] (2013) | STB (13) | ● | SVM | ACC | Feed |
| [76] (2013) | STB (?) | ○ | OC-SVM, SVM | FAR, FRR, ACC | Trust |
| [69] (2013) | STB (28) | ● | LR, SVM, RF, NB, NN, kNN, BN, SM, Euclidian, DT | EER | Feed |
| [68] (2014) | STB (?) | ○ | kNN | ACC | Feed |
| [151] (2014) | STB (?) | ◐ | IF, SVM, NB, BN, RF | ACC | Feed |
| [152] (2014) | SSB (8) | — | DT, NB, RBFN, PSO-RBFN, NN, kNN | FAR, FRR | Other |
| [153] (2014) | STB (31) | ● | HMM | EER, FAR, FRR | Mean |
| [124] (2014) | STB (37) | ● | SVM | ACC, AER | Mean |
| [154, 155] (2014) | IB | — | Proprietary | ACC, EER | Other |
| [156] (2015) | STB (58) | ● | kNN, SVM, NN, RF | ROC, FRR, FAR | Feed |
| [157] (2015) | STB (27) | ● | SVM, KSRC, KDTGR | EER | Mean |
| [67] (2015) | STB (15) | ● | kNN, RF, SVM | ACC | Trust |
| [158] (2015) | STB (15) | ● | NN, CPANN | ANGA, ANIA | Trust |
| [159] (2015) | STB (5) | ● | StrOUD | ROC, EER | N/A |
| [160] (2016) | SSB (5) | — | kNN, RF | FAR, FRR, ACC | Other |
| [136] (2016) | STB (24) | ● | kNN, SVM, NB, LR, RF, GB | EER | Mean |
| [137] (2017) | IB | — | SVM, DT, RF, NB | ACC | Other |
| [161] (2017) | STB (59) | ◐ | SVM, RF, DT | AUC | N/A |
| [162] (2018) | SSB (5) | — | AB, NB, kNN, LDA, LR, NN, RF, SVM, OC-SVM, LOF, IF, EE | FAR, FRR, HTER, AUC | N/A |
| [140] (2018) | SSB (21) | — | DT, NB, Kstar, RBFN, NN, PSO-RBFN | FAR, FRR, AER | Other |
| [163] (2018) | STB (8) | ● | IF | ANGA,ANIA | Trust |
| [164] (2019) | STB (28) | ● | NN | ACC, EER | N/A |
| [165] (2019) | STB (18) | ● | RF, SVM, LR, NB, NN | EER | Vote |
| [166] (2019) | STB (18) | ◐ | NB, NN, RC, RF, BN, DT | ACC | N/A |
| [143] (2019) | STB (16) | ◐ | IF, OC-SVM | ACC | Trust |
| [167] (2020) | STB (?) | ○ | NN | FAR, FRR, EER | N/A |
| [168] (2021) | STB (12) | ● | NN | ACC, AUC, FRR, FAR | Mean |
| [169] (2021) | STB (30) | ● | OC-SVM, kNN, NN, DT, RF, NB | ACC, FAR, FRR, EER, ROC | N/A |

### 4.2.2 Findings

The results of the survey we conducted have been succinctly summarised in Table 4.1. We deliberately do not include the performance reported by each study in the table. That is due to the variety of metrics and datasets used, making the exact performance data meaningless to compare across the studies, hence our experimental work in this chapter.

To provide a more comprehensive understanding of our findings, we have organised them into four distinct sections. Each section encompasses the results pertaining to a specific step in the continuous touch-based authentication lifecycle. These four sections are features, classifiers, aggregations, and metrics. The section on features deals with the different approaches for extracting information from strokes. The classifiers section covers the various machine learning algorithms that have been employed to identify users based on their touch-based inputs. The aggregations section looks at the methods for combining multiple strokes in order to improve the overall accuracy of the authentication system. Finally, the metrics section delves into how the results of the authentication systems have been reported and evaluated.

#### Features

We carefully examined the related work and what characteristics are typically used to describe strokes. We found that we can broadly categorise features according to three classes:

- *Stroke-based*: These features are based on data derived from individual strokes. Typically, the features are generated by examining a list of (X, Y, pressure, area) points that form a complete stroke. Examples of such features include the starting X or Y position, the length of the stroke trajectory, the average pressure of the stroke, etc.

- *Image-based*: These methods are based on generating an image that represents the stroke on a 2D plane. The images are then fed into image processing

pipelines for texture and shape extraction [137] or to compute a difference score between images [154].

- *Session-based*: These methods are based on the properties of whole sessions rather than a single or small group of stroke-based features. Examples of such features include the number of strokes per session, the average time duration of strokes per session, the average time duration between strokes per session, etc.

Most studies make use of *stroke-based* features (80%). The rest are split into 13% *session-based* features and 7% *image-based* ones. The prevalence of *stroke-based* features can be explained by the high computational cost associated with image processing and the long feature accumulation period of *session-based* methods, during which the device is left unprotected.

In total, we found that only 16 (67%) of the 24 *stroke-based* studies defined their features and extraction methods sufficiently enough to be reproducible. Another 4 (17%) of the *stroke-based* studies have feature sets that can be only partially reproduced as several features do not have clear and non-ambiguous descriptions. For instance, one of the articles has a feature described as "the angle of moving during swiping" [161], without detailing how and which angle is calculated. This makes it difficult to implement such features, and we only partially reproduce the studies. For a further 4 (17%) of the *stroke-based* studies, we could not infer the individual features used for authentication. In this case, the feature details were delivered in broad category definitions rather than specific descriptions, or the information was not provided by the authors at all.

In total, we identified 149 *stroke-based* unique features that we could re-implement from all papers investigated. The list of features can be seen in Table 4.4. The average number of features per paper is 24, where the largest number of *stroke-based* features identified in a single paper is 59 [161], and the smallest is 5 [160]. In this chapter, we focus on the *stroke-based* feature extraction due to being the most frequently used method in the field. Furthermore, we argue that it is the

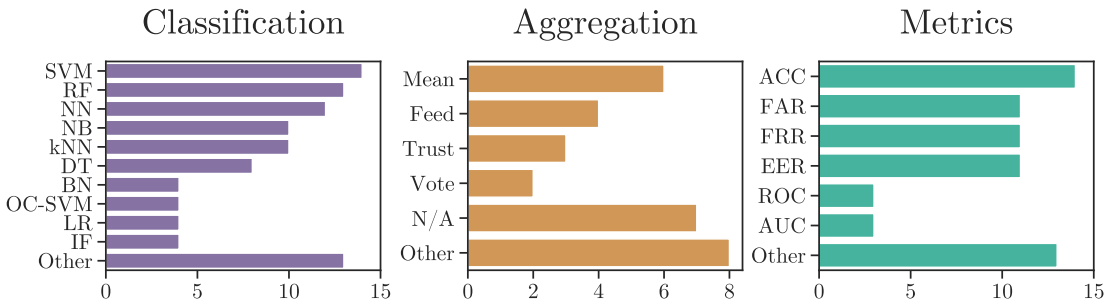



**Figure 4.1:** The prevalence of classifiers, aggregation methods, and performance metrics in continuous touch-based authentication studies. The "Other" category means the particular methods have been used less than three times in the case of Classification and Metrics and less than two times in the case of Aggregation.

most realistic approach given the computational, time, and security constraints of continuous mobile authentication systems.

### Classification

The studies included in Table 4.1 comprise a comprehensive examination of a variety of classification approaches in touch-based authentication systems, with a total of 27 unique methods represented among them. Many of these approaches are readily accessible and can be easily implemented using the pre-built, out-of-the-box functionality provided by popular machine-learning libraries. We present an analysis of the prevalence of various classification models used in the studies examined in Figure 4.1. The results indicate that the most frequently used models are Support Vector Machine, Random Forest, and Neural Networks, with 47%, 43%, and 40% of studies, respectively, including them as part of their analysis. It is worth noting that the maximum number of classifiers included in a single study is 12 [162]. Additionally, it is notable that 12 (or 44%) of the classifiers appear only once across all studies.

### Aggregation

In our analysis of the existing literature on continuous touch-based authentication systems, we found that a significant portion of the studies (77%) incorporated the optional aggregation step as part of their analysis. There are multiple ways

in which we can process a group of strokes, or a sequence of touch events, in order to extract the most relevant information and improve the performance of the authentication system. We found that we can broadly categorise the methods into the following four classes:

- *Mean/Median*: In this modality, we take the average or median value of the predictions for each stroke generated by the classifier. We use the resulting value to make classification decisions about the group of strokes.

- *Vote*: Here, we take the most common binary prediction. In this case, the classifier would be choosing between a stroke belonging to the legitimate user or anyone else. Once we have selected the most used prediction, we make a final conclusion about the group of strokes.

- *Feed*: In this approach, we combine the features of all strokes in the group and feed them together into the model at once. A single prediction is obtained by the classifier. For instance, if there are 10 features with a window size of 5 (i.e. group of five consecutive strokes), we input all 50 features into the model at once and receive one probabilistic output, as opposed to 5.

- *Trust*: There is a large variation in this category, however, in general, the approach relies on a statistical formula that outputs a score as new strokes are considered. The score is updated by rewarding positive predictions and penalising negative predictions proportionally to the individual classifier predictions. An instance of such aggregation methods is the dynamic trust model [170], which is tailored to continuous authentication biometric systems. This specific implementation has been used in [158] and [143].

We present the prevalence of each of these methods in Figure 4.1. The *Mean/Median* aggregation approach is the most frequently used one (20%). The *Vote*, *Feed*, *Trust* methods are used in 6%, 13% and 10% of the studies, respectively. As mentioned, 23% of the studies do not use aggregation at all, and a further 27% use solutions that do not fall into the categories described above. For

instance, the systems using session-based features are making decisions based on a large aggregation of strokes but cannot be included in any of the other categories we describe.

#### Metrics

Depending on the needs of a particular system, there are a variety of metrics that can be used to measure the performance of a model for continuous touch-based authentication. These include FAR, FRR, EER, Accuracy, ROC curve, and others. We have a thorough description of each one in Chapter 1. Statistics for the prevalence of these metrics in the field can be found in Figure 4.1. The variety of metrics shown illustrates the difficulty in comparing results reported in continuous touch-based authentication research. In this chapter, we aim to ease this comparison by reporting on differences in approaches when they are examined under the same conditions and by reporting the results using the same metric.

As mentioned in this thesis, we report our results using the EER metric. The EER is the point at which the FAR and FRR are equal on the ROC curve. The ROC curve is obtained by varying the threshold for acceptance into the biometric system. Therefore, there is a value of the threshold which corresponds to the EER. While some systems might benefit from choosing thresholds for optimising better FAR or FRR, we believe EER is the most representative of the general performance of a system. This is also supported by the related work [74] and Chapter 3. In particular, in Chapter 3, we show that when comparing two continuous touch-based authentication models, the performance differences between them on the ROC curve are largely consistent with the difference at the EER point.

### 4.2.3 Datasets

In this section, we include a list of publicly available datasets which comprise of touchscreen interactions. There are dozens of studies that design and implement their own experiments for data collection, as shown in Chapter 3. However, the vast majority do not share the resulting dataset upon publication. We present 9

publicly available touch-based authentication datasets in Table 4.2. We use the following criteria derived from Chapter 3 to select the datasets applicable to our investigation. Naturally, the data in question needs to be accessible at the time of requesting it. That is not always the case, as some of the servers hosting the data have gone down since paper publication. The dataset itself should contain a group of users using the same smartphone model. The users should have performed at least two separate sessions of the experimental tasks. Furthermore, each stroke needs to contain information about its (X, Y) coordinates as well as touch area and pressure values. These are required for the majority of feature extraction approaches. The only publicly available datasets which we consider usable under these conditions are Touchalytics [1], Bioident [67] and CEP, which is the dataset collected in Chapter 3. We focus on these three datasets in the rest of this chapter and describe the reason for not using the other datasets in the "Notes" column in Table 4.2. We use the data from the image gallery task in the CEP dataset and the raw data from the Touchalytics and Bioident datasets. For each of the three datasets, we select the largest subset of users who use the same phone model and perform two or more sessions.

## 4.3 Performance Evaluation

The objective of this performance evaluation is to determine the best-performing existing feature sets, classifiers, and aggregation methods. Furthermore, we aim to identify a set of novel techniques and compare them to the current state-of-the-art. Finally, our work aims to understand whether the results obtained are valid across multiple publicly available datasets. To this end, we examine how each classifier performs on different feature sets and then compare aggregation methods independently.

Throughout this chapter, we follow the best practices from Chapter 3 for fair evaluation of continuous touch-based authentication systems. We create the following model for each user and record the mean EER across all users at the end. We only select users who have performed at least two sessions and use the same

**Table 4.2:** Publicly available touch-based datasets. The following symbols are used in the table: ● - currently accessible without additional processes, ◐ - can be accessed through email or special process, ○ - link or instructions currently not working. Links accessed on 4 January 2023. The "Usable" column denotes the largest group of users with the same phone model, at least two sessions, and data for coordinates, pressure, and area.

| Dataset | Year | Total Users (Usable) | Sessions | Accessible | Notes |
|---|---|---|---|---|---|
| Touchalytics [1] | 2012 | 41 (15) | 3 | ● | - |
| WVW [69] | 2013 | 190 (0) | 2 | ○ | Data currently not accessible |
| TCPA [124] | 2014 | 32 (0) | 1 | ◐ | Only a single session |
| UMDAA-02 [136] | 2016 | 48 (0) | 11-429 | ● | Touch area values unavailable |
| Bioident [67] | 2016 | 71 (26) | 1-4 | ● | - |
| TGA [165] | 2019 | 31 (0) | 8 | ◐ | Data contains only extracted features |
| Brainrun [171] | 2020 | 2344 (0) | 1-1105 | ● | Pressure and area values unavailable |
| HuMIdb [145, 172] | 2020 | 600 (0) | 1-5 | ◐ | Session contains only a single stroke |
| CEP (Chapter 3) | 2022 | 470 (64) | 1-30 | ● | - |

phone model. At first, we split the data of a target user, selecting their first 80% of sessions for positive training data and the remaining 20% for positive testing data. We split the rest of the users into independent training and testing groups at random. The users in each group never overlap. The negative data for training or testing is then obtained by selecting a stroke at random while cycling through the respective group of users until the number of negative training or testing strokes is equal to the positive one. The combined training set is then used to train a binary model, and the testing set for evaluating the performance of the model. This whole process is repeated 10 times for each experiment, and we report the mean of the results from each repetition. At each of these iterations, we randomly select the training and testing user groups. The one-class classifiers employ the same process, however, the negative training data is not used.

The SVM, RF, NB, kNN, DT, OC-SVM, LR, and IF classifiers we investigated

were implemented using the `scikit-learn` [173] machine learning library. The Neural Networks were implemented using `Tensorflow` [174] and the `Keras` [175] API. The Bayesian Network implementation was done on the WEKA [176] machine learning library using a Python wrapper. The implementation details of each classification algorithm were left as close to the default as possible. Where we had to make decisions (e.g., in the case of kNN and Neural Networks), we considered the related work and performed preliminary experiments to decide on the hyperparameters. The final parameters for each classifier are given below:

- Support Vector Machine (SVM) - RBF kernel with a 'scale' coefficient.
- Random Forests (RF) - 100 estimators and max depth of 20.
- Neural Network (NN) - feed-forward with three hidden layers of 150, 150, and 75 with a 'ReLU' activation function. The output layer has a 'Sigmoid' activation function which outputs a probability of a match between 0 and 1. Batch-normalization is applied at each layer, and a 0.3 dropout between the hidden layers. The optimiser is 'Adam' with a 'binary cross-entropy' loss function. The network is trained with a batch size of 20 over 50 epochs.
- Naive Bayes (NB) - gaussian naive bayes implementation.
- k Nearest Neighbors (kNN) - number of neighbours: 18.
- Decision Trees (DT) - Gini criterion and no maximum depth.
- Bayesian Network (BN) - K2 for learning and Simple Estimator for predictions.
- One-Class Support Vector Machine (OC-SVM) - RBF kernel with a 'scale' coefficient.
- Logistic Regression (LR) - LBFGS solver with L2 penalty and max iterations of 1000.
- Isolation Forest (IF) - 100 estimators.

**Table 4.3:** Reproducible feature sets used in the performance comparison. The additional (non-touch-based) features were used in the final proposed model by the original paper. However, we do not re-implement them due to the lack of such data in all of our datasets.

| Study | Year of Proposal | Features Count | Additional Features |
|---|---|---|---|
| Frank et al. [1] | 2013 | 30 | ○ |
| Li et al. [2] | 2013 | 14 | ○ |
| Serwadda et al. [69] | 2013 | 28 | ○ |
| Xu et al. [124] | 2014 | 37 | ○ |
| Murmuria et al. [159] | 2015 | 5 | (*sensors, power*) ● |
| Antal et al. [67] | 2015 | 15 | ○ |
| Mahbub et al. [136] | 2016 | 24 | ○ |
| Shen et al. [156] | 2016 | 58 | ○ |
| Filippov et al. [163] | 2018 | 11 | ○ |
| Syed et al. [165] | 2019 | 18 | ○ |
| Rocha et al. [168] | 2021 | 12 | ○ |
| Incel et al. [169] | 2021 | 30 | (*sensors*) ● |

### 4.3.1 Comparison

In order to compare the performance of selected feature sets, we reproduced the 16 feature sets marked as "Feature Reproducible" in Table 4.1. Four of the studies [153, 157, 158, 164] implement the same group of features as other ones in the set, leaving us with 12 unique and complete feature sets. We focus specifically on these feature sets as they are the only completely reproducible ones in related work. More details for each feature set are given in Table 4.3. Some of the studies enhance their touch-based features with auxiliary data, such as ones coming from the accelerometer or gyroscope, however, we do not reproduce these features due to the lack of such data across all datasets. We compared the 9 most frequently used classifiers in continuous touch-based authentication studies as shown in Section 4.2.2 across all of the 12 feature sets and report their performance in EER. We also compared all five aggregation techniques described in Section 4.2.2 to highlight the best-performing method. In addition, we include the analysis of the *Median* of

scores as an alternative to the *Mean* approach. The aggregation window we chose in this set of experiments is 5 based on the availability of data and the diminishing returns of larger window sizes as shown in the related literature [1] and also by our experiments in Chapter 3. For this comparison, we use the novel classifier and feature set described below. Based on our findings in the previous section, we also propose two novel techniques which were not identified in other continuous touch-based authentication studies. We include these in our final analysis.

**Novel feature set**

We compiled a new feature set by implementing all *stroke-based* features from our literature review. These are derived from the X, Y, pressure, and area values of a stroke as described in Table 4.4. In addition to this, we utilised a feature selection algorithm that reduces the total number of features from the dataset. The goal of such approaches is to ensure better computational performance and overall results. For instance, this can be achieved by pruning features that contribute little to the output of the classifier or even have a negative effect on it. The feature selection algorithm we use is Analysis of Variance (ANOVA) using the F-value between features and labels. In order to ensure the method generalises well, we used the three datasets (CEP, Bioident [67] and Touchalytics [1]). We first selected the $n$ number of features for each dataset using ANOVA. Then, we only kept features that were sampled in at least two of the three datasets. We experimented with sizes 50, 75, 100, and 125 for the parameter $n$. In our preliminary results, we established that $n = 125$ is the best-performing one in our case, and we use it for the rest of this chapter. However, we highlight that, in this case, the general method for feature selection is more important for further research or industry applications rather than the individual features we chose.

**Novel classifier**

We propose an ensemble method for classification based on a combination of results from other classifiers. Ensemble methods are a well-known strategy used to combine multiple machine learning models which produce a result better than the

**Table 4.4:** Stroke-based features found in related work. "Perc." stands for percentile and "Std. Dev." for standard deviation. Full details about each of the features can be found in the corresponding papers. Note that [153, 157, 164] use the same features as [1] and [158] uses the same as [67] except they omit the mid-stroke pressure.

| Feature | Studies | Feature | Studies |
|---|---|---|---|
| 1-2. Start X,Y | [1, 2, 67, 124, 156] [136, 163, 165, 169] | 43. Std. Dev. acceleration | [69, 156] |
| 3-4. Stop X,Y | [1, 67, 124, 136, 156] [163, 165, 169] | 44-47. First Quartile pressure, area, velocity, acceleration | [69] |
| 5. Stroke duration | [1, 2, 67, 69, 124, 156] [136, 159, 163, 165, 169] | 48-51. Third Quartile pressure, area, velocity, acceleration | [69] |
| 6. End-to-end distance | [1, 67, 69, 124, 159] [136, 163, 165, 169] | 52-55. Extreme point 1,2 - X,Y | [69] |
| 7. Mid-stroke pressure | [1, 67, 69, 156, 165, 169] | 56. Last 2 points tangent | [69] |
| 8. Mid-stroke area | [1, 67, 69, 169] | 57. Velocity at first point | [124] |
| 9. Length of Trajectory | [1, 2, 69, 124, 156] [67, 163, 165, 169] | 58-60. Area, Pressure, Velocity at last point | [124] |
| 10. Inter-stroke time | [1, 136, 165] | 61. Last moving direction | [124] |
| 11. Mean Resultant Length | [1, 67, 136, 169] | 62. Average points distance | [124, 156, 168] |
| 12. Median acceleration at first 5 points | [1, 136, 169] | 63. Std. Dev. points distance | [124, 156] |
| 13. Median velocity at last 3 points | [1, 136, 169] | 64-68. LDP X, Y, Area, Pressure, Velocity | [124, 143] |
| 14. Average velocity | [1, 69, 124, 156] [67, 163, 165, 169] | 69-71. Start to LDP Latency, Length, Direction | [124] |
| 15. Up/Down/Left/Right | [1, 67, 136] | 72-74. LDP to Stop Latency, Length, Direction | [124] |
| 16. Direction of direct line | [1, 67, 124, 159, 165, 169] | 75. Ratio distance to LDP Length | [124] |
| 17. Average direction | [1] | 76. Total displacement length | [156] |
| 18. Ratio of direct distance to trajectory length | [1, 124, 136, 165, 169] | 77. Ratio of displacement and trajectory length | [156] |
| 19. 20% perc. velocity | [1, 136, 165, 169] | 78-81. Median, IQR, Skewnsess, Kurtosis of distance | [156] |
| 20. 50% perc. velocity | [1, 69, 136, 156, 165, 169] | 82-86. Avg, Std. Dev, IQR, Skewness, Kurtosis of deviation | [156] |
| 21. 80% perc. velocity | [1, 136, 165, 169] | 87-92. Avg, Median, Std Dev, IQR, Skewness, Kurtosis of pairwise angles | [156] |
| 22. 20% perc. acceleration | [1, 136, 169] | 93-98. Avg, Median, Std. Dev, IQR, Skewness, Kurtosis of phase-angles | [156] |
| 23. 50% perc. acceleration | [1, 69, 136, 156, 169] | 99. Displacement to duration ratio | [156] |
| 24. 80% perc. acceleration | [1, 136, 169] | 100-102. IQR, Skewness, Kurtosis of velocities | [156] |
| 25. 20% perc. deviation | [1, 136] | 103-108. Avg, Median, Std. Dev., IQR, Skewness, Kurtosis of angular-velocities | [156] |
| 26. 50% perc. deviation | [1, 136, 156] | 109-111. IQR, Skewness, Kurtosis of accelerations | [156] |
| 27. 80% perc. deviation | [1, 136] | 112-114. IQR, Skewness, Kurtosis of pressures | [156] |
| 28. Largest deviation | [1, 67, 136] | 115-116. Min, Max pressure | [161, 168] |
| 29. Pressure at first point | [2, 124] | 117-118. Min, Max area | [161, 168] |
| 30. Area at first point | [2, 124] | 119-120. Min, Max velocity | [143, 161] |
| 31. First moving direction | [2, 124] | 121-124. Min, Max, Mean, Median of pressure changes | [161] |
| 32. Average moving direction | [2, 67, 136, 169] | 125-128. Min, Max, Mean, Median of area changes | [161] |
| 33. Average moving curvature | [2] | 129-130. X, Y at max velocity | [143] |
| 34. Average curvature distance | [2] | 131-132. X, Y at min velocity | [143] |
| 35. Average pressure | [2, 69, 124, 156, 159] | 133-135. Quadratic fit pressure x2, x, n | [168] |
| 36. Average touch area | [2, 69, 124, 159, 163, 168] | 136-138. Min, Max, Avg time duration between points | [168] |
| 37. Max-area portion | [2] | 139-140. Max deviation of mean X,Y | [169] |
| 38. Min-pressure portion | [2] | 141-142. 20% perc. deviation of mean X,Y | [169] |
| 39. Average acceleration | [69, 156] | 143-144. Median deviation of mean X,Y | [169] |
| 40. Std. Dev. pressure | [69, 124, 156] | 145-146. 80% perc. deviation of mean X,Y | [169] |
| 41. Std. Dev. area | [69, 124] | 147-148. Direction vector X,Y | [163] |
| 42. Std. Dev. velocity | [69, 136, 156] | 149. Horizontal/Vertical flag | [165] |

outcome of each individual classifier. This is due to the fact that on some examples, some classifiers might perform poorly, but on average, models will agree on the correct decision. The algorithm we use outputs a final score by averaging out the probabilities from the predictions of the best-performing individual classifiers. We

**Table 4.5:** Performance of classifiers applied to different feature sets on the CEP dataset. No aggregation is used and the results are reported in EER (%). The average of each row and column is given. Our feature set consists of all extracted features from related work using and the ANOVA feature selection algorithm. Our classifier consists of an ensemble method using SVM, RF and NN.

| Set | SVM | RF | NN | NB | BN | KNN | DT | LR | OC-SVM | IF | Our | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [1] | 14.15 | 13.75 | 13.48 | 21.30 | 18.67 | 16.76 | 22.41 | 18.06 | 25.80 | 26.22 | 12.86 | 18.50 |
| [2] | 15.09 | 14.64 | 14.60 | 21.77 | 18.51 | 17.48 | 23.37 | 20.50 | 24.59 | 26.74 | 13.91 | 19.20 |
| [69] | 14.10 | 14.56 | 13.57 | 20.54 | 18.07 | 16.26 | 22.88 | 17.84 | 23.97 | 25.39 | 13.10 | 18.21 |
| [124] | 13.50 | 13.40 | 13.22 | 19.94 | 16.12 | 16.12 | 22.08 | 17.79 | 23.75 | 23.90 | 12.46 | 17.48 |
| [156] | 15.79 | 15.36 | 14.97 | 23.17 | 20.46 | 19.23 | 24.00 | 19.01 | 27.46 | 29.19 | 14.36 | 20.27 |
| [67] | 14.00 | 14.39 | 13.95 | 20.79 | 18.30 | 16.34 | 22.88 | 19.02 | 23.66 | 24.41 | 13.32 | 18.28 |
| [159] | 20.43 | 18.78 | 19.60 | 24.45 | 22.16 | 21.70 | 25.62 | 25.41 | 26.58 | 26.13 | 18.62 | 22.68 |
| [136] | 15.83 | 15.69 | 15.28 | 22.55 | 19.72 | 18.07 | 24.23 | 20.43 | 27.27 | 27.45 | 14.60 | 20.10 |
| [163] | 15.17 | 15.88 | 15.22 | 21.15 | 19.71 | 16.86 | 23.78 | 21.51 | 23.17 | 24.29 | 14.67 | 19.22 |
| [165] | 16.71 | 15.71 | 16.00 | 23.27 | 19.07 | 18.99 | 23.83 | 21.78 | 27.68 | 27.09 | 15.13 | 20.48 |
| [168] | 25.54 | 24.74 | 24.70 | 29.99 | 26.39 | 26.50 | 31.07 | 28.69 | 33.06 | 33.48 | 24.02 | 28.02 |
| [169] | 13.68 | 14.25 | 13.32 | 21.19 | 19.99 | 16.52 | 22.72 | 17.71 | 24.31 | 25.62 | 12.79 | 18.37 |
| Our | 12.57 | 13.00 | 12.36 | 19.84 | 16.37 | 15.63 | 21.94 | 15.81 | 23.45 | 24.96 | 11.67 | 17.05 |
| | 15.89 | 15.70 | 15.41 | 22.30 | 19.50 | 18.19 | 23.91 | 20.27 | 25.75 | 26.53 | 14.73 | |

performed preliminary experiments with three different combinations of classifiers of sizes 3 (SVM, RF, NN), 5 (SVM, RF, NN, kNN, LR), and 7 (SVM, RF, NN, kNN, LR, NB, DT) and found that the best-performing one in our case is the one consisting of SVM, Random Forest, and Neural Network. Similar to the novel feature set selection, the specific group of classifiers that we chose is less of interest than the proposed method itself.

### 4.3.2 Results

The results from the feature set and classifier comparisons can be found in Table 4.5. On average, the best-performing feature set is the one generated from the proposed ANOVA method with an average of 17.05% EER across all classifiers. The best

**Table 4.6:** Performance of aggregation methods on the CEP dataset with our novel feature set and an ensemble classifier consisting of an SVM, Random Forest and a Neural Network). The aggregation window used is 5 and the results are reported in EER (%).

| Mean | Median | Vote | Feed | Trust |
|---|---|---|---|---|
| 6.35 | 6.47 | 10.59 | 7.07 | 6.55 |

performing of the studies we re-implemented was Xu et al. [124] with an average of 17.48% EER over all the machine learning algorithms examined. We attribute the low performance of some feature sets, such as Rocha et al. [168] (28.02%), to the small number of features included. However, the final model of the study uses additional features from sensor data which could result in much better overall performance.

In terms of classifiers, on average, our ensemble method was the best performing, with an average of 14.73% EER across all feature sets. The three individual classifiers in the ensemble (SVM, RF, and NN) also form a well-performing group with 15.89%, 15.70%, and 15.41%, respectively. The one-class classifiers (OC-SVM and IF) produced the worst results in our experiments. Overall the best-performing single-stroke model consisted of our proposed feature set and ensemble classifier with an overall performance of 11.67% EER for a single stroke. However, continuous touch-based systems do not operate on a single stroke as it can be insufficient for secure authentication. Aggregation methods, based on multiple consecutive strokes, result in improved system performance. The results from the aggregation experiment can be found in Table 4.6. The *Mean* (6.35%), *Median* (6.47%), *Trust* (6.55%) and *Feed* (7.07%) methods resulted in very similar performance. The worst performing aggregation method in our experiments was *Vote* with 10.59% EER. Nevertheless, all of the aggregation methods performed better than using a single stroke to make authentication decisions.

### Dataset comparison

In order to determine the generality of our results, we replicated our experiments across three datasets: CEP (Chapter 3), Bioident [67] and Touchalytics [1]. When examining each category (features, classifiers, and aggregations), we choose the

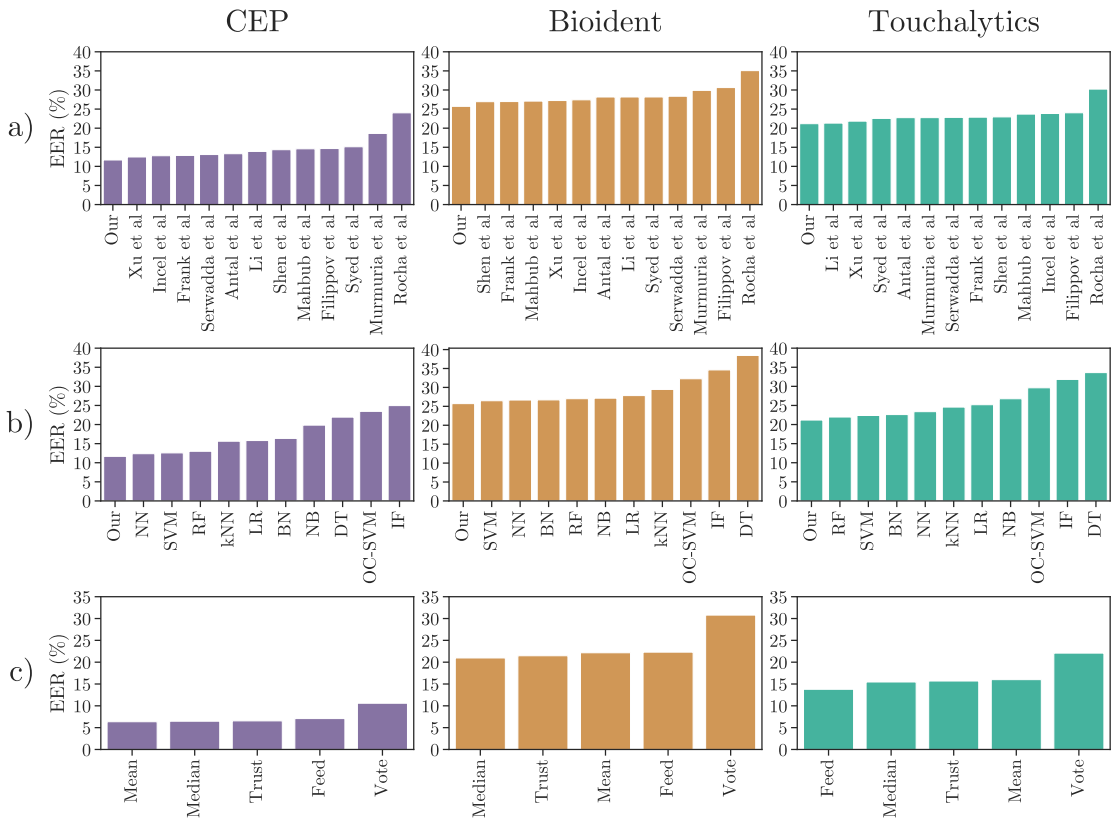



**Figure 4.2:** Performance of single-stroke models using (a) feature sets, (b) classifiers, and (c) aggregation methods across CEP, Bioident and Touchalytics touch-based datasets.

best-performing methods determined in our previous experiments and keep them constant (e.g., for all classifiers comparisons, we use the best-performing feature set). The results of our comparison across all three datasets can be found in Figure 4.2. Our novel feature set and the ensemble classifier consistently outperform the other methods across all three datasets. The best-performing aggregation method varies across the three datasets. However, the differences between the *Mean*, *Median*, *Trust* and *Feed* approaches are negligible, and either one can be used with an acceptable performance. The *Vote* aggregation method was the worst performing on all datasets. Xu et. al. [124] and Frank et al. [1] are other consistently well-performing feature sets with 37 and 31 features, respectively. Similarly, the individual SVM, RF, and NN classifiers provide stable performance across all the datasets examined.

## 4.4 Discussion

In this chapter, we presented a systematic analysis of continuous touch-based authentication techniques. To this end, we investigated a large number of studies focusing on continuous touch-based authentication to establish the most commonly used methods and grouped them into categories. In Section 4.3, we compared the performance of the techniques used in the field under fair conditions and established the best-performing methods. We found that the SVM, RF, and NN were the most robust and well-performing classifiers. Creating an ensemble using these methods resulted in a strong model outperforming the current state-of-the-art. Similarly, we used all features aggregated from related work and used the ANOVA feature selection algorithm to improve upon the current best-performing feature sets.

The chapter serves as a performance benchmark for the techniques for feature extraction, classification, and aggregation used in continuous touch-based authentication studies. The experiments conducted in Section 4.3 suggest that when performance is the only concern for a continuous touch-based authentication system, the optimal model would make use of our proposed feature set, ensemble classifier, and a mean aggregation approach which achieves an EER of 6.35% using a window of 5 strokes on the large CEP dataset. However, the lowest EER we achieved using this model is 4.80% when the aggregation window is increased to 16 strokes. The relative performance benefits of each novel technique are shown in Table 4.7. We highlight the next best and the median performing techniques while featuring the difference in EER with the novel ones proposed in this chapter. While the improvements might be perceived as marginal compared to the second-best methods, they are significant compared to the median. These results highlight the importance of fair comparison between models, which can be helpful for decision-making in the broader continuous touch-based authentication community.

The computational performance of the best-performing models might have a significant impact on their practical usage, particularly in mobile environments. When deploying a model to a mobile device, it is important to consider the computational resources available on the device, as well as the power consumption

required to run the model. If the best performing model is too computationally intensive, it may not be feasible to use in practice. In such a scenario, it may be beneficial to consider using a less sophisticated model architecture which can still deliver similar results but with a more cost-efficient computational performance. This can be achieved by reducing the number of parameters in the model, changing the classifier used or using optimised implementations for the specific environment.

Furthermore, we found that there is some variation between the results we obtained on each dataset. For this reason, selecting consistently well-performing models might be preferred for some applications. Selecting a model with consistent performance across datasets may be preferred for some applications, while for others, a model fine-tuned for a specific task may be more appropriate. The decision should be made considering the specific requirements and characteristics of the application and dataset at hand.

While EER is a good measure for the overall performance of biometric systems, in continuous authentication, the focus can be on guaranteeing a low False Negative Rate to ensure adequate usability of the system. However, that is application-specific and requires further examination, which is beyond the scope of this thesis.

It is worth noting that the results in our experiments might not match the results originally reported by a particular study, sometimes by a large margin. For instance, the EER we obtain using the Touchalytics [1] feature set on their dataset is multiple times higher than the one attained in the original study. This is due to the fair evaluation practices we follow as described in Chapter 3. Substantially less (16) of the original 41 users in the dataset fit into our criteria and were used in our evaluation. Many of them had done only two sessions, resulting in training and testing data skew closer to 50%/50% rather than the target of 80%/20%. Furthermore, we report the mean EER, while the original study reports the median. Even though we ultimately performed the comparison on all three datasets, we believe the results on the CEP dataset are the most representative. That is due to the larger size in terms of users, sessions performed, and the length of each session.

**Table 4.7:** Performance difference between the novel techniques proposed in this paper and the next best and median methods in related work. The differences in EER are reported in percentage points (%). The best performing feature set and classifier methods are the ones proposed in our study.

| | CEP | | Bioident | | Touchalytics | |
|---|---|---|---|---|---|---|
| | Next Best | Median | Next Best | Median | Next Best | Median |
| Features | *Xu et al.* (+0.79) | +1.94 | *Shen et al.* (+1.22) | +2.08 | *Li et al.* (+0.21) | +1.62 |
| Classifiers | NN (+0.69) | +4.14 | SVM (+0.76) | +1.44 | RF (+0.79) | +3.71 |

### 4.4.1 Limitations

There are several limitations to our experimental approach and results. Firstly, the implementation details of some features, classifiers and aggregation methods might not be perfectly reproduced from related work, despite our best effort. Furthermore, the categories we have grouped techniques in might be quite broad, with many internal differences between studies. For instance, implementing a generic trust model algorithm will not necessarily represent the nuances of all models falling under this category. Similarly, Neural Network implementations may vary between papers that differ from our architecture, and optimising the hyperparameters of other classifiers might lead to better overall performance. We believe that one-class classifiers, in particular, can achieve better results by fine-tuning their system parameters. Furthermore, some of the classification algorithms and feature sets that are not as prevalent in the field or are not reproducible might outperform the more widely available methods we examine. In that sense, there is also scope to improve the best-performing aggregation techniques with methods beyond the somewhat trivial ones described in this chapter. Finally, the fact that the methods we examine are mostly consistent throughout the three datasets is encouraging. However, application to other continuous touch-based authentication datasets might result in much different behaviour.

## 4.5 Conclusion

In this chapter, we performed a comprehensive review of the approaches for feature extraction, classification, and aggregation in the field.

To address RQ2.1 we investigated the prevalence of each technique in the relevant literature and categorised the feature extraction and aggregation methods. We found that the predominant feature extraction method is the stroke-based approach. Support Vector Machines were the most prevalent classifiers and "Mean" was the most widespread aggregation method. In terms of metrics, the most commonly used one is "accuracy" but we have strong evidence that EER is a better choice in this domain from the work in [75] and Chapter 3. As part of this investigation, we further presented and described a set of 149 unique features extracted from related work and identified 9 publicly available datasets for continuous touch-based authentication.

In order to address RQ2.2, we benchmarked the performance of the most common feature sets, classifiers, and aggregation methods in the field with a set of experiments consisting of a total of 204 model configurations. We ensured that the techniques are fairly compared by only modifying the variable examined and keeping the rest of the configuration constant. Such comparison is not possible across published studies as there is excessive variation in the configurations. Our results indicated that the most performant model makes use of the feature set described in [124], a Neural Network classifier and the "Mean" aggregation method.

In our efforts to tackle RQ2.3, we introduced a novel feature set based on the collection of 149 features with ANOVA feature selection and a novel ensemble-based classifier based on SVM, NN and RF classifiers. These methods outperform the state-of-the-art in their categories by 0.79% and 0.69% EER, respectively.

Finally, we concluded that our findings are largely similar across multiple datasets and provided a discussion of our results, including the limitation of the investigation.

*Il n'y a de réalité que dans l'action*

— Jean-Paul Sartre

# 5

# Privacy Concerns in Touch-Based Systems

## Contents


**5.1 Introduction . . . . . 116**
**5.2 Related Work . . . . . 117**
**5.3 Dataset and Features . . . . . 119**
**5.4 Fingerprinting . . . . . 120**
5.4.1 Evaluation Approaches . . . . . 121
5.4.2 Formalising our Approach . . . . . 122
5.4.3 Method . . . . . 124
5.4.4 Results . . . . . 126
**5.5 Personal Information Leakage . . . . . 130**
5.5.1 Method . . . . . 131
5.5.2 Results . . . . . 135
**5.6 Discussion . . . . . 137**
5.6.1 Countermeasures . . . . . 139
5.6.2 Limitations . . . . . 141
**5.7 Conclusion . . . . . 143**


In this chapter, we aim to understand and quantify the privacy threat landscape of touchscreen biometrics. These types of attacks are particularly alarming because touch interactions from mobile devices are ubiquitous and do not require additional permissions to collect. Two privacy threats were examined: user tracking and personal information leakage. First, we designed a practical fingerprinting simulation experiment and executed it on our large touch interactions dataset. We found

that touch-based strokes can be used to fingerprint users with high accuracy, and performance can be further increased by adding only a single extra feature. The system can distinguish between new and returning users with up to 75% accuracy and match a new session to the user it originated from with up to 74% accuracy. In the second part of the chapter, we investigate the possibility of predicting personal information attributes through the use of touch interaction behaviour. The attributes we investigated were age, gender, dominant hand, country of origin, height, and weight. We found that our model can predict the age group and gender of users with up to 66% and 62% accuracy, respectively. Finally, we discuss countermeasures and limitations and provide suggestions for future work in the field.

## 5.1 Introduction

In Chapters 3 and 4, we established best practices for evaluating continuous touch-based authentication systems and clarified the state-of-the-art in terms of techniques for improving performance. Developing a highly effective system is, of course, a crucial aspect of any technological advancement. However, as the technology matures and becomes more prevalent, it can raise important questions about its potential applications, as well as any unintended negative consequences that may arise. One area of concern that we would like to examine is the potential for continuous touch-based biometric technology to be used in ways that threaten the privacy of individuals. In this chapter, we will explore the various ways in which touch-based biometrics can be used to invade the privacy of users without their explicit knowledge and discuss the steps that can be taken to mitigate these risks.

To this end, we make use of a two-fold approach. First, we investigate the feasibility of fingerprinting users based on the way they interact with their mobile devices. Fingerprinting, also known as stateless tracking, can be a major concern for users as it can be used for a number of malicious purposes, such as discrimination and surveillance [80]. However, it can also be beneficial in cases such as personalising the user experience in mobile websites and applications. In the second part of the chapter, we examine the possibility of extracting personal information from

mobile users through the use of touch-based interactions. By personal information, we refer to the physical and intrinsic characteristics of humans. We evaluate the potential to reveal the following six attributes: age, gender, dominant hand, country of origin, height, and weight. We perform a series of experiments on a large publicly available touch interactions dataset to quantify the feasibility of the two privacy threats and give directions for future work.

To summarise, in this chapter, we aim to examine the practicality of using touch-based interactions for user fingerprinting. We introduce a realistic evaluation method and test two approaches for identifying new users, as well as re-identifying returning users. Additionally, we explore the potential of touchscreen interaction models to reveal information about a user's age, gender, dominant hand, country of origin, height, and weight. To do this, we experiment with three different data processing approaches and three machine learning algorithms. Finally, we discuss our findings, proposed countermeasures, and potential directions for future research in this area.

## 5.2 Related Work

Touchscreen interactions have been studied in the context of continuous authentication since the early 2010s. Several papers survey the development of touch-based authentication [177, 178] and investigate the approaches used in the field [69] including the work done in Chapter 4. However, studying the privacy implications of touch-based biometrics has been limited. Our work examines and quantifies the practical touch-based biometric implications on privacy with a focus on fingerprinting mobile users and revealing their personal information using touchscreen interactions. While traditional methods of tracking and personal information leakage, such as the ones described in Chapter 2, have known defence mechanisms that can be implemented to protect against them, the use of touch-based technology for tracking and identifying individuals may prove more challenging. The technology relies on the unique characteristics of an individual's touch interactions, which may be more difficult to obscure or alter than other types of identifying information. As a

result, it may be more difficult to protect against tracking and potential leakage of personal information through the use of touch-based biometrics.

The research work on touchscreen interactions as a method for fingerprinting users has been limited. To the best of our knowledge, the work of Masood et al. is the only one closely related to this particular area of research [179]. They have looked at the uniqueness of touchscreen interactions, thus showing that touch gesture features carry highly identifiable information about users and have a potential for fingerprinting. The study has similar goals to the first part of this chapter. However, we employ different evaluation methods with a focus on practicality and understanding.

Predicting personal information using touch interactions has received more attention from researchers in the field. Bevan et al. conducted a study of 178 users and showed that the differences in some features of strokes could reveal information about the handedness, thumb length, and gender of users [180]. Antal et al. created models to predict the gender and phone experience level of users. They report high accuracy of between 88-100% for gender and 80%-100% for phone experience requiring up to 20 strokes to make a decision [67]. Miguel-Hurtado et al. focused exclusively on gender prediction achieving up to 78% accuracy [181]. Similarly, Jain et al. focused specifically on gender prediction improving on previous work and achieving ∼93% accuracy [182]. Acien et al. [183], Nguyen et al. [184], and Cheng et al. [185] used touch interaction data of children and adults to differentiate between the two groups. All three studies report accuracies of above 96%.

Davis et al. used stroke data to predict both the gender of users and their age [186]. In the gender predicting scenario, they report a more modest 70% accuracy average across the different classifiers used. Predicting age groups above or below 40 years resulted in 80% accuracy. More recently, Wiliams [187] examined the feasibility of predicting location (the state within the United States), gender, race, and education level with 46%, 73.3%, 73.3%, and 26.7% accuracy respectively.

A variety of machine learning models have been used in these studies, including Support Vector Machine (SVM), Neural Network (NN), Random Forest (RF), Naive Bayes, Decision Tree, Logistic Regression, Nearest Neighbor, and others.

We differentiate the second part of this chapter from previous work by conducting a comprehensive evaluation of 6 personal traits (age, gender, handedness, height, weight, and location) using our constrained (9 iOS models) but large dataset in the context of privacy threats in touch-based biometrics.

## 5.3 Dataset and Features

In order to evaluate the privacy concerns stemming from the use of touch interactions, we conducted our experiments on the large open-source touch-based dataset (CEP) introduced in Chapter 3. We only use the remotely collected iOS portion of the dataset. To briefly remind of the details, the dataset consists of 470 users with up to 31 and an average of 13 sessions per user. The participants were required to perform two sessions daily, consisting of tasks aiming to mimic natural scrolling and swiping behaviour with each session lasting slightly less than 2 minutes on average. The social media task required users to scroll up and down through a randomly generated feed of posts and find an article related to a particular question. The image gallery task required users to swipe left and right through a series of images and count the number of specified objects. The data is limited to 9 iOS device models, and there are a total of 6,006 usable sessions for each of the tasks. The dataset has been collected over a relatively long period of time for such a study, meaning that the effects on the stability of the fingerprinting and personal information leakage methods are taken into account.

For our experiments, we used the set of 149 features introduced in Chapter 4. These are stroke-based (geometric) features describing the properties of a particular touchscreen gesture done by the user. The features are extracted from a series of points consisting of X, Y, pressure, and area values which describe a complete stroke. Since the authors found that this set of features works best for authentication, we hypothesised that it would also perform well on our tasks.

While we complete our investigation on an app-based dataset, we believe the findings in our work are relevant to websites as well because the recording of touchscreen strokes is also feasible on a mobile browser. For instance, it is possible to collect touch interaction data using JavaScript and the TouchEvents API [1]. This API is available on all modern mobile web browsers and extra permissions from the user are not required to access it. We do, however, acknowledge that there might be some limitations in place. For example, the API is clear that it cannot guarantee a specific touch sampling rate: *"The rate at which touchmove events are sent is browser-specific and may also vary depending on the capability of the user's hardware. You must not rely on a specific granularity of these events."* That is in contrast to the mobile application API, where the sampling rate is equal to the screen refresh rate (60Hz in most cases). Furthermore, in practice, there might be differences in the reported values for pressure and X, Y coordinates between APIs. Despite that, we believe that our findings are relevant for mobile web browsing as well. Similarly, our experiments are done on an iOS-exclusive dataset, however, there is no reason to believe there would be major differences with the Android operating system as the same methods for data collection are available.

## 5.4 Fingerprinting

Touch-based interactions can be used for tracking users on the web and mobile applications due to the unique differences in the way people interact with their touchscreens, as illustrated by continuous authentication research. In this section, we investigate the feasibility of tracking users by the way they interact with their smartphone screens. Unlike fingerprinting with sensors such as an accelerometer, gyroscope, and magnetometer, touch-screen-based tracking is invisible to the user and does not require extra permission requests. We briefly describe the evaluation methods used in fingerprinting systems, formalise our approach and present our findings.

[1] https://developer.mozilla.org/en-US/docs/Web/API/Touch_events

### 5.4.1 Evaluation Approaches

There are various methods for evaluating fingerprinting systems, each with its own strengths and weaknesses. We categorise and briefly describe each of the evaluation methods for fingerprinting systems. Additionally, we highlight why some approaches may be preferable for touch-based tracking as they are better suited to evaluating the performance of systems under realistic conditions. Ultimately, the choice of evaluation method will depend on the specific application of the fingerprinting system, and we focus on one method that is most suitable for our approach.

**Identification / Multi-Class Classification.** This approach assumes a closed set of users where a model is trained to predict the class a session belongs to. A class, in this case, is a user or a device. Whenever a new session is performed, the model can be used to match it to the correct class. This method, however, assumes that we have full knowledge about the number of users in the system and hence the classes a session can belong to. In other words, the evaluation does not take into account that a new session could be coming from a user that has never used the system before. This becomes an identification challenge, which is closer to authentication in nature than fingerprinting. However, the method can still be applicable in some narrow cases. For instance, it can be useful in the detection of multiple user accounts in a closed set of registered users. Das et al. [98] use this approach to evaluate their web tracking paper based on mobile motion sensors.

**Entropy and anonymity set.** The use of entropy as a measure of identifying information in a fingerprint has been widely adopted in the field, particularly in desktop settings [87, 188]. However, the notion of Shannon Entropy loses its usefulness when making decisions based on similarity rather than equality which is the case in fingerprinting systems using continuous feature values (e.g. motion sensors and touchscreen interactions). In other words, the minimum number of bits required to distinguish a user is not an important metric in our investigation, as nearly all of our sessions will be unique. However, a fingerprinting system needs to also recognise when sessions are performed by the same user. Similarly, the

anonymity set, which describes the size of groups with identical fingerprints, is also irrelevant in this case, as there are practically no anonymity sets larger than one. It is possible to use bins to create categorical data from continuous in order to use this evaluation method. However, losing the granularity of the data might lead to poorer performance. That is the approach taken by Masood et al. [179].

**Simulation.** In this approach, a simulation of user sessions visiting a website (or application) is modelled. The evaluation is performed in two steps: discrimination and re-identification. In the first step (discrimination), the system decides whether a session is coming from a new user who has never visited the website or a returning one that the system has observed already. The next step (re-identification) is performed only when the user is a returning one. At this stage, the system matches the session with the correct existing user. Both steps are typically done by measuring the distance between the features of the new session and the existing ones on record. Simulation fingerprinting systems have been used for motion sensor mobile fingerprinting by Hupperich et al. [92]. While this approach is suitable for continuous data, it can also be used for categorical data. For instance, Kurtz et al. [93] use the Jaccard similarity coefficient to measure the distance between sessions on categorical data.

### 5.4.2 Formalising our Approach

In order to evaluate the fingerprinting system based on touch interactions, we chose to use the simulation approach. We believe a simulation fits the continuous nature of our data and provides a better examination of how the technology can be used in practice. An example of how this technology can be applied is a web store where a number of users have placed an order in the past. The users consent to being fingerprinted and do not register an account. The goal of the system then is to identify returning users in order to upsell or advertise based on previous behaviour. In this case, the discrimination step is in place to avoid treating new users as returning ones, and the re-identification step to avoid advertising to the wrong person. Although we use this as an example of how the technology can be

used, the system has much wider applicability, both for benevolent and malicious purposes. We formally define our simulation as follows.

There is a set of users $G = \{U_1, U_2...U_n\}, n \in \mathbb{N}$. Each user in turn is a set of sessions $U = \{S_1, S_2...S_m\}, m \in \mathbb{N}$ and each session is a set of features $S = \{F_1, F_2...F_l\}, l \in \mathbb{N}$. For instance, $S$ can be the set of features described in Section 5.3.

The goal of our system is, then, to correctly classify a new session $S_u$ which either belongs to an existing user $U_n$ or a new user $U_{n+1}$. In order to make this decision, we use a dissimilarity function $D(S_1, S_2)$ where $S_1$ and $S_2$ are generic feature sets of the same size. In other words, $D$ measures how much two vectors differ from each other. Many functions can be used as a dissimilarity measure, including Euclidean distance, Manhattan distance, and machine learning algorithms such as Support Vector Machines and Neural Networks. Using the dissimilarity function $D$, we define $d_{min}$ to be the distance to the session most similar to $S_u$.

$$d_{min}(S_u) = \arg\min_{S \in U, U \in G} D(S, S_u)$$

We then define a threshold $\delta$, and if $d_{min}(S_u) > \delta$ we classify $S_u$ as belonging to a new user $U_{n+1}$. Else if $d_{min}(S_u) \leq \delta$ we classify the session as a returning one and mark the session as belonging to an existing user $U_i \in G$, which contains the session closest to $S_u$. Correct classification of a new session $S_u$ is then either of the following:

- $S_u$ is performed by a new user $U_{n+1}$ and we classify it as such
- $S_u$ is performed by a returning user, and we match it to the correct user $U \in G$

It is worth noting that we do not update the sets $G$ and $U$ while running this simulation. This is because we carefully balance the training and testing sets in our experiments for fair evaluation, and updates to these sets will make

results difficult to interpret. However, in practice, adding new users and sessions to the system is technically trivial.

### 5.4.3 Method

In order to evaluate the performance of the proposed fingerprinting approach, we executed the simulation described in the previous section on both the image gallery and social media tasks provided in our dataset. We investigated the performance of our system on the following three tasks:

- **Discrimination** - In this case, we choose a threshold that optimises the accuracy of the discrimination model which decides whether a session is coming from a new or returning user. This is a binary problem with a baseline accuracy of 50%.

- **Re-identification** - This task measures the accuracy of the model to match new sessions to users already known by the system. In this case, we only consider the returning sessions, hence the threshold here is not relevant. The baseline accuracy of this model is $1/n$, where $n$ is the number of users who are already known by the system.

- **Combined** - This task evaluates the correct classification of the system exactly as described in the simulation. In this case, both discrimination and re-identification steps need to be correct to mark a decision as accurate. The baseline accuracy is 50% since we can mark all sessions as new which is exactly half of our testing set. However, this baseline can be misleading as it does not measure the real purpose of the system, which also includes matching returning sessions to the original user.

In order to evaluate the system correctly, we split the session data such that there is an equal number of sessions coming from new and returning users. That is done by selecting $n$ number of users and setting aside half of their sessions for training and the other half as returning sessions for testing. The remaining users'

sessions are labelled as new and also used for testing. Users who have performed only a single session are always treated as new users because they do not have a second session that can be used for testing. There are a total of 61 users with only one session. The parameter $n$ is selected such that the amount of returning and new sessions used for testing the system is as close as possible. Then sessions are dropped to ensure the sets are exactly equal in size.

In our experiments, we started with the set of 149 features for each stroke as described in Section 5.3. Each value of the final feature vector is acquired by calculating the mean of each stroke feature throughout a whole session. In addition, we included an extra feature — the total number of strokes in a session, which results in a total of 150 features for each session. Then we applied feature selection algorithms to reduce the overall dimensionality of our data and achieve better computational performance and results. We applied the Analysis of Variance feature selection algorithm using the F-value between features and labels. The label in this case is the user id of the participant performing the session. We fit the algorithm only on training users and transform both the training and testing sets. We conducted preliminary experiments with varying the number of features $k$ and found that the best-performing $k$ for our purposes was 100.

We also performed a separate experiment where the phone model of the device performing the session (e.g. iPhone 7) is known, and we used it as an extra feature in our analysis. This is a realistic scenario as the phone model of a device is easily accessible by standard mobile application and web APIs.

For the dissimilarity function, we experimented with two approaches — one based on a vector distance metric and one based on a machine learning algorithm. The first approach uses the cosine distance ($D_c$), which is $1 - S_c$, where $S_c$ is the cosine similarity. It belongs to the interval $[0, 2]$. Given two feature vectors $S_1, S_2$, the cosine distance is defined as follows:

$$D_c = 1 - S_c = 1 - \frac{S_1 \cdot S_2}{||S_1|| \cdot ||S_2||}$$

The second approach we tested is based on a machine learning algorithm rather than a vector distance function. First, we trained an individual SVM classifier for each of the users in the training set. The positive class consists of the sessions of a particular user, and the negative class consists of sessions from the other users in the simulation training set. We keep the positive and negative classes balanced. These are small classifiers, as there are typically only a few positive examples per user. In the discrimination scenario, whenever we want to classify a session as new or returning, we make a prediction from each of the classes and record the maximum distance to the SVM hyperplane. If it is above the threshold ($\delta$), we mark it as a returning session and, otherwise, as a new user session. Similarly, in the re-identification scenario, we classify the session to belong to the user, with the SVM model producing the largest distance to the hyperplane. This is in contrast to the vector distance approach, where we choose the user with the lowest distance value. That is because the distance to the hyperplane represents how confident the model is about a prediction. The larger it is, the more likely the session is originating from the user the model has been trained on.

In practice, when a new user visits a website, it is unrealistic for the system to wait until the whole data for a particular session is available to make a decision. It should be possible to do that with a fraction of the data available. For this reason, we also repeated our experiments with a portion of the whole session data. We tested the system by only using the first 10% to 90% of the session data at 10 percentage point intervals.

For each of our experiments described in this section, we executed the simulation a total of 10 times, where each time, the group of $n$ users is randomised.

### 5.4.4 Results

The results for the discrimination, re-identification and combined scenarios on both image gallery and social media tasks are shown in Table 5.1. In addition, we highlighted the baseline performance of each scenario. The accuracy of our

**Table 5.1:** Performance of fingerprinting users with all strokes from a session. The cosine distance and SVM approaches are compared. The discrimination model differentiates between a new or returning user, the re-identification model matches a session to the user it originated from and the combined model fulfills both conditions. The image gallery task predominantly consists of swiping (left/right) behavior and the social media task mainly consists of scrolling (up/down) behavior. The threshold ($\delta$) is given in parentheses where relevant.

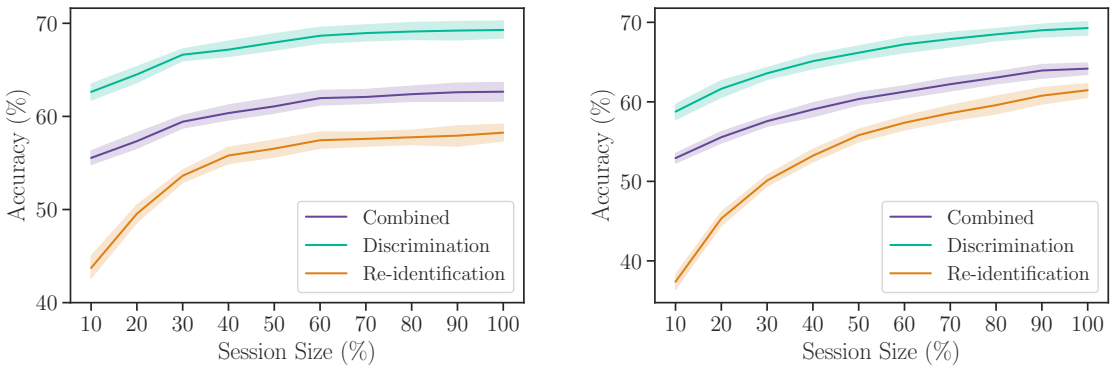
| | | Image Gallery Task | | | Social Media Task | | |
|---|---|---|---|---|---|---|---|
| | | Accuracy ($\delta$) | Phone Model Feature ($\delta$) | Baseline | Accuracy ($\delta$) | Phone Model Feature ($\delta$) | Baseline |
| Distance | Discrimination | 60.6% (0.25) | 64.2% (0.30) | 50% | 59.7% (0.28) | 65.8% (0.34) | 50% |
| | Re-identification | 40.8% (-) | 53.2% (-) | 0.36% | 41.9% (-) | 59.2% (-) | 0.36% |
| | Combined | 54.0% (0.17) | 58.1% (0.24) | 50%* | 53.4% (0.19) | 60.2% (0.28) | 50%* |
| SVM | Discrimination | 69.3% (0.85) | 72.3% (0.77) | 50% | 69.3% (0.84) | 75.0% (0.72) | 50% |
| | Re-identification | 58.3% (-) | 68.4% (-) | 0.36% | 61.5% (-) | 74.0% (-) | 0.36% |
| | Combined | 62.7% (0.94) | 67.3% (0.84) | 50%* | 64.2% (0.90) | 71.4% (0.78) | 50%* |

*Can be achieved by predicting all testing sessions as coming from a new user, however that does not represent the real purpose of the system.

system with the additional phone model feature is also given, and the distance and machine learning-based approaches are compared.

Overall, the SVM-based distance approach performed significantly better than the cosine distance measure with 62.7% and 64.2% accuracy compared to 54.0% and 53.4% for the two tasks in the combined scenario without the phone model feature. We discuss the results from the SVM approach for the rest of this section.

The touch-based fingerprinting method performs well in the re-identification scenario where returning sessions are matched to their original users. It achieves 58.3% and 61.5% accuracy on the image gallery and social media tasks, respectively. That is significantly higher than the baseline of 0.36%. The good performance of touch-based interactions for re-identification purposes is supported by the research in continuous touch-based authentication, which reports strong results in tasks with similar goals. In the discrimination step, where a session is labelled as originating from a new or returning user, the fingerprinting model also performs well. It achieves an accuracy of 69.3% on both tasks compared to the baseline of 50%.




**(a)** Image Gallery Task **(b)** Social Media Task

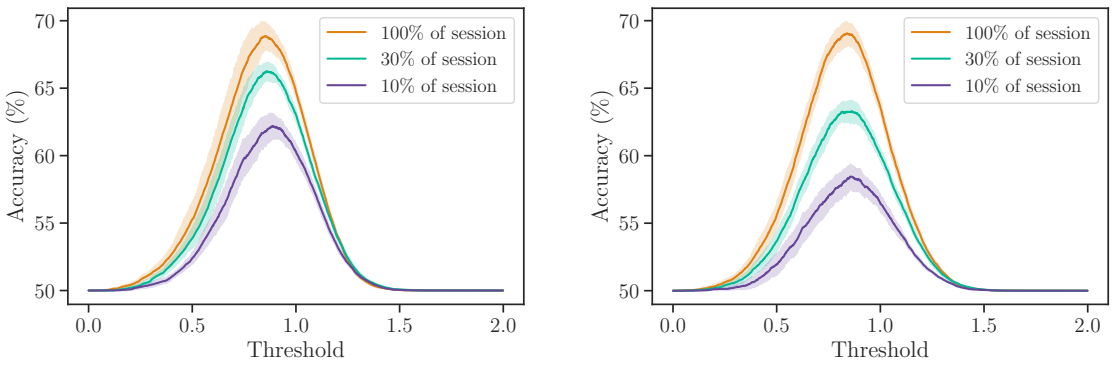
**Figure 5.1:** Performance of the fingerprinting simulation when using a fraction of the whole session data. The results are for both tasks without the additional phone model feature and using the SVM fingerprinting approach. The shaded area represents a 95% confidence interval.

The discrimination scenario represents an upper bound to the performance of the combined scenario, where we achieved 62.7% and 64.2% accuracy for the image gallery and social media tasks, respectively.

The performance of our system on both image gallery and social media tasks is equal in the discrimination case and differentiates only by 3.2% and 1.5% for the re-identification and combined scenarios respectively. The image gallery task results are slightly better in each case. This similarity in performance is encouraging for the applicability of the system for a variety of purposes and settings both on websites and mobile applications.

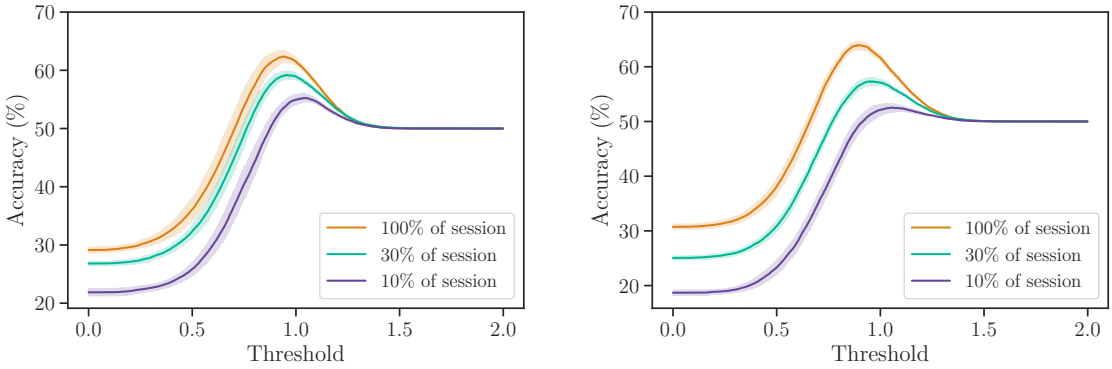
Including the phone model of the device as a single extra feature in our system resulted in better accuracy in all tasks. The increase in performance is by 5.7%, 12.5%, and 7.2% for the discrimination, re-identification, and combined scenarios in the social media task, respectively. The increase is similar (3%, 10.1%, and 4.6%) for the social media task. This is encouraging for the practical use of the system as it suggests that including a few conventional fingerprinting features in conjunction with the touch-based approach can result in a highly effective system.

The results of varying the session size used in our experiments are shown in Figure 5.1. The accuracy of the model in all scenarios is lower when less




**(a)** Image Gallery Task **(b)** Social Media Task

**Figure 5.2:** Discrimination model accuracy line plot at varying thresholds ($\delta$). Results are shown on both tasks and presented with session sizes of 10%, 30%, and 100%. The SVM fingerprinting approach without the additional phone model is used. The shaded area represents a 95% confidence interval.




**(a)** Image Gallery Task **(b)** Social Media Task

**Figure 5.3:** Combined model accuracy line plot at varying thresholds ($\delta$). Results are shown on both tasks and presented with session sizes of 10%, 30%, and 100%. The SVM fingerprinting approach without the additional phone model is used. The shaded area represents a 95% confidence interval.

data is used. However, the increase in performance when more data is used is marginal, particularly with session sizes above ∼30%. This means that users can be fingerprinted during the early stages of their session without sacrificing much accuracy.

We show the performance of the discrimination and combined scenarios at varying thresholds ($\delta$) at three session sizes (10%, 30% and 100%) in Figure 5.2

and Figure 5.3 respectively. These figures highlight the importance of selecting the correct threshold to achieve optimal performance. In practice, selecting a well-performing threshold is a difficult task as access to the testing data is not available. It is possible to make approximate decisions about the threshold using training data only. Another possible solution is to use well-performing thresholds in similar domains, considering that the social media and image gallery tasks have comparable threshold values. However, investigating the optimal threshold selection method is beyond the scope of this thesis.

We believe this practical investigation of fingerprinting using touch interactions is an important step in uncovering the privacy implications of touch-based biometrics. The bottleneck of our models is in the discrimination scenario where user sessions are marked as new or returning. Further improvements, such as more sophisticated machine learning models or dynamic threshold selection might result in much better performance. The touch interaction distance we described in this chapter can be used in conjunction with other easily accessible features to achieve a more complete fingerprinting system. For instance, Bojinov et al. [95] achieve 8% accuracy using their fingerprinting method but the performance increases to 53% by including only a single extra feature to the model: the user agent.

## 5.5 Personal Information Leakage

The way individuals interact with touchscreen devices has been extensively studied for authentication purposes. Similarly, we believe that there are variations in the way groups of people behave on touchscreen devices. For instance, left-handed people hold the phone slightly differently from the way right-handed people do. This could result in scrolls occurring on one side of the screen more often than on the other. A machine learning algorithm can establish these differences and predict whether a session belongs to users from one group or another. For this reason, we conducted a series of experiments to predict personal information attributes based on these unique interaction characteristics.

**Table 5.2:** Personal information attributes considered for touch interactions inference. The binary classes and the number of users in each class are given. The number of features used in the image gallery task models is shown on the left and the social media task models on the right.

| Attribute | Class 0 | Class 1 | # Features |
|---|---|---|---|
| Gender | Male (187) | Female (146) | 130 \| 130 |
| Handedness | Right (295) | Left (38) | 30 \| 12 |
| Country | USA (234) | India (19) | 100 \| 10 |
| Age | $\leq$ 25 years (73) | $\geq$ 45 years (57) | 10 \| 35 |
| Height | $\leq$ 159cm (30) | $\geq$ 183cm (40) | 27 \| 50 |
| Weight | $\leq$ 50kg (57) | $\geq$ 91kg (43) | 10 \| 18 |

Effective personal information leakage systems can lead to major issues in terms of censorship, tracking, and discrimination of people. However, in some cases, they might have beneficial uses such as restricting certain content from children or improving the mobile user experience. For example, one positive use case of this technology is to request additional age verification to visitors of a gambling website where the touch-based model predicts that the user is younger than 18 years with a high probability. We decided to explore a number of different personal information attributes that might be revealed from the way users interact with their touchscreens: age, gender, dominant hand, country of origin, height, and weight. Furthermore, we evaluated a number of data selection methods and machine learning classifiers to establish the best-performing ones for this purpose.

### 5.5.1 Method

The dataset we used to evaluate the personal information leakage potential has been collected remotely, and the demographic information shared by participants has been self-reported. Due to this, we performed a pre-processing step to clean the data from outliers. Firstly, we removed all users who have reported an unreasonable height (less than 100cm or above 250cm), weight (less than 20kg and above 250kg), and age (less than 18 years and above 90 years). Users below 18 years of age are

not allowed on the platform used for remote collection. Furthermore, for the weight and height attributes, we only considered users within 2 standard deviations of the average. In terms of gender, we only investigated people identifying as males or females and removed the other 5 users. Although it is certainly possible for some of the users we pruned to have reported true values, we decided to minimise the risk of polluting the prediction data. This pre-processing step reduced the total number of users we investigated in this section from 470 to 333.

For ease of comparison and a better understanding of the potential for leakage of personal information from touch interactions, we treated each of the predictions as a binary classification problem. Using the data we have, predicting the gender and dominant hand of participants is already a binary classification problem. However, for the country of origin attribute, we divided the dataset into participants originating from the USA and India which were the first and second largest country groups respectively. Furthermore, we converted the continuous age, weight, and height attributes into binary classes. In order to achieve this split (e.g. younger and older users), we separated the classes into groups above and below one standard deviation of the mean in each category. The binary classes for each of the six personal information attributes are shown in Table 5.2. In addition, we show the number of users in each of the categories.

Since some of the 150 features we have extracted are likely not relevant for each of the attributes examined, we applied a feature selection algorithm to reduce the dimensionality of our data. This approach ensures better computational performance but can also improve the overall accuracy of the model by removing irrelevant features. Similar to the fingerprinting scenario, we used the ANOVA feature selection algorithm. After some preliminary experiments, we chose different values for the number of $k$ features selected in each attribute prediction scenario. These are shown in Table 5.2 for both social media and image gallery tasks. We fit the feature selection algorithm on training data only. It is worth noting that in the privacy leakage experiments, we do not use the additional phone model feature mentioned in Section 5.4.3. However, we have shown in Chapter 3 that phone models can be

identified by the touch-based interactions performed on them. In that sense, part of that phone model data is intrinsic to the other features in the dataset.

In order to fairly compare the performance of our models, we balance the two attribute classes such that there is an equal number of users in both. The users are split into 80% training and 20% testing groups, where again, we ensure the two classes are equally balanced. We only include user sessions in the testing set if sessions from the same user have not been used for training. This is critical since the model can learn the identities of users and correctly identify the particular group of users (e.g. males in our dataset) a session belongs to instead of learning the attribute in question (e.g. gender). We also normalise the features by subtracting the mean and dividing by the standard deviation of the training data.

We investigated three data selection approaches for predicting personal information based on touch interactions:

- **Single-stroke** - The personal information predictions are made using the features from a single stroke. This is the most challenging scenario where the least amount of information is available, and there is likely a large deviation in prediction performance across individual strokes. However, that is also the fastest way to make a decision about personal information attributes during a session.

- **Multi-stroke** - In this scenario, we use multiple single-stroke predictions to make a final decision. Similar to an aggregation step in touch-based authentication systems described in depth in Chapter 4, we use the mean of the individual stroke predictions to form a prediction. In these experiments, we use 10 consecutive strokes to make a decision. It takes an average of 7 seconds of swiping to collect this number of strokes and thus make a prediction.

- **Session** - This approach is similar to the fingerprinting feature extraction. We use the data from a whole session in order to reach a conclusion, and the features are averaged across the whole session as described in Section 5.4.3. Naturally, this is the method that requires the most amount of time before a

final decision can be made. It is possible to use a portion of the session to shorten the time needed, however, we do not examine this scenario as it will likely reduce the accuracy, as is the case with fingerprinting.

Furthermore, we examined the performance of our models using a number of machine learning classifiers: Support Vector Machine, Random Forest, and Neural Network. The three classifiers were chosen as they are the best performing for touch-based authentication as shown in Chapter 4 and are commonly found in related work [67, 184–187].

- **Support Vector Machine (SVM)** - We use an SVM with an RBF kernel with a 'scale' coefficient, and probability estimations enabled.

- **Random Forest (RF)** - Our implementation uses 100 estimators and has a maximum depth of 20. Probability estimations are enabled.

- **Neural Network (NN)** - The feed-forward network we use consists of two layers of sizes 150 and 75, respectively. We use a 'ReLU' activation function for the hidden layers and a 'Sigmoid' activation function for the output, which predicts probability between 0 and 1 for each of the binary outputs. There is batch-normalisation at each layer and a dropout (0.3) between the hidden layers. The optimiser is 'adam', and the loss function is a 'binary cross-entropy'. The network is trained with a batch size of 32 over 20 epochs.

The Support Vector Machine and Random Forest classifiers were implemented using the `scikit-learn` [173] machine learning library, and the Neural Networks were implemented using `Tensorflow` [174] with the `Keras` [175] API. All the experiments are repeated 10 times while randomising the groups of testing and training users at each iteration. The mean accuracy across the repetitions is reported.

**Table 5.3:** Results for personal information leakage from touch interactions. The attributes investigated are Gender (male or female), Dominant hand (left or right), Country of origin (USA or India), Age (≤25 or ≥45), Height (≤159cm or ≥183cm) and Weight (≤50kg or ≥91kg) Each experiment is repeated 10 times and the average of all iterations is given. The standard deviation is shown in parentheses.

| | | SVM | | | Random Forest | | | Neural Network | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Single | Multi | Session | Single | Multi | Session | Single | Multi | Session |
| Image Gallery Task | Gender | 59% (±3) | 62% (±4) | 62% (±6) | 59% (±3) | 61% (±4) | 63% (±5) | 59% (±3) | 61% (±3) | 63% (±5) |
| | Hand | 60% (±5) | 63% (±6) | 61% (±7) | 61% (±8) | 62% (±10) | 57% (±8) | 59% (±7) | 64% (±7) | 61% (±8) |
| | Country | 53% (±14) | 53% (±17) | 55% (±21) | 52% (±16) | 52% (±19) | 50% (±23) | 54% (±14) | 56% (±15) | 52% (±19) |
| | Age | 54% (±3) | 56% (±4) | 58% (±6) | 53% (±4) | 54% (±6) | 58% (±4) | 55% (±4) | 56% (±5) | 59% (±6) |
| | Height | 60% (±2) | 62% (±3) | 60% (±5) | 59% (±2) | 61% (±4) | 61% (±4) | 59% (±2) | 61% (±4) | 58% (±4) |
| | Weight | 56% (±6) | 57% (±6) | 56% (±5) | 54% (±3) | 56% (±6) | 55% (±5) | 55% (±4) | 56% (±5) | 57% (±5) |
| Social Medial Task | Gender | 57% (±3) | 59% (±4) | 58% (±4) | 56% (±4) | 58% (±5) | 59% (±6) | 56% (±3) | 59% (±4) | 58% (±4) |
| | Hand | 54% (±7) | 55% (±9) | 57% (±10) | 54% (±9) | 56% (±11) | 48% (±10) | 52% (±7) | 55% (±9) | 54% (±11) |
| | Country | 50% (±10) | 47% (±17) | 53% (±17) | 49% (±10) | 46% (±16) | 57% (±18) | 48% (±11) | 45% (±18) | 53% (±13) |
| | Age | 62% (±2) | 65% (±3) | 65% (±4) | 60% (±3) | 63% (±4) | 65% (±4) | 62% (±2) | 65% (±3) | 64% (±4) |
| | Height | 54% (±2) | 54% (±3) | 53% (±4) | 53% (±2) | 53% (±3) | 54% (±5) | 53% (±2) | 53% (±3) | 52% (±5) |
| | Weight | 55% (±3) | 58% (±4) | 55% (±5) | 54% (±3) | 56% (±4) | 56% (±4) | 55% (±3) | 58% (±4) | 55% (±4) |

### 5.5.2 Results

We introduce our results for personal information prediction based on touch interactions in Table 5.3. The results are presented across the machine learning, and data selection approaches on the image gallery and social media tasks. The baseline performance for each of these experiments is 50% as the output is binary, and this accuracy is achievable by guessing that all examples belong to one of the classes.

The models predicting the gender of users performed well on the image gallery task and consistently achieved more than 60% accuracy with a maximum of 62% using the multi-stroke approach. The same experiments resulted in ∼3 percentage points lower on the social media task. The model predicting the dominant hand of a user performed well on the image gallery task, achieving over 60% accuracy consistently. However, it does not perform comparably on the social media task, suggesting swiping behaviour is more distinguishable between left and right-handed

users. The model predicting the country of origin attribute fails to achieve any reasonable performance on both tasks with high standard deviation across iterations. This could be due to the small number of samples for one of the classes, but it is also possible that there are no intrinsic differences in touch behaviour between countries and cultures. The age group prediction models achieved reasonably high results on the social media task with up to 65% accuracy, but results are closer to ∼55% on the image gallery task. That is the opposite of the height prediction model, which performed well on the image gallery task but poorly on the social media task. The weight model is consistent across the two tasks but only performed slightly above the baseline at ∼55% on both tasks.

It is worth noting that some of the attributes we are trying to predict are related to each other. For instance, the height and weight of users are likely correlated, and both are likely correlated with the gender since females tend to be shorter and, therefore, lighter on average.

Overall, the single-stroke approach performed the worst, with an average accuracy of 56.7% across all modalities. We excluded the country of origin in this analysis as the results on this task were not meaningful. Averaging out each feature across the whole session performed better at 58.1%. We found that the best performing method was the multi-stroke with 58.6% accuracy on average. In general, the SVM model performs marginally better than the rest of the classifiers, with an average of 58.2% accuracy across all modalities (excluding the country of origin). The second best is the Neural Network with 57.8% accuracy and then the Random Forest with 57.4%. The differences are not large enough to strongly recommend using one model over another. However, the SVM model struggles with scaling to a large number of examples and might be undesirable for practical use. The attribute prediction models performed slightly better on the image gallery task, which mainly consists of swiping (left/right) behaviour with 58.7% accuracy on average across all modalities tested (excluding the country of origin). The social media task achieved 56.8% accuracy on average on the same tasks.

Since the multi-stroke scenario was the best-performing method in our experiments, we decided to test the approach at varying window sizes of strokes. The group sizes we considered were 5, 10, 15, 20, 30, and one, where we used the whole session data available. We use the Neural Network classifier for this comparison. The results of this experiment are shown in Figrue 5.4. Overall, using more strokes resulted in better performance and less variation of performance across multiple iterations. However, differences in performance are small and inconsistent across the modalities.

## 5.6 Discussion

Our results show the feasibility of touch interactions to be used as a method for tracking users and revealing their personal information. While some of the results in both the fingerprinting and personal information leakage scenarios are good, the immediate threat to the privacy of mobile users is limited. However, considering no permissions are needed to collect touch data, we believe the technology can be applied in conjunction with other methods to achieve much better performance. For instance, we can use additional available data such as system and hardware attributes for the fingerprinting scenario [92] and keystroke behaviour for the personal information leakage scenario [113].

As mentioned, the technology we describe in this chapter can be used for malicious and undesirable goals, such as discrimination, surveillance, and even identity theft. These can be manifested in many ways: banks assessing your creditworthiness, employers discriminating based on gender, and governments oppressing dissidents. However, it can also be used beneficially to personalise the experience of users, particularly if they knowingly consent to such use.

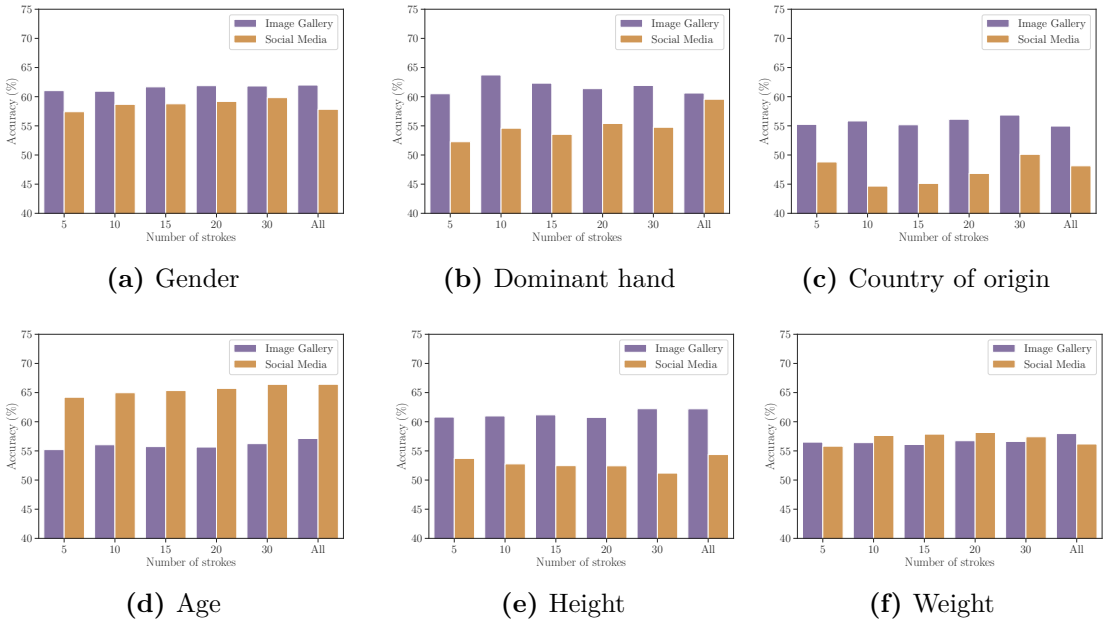
We found that, in general, personal information prediction studies in related work report much higher accuracy than the ones we achieved. This could be due to a variety of reasons, including dataset quality, experimental protocols, and machine learning processing. However, we want to highlight, once again, the issue of using data from the same user in both training and testing sets. This is problematic even if the test data is coming from different sessions than the training one. Often,

**Table 5.4:** Performance of personal information leakage models including data from the same user in the training and testing sets (i.e. no user separation between sets). The performance of a realistic evaluation is also given and the comparison is done on the Neural Network model using the multi-swipe approach on the whole session data.

| | Image Gallery Task | | Social Media Task | |
|---|---|---|---|---|
| Attribute | No user separation | Realistic | No user separation | Realistic |
| Gender | 85.3% | 62.0% | 81.4% | 57.8% |
| Handedness | 81.9% | 60.6% | 70.4% | 59.6% |
| Country | 81.0% | 55.0% | 63.6% | 48.1% |
| Age | 64.3% | 57.1% | 75.3% | 66.4% |
| Height | 74.6% | 62.2% | 75.7% | 54.4% |
| Weight | 65.2% | 58.0% | 69.9% | 56.2% |

in related work, it is not clear whether this division of data is maintained [67, 181, 184, 186]. In order to exemplify this pitfall, we conducted the experiments described in Section 5.5.1 without fulfilling the condition of separating users into training and testing groups. We used a Neural Network classifier with the multi-stroke approach on the whole session data. The results of the experiments are shown in Table 5.4 and are compared to the realistic evaluation method. The unrealistic approach produces higher results in all of the cases we tested, with an increase of between 7.2% and 23.6%.

The SVM approach to the fingerprinting problem resulted in much better performance than the vector distance approach. There might be computational performance concerns over creating single models for each user. However, since each model is relatively small, we found that the SVM approach was not computationally heavy when applied to the dataset we used. In fact, without formally analysing performance, we recorded the following times for model training and decision-making on a commercial off-the-shelf computer. Training all of the SVM user models took 1.1 seconds in total, and a decision about a session was made on average in 52ms. These are similar to the decision function approach where there is no training, however, a decision is made on average in 60ms.




**(a)** Gender **(b)** Dominant hand **(c)** Country of origin

**(d)** Age **(e)** Height **(f)** Weight

**Figure 5.4:** Performance of multi-stroke models on personal information leakage predictions when varying the number of strokes considered. The results are shown on the image gallery task with a Neural Network classifier.

It is also worth noting that the simulation framework described in Section 5.4.2 can be used in other fingerprinting approaches beyond touch interactions. It can be particularly useful for continuous feature values such as the ones coming from sensors.

### 5.6.1 Countermeasures

Countermeasures to the privacy issues described in this chapter are difficult to implement, considering the pervasiveness of the technology. We believe at the current stage of this investigation, the ways in which users can be protected from fingerprinting and personal information leakage is limited. Firstly, we briefly present some technical approaches that can be taken to minimise the threats but also discuss the issues relating to their use. Some of the methods we describe have also been applied to similar threats on the web.

- Permission requests - This approach requires users to accept a permission request before allowing the system to access and use specific sensors. It is often proposed as a countermeasure in related work based on mobile device sensors [96, 103]. However, asking for permissions is not possible in the

touch-based case as interacting with the touchscreen is a necessity for using the mobile device in the first place.

- Limiting sampling rate - Another approach could be to limit the touch sampling rate of the browser/application APIs. This might be able to reduce the accuracy of our models, however, it is also likely to reduce the smoothness of operation of the mobile device and hence have an impact on the user experience.

- Noise injection - This method entails introducing random variations into the collected touch data in order to make it more challenging to identify and track users by their touch behaviour. However, excessive noise can negatively impact the user experience by considerably reducing the accuracy of the authentication model. It is unknown how much noise is necessary to avoid fingerprinting. Further investigation is needed to determine how we can best apply the approach and to what extent it impacts usability. Moreover, it is possible that this strategy might not be effective against more sophisticated fingerprinting methods that can draw out delicate patterns from noisy data. For instance, an adversary might use a model which learns and extracts the noise pattern to improve performance.

- Disabling JavaScript - It might not be possible to collect touch interactions on a mobile browser without JavaScript. However, removing this core functionality of many websites would impact the user experience and completely prevent the use of some functionality. Furthermore, it would still not be possible to disable collection on mobile application APIs.

- Software detection and prevention - The browser, operating system or third-party extensions can check for malicious patterns of use in the touch APIs. However, data and behaviour can be obfuscated by website and application developers. This approach can become a race between security specialists and malicious actors.

Apart from technical solutions, it is also possible to prevent the use of these privacy attacks through other means. In recent times, individuals using the internet have become increasingly aware of the importance of protecting their personal information and privacy. The discovery that a large internet company is utilising the methods discussed in this chapter could result in significant damage to its reputation and could serve as a deterrent for other companies to avoid such strategies. Therefore, the potential negative impact on a business's reputation may prevent large players from utilising these techniques.

Furthermore, relatively new regulations such as The General Data Protection Regulation (GDPR) and The California Consumer Privacy Act (CCPA) are designed to give citizens more control over their personal data. Such initiatives regulate the collection, storage, and use of personal data of individuals and give them the right to know what personal information businesses collect about them. Under such laws, large companies would once again be deterred from using privacy invasive techniques as the ones described in this chapter without letting users know. That is because it will make them liable to large fines.

### 5.6.2 Limitations

While we present strong results in using touch interactions as a method for privacy invasion, there are a number of possible improvements to our work.

First of all, our investigation could benefit from collecting a larger dataset to ensure the validity of our results. This is particularly important for the fingerprinting section, where in practice, the number of users is an order of magnitude larger, and scalability might be an issue. Furthermore, it would be more realistic to also collect data on mobile browsers, which can be slightly different from the native touch API data we use. In fact, the limitations and differences in sampling imposed by the browser APIs themselves could be used for device fingerprinting.

The features we use in the fingerprinting scenario are based on averaging out individual stroke features over the whole session. This might not be the optimal approach for feature extraction, and better feature engineering might increase the

performance of the model. For instance, that can include methods suitable for reducing the dimensionality of the feature data, such as autoencoders or time-series analysis. Furthermore, using more sophisticated distance measures or machine learning algorithms as a dissimilarity function could yield better overall results. In particular, ones tailored for high-dimensional vector processing.

We have shown that threshold selection is an important part of the fingerprinting system, and finding an optimal threshold can be difficult and imprecise. We believe that further research and quantitative results using different approaches are needed.

The dataset we use has been collected remotely, and the personal information associated with the users has been self-reported, as mentioned in Section 5.5.1. Some of the personal information, such as date of birth and country of origin, can be considered too sensitive and personally identifiable. It is not unreasonable to assume that some of the participants have opted out to obscure their real personal information. However, users do not have an incentive to 'fake' their behaviour while using the application. That is why we excluded users from our experiments only in the personal information leakage section and not in the fingerprinting section of this chapter.

The usefulness of splitting the age, height, and weight attributes into binary classes is limited. Creating regression models or bucketing the values for multi-class classification would be preferable. Furthermore, the number of features selected for each attribute (i.e. gender, age, dominant hand) is based on preliminary experiments and can be somewhat arbitrary. The optimal feature count might differ between the single stroke and whole session scenarios. A more thorough analysis of feature selection in the personal information leakage experiments is needed.

Finally, the countermeasures we propose are somewhat superficial or impractical for implementation without understanding their impact. Further work is needed to develop better countermeasures to combat the issues described in this thesis and quantify their effectiveness. It is also unclear whether our results are applicable to larger mobile devices such as tablets, and more research is required to establish that.

## 5.7 Conclusion

In this chapter, we illustrated how the privacy of users could be compromised without their explicit knowledge by using touchscreen interactions.

In the first part of this chapter, we investigated the potential for touch-based systems to track user behaviour, hence addressing RQ3.1. We introduced a simulation system to measure the extent to which swipes and scrolls can be used to track mobile users online. Our results demonstrated that a user could be fingerprinted using touch interactions with high accuracy, and we showed that the technology could be used in conjunction with other features for additional performance.

In the second part of the chapter, as part of RQ3.2, we investigated how touchscreen interactions can be used to reveal personal information of users, such as their age, gender, dominant hand, country of origin, height, and weight. Our findings suggest that age, gender, dominant hand, and height can be consistently predicted with accuracies of over 60% on certain tasks. While these results may be low for some practical applications, this chapter presents one of the first investigations of this phenomenon. We believe that improvements in the underlying technology as well as integration with other privacy-sensitive signals can result in much more accurate personal information leakage systems.

Finally, we showed that imprecise evaluation methods could lead to an artificial increase in the performance of models by up to 23.6%. We briefly discussed potential countermeasures to the threats posed in this chapter and described the limitations of our work.

*A story has no beginning or end: arbitrarily one chooses that moment of experience from which to look back or from which to look ahead.*

— Graham Greene

# 6 Conclusion

## Contents


**6.1 Summary and Key Findings . . . . . 145**
**6.2 Directions for Future Work . . . . . 147**
**6.3 Final Remarks . . . . . 149**


This chapter concludes our work and presents our most significant results. We summarise the work done in the preceding chapters and lay out our key findings. Then we provide potential directions for future work based on our outcomes and offer some final thoughts in conclusion of the thesis.

## 6.1 Summary and Key Findings

In Chapter 1, we introduced the goal of this thesis — to advance the field of continuous touch-based biometrics by creating a more robust authentication system, establishing best practices for data collection, consolidating authentication techniques and understanding the related privacy concerns. Additionally, we set out a number of research questions which were concretely answered through a series of systematic experiments.

In Chapter 3, we established that it is possible to collect large-scale touch interactions data remotely, although differences persist between remote and in-lab

experiments. Using the dataset, we discovered six evaluation pitfalls (small sample size, phone model mixing, non-contiguous training data selection, attacker data in training, aggregation window size, dataset and code availability) and other issues which could lead to an overestimation of performance and are present across the majority of previous work on the topic. We provided a set of best practices for evaluating such systems to avoid issues and ease estimation of the real authentication performance of touch-based authentication systems.

In Chapter 4, we conducted a systematic literature review in order to understand the techniques used in continuous touch-based authentication systems. In addition to establishing the prevalence of features, classifiers, metrics and aggregation methods, we categorised these into common types. Using this information, we compared the performance of the most prevalent techniques in each category while keeping all other variables constant. We found that the best-performing feature set was the one proposed by [124], the best-performing classifier was a Neural Network, and the best aggregation method was the one using the *Mean* of each stroke prediction. However, based on our findings, we also introduced our own novel feature set based on 149 unique features as well as an ensemble classifier based on the outputs from an SVM, Neural Network and Random Forest. The novel techniques outperformed the state-of-the-art methods in their respective categories. These conclusions were also validated on three distinct publicly available datasets.

The overall best-performing model achieved using our most promising techniques and ensuring it is fairly evaluated achieved 4.8% EER using 16 consecutive strokes. Considering our efforts to correctly evaluate the system, we believe that it is possible to develop continuous touch-based biometric systems which enhance the security of users and applications.

In Chapter 5, we established that a well-performing continuous touch-based biometric system could lead to important privacy implications for the end user. Two potential issues were investigated in this thesis. Users can be fingerprinted and tracked without their permission by the way they interact with their phones. We found that we can achieve an accuracy of up to 71.4% when correctly matching

which user a session originates from or whether it is from an entirely new one. The accuracy is relatively stable even when we reduce the amount of data used to make the decision. Using the same stroke interaction data, we can also predict personal information such as gender, age, dominant hand, height and weight. We achieved accuracies between 60% and 70% on each of these tasks.

## 6.2 Directions for Future Work

In light of our results, we suggest a number of ideas for future work. These are divided into two parts: continuous touch-based authentication systems and the privacy concerns related to the biometric technology. We believe that pursuing these directions of research will build upon our findings and improve the research landscape.

### Authentication

In terms of continuous touch-based authentication, we believe that there is a need for further research in order to examine the practicalities of deploying the system to end users. In particular, it would be interesting to explore the feasibility of processing data in real time on the phone and making authentication decisions without the need for connectivity. The alternative would be to design a server-based architecture which is efficient and preserves the privacy of the users. Currently, the majority of research focuses on experiments concerning continuous touch-based authentication while using a single application. It would be beneficial to investigate whether the system can operate globally (i.e. on the OS level) when using the phone and what the corresponding performance would be. Other practical issues, such as battery usage and integration methods, should also be explored. Considering its current performance, it is also unclear whether the system can be used as a single security measure on a device or whether it will always remain as a second-factor or even third-factor authentication. Finally, we believe that there is a need for a longitudinal study to track the usability of the system in a practical setting. This will help us understand the experience from the perspective of the end users and might lead to further research ideas.

While optimising the EER might prove useful, particularly if there are large margins, we ultimately believe that the system performance in a lab setting the system achieves is sufficient for its practical applications. Hence we advise focusing on other, more practical research questions.

We believe that a lot of the ideas we presented in this thesis are applicable to other biometric systems as well. In particular, authentication using interactions in 3D space based on AR (Augmented Reality) and VR controller systems might be a great avenue for impactful research building on top of our work. A lot of the ideas described in this thesis should be applicable beyond what we have described to any hardware devices where continuous interaction is observed.

### Privacy Concerns

We provided some insights into the future work of our privacy study in Chapter 5 in the context of its limitations. To reiterate, we believe that there is a need for a larger dataset, ideally collected on mobile web browsers and ensuring participants report correct personal information data. Moreover, better feature extraction and threshold selection for this particular attack might result in higher accuracies than what we have achieved. A large amount of work remains in devising suitable countermeasures to the privacy concerns we described as well as ways to inform and educate the public about them.

Our study was one of the first to look into the privacy issues stemming from the use of touch-based biometric systems. However, it would not be surprising if there are other possible avenues for infringing on the privacy of users with similar techniques. More research is needed to investigate further privacy issues related to the technology. For instance, we believe that the fingerprinting method we propose can also be used for cross-browser and cross-device tracking as the behaviours of the users themselves become the fingerprint. This has been suggested in related work [179], however, further experimentation is needed to determine its validity.

We believe that combining multiple modalities for tracking or personal information leakage could also lead to much higher accuracy. Further work, including specific data collection and practical experimentation, is needed to explore these interactions.

The models we explored in Chapter 5 achieve substantially high accuracies that, in combination with other techniques, could warrant their deployment in practical tracking solutions on the web. It is compelling to devise ways to establish whether the technology has been used in the wild, including for how long it has been around and its prevalence on the most used websites and applications. Furthermore, such work should investigate to what extent users are impacted and how they can protect themselves from what has happened and how to avoid it in the future.

## 6.3 Final Remarks

The technology of touch-based biometrics has tremendous potential, and we believe that our thesis has made a significant contribution to its future development and to the advancement of the field of authentication beyond touch-based systems. We hope that by establishing best practices and consolidating the field, we can facilitate the integration of touch-based biometric technology into our devices, leading to a more usable and secure world. While touch-based biometrics can provide an extra layer of security, there are also certain privacy concerns that must be considered to ensure their safe and ethical use in the digital world.

In conclusion, touchscreen biometrics systems are a promising technology that has the potential to revolutionise the way we interact with our devices and secure our personal information. We hope that our thesis will pave the way for future research and development in the field and contribute to the integration of touch-based biometrics in everyday life.

*You do not rise to the level of your goals. You fall to the level of your systems.*

— James Clear

# References


[1] M. Frank et al. "Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication". In: *IEEE Transactions on Information Forensics and Security* 8.1 (2013), pp. 136–148.

[2] Lingjun Li, Xinxin Zhao, and Guoliang Xue. "Unobservable Re-authentication for Smartphones". In: *20th Annual Network and Distributed System Security Symposium, NDSS 2013, San Diego, California, USA, February 24-27, 2013*. The Internet Society, 2013.

[3] Sanka Rasnayaka and Terence Sim. "Who wants Continuous Authentication on Mobile Devices?" In: *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. 2018, pp. 1–9.

[4] Martin Georgiev et al. "Common Evaluation Pitfalls in Touch-Based Authentication Systems". In: *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*. ASIA CCS '22. Nagasaki, Japan: Association for Computing Machinery, 2022, pp. 1049–1063. URL: https://doi.org/10.1145/3488932.3517388.

[5] Martin Georgiev et al. *FETA: Fair Evaluation of Touch-based Authentication*. 2022. URL: https://arxiv.org/abs/2201.10606.

[6] Martin Georgiev, Simon Eberz, and Ivan Martinovic. "Techniques for Continuous Touch-Based Authentication". In: *International Conference on Information Security Practice and Experience*. Springer. 2022, pp. 409–431.

[7] Martin Georgiev, Simon Eberz, and Ivan Martinovic. "Fingerprinting and Personal Information Leakage from Touchscreen Interactions". In: *Proceedings of the 21st Workshop on Privacy in the Electronic Society*. 2022, pp. 145–157.

[8] L. Zhang-Kennedy, S. Chiasson, and P. van Oorschot. "Revisiting password rules: facilitating human management of passwords". In: *2016 APWG Symposium on Electronic Crime Research (eCrime)*. 2016, pp. 1–10.

[9] William Melicher et al. "Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks". In: *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 175–191. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/melicher.

[10] Markus Dürmuth et al. "OMEN: Faster Password Guessing Using an Ordered Markov Enumerator". In: *Engineering Secure Software and Systems*. Ed. by Frank Piessens, Juan Caballero, and Nataliia Bielova. Cham: Springer International Publishing, 2015, pp. 119–132.

[11] Florian Schaub, Ruben Deyhle, and Michael Weber. “Password Entry Usability and Shoulder Surfing Susceptibility on Different Smartphone Platforms”. In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*. MUM ’12. Ulm, Germany: Association for Computing Machinery, 2012. URL: https://doi.org/10.1145/2406367.2406384.

[12] Sanam Ghorbani Lyastani et al. “Better managed than memorized? Studying the Impact of Managers on Password Strength and Reuse”. In: *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, Aug. 2018, pp. 203–220. URL: https://www.usenix.org/conference/usenixsecurity18/presentation/lyastani.

[13] Kat Krol et al. “"They brought in the horrible key ring thing!" Analysing the Usability of Two-Factor Authentication in UK Online Banking.” In: *CoRR* abs/1501.04434 (2015). URL: http://dblp.uni-trier.de/db/journals/corr/corr1501.html#KrolPCS15.

[14] F. Aloul, S. Zahidi, and W. El-Hajj. “Two factor authentication using mobile phones”. In: *2009 IEEE/ACS International Conference on Computer Systems and Applications*. 2009, pp. 641–644.

[15] Xinyi Huang et al. “A generic framework for three-factor authentication: Preserving security and privacy in distributed systems”. In: *IEEE Transactions on Parallel and Distributed Systems* 22.8 (2010), pp. 1390–1397.

[16] Bruce Schneier. “Two-Factor Authentication: Too Little, Too Late”. In: *Commun. ACM* 48.4 (Apr. 2005), p. 136. URL: https://doi.org/10.1145/1053291.1053327.

[17] J.A. Unar, Woo Chaw Seng, and Almas Abbasi. “A review of biometric technology along with trends and prospects”. In: *Pattern Recognition* 47.8 (2014), pp. 2673–2688. URL: http://www.sciencedirect.com/science/article/pii/S003132031400034X.

[18] Shaxun Chen, Amit Pande, and Prasant Mohapatra. “Sensor-Assisted Facial Recognition: An Enhanced Biometric Authentication System for Smartphones”. In: *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*. MobiSys ’14. Bretton Woods, New Hampshire, USA: Association for Computing Machinery, 2014, pp. 109–122. URL: https://doi.org/10.1145/2594368.2594373.

[19] Mohammad Omar Derawi, Bian Yang, and Christoph Busch. “Fingerprint Recognition with Embedded Cameras on Mobile Phones”. In: *Security and Privacy in Mobile Information and Communication Systems*. Ed. by Ramjee Prasad et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 136–147.

[20] Apple Siri Team. “Personalized Hey Siri”. In: *Apple Machine Learning Journal* (2018).

[21] Shabab Bazrafkan, Shejin Thavalengal, and Peter Corcoran. “An end to end Deep Neural Network for iris segmentation in unconstrained scenarios”. In: *Neural Networks* 106 (2018), pp. 79–95. URL: http://www.sciencedirect.com/science/article/pii/S089360801830193X.

[22] John Chuang et al. "I think, therefore I am: Usability and security of authentication using brainwaves". In: *International conference on financial cryptography and data security*. Springer. 2013, pp. 1–16.

[23] Tyler Kaczmarek, Ercan Ozturk, and Gene Tsudik. "Assentication: user de-authentication and lunchtime attack mitigation with seated posture biometric". In: *International Conference on Applied Cryptography and Network Security*. Springer. 2018, pp. 616–633.

[24] Andrew Teoh Beng Jin, David Ngo Chek Ling, and Alwyn Goh. "Biohashing: two factor authentication featuring fingerprint data and tokenised random number". In: *Pattern Recognition* 37.11 (2004), pp. 2245–2255. URL: http://www.sciencedirect.com/science/article/pii/S0031320304001876.

[25] T. Boult. "Robust distance measures for face-recognition supporting revocable biometric tokens". In: *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*. 2006, pp. 560–566.

[26] M. O. Derawi et al. "Unobtrusive User-Authentication on Mobile Phones Using Biometric Gait Recognition". In: *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. 2010, pp. 306–311.

[27] A. Fazel and S. Chakrabartty. "An Overview of Statistical Pattern Recognition Techniques for Speaker Verification". In: *IEEE Circuits and Systems Magazine* 11.2 (2011), pp. 62–81.

[28] M. Abramson and D.W. Aha. "User authentication from web browsing behavior". In: *FLAIRS 2013 - Proceedings of the 26th International Florida Artificial Intelligence Research Society Conference* (Jan. 2013), pp. 268–273.

[29] K. S. Killourhy and R. A. Maxion. "Comparing anomaly-detection algorithms for keystroke dynamics". In: *2009 IEEE/IFIP International Conference on Dependable Systems Networks*. 2009, pp. 125–134.

[30] Zach Jorgensen and Ting Yu. "On Mouse Dynamics as a Behavioral Biometric for Authentication". In: *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. ASIACCS '11. Hong Kong, China: Association for Computing Machinery, 2011, pp. 476–482. URL: https://doi.org/10.1145/1966913.1966983.

[31] Ken Pfeuffer et al. "Behavioural Biometrics in VR: Identifying People from Body Motion and Relations in Virtual Reality". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–12. URL: https://doi.org/10.1145/3290605.3300340.

[32] Dirk Van Bruggen et al. "Modifying Smartphone User Locking Behavior". In: *Proceedings of the Ninth Symposium on Usable Privacy and Security*. SOUPS '13. Newcastle, United Kingdom: ACM, 2013, 10:1–10:14. URL: http://doi.acm.org/10.1145/2501604.2501614.

[33] Serge Egelman et al. "Are You Ready to Lock?" In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS '14. Scottsdale, Arizona, USA: ACM, 2014, pp. 750–761. URL: http://doi.acm.org/10.1145/2660267.2660273.

[34] Adam J. Aviv et al. “Smudge Attacks on Smartphone Touch Screens”. In: *Proceedings of the 4th USENIX Conference on Offensive Technologies*. WOOT’10. Washington, DC: USENIX Association, 2010, pp. 1–7.

[35] Yang Zhang et al. “Fingerprint Attack against Touch-Enabled Devices”. In: *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*. SPSM ’12. Raleigh, North Carolina, USA: Association for Computing Machinery, 2012, pp. 57–68. URL: https://doi.org/10.1145/2381934.2381947.

[36] P. Markert et al. “This PIN Can Be Easily Guessed: Analyzing the Security of Smartphone Unlock PINs”. In: *2020 IEEE Symposium on Security and Privacy (SP)*. 2020, pp. 286–303.

[37] E. Cheon et al. “Gesture Authentication for Smartphones: Evaluation of Gesture Password Selection Policies”. In: *2020 IEEE Symposium on Security and Privacy (SP)*. Los Alamitos, CA, USA: IEEE Computer Society, May 2020, pp. 249–267. URL: https://doi.ieeecomputersociety.org/10.1109/SP40000.2020.00034.

[38] N. Karimian, M. Tehranipoor, and D. Forte. “Non-fiducial PPG-based authentication for healthcare application”. In: *2017 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*. 2017, pp. 429–432.

[39] Giulio Lovisotto et al. “Seeing Red: PPG Biometrics Using Smartphone Cameras”. In: *IEEE 15th Computer Society Workshop on Biometrics*. IEEE. Seattle, Washington, June 2020.

[40] W. Meng et al. “Surveying the Development of Biometric User Authentication on Mobile Phones”. In: *IEEE Communications Surveys Tutorials* 17.3 (thirdquarter 2015), pp. 1268–1293.

[41] J. Galbally et al. “Evaluation of direct attacks to fingerprint verification systems”. In: *Telecommunication Systems* 47.3 (Aug. 2011), pp. 243–254. URL: https://doi.org/10.1007/s11235-010-9316-0.

[42] Raghavendra Ramachandra and Christoph Busch. “Presentation Attack Detection Methods for Face Recognition Systems: A Comprehensive Survey”. In: *ACM Comput. Surv.* 50.1 (Mar. 2017). URL: https://doi.org/10.1145/3038924.

[43] V. M. Patel et al. “Continuous User Authentication on Mobile Devices: Recent progress and remaining challenges”. In: *IEEE Signal Processing Magazine* 33.4 (2016), pp. 49–61.

[44] Markus Jakobsson et al. “Implicit Authentication for Mobile Devices”. In: *Proceedings of the 4th USENIX Conference on Hot Topics in Security*. HotSec’09. Montreal, Canada: USENIX Association, 2009, p. 9.

[45] Fudong Li et al. “Active authentication for mobile devices utilising behaviour profiling”. In: *International Journal of Information Security* 13.3 (June 2014), pp. 229–244. URL: https://doi.org/10.1007/s10207-013-0209-6.

[46] T. J. Neal, D. L. Woodard, and A. D. Striegel. “Mobile device application, Bluetooth, and Wi-Fi usage data as behavioral biometric traits”. In: *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. Sept. 2015, pp. 1–6.

[47] Y. Zhong, Y. Deng, and G. Meltzner. "Pace independent mobile gait biometrics". In: *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. 2015, pp. 1–8.

[48] J. Mantyjarvi et al. "Identifying users of portable devices from gait pattern with accelerometers". In: *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* Vol. 2. 2005, ii/973–ii/976 Vol. 2.

[49] M. E. Fathy, V. M. Patel, and R. Chellappa. "Face-based Active Authentication on mobile devices". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 1687–1691.

[50] P. Samangouei, V. M. Patel, and R. Chellappa. "Attribute-based continuous user authentication on mobile devices". In: *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. 2015, pp. 1–8.

[51] Oriana Riva et al. "Progressive Authentication: Deciding When to Authenticate on Mobile Phones". In: *21st USENIX Security Symposium (USENIX Security 12)*. Bellevue, WA: USENIX Association, Aug. 2012, pp. 301–316. URL: https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/riva.

[52] W. Shi et al. "SenGuard: Passive user identification on smartphones using multiple sensors". In: *2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2011, pp. 141–148.

[53] Francesco Bergadano, Daniele Gunetti, and Claudia Picardi. "User Authentication through Keystroke Dynamics". In: *ACM Trans. Inf. Syst. Secur.* 5.4 (Nov. 2002), pp. 367–397. URL: https://doi.org/10.1145/581271.581272.

[54] Hui Xu, Yangfan Zhou, and Michael R. Lyu. "Towards Continuous and Passive Authentication via Touch Biometrics: An Experimental Study on Smartphones". In: *Proceedings of the Tenth USENIX Conference on Usable Privacy and Security*. SOUPS '14. Menlo Park, CA: USENIX Association, 2014, pp. 187–198.

[55] Shatha J. Alghamdi and Lamiaa A. Elrefaei. "Dynamic Authentication of Smartphone Users Based on Touchscreen Gestures". In: *Arabian Journal for Science and Engineering* 43.2 (Feb. 2018), pp. 789–810. URL: https://doi.org/10.1007/s13369-017-2758-x.

[56] Cristiano Giuffrida et al. "I Sensed It Was You: Authenticating Mobile Users with Sensor-Enhanced Keystroke Dynamics". In: *Detection of Intrusions and Malware, and Vulnerability Assessment*. Ed. by Sven Dietrich. Cham: Springer International Publishing, 2014, pp. 92–111.

[57] D. Deb et al. "Actions Speak Louder Than (Pass)words: Passive Authentication of Smartphone* Users via Deep Temporal Features". In: *2019 International Conference on Biometrics (ICB)*. 2019.

[58] N. Zheng et al. "You Are How You Touch: User Verification on Smartphones via Tapping Behaviors". In: *2014 IEEE 22nd International Conference on Network Protocols*. 2014, pp. 221–232.

[59] Yunpeng Song, Zhongmin Cai, and Zhi-Li Zhang. "Multi-touch Authentication Using Hand Geometry and Behavioral Information". In: *2017 IEEE Symposium on Security and Privacy (SP)*. 2017, pp. 357–372.

[60] T. Feng et al. “Continuous mobile authentication using touchscreen gestures”. In: *2012 IEEE Conference on Technologies for Homeland Security (HST)*. Nov. 2012, pp. 451–456.

[61] Yuxin Meng et al. “Touch Gestures Based Biometric Authentication Scheme for Touchscreen Mobile Phones”. In: *Information Security and Cryptology*. Ed. by Mirosław Kutyłowski and Moti Yung. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 331–350.

[62] X. Zhao et al. “Mobile User Authentication Using Statistical Touch Dynamics Images”. In: *IEEE Transactions on Information Forensics and Security* 9.11 (2014), pp. 1780–1789.

[63] C. Wu et al. “ICAuth: Implicit and Continuous Authentication When the Screen Is Awake”. In: *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. 2019, pp. 1–6.

[64] U. Mahbub et al. “Active user authentication for smartphones: A challenge data set and benchmark results”. In: *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. 2016, pp. 1–8.

[65] D. Deb et al. “Actions Speak Louder Than (Pass)words: Passive Authentication of Smartphone* Users via Deep Temporal Features”. In: *2019 International Conference on Biometrics (ICB)*. 2019.

[66] Rahul Murmuria et al. “Continuous Authentication on Mobile Devices Using Power Consumption, Touch Gestures and Physical Movement of Users”. In: *Research in Attacks, Intrusions, and Defenses*. Ed. by Herbert Bos, Fabian Monrose, and Gregory Blanc. Cham: Springer International Publishing, 2015, pp. 405–424.

[67] Margit Antal, Zsolt Bokor, and László Zsolt Szabó. “Information revealed from scrolling interactions on mobile devices”. In: *Pattern Recognition Letters* 56 (2015), pp. 7–13. URL: http://www.sciencedirect.com/science/article/pii/S0167865515000355.

[68] Tao Feng et al. “TIPS: Context-Aware Implicit User Identification Using Touch Screen in Uncontrolled Environments”. In: *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*. HotMobile ’14. Santa Barbara, California: Association for Computing Machinery, 2014. URL: https://doi.org/10.1145/2565585.2565592.

[69] A. Serwadda, V. V. Phoha, and Z. Wang. “Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms”. In: *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. 2013, pp. 1–8.

[70] Daniel Buschek, Alexander De Luca, and Florian Alt. “Evaluating the Influence of Targets and Hand Postures on Touch-Based Behavioural Biometrics”. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI ’16. San Jose, California, USA: Association for Computing Machinery, 2016, pp. 1349–1361. URL: https://doi.org/10.1145/2858036.2858165.

[71] Yuji Watanabe and Liu Kun. “Long-Term Influence of User Identification Based on Touch Operation on Smart Phone”. In: *Procedia Comput. Sci.* 112.C (Sept. 2017), pp. 2529–2536. URL: https://doi.org/10.1016/j.procs.2017.08.196.

[72] Margit Antal and László Zsolt Szabó. "Biometric Authentication Based on Touchscreen Swipe Patterns". In: *Procedia Technology* 22 (2016). 9th International Conference Interdisciplinarity in Engineering, INTER-ENG 2015, 8-9 October 2015, Tirgu Mures, Romania, pp. 862–869. URL: http://www.sciencedirect.com/science/article/pii/S2212017316000621.

[73] J. Fierrez et al. "Benchmarking Touchscreen Biometrics for Mobile Authentication". In: *IEEE Transactions on Information Forensics and Security* 13.11 (2018), pp. 2720–2733.

[74] Simon Eberz et al. "Evaluating Behavioral Biometrics for Continuous Authentication: Challenges and Metrics". In: *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ASIA CCS '17. Abu Dhabi, United Arab Emirates: ACM, 2017, pp. 386–399. URL: http://doi.acm.org/10.1145/3052973.3053032.

[75] Shridatt Sugrim et al. "Robust Performance Metrics for Authentication Systems". In: *Network and Distributed Systems Security (NDSS) Symposium 2019* ().

[76] Cheng Bo et al. "SilentSense: Silent User Identification via Touch and Movement Behavioral Biometrics". In: *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*. MobiCom '13. Miami, Florida, USA: Association for Computing Machinery, 2013, pp. 187–190. URL: https://doi.org/10.1145/2500423.2504572.

[77] Abdul Serwadda and Vir V. Phoha. "When Kids' Toys Breach Mobile Phone Security". In: *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*. CCS '13. Berlin, Germany: ACM, 2013, pp. 599–610. URL: http://doi.acm.org/10.1145/2508859.2516659.

[78] Neil Zhenqiang Gong et al. "Forgery-Resistant Touch-based Authentication on Mobile Devices". In: *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ASIA CCS '16. Xi'an, China: ACM, 2016, pp. 499–510. URL: http://doi.acm.org/10.1145/2897845.2897908.

[79] Ada Lerner et al. "Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016". In: *25th USENIX Security Symposium (USENIX Security 16)*. 2016.

[80] Tomasz Bujlow et al. "A survey on web tracking: Mechanisms, implications, and defenses". In: *Proceedings of the IEEE* 105.8 (2017), pp. 1476–1510.

[81] Craig E Wills and Can Tatar. "Understanding what they do with what they know". In: *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*. 2012, pp. 13–18.

[82] Aniko Hannak et al. "Measuring personalization of web search". In: *Proceedings of the 22nd international conference on World Wide Web*. 2013, pp. 527–538.

[83] Thomas Vissers et al. "Crying wolf? on the price discrimination of online airline tickets". In: *7th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2014)*. 2014.

[84] Jakub Mikians et al. "Detecting price and search discrimination on the internet". In: *Proceedings of the 11th ACM workshop on hot topics in networks*. 2012, pp. 79–84.

[85] Pierre Laperdrix et al. “Browser fingerprinting: A survey”. In: *ACM Transactions on the Web (TWEB)* 14.2 (2020), pp. 1–33.

[86] Peter Eckersley. “How unique is your web browser?” In: *International Symposium on Privacy Enhancing Technologies Symposium*. Springer. 2010, pp. 1–18.

[87] Pierre Laperdrix, Walter Rudametkin, and Benoit Baudry. “Beauty and the beast: Diverting modern web browsers to build unique browser fingerprints”. In: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2016, pp. 878–894.

[88] Gunes Acar et al. “The web never forgets: Persistent tracking mechanisms in the wild”. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 2014, pp. 674–689.

[89] Steven Englehardt and Arvind Narayanan. “Online tracking: A 1-million-site measurement and analysis”. In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016, pp. 1388–1401.

[90] Tao Wang and Ian Goldberg. “Improved website fingerprinting on tor”. In: *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*. 2013, pp. 201–212.

[91] Tao Wang and Ian Goldberg. “On Realistically Attacking Tor with Website Fingerprinting.” In: *Proc. Priv. Enhancing Technol.* 2016.4 (2016), pp. 21–36.

[92] Thomas Hupperich et al. “On the robustness of mobile device fingerprinting: Can mobile users escape modern web-tracking mechanisms?” In: *Proceedings of the 31st Annual Computer Security Applications Conference*. 2015, pp. 191–200.

[93] Andreas Kurtz et al. “Fingerprinting Mobile Devices Using Personalized Configurations”. In: *Proc. Priv. Enhancing Technol.* 2016.1 (2016), pp. 4–19.

[94] Vimal K Khanna. “Remote fingerprinting of mobile phones”. In: *IEEE Wireless Communications* 22.6 (2015), pp. 106–113.

[95] Hristo Bojinov et al. “Mobile device identification via sensor fingerprinting”. In: *arXiv preprint arXiv:1408.1416* (2014).

[96] Jiexin Zhang, Alastair R Beresford, and Ian Sheret. “Sensorid: Sensor calibration fingerprinting for smartphones”. In: *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2019, pp. 638–655.

[97] Zhiju Yang, Rui Zhao, and Chuan Yue. “Effective Mobile Web User Fingerprinting via Motion Sensors”. In: *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE. 2018, pp. 1398–1405.

[98] Anupam Das, Nikita Borisov, and Matthew Caesar. “Tracking Mobile Web Users Through Motion Sensors: Attacks and Defenses.” In: *NDSS*. 2016.

[99] Anupam Das, Nikita Borisov, and Edward Chou. “Every Move You Make: Exploring Practical Issues in Smartphone Motion Sensor Fingerprinting and Countermeasures.” In: *Proc. Priv. Enhancing Technol.* 2018.1 (2018), pp. 88–108.

[100] Sara Amini et al. “Deepfp: A deep learning framework for user fingerprinting via mobile motion sensors”. In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 84–91.

[101] Gianmarco Baldini and Gary Steri. "A survey of techniques for the identification of mobile phones using the physical fingerprints of the built-in components". In: *IEEE Communications Surveys & Tutorials* 19.3 (2017), pp. 1761–1789.

[102] Gianmarco Baldini et al. "Identification of mobile phones using the built-in magnetometers stimulated by motion patterns". In: *Sensors* 17.4 (2017), p. 783.

[103] Nikolay Matyunin et al. "Magneticspy: Exploiting magnetometer in mobile devices for website and application fingerprinting". In: *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*. 2019, pp. 135–149.

[104] Antitza Dantcheva, Petros Elia, and Arun Ross. "What else does your biometric data reveal? A survey on soft biometrics". In: *IEEE Transactions on Information Forensics and Security* 11.3 (2015), pp. 441–467.

[105] Na Cheng et al. "Gender identification from e-mails". In: *2009 IEEE Symposium on Computational Intelligence and Data Mining*. IEEE. 2009, pp. 154–158.

[106] Na Cheng, Rajarathnam Chandramouli, and KP Subbalakshmi. "Author gender identification from text". In: *Digital investigation* 8.1 (2011), pp. 78–88.

[107] Francisco Rangel and Paolo Rosso. "Use of language and author profiling: Identification of gender and age". In: *Natural Language Processing and Cognitive Science* 177 (2013).

[108] John D Burger et al. *Discriminating gender on Twitter*. Tech. rep. MITRE CORP BEDFORD MA BEDFORD United States, 2011.

[109] James Marquardt et al. "Age and gender identification in social media". In: *Proceedings of CLEF 2014 Evaluation Labs* 1180 (2014), pp. 1129–1136.

[110] Eran Eidinger, Roee Enbar, and Tal Hassner. "Age and gender estimation of unfiltered faces". In: *IEEE Transactions on information forensics and security* 9.12 (2014), pp. 2170–2179.

[111] Antitza Dantcheva, Francois Bremond, and Piotr Bilinski. "Show me your face and I will tell you your height, weight and body mass index". In: *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE. 2018, pp. 3555–3560.

[112] Syed Zulkarnain Syed Idrus et al. "Soft biometrics for keystroke dynamics: Profiling individuals while typing passwords". In: *Computers & Security* 45 (2014), pp. 147–155.

[113] Attaullah Buriro et al. "Age, gender and operating-hand estimation on smart mobile devices". In: *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE. 2016, pp. 1–5.

[114] Yasin Uzun, Kemal Bicakci, and Yusuf Uzunay. "Could we distinguish child users from adults using keystroke dynamics?" In: *arXiv preprint arXiv:1511.05672* (2015).

[115] Avar Pentel. "Predicting age and gender by keystroke dynamics and mouse patterns". In: *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*. 2017, pp. 381–385.

[116] Joanne Hinds and Adam N Joinson. "What demographic attributes do our digital footprints reveal? A systematic review". In: *PloS one* 13.11 (2018), e0207112.

[117] Eric Malmi and Ingmar Weber. "You are what apps you use: Demographic prediction based on user's apps". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 10. 1. 2016, pp. 635–638.

[118] Tempestt J Neal and Damon L Woodard. "A gender-specific behavioral analysis of mobile device usage data". In: *2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)*. IEEE. 2018, pp. 1–8.

[119] Tempestt J Neal and Damon L Woodard. "You are not acting like yourself: A study on soft biometric classification, person identification, and mobile device use". In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* 1.2 (2019), pp. 109–122.

[120] Kaixiang Mo et al. "Report of task 3: your phone understands you". In: *Nokia mobile data challenge 2012 workshop, Newcastle, UK*. 2012, pp. 18–19.

[121] Vanessa Frias-Martinez, Enrique Frias-Martinez, and Nuria Oliver. "A gender-centric analysis of calling behavior in a developing economy using call detail records". In: *2010 AAAI Spring Symposium Series*. 2010.

[122] Tempestt J Neal and Damon L Woodard. "On the use of mobile calling patterns for soft biometric classification". In: *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE. 2018, pp. 1–6.

[123] R. Kumar, V. V. Phoha, and A. Serwadda. "Continuous authentication of smartphone users by fusing typing, swiping, and phone movement patterns". In: *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. 2016, pp. 1–8.

[124] Hui Xu, Yangfan Zhou, and Michael R. Lyu. "Towards Continuous and Passive Authentication via Touch Biometrics: An Experimental Study on Smartphones". In: *Proceedings of the Tenth USENIX Conference on Usable Privacy and Security*. SOUPS '14. Menlo Park, CA: USENIX Association, 2014, pp. 187–198.

[125] N. Zheng et al. "You Are How You Touch: User Verification on Smartphones via Tapping Behaviors". In: *2014 IEEE 22nd International Conference on Network Protocols*. 2014, pp. 221–232.

[126] T. Feng et al. "Continuous mobile authentication using touchscreen gestures". In: *2012 IEEE Conference on Technologies for Homeland Security (HST)*. 2012, pp. 451–456.

[127] Hassan Khan and Urs Hengartner. "Towards Application-Centric Implicit Authentication on Smartphones". In: *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*. HotMobile '14. Santa Barbara, California: Association for Computing Machinery, 2014. URL: https://doi.org/10.1145/2565585.2565590.

[128] Premkumar Saravanan et al. "LatentGesture: Active User Authentication through Background Touch Analysis". In: *Proceedings of the Second International Symposium of Chinese CHI*. Chinese CHI '14. Toronto, Ontario, Canada: Association for Computing Machinery, 2014, pp. 110–113. URL: https://doi.org/10.1145/2592235.2592252.

[129] H. Zhang et al. "Touch Gesture-Based Active User Authentication Using Dictionaries". In: *2015 IEEE Winter Conference on Applications of Computer Vision*. 2015, pp. 207–214.

[130] J. Nader et al. “Designing Touch-Based Hybrid Authentication Method for Smartphones”. In: *Procedia Computer Science* 70 (2015). Proceedings of the 4th International Conference on Eco-friendly Computing and Communication Systems, pp. 198–204. URL: http://www.sciencedirect.com/science/article/pii/S1877050915032366.

[131] V. Zaliva et al. “Passive user identification using sequential analysis of proximity information in touchscreen usage patterns”. In: *2015 Eighth International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*. 2015, pp. 161–166.

[132] M. Temper, S. Tjoa, and M. Kaiser. “Touch to Authenticate — Continuous Biometric Authentication on Mobile Devices”. In: *2015 1st International Conference on Software Security and Assurance (ICSSA)*. 2015, pp. 30–35.

[133] Rahul Murmuria et al. “Continuous Authentication on Mobile Devices Using Power Consumption, Touch Gestures and Physical Movement of Users”. In: *Research in Attacks, Intrusions, and Defenses*. Ed. by Herbert Bos, Fabian Monrose, and Gregory Blanc. Cham: Springer International Publishing, 2015, pp. 405–424.

[134] C. Shen et al. “Performance Analysis of Touch-Interaction Behavior for Active Smartphone Authentication”. In: *IEEE Transactions on Information Forensics and Security* 11.3 (2016), pp. 498–513.

[135] Margit Antal and László Zsolt Szabó. “Biometric Authentication Based on Touchscreen Swipe Patterns”. In: *Procedia Technology* 22 (2016). 9th International Conference Interdisciplinarity in Engineering, INTER-ENG 2015, 8-9 October 2015, Tirgu Mures, Romania, pp. 862–869. URL: http://www.sciencedirect.com/science/article/pii/S2212017316000621.

[136] U. Mahbub et al. “Active user authentication for smartphones: A challenge data set and benchmark results”. In: *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. 2016, pp. 1–8.

[137] Jamil Ahmad et al. “Analysis of interaction trace maps for active authentication on smart devices”. In: *Multimedia Tools and Applications* 76.3 (Feb. 2017), pp. 4069–4087. URL: https://doi.org/10.1007/s11042-016-3450-y.

[138] Xiao Wang et al. “Towards Continuous and Passive Authentication across Mobile Devices: An Empirical Study”. In: *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. WiSec ’17. Boston, Massachusetts: Association for Computing Machinery, 2017, pp. 35–45. URL: https://doi.org/10.1145/3098243.3098244.

[139] Shatha J. Alghamdi and Lamiaa A. Elrefaei. “Dynamic Authentication of Smartphone Users Based on Touchscreen Gestures”. In: *Arabian Journal for Science and Engineering* 43.2 (Feb. 2018), pp. 789–810. URL: https://doi.org/10.1007/s13369-017-2758-x.

[140] Weizhi Meng et al. “TouchWB: Touch behavioral user authentication based on web browsing on smartphones”. In: *Journal of Network and Computer Applications* 117 (2018), pp. 1–9. URL: https://www.sciencedirect.com/science/article/pii/S1084804518301723.

[141] Zahid Syed et al. "Touch gesture-based authentication on mobile devices: The effects of user posture, device size, configuration, and inter-session variability". In: *Journal of Systems and Software* 149 (2019), pp. 158–173. URL: https://www.sciencedirect.com/science/article/pii/S0164121218302516.

[142] Michail D. Papamichail et al. "BrainRun: A Behavioral Biometrics Dataset towards Continuous Implicit Authentication". In: *Data* 4.2 (2019). URL: https://www.mdpi.com/2306-5729/4/2/60.

[143] Yafang Yang et al. "BehaveSense: Continuous authentication for security-sensitive mobile apps using behavioral biometrics". In: *Ad Hoc Networks* 84 (2019), pp. 9–18. URL: https://www.sciencedirect.com/science/article/pii/S1570870518306899.

[144] Rodrigo Rocha, Davide Carneiro, and Paulo Novais. "Continuous authentication with a focus on explainability". In: *Neurocomputing* (2020). URL: http://www.sciencedirect.com/science/article/pii/S0925231220307323.

[145] Alejandro Acien et al. "BeCAPTCHA: Behavioral bot detection using touchscreen and mobile sensors benchmarked on HuMIdb". In: *Engineering Applications of Artificial Intelligence* 98 (2021), p. 104058. URL: https://www.sciencedirect.com/science/article/pii/S0952197620303274.

[146] Henry Turner, Simon Eberz, and Ivan Martinovic. "Recurring Turking: Conducting Daily Task Studies on Mechanical Turk". In: *CoRR* abs/2104.12675 (2021). arXiv: 2104.12675. URL: https://arxiv.org/abs/2104.12675.

[147] *NewsUSA: Copyright Free Content.* https://www.copyrightfreecontent.com/category/newsusa/. Accessed: 15 November 2021.

[148] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, 2014, pp. 740–755.

[149] David Moher et al. "Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement". In: *PLoS medicine* 6.7 (2009), e1000097.

[150] Tao Feng et al. "Continuous mobile authentication using touchscreen gestures". In: *2012 IEEE Conference on Technologies for Homeland Security (HST)*. 2012, pp. 451–456.

[151] Premkumar Saravanan et al. "LatentGesture: Active User Authentication through Background Touch Analysis". In: *Proceedings of the Second International Symposium of Chinese CHI*. Chinese CHI '14. Toronto, Ontario, Canada: Association for Computing Machinery, 2014, pp. 110–113. URL: https://doi.org/10.1145/2592235.2592252.

[152] Yuxin Meng, Duncan S. Wong, and Lam-For Kwok. "Design of Touch Dynamics Based User Authentication with an Adaptive Mechanism on Mobile Phones". In: *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. SAC '14. Gyeongju, Republic of Korea: Association for Computing Machinery, 2014, pp. 1680–1687. URL: https://doi.org/10.1145/2554850.2554931.

[153] Aditi Roy, Tzipora Halevi, and Nasir Memon. "An HMM-based behavior modeling approach for continuous mobile authentication". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, pp. 3789–3793.

[154] Xi Zhao et al. "Mobile User Authentication Using Statistical Touch Dynamics Images". In: *IEEE Transactions on Information Forensics and Security* 9.11 (2014), pp. 1780–1789.

[155] Xi Zhao, Tao Feng, and Weidong Shi. "Continuous mobile authentication using a novel Graphic Touch Gesture Feature". In: *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. 2013, pp. 1–6.

[156] Chao Shen et al. "Performance Analysis of Touch-Interaction Behavior for Active Smartphone Authentication". In: *IEEE Transactions on Information Forensics and Security* 11.3 (2016), pp. 498–513.

[157] Heng Zhang et al. "Touch Gesture-Based Active User Authentication Using Dictionaries". In: *2015 IEEE Winter Conference on Applications of Computer Vision*. 2015, pp. 207–214.

[158] Soumik Mondal and Patrick Bours. "Swipe gesture based Continuous Authentication for mobile devices". In: *2015 International Conference on Biometrics (ICB)*. 2015, pp. 458–465.

[159] Rahul Murmuria et al. "Continuous Authentication on Mobile Devices Using Power Consumption, Touch Gestures and Physical Movement of Users". In: *Research in Attacks, Intrusions, and Defenses*. Ed. by Herbert Bos, Fabian Monrose, and Gregory Blanc. Cham: Springer International Publishing, 2015, pp. 405–424.

[160] Rajesh Kumar, Vir V. Phoha, and Abdul Serwadda. "Continuous authentication of smartphone users by fusing typing, swiping, and phone movement patterns". In: *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. 2016, pp. 1–8.

[161] Xiao Wang et al. "Towards Continuous and Passive Authentication across Mobile Devices: An Empirical Study". In: *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. WiSec '17. Boston, Massachusetts: Association for Computing Machinery, 2017, pp. 35–45. URL: https://doi.org/10.1145/3098243.3098244.

[162] Rajesh Kumar, Partha Pratim Kundu, and Vir V. Phoha. "Continuous authentication using one-class classifiers and their fusion". In: *2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)*. 2018, pp. 1–8.

[163] Alexander I. Filippov, Artem V. Iuzbashev, and Alexey S. Kurnev. "User authentication via touch pattern recognition based on isolation forest". In: *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. 2018, pp. 1485–1489.

[164] Hasan Can Volaka et al. “Towards Continuous Authentication on Mobile Phones using Deep Learning Models”. In: *Procedia Computer Science* 155 (2019). The 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2019),The 14th International Conference on Future Networks and Communications (FNC-2019),The 9th International Conference on Sustainable Energy Information Technology, pp. 177–184. URL: https://www.sciencedirect.com/science/article/pii/S187705091930941X.

[165] Zahid Syed et al. “Touch gesture-based authentication on mobile devices: The effects of user posture, device size, configuration, and inter-session variability”. In: *Journal of Systems and Software* 149 (2019), pp. 158–173. URL: https://www.sciencedirect.com/science/article/pii/S0164121218302516.

[166] Saeed Samet et al. “TouchMetric: a machine learning based continuous authentication feature testing mobile application”. In: *International Journal of Information Technology* 11.4 (Dec. 2019), pp. 625–631. URL: https://doi.org/10.1007/s41870-019-00306-w.

[167] Mohammed Abuhamad et al. “AUToSen: Deep-Learning-Based Implicit Continuous Authentication Using Smartphone Sensors”. In: *IEEE Internet of Things Journal* 7.6 (2020), pp. 5008–5020.

[168] Rodrigo Rocha, Davide Carneiro, and Paulo Novais. “Continuous authentication with a focus on explainability”. In: *Neurocomputing* 423 (2021), pp. 697–702. URL: https://www.sciencedirect.com/science/article/pii/S0925231220307323.

[169] Özlem Durmaz Incel et al. “DAKOTA: Sensor and Touch Screen-Based Continuous Authentication on a Mobile Banking Application”. In: *IEEE Access* 9 (2021), pp. 38943–38960.

[170] Soumik Mondal and Patrick Bours. “A computational approach to the continuous authentication biometric system”. In: *Information Sciences* 304 (2015), pp. 28–53. URL: https://www.sciencedirect.com/science/article/pii/S0020025514011979.

[171] Michail D. Papamichail et al. “BrainRun: A Behavioral Biometrics Dataset towards Continuous Implicit Authentication”. In: *Data* 4.2 (2019). URL: https://www.mdpi.com/2306-5729/4/2/60.

[172] Alejandro Acien et al. “BeCAPTCHA: Bot Detection in Smartphone Interaction using Touchscreen Biometrics and Mobile Sensors”. In: *CoRR* abs/2005.13655 (2020). arXiv: 2005.13655. URL: https://arxiv.org/abs/2005.13655.

[173] Lars Buitinck et al. “API design for machine learning software: experiences from the scikit-learn project”. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 2013, pp. 108–122.

[174] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[175] François Chollet et al. *Keras*. https://keras.io. 2015.

[176] Ian H Witten et al. “Practical machine learning tools and techniques”. In: *DATA MINING*. Vol. 2. 2005, p. 4.

[177] Vishal M. Patel et al. "Continuous User Authentication on Mobile Devices: Recent progress and remaining challenges". In: *IEEE Signal Processing Magazine* 33.4 (2016), pp. 49–61.

[178] Pin Shen Teh et al. "A survey on touch dynamics authentication in mobile devices". In: *Computers & Security* 59 (2016), pp. 210–235. URL: `https://www.sciencedirect.com/science/article/pii/S0167404816300256`.

[179] Rahat Masood et al. "Touch and You're Trapp (ck) ed: Quantifying the Uniqueness of Touch Gestures for Tracking." In: *Proc. Priv. Enhancing Technol.* 2018.2 (2018), pp. 122–142.

[180] Chris Bevan and Danaë Stanton Fraser. "Different strokes for different folks? Revealing the physical characteristics of smartphone users from their swipe gestures". In: *International Journal of Human-Computer Studies* 88 (2016), pp. 51–61.

[181] Oscar Miguel-Hurtado et al. "Predicting sex as a soft-biometrics from device interaction swipe gestures". In: *Pattern Recognition Letters* 79 (2016), pp. 44–51.

[182] Ankita Jain and Vivek Kanhangad. "Gender recognition in smartphones using touchscreen gestures". In: *Pattern Recognition Letters* 125 (2019), pp. 604–611.

[183] Alejandro Acien et al. "Active detection of age groups based on touch interaction". In: *IET Biometrics* 8.1 (2019), pp. 101–108.

[184] Toan Nguyen, Aditi Roy, and Nasir Memon. "Kid on the phone! Toward automatic detection of children on mobile devices". In: *Computers & Security* 84 (2019), pp. 334–348.

[185] Yushi Cheng et al. "Identifying child users via touchscreen interactions". In: *ACM Transactions on Sensor Networks (TOSN)* 16.4 (2020), pp. 1–25.

[186] Storm P Davis, Alireza Ashayer, and Nasseh Tabrizi. "Predicting Sex and Age using Swipe-Gesture Data from a Mobile Device". In: *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*. IEEE. 2020, pp. 1136–1143.

[187] Baylea Denise Williams. *Tactile Demographics: Predicting Demographic Information Using Touch Data from Mobile Devices*. East Carolina University, 2021.

[188] Ting-Fang Yen et al. "Host Fingerprinting and Tracking on the Web: Privacy and Security Implications." In: *NDSS*. Vol. 62. 2012, p. 66.