# The Signature Verification Project: Authentication of Handwritten Signatures on Mobile Devices and Touchpads

Linus Claus Heimböck

MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

Mobile Computing

in Hagenberg

im November 2020

© Copyright 2020 Linus Claus Heimböck

This work is published under the conditions of the Creative Commons License *Attribution-NonCommercial-NoDerivatives 4.0 International* (CC BY-NC-ND 4.0)—see https://creativecommons.org/licenses/by-nc-nd/4.0/.

# Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere. This printed thesis is identical with the electronic version submitted.

Hagenberg, November 23, 2020

Linus Claus Heimböck

# Contents

**Declaration** iii

**Acknowledgements** vii

**Abstract** viii

**Kurzfassung** ix

**1 Introduction** 1
- 1.1 Handwritten Signatures in the Digital World . . . 1
- 1.2 Research Questions . . . 2
- 1.3 Thesis Organization . . . 3

**2 Personal Identification** 4
- 2.1 Methods of Digital Identification . . . 4
  - 2.1.1 Knowing Something . . . 5
  - 2.1.2 Being Something . . . 7
  - 2.1.3 Having Something . . . 8
  - 2.1.4 Doing Something . . . 8
- 2.2 Digital Handwritten Signatures . . . 9
  - 2.2.1 Static and Dynamic . . . 10
  - 2.2.2 Local and Global . . . 11
  - 2.2.3 Reference-Based and Model-Based . . . 12
  - 2.2.4 Writer-Dependent and Writer-Independent Verification . . . 12
  - 2.2.5 Skilled and Random Forgeries . . . 13

**3 Related Work** 14
- 3.1 Static Signatures . . . 14
- 3.2 Dynamic Signature Features . . . 15
  - 3.2.1 X-Y-Coordinates . . . 15
  - 3.2.2 Pressure . . . 16
  - 3.2.3 Pen-Based features . . . 17
  - 3.2.4 Movement-Based Features . . . 17
- 3.3 Dynamic Signature Verification . . . 18
  - 3.3.1 Pen-Tablet . . . 19
  - 3.3.2 Phone . . . 19

3.3.3 Touchpad . . . 28
3.4 Signature Datasets . . . 29
3.4.1 Existing Datasets . . . 29
3.4.2 Signature Recording . . . 30
3.5 Usability . . . 31
3.6 Summary . . . 32

**4 Dataset** **34**
4.1 Mobile Dataset . . . 34
4.1.1 Features . . . 35
4.1.2 Sampling . . . 36
4.1.3 Processing . . . 37
4.1.4 Setting . . . 37
4.1.5 Participant Demographics . . . 42
4.1.6 Forgeries . . . 43
4.2 Touchpad Dataset . . . 46
4.2.1 Features . . . 46
4.2.2 Sampling and Processing . . . 47
4.2.3 Setting . . . 48
4.2.4 Participant Demographics . . . 50
4.2.5 Forgeries . . . 51
4.3 Usability . . . 54
4.4 Dataset Structure . . . 54
4.5 Additional data . . . 56

**5 Approach** **57**
5.1 Signature Processing . . . 59
5.2 Local Feature Extraction . . . 61
5.3 Global Feature Extraction . . . 62
5.4 Resampling and Normalization . . . 63
5.5 Classification . . . 64
5.5.1 Dynamic Time Warping . . . 65
5.5.2 Multi-Dimensional Dynamic Time Warping . . . 67
5.5.3 Machine Learning . . . 68
5.6 Model Updates . . . 71

**6 Implementation** **72**
6.1 Recording . . . 73
6.1.1 Time System Conversions . . . 73
6.1.2 Alignment . . . 74
6.1.3 Interpolation . . . 76
6.1.4 Sampling . . . 76
6.2 Processing . . . 79
6.3 Classification . . . 80
6.3.1 Multi-Dimensional Dynamic Time Warping . . . 80
6.3.2 Classification and EER Estimation . . . 81

**7 Results** **83**
7.1 Scenarios . . . 83
7.2 Enrollment Strategies . . . 85
7.3 Touchpad Signature Verification . . . 86
7.4 Mobile Signature Verification . . . 90
7.4.1 XY-Performance . . . 90
7.4.2 Inclusion of Movement Data . . . 93
7.4.3 Individual Movement Features . . . 98
7.4.4 Touchsize . . . 103
7.4.5 Interoperability . . . 105
7.5 Multi-Dimensional Dynamic Time Warping . . . 106
7.6 Global Models . . . 107
7.6.1 N-Class Classification . . . 108
7.6.2 Subject-Others Classification . . . 109
7.6.3 Forgery Detection . . . 114
7.7 Usability . . . 115

**8 Conclusion** **121**
8.1 Summary . . . 121
8.2 Contributions . . . 122
8.3 Answers to the Research Questions . . . 123
8.4 Future Work . . . 125

**A Participant Instructions (English)** **126**
A.1 Participation . . . 126
A.2 Illustrations . . . 128

**B Participant Instructions (German)** **130**
B.1 Teilnahme . . . 130
B.2 Illustrationen . . . 132

**C Usability Questionnaires (English)** **133**
C.1 Mobile Questionnaire . . . 133
C.2 Touchpad Questionnaire . . . 135

**D Usability Questionnaires (German)** **137**
D.1 Fragebogen Smartphone . . . 137
D.2 Fragebogen Touchpad . . . 139

**References** **141**
Literature . . . 141
Online sources . . . 152

# Acknowledgements

After now nearly a whole two years of working on this thesis, these are the final words to write. During all this time, I have had the chance to meet, work, and study with many amazing people, sometimes under the craziest of circumstances, and I am grateful that I can call many of them friends. They all helped and supported me in my research in their own way, and I would like to take the time to say thank you.

First and foremost, I would like to thank Rainhard Findling for his continued advice and support. With his knowledge and valuable feedback, he has helped me organize my many messy ideas into a coherent thesis. He has given me the freedom to explore when I wanted to, offered his help and expertise when I needed it, and has gone to great lengths to assist me in any way he could. He was the best advisor anyone could wish for, and I can truly say I've learned a lot. But perhaps even more importantly than that, he has been encouraging and understanding, gave me confidence in my own work, and our meetings and discussions sparked not only motivation and excitement to continue, but also deepened my interest in and understanding of science.

I can also only thank the many people who took part in the recording of our database. Nikolas Pacik, who as a my beta-tester and more importantly, friend, found all the bugs I've missed. To the fellow students at our Christmas party in Idun who trusted me, a stranger to some of them, with their signatures to build my first preliminary dataset. To my friends and family from home and abroad, who each took the time to sign and made it grow much bigger than I imagined it to be. And to the forgers, Berta, Helga, Johannes, Johannes, and Martin, who for hours and hours on end imitated hundreds of signatures so I could continue my work. To all of you, it was not only a great help, but also some much needed fun in the calm between the waves of a pandemic.

Last, but most certainly not least, cordial thanks go to the many people who have been there for me not only in my studies, but throughout my life. To you, Mama, for the parcels with gingerbread that always arrive when I need them the most, for your unrelenting support, and for everything else. To you, Däta, for your unconventional thinking, for our times together, and for your help whenever I ask. To my brother, Martin Heimböck, for our many discussions, and for much more. To my aunt, Helga Türtscher, for being the best Gota in the world. And to Johannes Hartmann, for almost 25 years of friendship. Finally, I would like to thank my friends, old and new, for always having my back, for all our experiences together, and for our Calls, Zooms, Skypes and Facetimes, without which the lockdowns just would not have been the same. You all have helped me focus on my work over the past two years by making the times outside as special as they were. It's been a ride, and I wouldn't want to miss a second of it.

# Abstract

A person's handwritten signature is a unique behavioral biometric trait that has been used for centuries as a means of proving identity and consent in many societies around the world. With the rise of touch-enabled electronic devices has come a transition towards signing digitally on a screen, and automated authentication has become possible as a result. The field of dynamic signature verification is concerned with determining authenticity by comparing time series recordings of the signature geometry and additional sensors between new samples and a genuine template.

This thesis investigates the verification of finger-drawn handwritten signatures on widely available mobile devices. In particular, it focuses on touchpads as a new platform for digital signing, and on the inclusion of movement features as additional biometric information for the authentication on smartphones. Those features are potentially valuable against targeted attacks, because they are harder to imitate than the signature image that can be seen and practiced well. With our research, we also introduce our database, which, to the best our our knowledge, is the largest and most comprehensive mobile signature database in the world. It comprises of more than 10,400 signatures by 73 subjects, and 2 biometric modalities from 7 different physical or software-based sensors are recorded. The scientific recording tools for touchpads and mobile phones we provide alongside allow for easy extension of the database in future work. For classification, we developed Multi-Dimensional Dynamic Time Warping (MD-DTW) as a variation of the commonly used Dynamic Time Warping algorithm. It warps all dimensions of a signature at once with the same warping path to determine the distance between two samples, instead of distorting them independently and combining the results.

Our evaluation shows that touchpads and mobile phones perform equally well under the same conditions, at about 5.75% EER against skilled forgeries. However, mobile phones have the potential to perform better in when used in different positions and when using additional features. We find that the inclusion of movement information results in improvements of 0.6% to 2% EER over only image-based signature data. The best average result under ideal conditions on combined data against skilled forgeries is at 2.59% EER, the best individual result at 1.79% EER. Our usability studies on both device classes show that while most users prefer signing on a mobile phone, 25% find the touchpad more comfortable, and an additional 13% did not have a preference. Over 55% reported that they would be likely to use touchpad signing in their daily life for signing digital documents. Overall, we find that movement data is valuable as additional biometric information for verification, and that the touchpad is a viable platform for digital signing.

# Kurzfassung

Die handschriftliche Unterschrift einer Person ist ein einzigartiges verhaltensbiometrisches Merkmal und wird weltweit als Identitätsnachweis und Zeichen der Zustimmung genutzt. Mit dem Aufstieg elektronischer Geräte mit Touchscreen geht ein Übergang zum digitalen Signieren einher, wodurch auch die automatisierte Authentifizierung der Unterschriften ermöglicht wird. Das Feld der dynamischen Unterschriften-Verifikation beschäftigt sich mit der Echtheitsprüfung handschriftlicher digitaler Signaturen anhand verschiedener Merkmale, die als Zeitserien zur Verfügung stehen.

Die vorliegende Arbeit untersucht die Verifizierung von Unterschriften mit dem Finger auf Mobilgeräten. Der Fokus liegt dabei auf Touchpads als neue Plattform für digitales Signieren, und auf der Verwendung von Bewegungsdaten als zusätzliche biometrische Merkmale für die Verifizierung auf Smartphones. Diese sind potentiell wertvoll für die Erkennung gezielter Fälschungen, weil sie nur schwer durch Angreifer geübt und imitiert werden können. Im Rahmen unserer Forschung stellen wir unsere Datenbank vor, die nach unserem Wissen größte und umfangreichste Datenbank mobiler Unterschriften weltweit. Sie besteht aus über 10,400 Unterschriften von 73 Teilnehmern und umfasst 2 biometrische Merkmale, die mit 7 physischen oder software-basierten Sensoren aufgenommen wurden. Zusätzlich stellen wir wissenschaftliche Aufnahme-Anwendungen bereit, um die Erweiterung unserer und die Aufnahme neuer Datenbanken zu vereinfachen. Zur Klassifizierung haben wir den Multi-Dimensional Dynamic Time Warping (MD-DTW)-Algorithmus entwickelt. Dieser ist eine Variante des häufig verwendeten DTW-Algorithmus. Er verschiebt alle Dimensionen einer Unterschrift auf einmal und findet einen einheitlichen Pfad mit minimaler Distanz zur Vorlage, anstatt jede Dimension unabhängig zu verzerren und die Ergebnisse zu kombinieren.

Unsere Evaluierung zeigt, dass Touchpads und Smartphones unter gleichen Bedingungen gleich gute Ergebnisse liefern. Diese liegen bei etwa 5.75% EER gegen gezielte Fälschungen. Jedoch können Smartphones in anderen Positionen und mit zusätzlichen Merkmalen bessere Resultate erzielen. Die Nutzung von Bewegungsdaten in der Verifikation ermöglicht Verbesserungen von etwa 0.6% bis 2% EER. Das beste durchschnittliche Ergebnis gegen gezielte Fälschungen liegt bei 2.59% EER, das beste Einzelergebnis bei 1.79% EER. Unsere Nutzbarkeits-Studien auf beiden Geräteklassen zeigen, dass die meisten Nutzer das Unterschreiben auf dem Smartphone bevorzugen. Etwa 25% der Teilnehmer empfinden jedoch das Touchpad als angenehmer, und weitere 13% haben keine Präferenz. Über 55% gaben an, dass sie das digitale Signieren auf Touchpads im Alltag verwenden würden. Zusammenfassend können wir aufgrund unserer Ergebnisse sagen, dass Bewegungsdaten wertvolle bimometrische Merkmale zur Verifikation sind, und dass Touchpads eine geeignete Platform für digitales Unterschreiben darstellen.

# Chapter 1

# Introduction

Signatures are an important cultural technique throughout the world. In almost all languages, alphabets, and societies, they are accepted as proof of identity and consent of an individual, and many of a person's most important decisions in life are confirmed by signing their name under a document. Considering their importance, it comes as no surprise that a whole profession has formed where highly skilled forensic handwriting experts try to judge the authenticity of signatures. Especially in disputed cases, they are called upon to determine whether a handwritten signature was drawn by the claimed signer. To do so, the questionable samples are compared to genuine samples of the supposed author by looking at various signature properties, many of which are not apparent at first. Those can be as simple as the number of strokes a writer set, or as complex as the speed and pressure that was applied while signing. In daily life, signatures are rarely verified in such a way.

## 1.1 Handwritten Signatures in the Digital World

Nowadays, information is often only exchanged electronically, as computerized workflows are both faster and easier than handling paperwork. However, as soon as a proof of agreement is required, things become complicated. Documents have to be printed only to be able to sign, handwritten signatures are scanned or photographed and added as images, or cryptographic signatures are used. The latter are highly secure, but often complicated to set up, incompatible with many established processes, and time consuming to include. For many people, they are also hard to understand and use. Businesses such as banks and delivery services therefore often resort to signing tablets for confirmation or transaction approval, where the handwritten signature is drawn with a digitized pen on a touch surface. In such an electronic signing process, the same information forensic handwriting experts use could be obtained from the recording device.

The field of behavioral biometrics attempts to use unique patterns in a person's conscious or unconscious behavior for identification. Handwritten signatures, as a product of a learned action, are one area of application. There is already a large amount of research, as well as commercial systems, that investigate and implement different schemes for automated authentication of digital signatures. They rely not only on the signature image, but use features such as pressure, speed, and more to improve classification ac-

curacy. Even if their error rates do not yet reach the high bar of performance set by physiological biometrics such as fingerprints, it is still a strict improvement over often not having any verification at all. With the ubiquity of touch-enabled mobile devices, which have become an integral part of our modern society, the application of such systems is no longer limited to banks or delivery services. Scenarios from easy and verifiable signing of documents at home to unlocking devices with a signature become possible, with no need for special hardware anywhere in the process.

In our research, we work on the authentication of signatures on mobile devices the users already own. In particular, we consider two device categories, smartphones and notebooks. Smartphones are well suited for signing with their large, high quality touchscreens and wide availability, and they have already been shown to be a viable platform in related work. Notebooks are also very widespread and frequently used for productivity. They come with integrated touchpads as pointing devices, but those have received little attention as signing surfaces so far. With both devices, we do not rely on any specific equipment, such as digitized pens, that might not be available to some people, and therefore exclusively focus on finger-drawn signatures.

Primarily, we are concerned with the task of verification, where the objective is to determine whether or not a signature is genuine. However, we also touch on the complementary subject of recognition, with the aim to identify who is signing from a set of known subjects. Our goal is to require only a few signatures samples of a person to be able to do so. To achieve it, we rely on different techniques in the realm of machine learning, and include different kinds of features to improve performance. On mobile phones, we investigate the use of unconscious phone movements that occur while signing as additional biometric information for verification. We combine movement information obtained from various embedded sensors that are common in most smartphones and use it for classification, both alone and in combination with different screen-based features of a signature. On notebooks, we focus on the overall verification performance, special characteristics such as the indirect visual feedback, and the comparison to our results on mobile phones. As to the best of our knowledge, ours is the first research on the authentication of signatures drawn on touchpads. We also discuss their usability for digital signing and attempt to assess their viability as a platform.

## 1.2 Research Questions

The work in this thesis is organized around the following research questions that are answered in chapter 8.3:

- How does the verification performance of signatures drawn with a finger on an integrated notebook touchpad compare to signatures drawn on a smartphone screen?
- Does the combination of image-based and movement-based dynamic features of a signature drawn on a mobile device improve verification performance over only image-based or only movement-based systems, and how do the individual features contribute to the final performance?
- How is the usability of signing with a finger on an integrated notebook touchpad, especially with regard to the indirect visual feedback and the narrow shape of the touchpad, and how does it compare to signing on a mobile phone screen?

To answer those research questions, this thesis investigates new ways of authenticating handwritten digital signatures within the realm of behavioral biometrics, introduces a database of more than 10,400 real mobile signatures, and presents some of the first scientific research on the application of dynamic signature verification on touchpads.

In addition to the research questions stated above, this work also contributes to or answers further questions that have been asked in related work:

- It explores the size of the finger contact area on screen (touchsize) as a potential additional screen-based feature, something that has been suggested as a potential substitute for pressure [122] and not been considered so far [36, 64].
- It studies the interoperability of signature templates and individual signature features between devices [64].
- It considers the influence of different signing positions and enrollment scenarios, which have not been evaluated in time-series based signature verification yet.

Finally, it provides ready-to-use, open-source applications for scientific signature recording on two platforms, mobile and touchpad, that allow for easy extension of our database or the creation of entirely new ones to facilitate further research.

## 1.3 Thesis Organization

The rest of this thesis is organized in two parts. Part I focuses on the background, related work, and current state of the art of signature verification. Part II contains our contributions, including the signature database, our approach and classification algorithm, details on the recording tools, and our evaluation results and findings. Chapter 2 starts with an introduction into the field of personal identification and dynamic signatures. It describes the different approaches to verification and the possible types of attacks on handwritten signatures. Chapter 3 discusses the previous work on mobile signature verification. It provides information about different sets of possible features, presents various approaches to verification, and reviews the current challenges in the field.

Part II focuses on our own contribution. We give an overview over our dataset in chapter 4. We provide details on the recording process, the usability study design, and technical properties of our measurements. In chapter 5, we present our own approach to signature verification. Following the three stages of our processing and classification chain, we first describe the processing of the individual sensor measurements and their aggregation into one signature file on device. Then, we discuss the preprocessing and feature extraction of the individual signatures. Finally, we go into depth on our own classification algorithm, Multi-Dimensional Dynamic Time Warping (MD-DTW). Chapter 6 continues with information on the algorithmic implementation of important components of our processing chain. Special focus lies on the accuracy of the recorded signatures as a prerequisite for successful classification. Our evaluation results are discussed in chapter 7. To answer our research questions, we start by exploring the viability of signature verification on notebook touchpads. We then compare those results with the same setting on a mobile device, before going into detail on improving mobile performance by including additional movement features. Having considered the technical aspects, we round up our results by reviewing both platforms from a usability perspective. Lastly, in chapter 8, we conclude and give an outlook on future work.

# Chapter 2

# Personal Identification

Personal identification - the identification of a person based on some form of information - is and always has been an essential part of our society. Humans do it unconsciously all the time. Whenever we see another person, we compare biometric traits such as face, movement or voice patterns to a database of known entities stored in our brain, and behave accordingly towards that person based on the result. With the introduction of identity documents such as passports, this process has been formalized. The issuing agency (e.g. a national state) acts as a common source of trust that guarantees the validity of the information and removes the requirement of recognizing other persons oneself to know who they are. Alongside, handwritten signatures have established themselves all over the world as permanent tokens of identity and consent that can be easily drawn on documents. One example of their importance is the signature template that is persisted in every persons passport for reference and comparison. Through the introduction of electronic passports with machine-readable sections, the traditional method of personal identification has even entered the digital world.

With the rise of computers and the internet, digital identification has become ubiquitous. Many smart devices we use and every account we have requires us to identify ourselves. Much like humans, these systems perform comparisons of identifiers, such as passwords or fingerprints, against known references. Depending on whether there is a match with an entry in a login database or a biometric template, they do or do not recognize us and choose an appropriate response. Online, the advent of authentication providers such as "Log in with Google" or "Log in with Facebook" can be seen as an equivalent to the introduction of passports. There, a trusted third party verifies the credentials of a user and provides their digital identity to others. Cryptographic signing, based on certificates ("signature templates") the validity of which can be guaranteed by a common source of trust, even serves as a legally binding equivalent to handwritten signatures. All of this requires some form of authentication, and a wide variety of methods have been developed for it. Some of them are described below.

## 2.1 Methods of Digital Identification

The methods of digital user identification are usually grouped by the way they infer the users identity into three groups [5, 63, 143]: the knowledge of some shared secret [72], the

recognition of a unique body feature (physiological biometrics) [17], or the possession of some object that is linked to the person [155]. The concept of vouching for someone ("someone you know") has been proposed as an additional option in [26] as a form of emergency authentication. A fourth group - the exhibition of a certain behavioral pattern (behavioral biometrics) - has been introduced as an additional way of authentication, but is usually grouped together with physiological biometrics under the term biometric authentication as one of the three factors [155].

The scenarios in which these methods are applied can also be subdivided into two groups of use cases. Both try to solve the same problem: To confirm an identity based on what is provided. Most commonly, digital identification is needed in the context of identification and access, such as for unlocking a phone, logging in to an account, or opening the door of a house or a car. Sometimes, though, users come across the second type of application as well - agreement and confirmation, where some form of permanent, identifiable mark is needed that proves agreement over time. Examples would be confirming the reception of a parcel to the postman, digitally signing a document, or agreeing to a contract. These cases can be realized by either using the method of authentication directly, as with drawing a handwritten signature on screen, or by using these methods of authentication indirectly in combination with cryptographic signatures.

### 2.1.1 Knowing Something

Knowledge-based authentication is the most widely employed method of authentication today. It comprises all systems where the user has to prove knowledge of some secret information to gain access to a resource. Common examples of this on mobile devices are unlock patterns, PIN-codes and passwords, while online, passwords are used almost exclusively. Not as well known are picture passwords (such as picture gesture authentication, Windows [67, 152, 153]), where the user draws a pattern over a favorite image, or knock codes (for example in LG smartphones [75]), where a phone is unlocked by knocking with the finger on screen in a specific pattern.

Conceptually, the common knowledge-based mechanisms are secure, as the number of possible combinations rises exponentially with the number of digits in the passcode. For an 8-character password, which is often the minimal required length, with a character set of 80 possible symbols, there are already $1.68 \cdot 10^{15}$ theoretically possible combinations or an entropy of $log(1.68 \cdot 10^{15})/log(2) = 50.6$ bit. The same exponential growth applies to PIN-codes, where a character set of 10 possible characters (0-9) results in $10^4$ possible combinations (entropy of 13.3 bit) for a common 4-digit code. While this is still a low number, a PINs security comes from the small number of possible tries and from it being local on the device, requiring physical access. Unlock patterns, in theory, are also relatively safe, although there are far less possible combinations than for a PIN code with the same number of points. On the commonly used nine-dot grid, a total of 389112 unique patterns can be drawn [86] when following the Android rules that every pattern must include at least four points, that points can not be reused, and that intermediate points between two points must be included unless previously used. The average pattern has a length of 5.6 points [86, 138], which results in about the same number of possible combinations as a PIN with four digits.

The problem with these commonly-used unlocking systems is that humans usually

do not use purely random combinations, which significantly improves the chances of an attacker of guessing the correct passcode [146]. For passwords, lists of the most popular ones can easily be found on the internet (an analysis of 70 million real Yahoo! passwords can be found in [25]), many of which are from large data breaches, and dictionary attacks use commonly used character patterns (words) to significantly reduce the number of possible combinations. Also, they can often be guessed from information about the target, such as their name, birthdate, or childrens or pets names. However, a recent study [140] has found that users perceptions of password security are generally consistent with the actual difficulty of cracking the password. An approach to prevent passwords from becoming too popular has been proposed in [124], where users would only be allowed to choose patterns that are not too commonly used yet. The reason why these simple passwords are chosen anyways is a conceptual problem with knowledge-based authentication called cognitive load. The more accounts and devices a user has, the more secrets they have to memorize. To improve usability, users therefore tend to reuse passwords and to use simpler, easier to remember passwords instead of long, random passphrases. As they are not local to one device, the use of a password manager could ease that burden of remembering everything. However, it was found that they are only rarely used due to reasons such as lack of time and technical knowledge [7], and that there still are a lot of misunderstandings of what a password manager even is [3, 30, 65]. The same problem exists for PINs, where people tend to use "beautiful" numbers or numbers with a meaning to them, such as birth dates, and patterns, where people draw easy to input or "meaningful" shapes. In [86], the top 100 most popular patterns constituted 42% of the total number of patterns collected, and in [75], the authors manage to unlock 36.6% of devices secured with a pattern and 28% of devices secured with a PIN after only 30 attempts, which should not be possible if the patterns used were truly random.

The security of the lesser-known alternative methods varies widely. LGs Knock codes have been shown [75] to be unsafe, with over 50% of codes being cracked after only 30 guesses. Somewhat safer is the Windows picture gesture authentication [67]. Microsoft states [159] that about 1.15 billion combinations are possible for three-gesture passwords, but due to it working like a PIN that is disabled after a five tries, brute-force-attacks are not possible. By pre-computing their five best guesses, however, researchers were able to gain access within five tries in 2.6% of test cases [153].

All of these knowledge-based identification methods can be attacked with a shoulder smurfing attack [123, 131], where the attacker observes the subject while entering the secret and gains knowledge about it. The shorter and more rigid the method is, the easier it is for the attacker. A four-digit pin on a number pad is easier to observe than a 10-digit password on a full keyboard, and a shape drawn on a fixed 3x3 grid of points is easier than shapes drawn on a picture without clear markings. Drawing-based authentication such as unlock patterns and, to a lesser degree, picture passwords, are also prone to smudge attacks [9, 145], where the oily residue left on smartphone touchscreens by the finger is used to deduce the owners unlock gesture, even if the attacker did not observe it directly. More technical attacks such as a motion-based keystroke inference attack [8, 31, 32] have also been shown to be possible for all three popular knowledge-based authentication mechanisms, password, PIN and pattern.

### 2.1.2 Being Something

Being something, or inherence-based authentication, has become widespread in recent years, as it does not have many of the downsides of knowledge-based authentication mentioned before. All these methods use something the user is - a biometric trait - that is unique and recognizable, such as a persons fingerprint, face, or iris. Everyone has these traits on them all the time, so one does not need to remember a pattern or passphrase, it requires effort to forge, and it is usually faster than entering a code manually. Because they are part of the body and not the same for any two persons, they can also not be lost or exchanged easily. However, these methods often still require a PIN or a password as backup in case the recognition does not work for any reason (e.g. a wet finger, broken sensor, injury). If that is the case, the biometric authentication can never be more secure than the chosen backup, because an attacker can also always choose the backup to attack.

To use biometric authentication, additional sensors are often required, depending on the biometric trait that is used. The most common one is the fingerprint reader, which is built into many smartphones nowadays. Other examples would be Apples FaceID, which projects a dot pattern in the invisible light spectrum on the users face and uses infrared cameras to detect it, or special retina scanners, for example built into some phones of Samsungs Galaxy line. But no matter the type, the authentication process is always the same. First, the sensor obtains data that contains the users biometric information. From that data, a set of discriminating features is extracted, based on which the user is identified. Using the feature set, the user is finally authenticated by the matcher, which compares the new input to a set of enrollment data [115]. Due to variations in the sensed input data, this can usually not be done by exact comparison as is done with passwords, but by using a distance measure or machine learning approach instead. The performance of such systems is therefore reported as false positive and false negative rates, indicating the percentages of unauthorized subjects that still manage to gain access by similarity or forgery, and the percentage of authorized subjects that is incorrectly not recognized as such.

The security of these different systems is actively being discussed. Already in 2001, [115] have identified eight points where such biometric systems can theoretically be attacked, discussed possible attacks, and suggested strategies to alleviate those. For fingerprints specifically, Apple has claimed that their TouchID system has a one in 50000 chance for a false match, but research [118] indicates that it is possible to generate masterprints that can be used to impersonate a large number of users. Also, the possibility of photo or replica attacks on biometric systems has to be considered, and liveness detection strategies are required for mitigation [139]. Puppet attacks, where an unwilling victim is forced to unlock, for example by placing their finger on the fingerprint reader, have to be considered as well, and behavioral patterns have been suggested to detect such forced unlock attempts [147]. Similarly, in Apples FaceID system, the subject is required to have their eyes open to unlock, in order to prevent unlocks while a user is sleeping. With some level of knowledge of and access to the system, more technical attacks, such as replay-attacks, where previously intercepted signals are replayed to the system, also become possible [139].

A final important security concern is the protection of biometric information. The

same aspect that makes biometrics so useful for authentication - the fact that it does not change over time - is also its greatest liability. Once biometric information is compromised, it is compromised forever, and cannot be changed easily like a password. To prevent this from happening, it is suggested [115] to apply repeatable, non-invertable transformations to the biometric signal, so that true biometrics can never be revealed, even in case of a breach.

### 2.1.3 Having Something

Having something is the third common class of authentication methods. The concept is the same as with any common door and key - whoever is in possession of the key can access what is behind the door. In the digital world, there are many common examples [5, 11, 18, 63, 143]: Smart Cards with a card reader, tokens that are synchronized with an authentication server and display a frequently changing number that has to be entered for access, proximity sensing with wearable devices such as smart watches to unlock a computer, the SMS-OTP system where a one-time code is sent to a specific phone number, cars that unlock when the key is nearby, or the Android smart lock system, where the user can define devices that, when they are connected via Bluetooth, keep the phone unlocked. Often, possession-based authentication is used together with a second factor, such as a PIN, to increase security.

As with biometric authentication, the advantage of this method is that it does not require the user to remember anything, and that it is in many cases fast and unobtrusive. Also, there is not the same concern of a key becoming compromised, because it can be disabled and exchanged rather easily compared to biometric markers. However, on the downside, possessions need to be brought along, can be forgotten, damaged, lost or stolen, and whoever has them gains access to the protected resource, provided that there is no two-factor authentication in place. This is also the most important security concern that is relevant for all the methods described above. Other attacks, such as power analysis attacks on smartcards [99], or wireless interception or trojans against smartphone OTPs [101], can also be possible, but require more effort and technical knowledge.

### 2.1.4 Doing Something

Doing something is usually not considered as a part of the classical three authentication factors, or is included as a part of being something. However, while traditional biometric authentication methods focus on what a persons physiological markers are, behavioral biometrics consider learned - or unconsciously executed - patterns that identify a user uniquely [49]. To better emphasize this separation, especially because our own work is part of the behavioral methods, we have decided to discuss it as a separate category.

The general idea of behavioral authentication is that not just what a person does, but how they do it can be unique. When writing on a keyboard, everyone types a bit different and has unique traits how and where they strike certain keys [41, 49]. When walking, everyone moves slightly different [78, 91, 105, 116]. When drawing an unlock pattern, everyone might draw the same shape, but everyone does it in their own distinctive way [98, 108]. And when signing, the way the signature is drawn will differ from subject to subject. All of these methods, and many more (eg. [27, 129, 154]), have

already been considered in related work. In many cases, behavioral biometrics are also considered as transparent methods of continuous re-authentication while the subject is using the device instead of being used as single-point logins, because systems such as gait-pattern detection require the user to be continuously performing an activity for a while and do not allow for an instantaneous login. Similar to other common biometric systems, the advantage of behavioral biometrics is that the user is not required to remember anything, and that it can be hard for others to imitate behavioral traits that the subject itself might not even be aware of [100].

Aside from the traditional points of attack that must be considered for all kinds of biometric systems [115], such as replay/imitation attacks or attacks against the verification system itself, behavioral biometrics suffer mostly from intra-user variance and time-variance (also called template-aging) that make reliable authentication more difficult. These two terms refer to the fact that, while they might exhibit similar patterns, no two executions of a behavior are exactly the same, and that the behaviors are also expected to change over time. Most physiological biometric markers have the same problem of aging (only the iris is not affected directly, but can be affected by health issues [79]), but for most modalities it occurs only after years instead of months [79, 137]. Intra-user variation can also be an issue, but not due to changes in the marker itself, as with behavioral patterns, but because of sensing variations. A good example for short-term changes in behavior is gait recognition. When the subject is tired after a day of work, their gait will likely be different from when they are well rested and full of energy (intra-class variance). And when the subject is 50 years old and healthy, their walk will likely be different compared to 20 years later, when they are 70 and might suffer from joint pain (time variance/template aging). In chapter 3, mitigation strategies for these problems are discussed in more detail. Our work - the verification of handwritten signatures - is one important example of behavioral biometrics. The following section gives a detailed introduction into the topic and discusses the important terminology, and chapter 3 gives a comprehensive review of the current state of research in the field.

## 2.2 Digital Handwritten Signatures

Handwritten signatures are a widespread technique of personal identification. It is the preferred method of authentication and authorization in the offline world, and users are accustomed to it. With the progress of technology, signatures are also being used with increasing frequency in the digital world, for example for paperless signing of contracts or to approve bank transactions. Because a persons signature is considered unique, signatures are also considered as a means of user authentication in research [29], similar to fingerprints or iris recognition, although not used in practice yet.

In many businesses, such as banks and the postal service, digital signing is already common practice, and special signing pads are used with a pen to offer the user an experience similar to signing on paper. At home, a common way to get a digital version of a handwritten signature is to just sign on a piece of paper, scan it or take a picture, and insert the signature image where it is needed. However, with the rise of touchpads and touchscreens, digital alternatives have become possible, where the user signs either with their finger or with a stylus on screen. In these cases, additional - so-called dynamic - data can be captured together with the signature, that provides more information on

how the signature was drawn. Either of these approaches can be verified, and although the methods are different, the overall concept of handwritten signature verification stays the same. In all verification systems, a user is registered by acquiring a number of samples of their signature. From those signatures, either the features themselves (reference-based enrollment) or a model representing the signature samples (model-base enrollment) are stored and used for the verification of new signatures [76]. If a signature is similar enough to the references, it is considered genuine and is accepted, else it is rejected.

Compared to authentication methods such as fingerprint readers or iris scanners that were described previously, the accuracy of signature recognition is still lower, but there have been significant improvements in error rates in the last few years due to ongoing research [23]. As a behavioral method of identification, one of the major challenges is that signatures underlie variations over time (personal entropy), caused by mood, aging, and other influencing factors, so new signatures to verify will vary from the templates after a while [23]. This is known as the template-aging effect [119], and different updating strategies have been investigated to mitigate it [137]. With mobile devices, there is also a wide variety of situations (e.g. sitting, walking, standing) and environments (e.g. temperature, humidity, light) in which the user can use the signing system. Also, it is yet to be investigated how the platform the user is on (e.g. at home, on a train) affects performance [122]. All these factors have the possibility of influencing signing behavior, making verification more difficult.

### 2.2.1 Static and Dynamic

Many different approaches to signature recognition have been developed trying to solve these problems and improve performance, and a variety of different algorithms are employed. However, all of them can be classified into one of two major sub-categories based on the information that is used:

- Static (also called offline) signature recognition
- Dynamic (also called online) signature recognition

#### Static Signature Recognition

Static signature recognition is based on the signature image alone, and no other data is available. Emphasis lies on signature image, because the signatures are usually digitized by scanning or photographing, so no time series information of the user signing is available from which additional features could be derived. Historically, only this method has been used, as this is the only possibility for traditional signatures on paper. However today, static signatures are still the most widely deployed form of digital signature, without any verification, but just to be able to avoid paper and store the image with a signed document [122].

Static approaches still have not reached the performance of a professional calligrapher. One reason is that humans can benefit from their experience and try different approaches when they feel it is needed, while algorithms do not have the same kind of intuition. Another one is that variations in the signature such as pen pressure and pen inclination are visible to a trained professional, but not easily detected by scanning the on-paper signature and completely lost when signing on a screen without capturing

additional data. Without having this information available, static approaches are more prone to forgeries and nowadays, dynamic signature recognition is becoming more popular, as it is considered to be better for automatic verification and offer larger fraud robustness [23, 122].

#### Dynamic Signature Recognition

Dynamic signature recognition, instead of just using a signature image, operates on time series data of the user signing. The signature geometry is given through a list of coordinates at a position in time, often together with various other sensor readings. Thanks to the availability of time information, there is not only a visual comparison, but the whole process of signing can be retraced, and in combination with the additional features, a signature becomes much harder to imitate. Evidently, this is only possible on electronic capture devices and not with traditional signatures on paper. The additional channels are recorded while the user is signing, making the process of data acquisition more complex. Below, some channels that have been used in related work are listed in no particular order:

- x-y-coordinates (the signature image) [23] and pen-up trajectory information (the movement of the pen while it doesn't touch the screen), if a pen is used [94]
- Pressure, usually recorded with a pressure-sensitive pen [23]
- Azimuth (pen inclination) [23]
- Elevation (height of pen over tablet) [38, 48]
- Accelerometer readouts [104]
- Gravity sensor readouts [29]
- Magnetometer readouts [29]

Nowadays, a big part of research focuses on trying a variety of these different features and feature combinations to improve verification performance [23]. There is also still more work to be done on which algorithms to use, especially in the mobile environment. In terms of information channels, forensic experts consider pressure information to be important, but it is still an open issue whether or not to acquire it on mobile devices. As most mobile devices are used without a stylus, there are investigations to find out if pressure information can be emulated, for example using the number of pixels touched while signing. It is not yet demonstrated if such emulation is is useful or equivalent to the pressure acquired by a pen. The inclusion of gyroscope and accelerometer were also shown to be a promising future research line [122] for mobile applications, even though these features are not commonly thought of being part of a signatures. It is suggested [122] that such a holistic approach of not only using the movements detected by the touchscreen, but also those of the device obtained by gyroscope could be a potential solution to the lack of resolution compared to traditional desktop applications with signing tablets, and to improve robustness against forgeries.

### 2.2.2 Local and Global

The field of dynamic signature recognition can further be divided into function-based (local) and feature-based (global) systems, that differ in how they use the captured

time-series information. Function-based systems rely on so-called local features for classification. Those are the actual time series of different information channels themselves, such as the screen coordinates or pressure at each point in time, as they are recorded. An easy way to think of it is that the actual signature is used directly, which means that the longer the signature is, the more features are available. Feature-based systems, on the other hand, use global features. Global features are statistical information about a signature that is extracted from the original time series data, with signature duration, average pen-speed and number of pen-ups just being three examples of a wide variety of possibilities. As opposed to local systems, this results in the same amount of features regardless of signature length. In [46], a 100 features global feature set is originally described, and it still used either as is or in adapted form in different studies [76, 93, 94]. The two approaches also differ in the algorithms that are most frequently used for classification. Function-based systems often use dynamic time warping or hidden markov models to compare time series [23], while feature-based systems tend to use standard machine learning approaches and statistical classifiers [94, 102].

### 2.2.3 Reference-Based and Model-Based

The distinction between reference-based and model-based enrollment specifies how a users signature templates are stored. In reference-based systems, the features of each of the individual signatures are stored as templates, and the signatures themselves are used to compare to an input signature in the matching process. In model-based systems, a statistical model of a set of enrollment signatures is computed, which is used for comparisons in the matching step [102]. This can be as simple as storing a set of global features of reference signatures, or as complex as pre-training machine learning models on the users enrollment signatures to distinguish genuine signatures from forgeries. While the chosen approach is mostly determined by the type of method used for verification, there are important issues with each of them that need to be considered. In a mobile environment, power and processing speed are constrained, so training more complex models might not be viable on device and could be done outside, such as in a cloud. For most biometric systems that only perform 1:few comparisons in a reference-based scenario, on the other hand, current processing power on mobile devices should be enough [122]. However, when the enrollment signatures are stored on device (or transmitted to and stored in the cloud) for such comparisons, template security becomes an issue, because a potential attacker would be able to access the real signatures, instead of just a statistical representation, as in a model-based system. In [151], it is suggested to secure templates with homomorphic encryption, which still enables comparison with no performance loss, but protects the users signature data from being compromised.

### 2.2.4 Writer-Dependent and Writer-Independent Verification

Verification methods can be either writer-dependent or writer-independent. This is true both in offline signature verification, where this distinction is more commonly made, but also in online scenarios. Writer-dependent systems need to be trained specifically for the person they should verify, and re-trained for anyone else [35]. Examples of such systems are the various kinds of one-class classifiers that are trained on users enrollment samples

to be able to detect deviations from the known pattern. They are specific to the one user they were trained for, and need to be trained again to verify someone else. More preferable are user-independent models. Those are algorithms that are able to compare two signatures, such as the siamese network approach in [35], or more commonly DTW-based systems that compare a new signature to a given template. Also, machine learning models that were trained as general forgery detectors that recognize the non-user-specific characteristics of a forgery would be considered user-independent, as they can be used everywhere without retraining. There are also combinations of both approaches, where for example a user-independent forgery detector is used together with a user-dependent anomaly detector for verification.

### 2.2.5 Skilled and Random Forgeries

The terms skilled and random forgeries describe the different attack scenarios against which a verification system must hold up. The simpler task, more recognition than verification, is the correct detection of random forgeries [47]. There, an attacker signs with their own (or any random) signature in a low-effort attempt to pass as another user. This is the baseline test scenario and all systems in related work perform relatively well, as signatures are usually different from one another and high similarity between signatures would be coincidental. Somewhat more difficult, and an intermediary step between random and skilled forgeries, are simple forgeries, where the attacker signs knowing the name of the user they want to attack, but without any knowledge of the real signature of that user [14, 69]. This can be a realistic attack scenario, and while luck is still needed to come close to the users real signature, the chances of passing are higher than in the random case. However, this is only rarely considered in related work at all.

The most important case, and a good benchmark of system performance, is the case against skilled forgeries [47]. There, an impostor has knowledge of a users real signature at least to some degree, and tries to impersonate that user to the best of their ability. The actual attack scenarios can vary: From an over-the-shoulder attack where the attacker can peek at the person signing one time, to one or multiple images of the signature or even videos of the signing process that can be replayed and allow for accurate imitation of not only the signature image, but the signing speed and other dynamic factors. Depending on the skill and effort of the forger and the amount of information available, this can result in forgeries that are highly similar to the real signature and hard to detect for an automated system.

# Chapter 3

# Related Work

In this chapter, we review the prior work and relevant literature on the subject of signature verification. We start by giving a brief overview over traditional static verification where the field has its origins, before going into detail on dynamic signatures. First, we describe the categories of dynamic features and the respective challenges from a mobile perspective. Then, we discuss different verification systems, beginning with a short overview of standard pen-tablet systems, and continuing with detailed descriptions of different mobile approaches that were published in recent years. Finally, we discuss the prior work on dataset recording and usability, and end with a summary of the key takeaways and open areas of research.

## 3.1 Static Signatures

There are static (offline) and dynamic (online) approaches to signature verification. In the field of static verification, studies have predominantly employed Hidden Markov Models [12, 39, 106], Support Vector Machines and One-Class-SVMs [15, 54, 56, 70, 77, 109, 126, 149], Neural Networks and Deep Learning [13, 35, 56, 62, 87, 114, 127, 130], and ensembles of classifiers with a combination of predictions [15, 106, 149]. Both writer-dependent and writer-independent classifiers are considered in related work, and combinations of both approaches have been investigated as well. Empirically, SVMs have been shown to be one of the most effective models for the task.

In many cases, the proposed systems are tested on one or more of three public offline signature datasets, GPDS [42, 43, 142], MCYT [45] and CEDAR [71]. Results on skilled forgeries range between 1.69% EER at the very best to about 15% EER, with a mean of about 10% EER over all studies that report it. It has to be noted that different studies use different experimental protocols and results can also vary between different test datasets. The best result on all three dataset has been achieved using an SVM with a feature learning approach [57]. As our work is in the field of dynamic recognition, where additional sensor and time information is available and used for verification together with different classification approaches such as DTW, we are not going further into depth on static verification. A more comprehensive review of offline literature can be found in [57].

## 3.2 Dynamic Signature Features

Dynamic signatures, compared to static ones, are considered to be better for automatic verification and to offer better robustness against forgeries [122] thanks to the additional information that is available. They are recorded as time-series of x- and y-coordinates, sometimes together with additional pressure, pen [76] and movement data [29], making them more complex than static signatures, where only the signature image is used. The available features are determined by the devices and device classes on which the signatures are recorded [23]. Normalizations of alignment, size and length, such as equispacing by linear interpolation [23] or aligning the starting point of the signature with the coordinate system center [104], and low pass filtering to remove slight noise variations introduced by the capture device [23], can be necessary on all time series data before use, depending on interoperability requirements and the type of system.

The recorded time series can either be used as such in a function-based (local) system, or they can be used to extract statistical features that describe the signature for a feature-based (global) system. In [94], it was investigated how the two types of systems compare in terms of verification performance in different scenarios. Especially for local features, the feature discriminative power was negatively affected for signatures recorded on mobile devices. It was found that the global feature approach is more robust against the degraded signature acquisition conditions, and also more robust against the lack of pen-up trajectory information, which resulted in a performance penalty in the local system. Somewhat in contrast to this, in [104] and [76], where mobile data is used, it is stated that function-based systems generally offer better verification performance than feature-based systems [104]. Within global systems, geometry features were found to have higher relevance in the mobile scenario, while time, speed, and acceleration related features were more relevant in pen-tablet scenarios. As geometry features are the easiest to forge in principle, their higher importance in a mobile dataset may lead to lower verification performance of global systems on mobile data compared to pen-tablet data [119].

### 3.2.1 X-Y-Coordinates

The X-Y-coordinates of the finger or pen on screen form the signature image, but because they are recorded as time series, it is possible to extract more information. As local features, contact/no-contact information of the finger with the touch surface, and therefore the order of strokes/pen-ups [119], can be obtained [2, 23] to get a stroke pattern of the signature. Also, it is possible to calculate the first- [23] and second-order derivatives to gain velocity and acceleration for both the X and Y axis [94], although second-order derivatives for both signature- and pen-features have been shown to not contribute much to improve prediction performance in [93]. In [93] and [94], a total of 21 different local features based on X and Y, such as additional geometric relations and different angular metrics, are proposed and used [93, 94]. Examples of global features are the number of strokes, the start- and end points [29] of the signature, averages and standard deviations, and minimum and maximum feature values [94]. An extensive list of possible global features can be found in [94].

While the X-Y-coordinates are the most common and important channel of features

for signatures, they are also considered to be the easiest to forge. The developers of the hold and sign system in [29] did not take the signature image itself into account because it can be mimicked well, and only relied on 13 global features in addition to movement sensor values recorded while signing [29]. In [10], the X-Y-dimension was omitted all together, instead requiring the user to draw the signature in the air with the phone and relying solely on movement sensors for classification. And in [94] it is stated that especially the geometry features of the X-Y-channel are in principle the easiest to forge, but derived features such as speed and acceleration are more robust [94].

### 3.2.2 Pressure

Pressure information can be obtained in different ways. The most common one is the use of a special pen that measures the pressure that is applied in fine-grained levels of accuracy. All datasets we are aware of recorded it this way [2, 16, 24, 38, 74, 90, 148]. However, it is possible to detect pressure on devices that do not offer a stylus. As this is the case in smartphones and notebook touchpads, it warrants some further investigation into pen-less alternatives.

Some mobile devices have screens or touchpads that can differentiate different pressure levels. The most notable examples are some iPhone and MacBook models that support Force Touch, where different pressure levels can be detected by measuring the degree of inward curvature of the display glass when force is applied [103]. Huawei has introduced an Android phone with a similar pressure sensitive screen. Using the iOS and Android APIs, apps have been developed that can report the percentage[1] (iOS, only possible with jailbreak, as iOS usually only provides three discrete pressure levels) or approximate weight[2] (Android) with which the pressure sensor is loaded, although very light loads can not be detected. However, to our knowledge, the usefulness of these measurements has not been tested in signature-related work yet, and it might not be feasible at all. These technologies are intended for clicks and not for drawing, and detecting pressure this way requires a bit of time after the touch for the pressure to be built up and held [103].

Some attempts were already made on smartphones to detect screen pressure without requiring any non-standard hardware. In a study by Google research, the device barometer was used to sense differences of atmospheric pressure inside sealed phones caused by exertion of force on the screen. While the relative changes in pressure could be measured accurately, the system can not deal with fast changes [113], making it unsuitable for application in signature recording. In [89] it is mentioned that another study [10] used the device accelerometer to obtain the pressure users make in the signing procedure, but we could not find any mention of that being the case in said study. One study [51] managed to detect three continuous pressure levels from accelerometer readings when pulsing the phones vibration motor during touch contact, and others [59] have managed to detect taps and force taps with the accelerometer alone, but not as continuous measurements.

As for the importance of pressure information in signature recognition, there are some contradictions that can be found. In [23], pressure is considered to be an important

[1] http://flexmonkey.blogspot.com/2015/10/the-plum-o-meter-weighing-plums-using.html
[2] Huawei Fun Scale App

discriminative signal, but because some devices did not provide reliable data or no data at all, it was not recorded [23]. Also, almost every dataset recorded with a pen tablet includes pressure information (cf. [2, 16, 24, 38, 74, 90, 148]), which also backs this claim. In contrast to this, in a study that investigated feature discriminative power and feature importance in mobile and signing-tablet-based systems [94], pressure related features are not present in the optimal feature set of the function-based system, suggesting that a signing tablet does not have a per-se advantage over handheld devices just because of the capture of pressure information [94], and in [73] it is stated that pressure information does not seem to be a useful feature to distinguish genuine signatures from forgeries.

### 3.2.3 Pen-Based features

Styluses, as used in signing tablets and some smartphones, can provide a variety of different information to be used in dynamic signature recognition. Pressure, which has already been discussed in depth above, can be one of them, but there are more, such as pen inclination (Azimuth) [24, 38, 48, 148], pen elevation (height of pen over tablet) [24, 38, 48, 85, 148], and pen-up trajectory information, as explicitly mentioned in [94], but probably implicitly recorded with the x-y-coordinates by most pen-based datasets.

These pen-up trajectories seem to be important. In [94], when the same dataset was used twice, once with pen-up-trajectories included, and once with interpolated values instead, no significant difference could be found for feature-based systems, but function-based systems performed notably worse on the interpolated dataset. The results suggest that pen-up trajectories contain important discriminative information against skilled forgeries because they are harder to imitate, putting handheld systems that do not have them at a disadvantage. However, other pen-orientation related features were not present in the function-based systems optimal feature set [94]. While that suggests that a lack thereof should not penalize the verification performance of pen-less systems [94], another study has stated that one problem of signing with finger on phone is that pen-tilt angle information is not being captured [150].

A final important finding is that there is no big difference in performance between signing with a finger or pen if no pen-based features are used [22]. While in real applications, pen-based features will be included if they are available, this suggests that the signature itself can be considered to be the same, whether it is drawn with a finger or a stylus. A follow-up study by the same authors comes to the same conclusion and corroborates that finger and stylus can be counted as the same biometric modality [23].

### 3.2.4 Movement-Based Features

Movement-based features are not a part of the signature itself, but rather a biometric trait on their own that is recorded while signing and can be used for identification alone or in combination with other information. They have been used for identification based on gait patterns measured with an accelerometer [91], based on how users answer their phone, using accelerometer and orientation sensor in [34] and accelerometer, gyroscope and orientation sensor, together with a slide to unlock pattern recognition and voice recognition, in [27]. Gesture patterns while using the phone [154], movement patterns while entering a PIN [28], and the micro-movements of the phone while holding and tapping on it in hand [129] have also already been used to identify users based on their

movements.

Related to signatures, [10] proposed a system based on accelerometer readings where the user "signs" their signature in the air while holding the phone in hand. Together with global features of a signature drawn on screen, [29] recorded magnetometer, gravity sensor and accelerometer to use the phones micro-movements as a second biometric trait, and [104] used the accelerometer in addition to the signature point coordinates on screen to record the phones movement. However, movement features are usually not a part of any traditional signing tablet datasets and also not relevant for notebook touchpad signatures, as these tablets do not have the sensors necessary for recording them and can not be used in a way where recording of movement data would make sense. Even in the handheld mobile area, they are often not or only partially recorded for signature datasets [22, 94, 104].

There are five main sensors that are used in sensor-based authentication that are present in most smartphones: accelerometer, gyroscope, gravity sensor, orientation sensor and magnetometer [29]. From the raw accelerometer values, two features can be derived. A low-pass-filter isolates the gravity component that acts on the device, and a high-pass filter isolates the linear acceleration, which is the actual acceleration applied by the user on the device [29]. The virtual gravity sensor essentially provides a more sophisticated version of the low-pass-filtered accelerometer readings. The magnetometer, although it is not a movement sensor, can still be used because of the change in magnetic field caused by a user action, such as tapping or drawing on screen [129]. Each of these sensors report x, y, and z, values. Usually, as a fourth dimension, the magnitude is added [28, 129], as it has been shown to be effective in classification accuracy in the context of smartphone user authentication [29].

In [129], where a continuous authentication system based on movements while typing on the phone was proposed, the magnetometer values were eventually left out because they consistently performed worse compared to accelerometer and gyroscope, and including them did not improve overall performance [129]. Also, sitting, standing and walking are treated as different conditions in some movement-based systems [29, 129]. In [29], this was explicitly investigated, and it was found that models trained in one stance could not be used to predict another stance well (65.82% TAR with unspecified FAR, compared to 86% TAR with 0.1% FAR). However, it was possible to train a system with all stances to predict all stances with satisfactory results (95% TAR, 3% FAR) [29]. In [129] it was also found that the system performed particularly well during walking (10.05% EER when sitting, compared to 7.15% EER when walking), possibly because the sensors could capture more information about body movements instead of just taps on the phone [129].

## 3.3 Dynamic Signature Verification

There are three scenarios in dynamic signature recognition that can be considered, each with different challenges and different sets of features that are available. Pen-Tablet systems usually have x-y, pressure, and pen-information available as dynamic features. Mobile phones usually do not have a pen for signing, and thus no pressure- or pen-data that can be used, but as handheld devices, movement data recorded by the embedded sensors can be considered in addition to the x-y-channel signature. Notebook touchpads,

as stationary devices without stylus, do not have either of the additional channels, and depend only on the signature image (x-y-channel) for verification.

### 3.3.1 Pen-Tablet

Pen-Tablet systems are the traditional scenario for dynamic signature recognition with both the most related work and datasets available, partially thanks to different signature recognition competitions that provide accessible signature datasets, such as [24, 85, 89, 90, 148]. These systems rely on combinations of X-Y-coordinate- [4, 40, 80, 88], pressure- [4, 40, 80, 88] and pen-related [40] features, the public datasets and competitions usually provide all three types. They are often recorded on drawing tablets, but also tablet computers with stylus as in [4] fall into this category. Classification is mostly done using Hidden Markov Models [4, 37, 148], Neural Networks [40, 88] or Dynamic Time Warping [37, 60, 80, 148]. Achieved equal error rates range between 2.2% and about 10% EER on skilled forgeries, with the highest-scoring one being a DTW-based verification system submitted to the BioSecure evaluation campaign [61]. However, it is important to note that there are significant differences in experimental protocols between studies, so these numbers can only serve as a rough performance reference. Because our works focus lies exclusively on mobile devices, we are not going into more detail on stationary signature verification with pen here, and include results that are relevant for mobile in the mobile section.

### 3.3.2 Phone

A substantial amount of work can be found on mobile signature recognition with smartphones and PDAs. In recent mobile datasets, Signatures are usually drawn with a finger or a stylus on high-quality touch screens while holding the device, while in older databases, mostly PDA devices with significantly worse screens and sometimes styluses were used, which is less relevant for our case. Therefore, most of the information presented below is from studies newer than 2010, as that was the time when the modern smartphones started to become available.

#### Local Systems

**DTW-based classification** There are different studies about signature recognition on mobile devices. In [97], the authors compare an SVM with a DTW-based classification approach. For their tests on devices with modern capacitive touch screens, they included X-Y-Coordinates and size (which is assumed to be size of the touch point, not specified) as features. After smoothing, normalization and equispacing, additional features were extracted from the recorded time series. A PCA was calculated over the dataset to improve performance, as both the manually reduced and the unreduced dataset did not perform as well as expected. The DTW algorithm performed better than the SVM in all scenarios, with 2.33% vs. 4.28% EER on the capacitive smartphone and 4.01% vs. 5.23% on the capacitive tablet. Results on devices with resistive touchscreens were slightly better than on capacitive ones because the pressure information available on them was more precise than the size information on capacitive screens. The users reported feeling uncomfortable because of the (too large) size when signing on the tablet, which explains

the worse results there. Interoperability was also tested by using a model built on a signing pad dataset to classify each of the mobile datasets, and it resulted in EER's between 8.16% and 13.58% using DTW, with the SVM performing worse [97].

Both [22] and the follow-up study [23] continue the study above [97], and different aspects of digital handwritten signing specifically in the mobile area are investigated. Using the DTW algorithm from above and with only X-Y-coordinates and derived features, at best an equal error rate of 0.29% on random forgeries and 10.10% on skilled forgeries is reached on the mobile phone with a finger. Results on the same device with a stylus (without additional data from it) were almost identical, and both were slightly better than the signing pad reference. The importance of good visual feedback is stressed, with an approximately 20% difference in Equal Error Rate (EER) between the two tested devices. Interoperability was tested by using the smartphone model on signatures drawn with the finger on different tablets, and results show that for two devices, EERs improved slightly compared to using a model trained on data from the same device, while for one other device they got a bit worse, within a range of EERs between 1% and 2.5% [22]. The follow-up study also found that generally, there is interoperabiltiy between fingertip-based devices, with no significant difference to intra-device results, with the exception of one device, where the system performed about 2% points worse in EER when its signatures were used as templates. While it can be hypothesized that this is due to its relatively low-resolution touch screen or it being the only device running on Blackberry OS instead of Android or iOS, the authors state the further work is required to obtain a reason for this small performance degradation [23].

DTW-based verification with signature complexity specific preprocessing A verification system that is adapted to different levels of signature complexity is proposed in [135] and tested amongst others on the tablet-signature subcorpus of the eBioSign database [144], where users signed with both finger and pen on a notebook display. A complexity detector classifies each signature in one of three complexity levels, with the number of strokes as computed by the Sigma LogNormal model as indicator of complexity. A higher number of strokes is considered to be a more complex signature, while a lower number of strokes indicates less complexity. Each user is assigned a complexity level based on the complexity of the majority of their enrollment signatures, and that complexity level is used when a new signature is compared with the claimed identity [135].

In preprocessing, set of 21 time-function features, as described in [93] and also used in [94] is extracted from the x/y-timeseries. Pressure and pen-up-trajectory features are removed beforehand to enable comparability with finger signatures, and normalization based on the mean and standard deviation is applied. Using sequential forward feature selection, an optimal subset of time functions is then selected for each level of complexity to serve as input. For classification, the same DTW approach is used for all complexity levels with four enrollment signatures [135].

On the eBioSign database, signatures that were classified as high complexity reached an absolute improvement of 5.5% EER over those classified as low complexity by the complexity detector, which is similar to related work and shows its effectiveness. The performance of the proposed system is then compared to baseline DTW systems [53, 133], with the only difference being the complexity classification and feature subsect selection for each complexity level in the proposed system, as opposed to the same overall

feature subset selection for all signatures in the baseline. Both systems perform near-perfect on random forgeries with an EER of 0.1% in all test scenarios, but the proposed system achieves an average absolute improvement in EER of 3.4% over the baseline for skilled forgeries. Although no additional pen features are considered, there is also a significant improvement on skilled forgeries of 4.3% EER when signatures are written with a pen, as compared to signatures written with a finger. In terms of absolute numbers, the performance on skilled forgeries of the proposed system ranges between 10.2% to 5.6% EER for pen signatures and 15.7% to 10.2% EER for finger-drawn signatures. In inter-modality tests, where signatures drawn with a pen were used to verify signatures drawn with a finger and the other way round, the performance of the proposed system on medium and high complexity signatures was similar to the finger-only scenario with EERs of 16.7% and 11.1%, while for low-complexity signatures, the performance dropped to around 25% EER. In general, the performance on higher complexity signatures was better, and especially the mixed input scenario is more feasible for higher complexity signatures. The complexity-based feature selection improved performance in almost all scenarios, especially for challenging lower complexity signatures [135].

The authors also find that the difference between pen- and finger signatures is that users tend to perform much larger versions of closed letters (eg. a, e, o, l, p, q) due to the lower precision with the finger compared to a pen, and those with a long name tend to simplify parts when signing with the finger. Also, the sampling distribution of signatures drawn with a pen is much more uniform over the whole signature compared to drawing with a finger, which is due to the lack of precision with the finger and the friction produced between finger and screen [135].

**DTW with homomorphic encryption** The authors of [151] focus on the security of a mobile signature verification system and its template data by using homomorphic encryption [52, 110]. This makes it possible to perform computations without having to restore the actual signature template, so its security is ensured. They use a client-server architecture, where the encrypted signature template is stored on and the computation of the DTW algorithm used for detection is done by the server. While the exact architecture is not relevant for our work, their findings are interesting. They use the x/y-features of signatures captured on mobile phones in two sessions over half a year period, and evaluate their system on both random and skilled forgeries. To improve performance, they use a restriction window on the DTW, and find that increasing its size from 10 to 35 for signature time series with between 100-800 data points reduces EERs on random forgeries form 12.8% to 1%, which is even better than the unrestricted DTW with 1.7% EER. Also, they find that there is virtually no performance degradation when the DTW is performed in the homomorphic encryption domain. On skilled forgeries, they reach an EER of 9.1% with the x/y-features alone [151].

**Local Autoencoder-based verification using signature and acceleration data** Acceleration data is also used as a dynamic feature, together with x-y-coordinates, in a function-based system where Convolutional Neural Networks (CNNs) are used as a means of feature extraction before training an autoencoder for one-class-classification [104]. The general idea is that a CNN which is trained to classify forged and genuine signatures will return the features that are important for that purpose as output, which can then be used as

input to the autoencoder. The autoencoder itself is trained using the same data (sample signatures of a subject) as input and output reference. In the test phase, it then generates an output corresponding to a test signature input, with highly similar output for trained data patterns, but not for unfamiliar patterns. To perform the actual classification, the error between the test signature and the autoencoder output is compared, and accepted or rejected based on a threshold.

The feature extracting CNN's were trained in four different scenarios: As two-class classifier between all original and all forged signatures (forgery detector), as N-Class-classifier to classify signatures as subjects, as two-class-classifier for subject/others classification, and as a two-class classifier to distinguish a subjects signatures and their forgeries (not applicable in practice). While the last CNN performed the best standalone in distinguishing forgeries, they got almost as good results with a concatenation of the feature vectors of the first and third CNN's (a possible scenario). In that case, they reach EERs of 2.7% on subject/others classification tests with random forgeries, and 4.4% on data including skilled forgeries. This is an improvement of 1.1% on subject others and 13.7% on skilled forgeries [104] compared to an MLP approach that was used in previous work [29].

**Local writer-independent signature verification using RNN with DTW preprocessing** In [134], the authors train a recurrent neural network (RNN) to detect general genuine and impostor patterns in signatures (a forgery detector), so it is one writer-independent model. The goal is to learn a dissimilarity metric for comparing two signatures that is small for pairs of genuine signatures of the same subject, but large for genuine-forgery pairs. They build upon their previous RNN architecture as described in [136], which uses a common set [93, 94, 135] of 23 time functions derived from X, Y and pressure signals, but add DTW as a preprocessing step. This way, the time series of the template signature and the signature under test are aligned as well as possible before feeding them to the RNN [134].

The approach is tested on a database built up from multiple databases of other studies they introduce in their study as well. Tests are conducted on random and skilled forgeries on both traditional signing tablet data with pen and mobile data recorded on tablets and phones with a finger. In the mobile scenario, there were no pen- or pressure features included. On skilled forgeries, their approach reached 13.8% EER and 11.3% EER on average on finger-drawn signatures and 4.2% and 3.3% EER on average on pen-tablet signatures, for 1 and 4 enrollment samples respectively. On random forgeries, EERs in all scenarios were below 1.5%. Compared to a DTW reference system in the mobile scenario, the the proposed system reached EERs that were about 3% (absolute) lower on skilled forgeries, and compared to an RNN without DTW alignment preprocessing, the proposed systems EERs were about 5% lower. On random forgeries, the DTW approach performed slightly better than the time-aligned RNN, with about 0.5% difference in EER. RNNs without time-alignment performed significantly worse than both and were therefore not included in the reported comparison results. Especially notable is the high performance degradation in the case of skilled forgeries in the mobile scenario, when no pen- and pressure-data was available [134].

### Hybrid Systems

**Comparison of local and global systems in mobile conditions** In [94], the authors build two systems to determine the importance of local and global features of signatures recorded on a Galaxy Note smartphone and reference signatures from the BioSecure database. The global system is a classifier based on the Mahalanobis-distance, while the local classifier is based on Hidden Markov Models. The optimal feature sets for handheld mobile devices were determined using PDA signatures from the reference dataset, which only includes x-y-information as input source. On the Galaxy Note data with these feature sets, they report EERs of 4.2% and 2.1% on the global system, and 6.2% and 6.8% on the local system on random forgeries. The two values per system represent the performance with two different optimization scenarios, for skilled forgeries (which were included in the PDA "feature optimisation" set, but not the Galaxy Note database, which was used for training and testing in the experiments) and random forgeries, respectively. Compared also to their non-mobile experiments, they find a decrease in feature discriminative power in mobile environments, especially for local features. The finding of [23] that screen size, a major concern in the mobile area, does not have significant impact on recognition performance, is also contradicted by [94], which states that a lack of space could lead to hesitation marks and loss of fluency and signature complexity with reference to related work [128].

**Hybrid local and global system employing DTW** A hybrid system that combines feature-based and function-based verification was developed in [76]. For the feature-based classification, 100 global features are extracted from each signature sample (time based, speed and acceleration based, direction based and geometry based) based on the global feature set described in [46], which is also used in [94]. Sequential forward feature selection that minimizes EER is used to reduce the dimensionality of the feature vector, normalization is performed on the whole feature set using tanh-normalization, and the Mahalanobis-Distance is computed for the similarity score. The function-based, local approach extracts additional time-functions and derivatives as described in [93] from the siganture samples. From the second derivatives, only an optimal subset is used based on their discriminative power, as they generally do not contribute so much. The score is generated using DTW. To obtain the final classification result, a weighted average of both scores is computed [76].

The experiments were conducted on signature data captured for the study in two sessions, with an average gap of 5 days in between, and 10 signatures per session. It is notable that each session was again split in two phases of 5 signatures each, with a short break for the participant in between. In total, 25 participants signed on a Samsung Galaxy Note with a pen, equating to 500 signatures. Two scenarios were tested: Experiment 1 with only the signatures from the first session for test (no intra-class variability), and experiment 2 with only signatures from the second session for test. Enrollment for both was done on the signatures of the first session. Only tests on random forgeries were conducted. In experiment 1, the system reaches 0% EER with 3 samples from the first and 2 from the second phase for enrollment. With 5 samples from the same phase for enrollment, it performs slightly worse with 0.58% EER, and with only 3 samples of the same phase for enrollment, EER goes up to 0.8%. In experiment two, 0.53% EER

was the best result with 5 same-phase samples for enrollment. The 3/2-split-version that performed best in experiment 1 reached 0.8% EER, and with only 3 samples for enrollment, 0.99% EER were reached on random forgeries [76].

**Hybrid local/global verification algorithm using DTW for low end devices** In [111], the authors suggest an algorithm specifically for low end mobile devices. In it, X/Y and Pen up/down (called pressure) are used as base features, together with a Time Series features, which is the sequence of equispaced sampling time instants (timestamps). The X/Y coordinates of the signature are translated, so that the mean coordinates are the new origin of the coordinate system, as signatures can be written anywhere on screen. From the signature, they derive a set of the same global features from the whole signature and from each stroke individually [111].

The verification itself is done as follows: In the registration phase, a user needs to draw their signature three times, with same amount of strokes each time. Noise is eliminated, and then the global features are calculated. To deal with different screen sizes and pixel densities, a tolerance factor is introduced for all affected metrics. Each feature of a new enrollment signature has to be within the tolerance factor to already existing enrollment signatures, otherwise it has to be repeated. Once all three signatures are recorded, the average of the enrollment signatures is calculated and an interval for each feature is defined with a threshold value. Also, for each feature (X, Y, P, TS) of a signature, 1D DTWs in all permutations of the three enrollment signatures are calculated for each stroke individually. This results in four distance vectors (one for each feature), each with one distance value per stroke of the signature. The four distance values of each stroke are then added up, so it results in one distance value per stroke, which is used in the template as intra-enrollment distance reference [111].

The verification is done in multiple steps. First, the number of strokes (pen-ups) is compared to the template, and if it is not the same, the signature is automatically rejected. Then, each global feature of the whole signature is compared, and if any one of them does not fall in the defined interval, the signature is rejected. Then, the global feature comparison is done for each stroke individually, and if a certain percentage of strokes does not have all its features within the reference intervals for that stroke, the signature is rejected. If all these tests are passed, DTW is applied for verification by calculating the distance of the new signature to each of the enrollment signatures the same way as described above, and the resulting distances are compared to the template reference. If the difference is above a threshold, the signature is rejected [111].

The system, although designed for mobile phones, was tested on data recorded on different pen-tablets, namely the SigComp2011 [85] and the SigComp2013 [89] datasets, only the computation of the algorithm was done on mobile phones. All datasets included skilled forgeries, and an f-measure of 94% was reached, with computation of the algorithm in under six seconds (in most cases) even on old devices [111].

### Global Systems

**Combination of global signature features with global movement features** In the Hold & Sign system proposed in [29], the users are identified based on how they hold the phone (micro-movements) during signing and on the characteristics of the points being pressed

on the touchscreen, but not the produced signature image itself. In addition to the X-Y-Coordinates, magnetometer, accelerometer and gravity sensor are recorded while the user is drawing their signature. The system is a global system that does not use the actual time series data, but extracts statistical features from it in preprocessing. They reduce their resulting 93-feature vector (16 features for each movement sensor including high- and low-pass filtered versions of accelerometer, plus 13 touch features) using feature subset selection with recursive feature elimination to 10-11 features for each stance. Different machine learning algorithms were evaluated, with 1-class Multi-Layer Perceptrons (MLPs) performing best with a TAR of 95% and a FAR of 3.1% on random forgeries in activity fusion, or about 86% TAR with 0.1% FAR for each stance individually (80% TAR without feature selection). It is found that the system performs best in walking state, maybe because the movement sensors can gather more data in that state. However, systems need to have training data for every different body stance (sitting, standing, walking) in which the user can sign [29].

Histogram-based statistical signature modelling Histograms are widely used to capture attribute statistics in recognition tasks, such as object or offline signature recognition. For online signature recognition, the information that can be used to derive useful histograms includes X-Y trajectories, speed, angles, pressure, and derivatives of all [119]. In [119], such a system was developed where a feature extractor computes a set of histograms for each signature from which a feature vector is derived. From the feature vectors of multiple signatures, a template generator constructs a user specific template for enrollment, and the template is then used to verify the signature by a matcher based on a simple Manhattan distance. The advantage is that this is an irreversible transformation, so the signature template cannot be used to reconstruct the actual signature, and it can be done in linear time and requires only a small fixed size of storage, making it well suitable for mobile environments. Also, it is a one-class classification approach like DTW, so no training set from a large number of users is required [119].

The features of the signatures are recorded at an inconsistent rate, but time normalized (sampled to uniform rate by interpolation) in preprocessing, and therefore carry implicit time information. Also in preprocessing, signature stroke concatenation is applied, where the next stroke is translated so that its origin lines up with the end of the current stroke. The processed time-series are converted to a sequence of vectors in the cartesian coordinate system and their derivatives are calculated. Then, each cartesian vector is converted to the polar coordinate system. The histograms are computed from a sequence of vectors that includes the cartesian and polar coordinates and the pressure information for each order of derivative. More information on the construction of the underlying vectors can be found in the paper directly [119]. One- and two-dimensional histograms are calculated to capture the distribution of individual attributes and the distribution of relationship between attribute pairs. It has been shown that 1D-histograms provide valuable additional information, but 2-D histograms are more effective in terms of discrimination power. 16 of the 21 histograms used are relative frequency histograms that ignore the length of the signature, but 5 were empirically chosen to use absolute frequency because it improved recognition rate. The feature vector of a signature is then the concatenation of the vectors of bin frequencies for each histogram. [119]

The user template is generated from multiple signatures that are collected during

user enrollment. From the enrollment signatures, a quantization step size vector is generated by computing the standard deviation of each feature in the feature vector and using a multiple of it as step size. Each of the enrollment signatures is then quantized by dividing the features through the according step sizes in the quantization vector. The average of these quantized signatures is stored together with the quantization vector as the signature template. To verify a new signature, the signature is transformed to a feature vector as described above, and then quantized using the quantization vector that is stored with the template. It is accepted if the Manhattan distance to the template is less than a predefined threshold, otherwise it is rejected. [119]

The method was tested on the SUSIG [74] and MCYT-100 [107] databases (included pressure information, non-smartphone), as well as on a dataset that was recorded by the authors of the study on a smartphone (did not include pressure information). On the MCYT-100 dataset, the EER on random and skilled forgeries was 1.43% and 5.74% respectively with 3 signatures for enrollment, and consistently lowered with more signatures to a minimum of 0.35% and 2.72% respectively with 20 signatures for enrollment. On the SUSIG database, EERs of 2.94% and 6.08% were reached. [119].

For the mobile dataset that was recorded for the study, it is important to note that participants were given a financial incentive to sign as consistently as possible by offering a $40 reward to the 10 participants with the most consistent signatures, which might have led to more consistent signature samples than in real life. The signatures were recorded remotely over six sessions of 5 signatures that were each between twelve and 96 hours apart. The recordings only included x-y-coordinates and a time stamp. The average EERs in leave-one-out intra-session classification - meaning 4 of 5 signatures of the subject in a session were used for training, one (positive) for test, five times, for each subject - were in a range between 3.64% and 2.67% for the six different sessions. In inter-session tests, the performance deteriorated in most cases, with an increase in EER from 3.18% on intra-session classification of session two to a range of 5.04% to 7.04% EER when using models built from signatures from the immediately preceding session to classify signatures of a session. The more preceding sessions the training set was sampled of, the better the performance became, with an EER of 2.89% EER when classifying session six with a model that included signatures from session one to five. In general, it was found that the further the train and test sessions were apart, the higher the EER tended to be. This suggests that template-aging is relevant and that a template-updating mechanism could be required to stabilize performance. The inter-session performance also significantly increased when signatures from more than one session used for training [119].

**HMM-based verification in an uncontrolled cloud scenario** In a cloud scenario, the users record signatures as they want, in different stances and environments, and holding the device differently in their hand or on a surface. The authors of [1] attempt to combat this problem of mobile conditions by enhancing the normal signature with additional handwritten information during enrollment, namely initials, name-surname, date and place of birth. The signatures are recorded without detailed instructions on how to sign for the users on purpose to achieve real-life conditions. The additional information is considered to be signature types as well, as it is written by hand and commonly appended to a contract in real-life situations. Recording is done on an iPad, only x/y-

features are recorded, and each type of signature is recorded on a separate blank screen, not all at once. For each signature type, there are also skilled forgeries recorded [1].

From the five individual types of signature, seven hybrid combinations of two or multiple types were created by concatenation, resulting in a single time series. The personal entropy - variations in the signature of one person - is the highest for initials, followed by the usual signature, and then the other three types of signature considered. For hybrids, the more types were included, the lower the personal entropy was found to be. To assess the approach, a HMM-based statistical verification system is used that was already tested on different other signature databases. It is trained on 5 signatures and tested on a set of ten skilled forgeries and ten genuine signatures [1].

The performance of their approach on just the usual signature lies between 7.17% EER on high personal entropy (shorter and simpler, more variable signatures) subjects and 5.93% EER on medium personal entropy subjects. The name-surname signature type performed very similar to the usual signature for medium- and low-PE subjects (more complex and stable signatures), but better for high-pe subjects. The (very short, high entropy) initials performed significantly worse in all cases with EERs between 14% and 16%, even in the case of high-entropy subjects with already short signatures. These results show that complexity, while being helpful in the case of the significantly less complex high-pe signatures, is not sufficient to enhance the security of a signature, as the (still less complex) medium and low-pe signatures perform equally well. They also highlight the importance of the ballistic aspect of signing for resisting attacks [1].

Hybrid combinations of the usual signature with different additional types resulted in better performance, especially when a total of three or more types were combined. In that case, the worst EER was 1.7% on high-pe subjects for one hybrid type, with all other scenarios and hybrid types below 0.5% EER, and even perfect scores on low-pe signatures. Adding date and place information to the usual signature was most powerful, as it adds additional information and keeps the ballistic signature aspect [1].

### Reducing the Template Aging Effect

In [137], the effects of long-term template aging and the efficacy of different strategies to mitigate these effects are studied on a long-term signature database with 6 sessions captured over a time span of 15 months. This is especially relevant as users usually do not have to sign something every day. Their idea is to update a template with new genuine signatures over time that the user has signed when using the system, but in a human supervised scenario (eg. signatures drawn in a bank where the employee has seen the ID) so it is made sure that none of these signatures are impostor signatures. Two different strategies are analyzed: Adding all available signatures across time to the original enrollment signatures, or not considering old genuine signatures anymore so the model is always up-to-date [137].

While the dataset was recorded on a signing tablet, pen-related features were dropped to come closer to the feature set that would be available on mobile devices, such as smartphones and tablets. Pressure however was still included. From the raw features, the same 23 time functions were extracted as in [93, 94, 134, 135], and optimal subsets were generated using SFFS subset selection. Results are obtained using three common verification algorithms, namely HMM, GMM (gaussian mixture models) and DTW. In

the end, a fusion of all three systems is also tested, where the sum of the matching scores is computed, as this is a successful approach in related work.

On skilled forgeries, a clear improvement in performance can be observed for all three systems when the elapsed time between training and testing signatures is reduced and only training samples from one session are used. DTW is performing the best, with a reduction in EER from 11% EER to below 1% EER from 15 to 0 months time difference. On random forgeries, this is not the case for GMM and HMM, where the only significant improvement in performance occurs when training and testing signatures are from the same session, and there are only small improvements for DTW, with a drop from 2% to 0% EER [137].

For template update strategies, the best strategy for HMM and GMM was found to be to use all available training signatures, or at least a few of each available session, for template generation. This performed better than using more signatures per session, but only from the sessions closer in time, and resulted in EERs of 4-5% on skilled forgeries, and near-perfect results on random forgeries. For DTW, in contrast, the optimal strategy is to only consider a few signatures from the last sessions closer in time to the test (8 signatures, two sessions performed best), and EERs of 5.5% and 0% were reached using it on skilled and random forgeries, respectively. Compared to baseline systems that only use the original four enrollment signatures, using the right template-update strategies resulted in an average relative improvement of 40.9% and 44.2% EER for random and skilled forgeries, respectively. A fusion of the three systems proved to be effective, with an additional absolute drop in EER of 2% to 2.1% compared to the best individual system on skilled forgeries, and the same performance on random forgeries. This fusion-system was deployed in a pilot project to the Spanish banking sector.

### 3.3.3 Touchpad

There is hardly any related work to be found on signature recognition on notebook touchpads. The only study that has investigated it is [6] from 2008, where the authors built a system to identify users over a network based on drawing captured on a Synaptics touchpad on a laptop, using x-y-channel features and pressure values reported by the touchpad, as well as the number of units of mouse motion per point. The users did not sign with their actual paper signatures, but with a personal paraph they were asked to invent and practice before. No information about if and how visual feedback to the user was given is included. Using a modified Hausdorff distance, which can deal with variable length series, on almost raw paraph time series data (without interpolation), they achieve an EER of about 12% on random forgeries, and about 15% on forgeries made by imitating the paraph after watching the signer over the shoulder [6].

In [22] and the follow-up study [23], a notebook was included in the database, but signatures were drawn on its resistive touch screen display, not on its touchpad. In a study from 2005 [82], the authors describe a system where signatures are drawn using the computer mouse, which is different from signing on a touchpad. In a United States patent application of 2006 [96], a system is suggested where the user signs on the touchpad to authenticate preferably using a pen that can detect multiple pressure levels. However, no details on how the system would actually work are provided, no tests or results are stated, and the system does not seem to have been implemented anywhere.

In another patent of 2008 [120], signing on a notebook display with a pressure sensitive pen is suggested, but not on a notebook touchpad. Again it is unclear if this system has ever been actually implemented.

Other than for the aforementioned work, we can only look to systems where only X-Y-channel features were used, but the results are not really translatable. For signatures on a display, as in [22] and [23], users sign with a finger on a small surface, but have good visual feedback, which was found to be an important performance factor in [23]. For signatures acquired on drawing tablets, users might not have visual feedback, but the signatures are still acquired on a surface intended for drawing and signing. Moreover, they are signing with a pen, and although finger and pen can be considered the same biometric modality [23], it is unclear if that extends to finger-drawing a signature without visual feedback, which is less familiar than writing it with a pen. Also, touchpad-specific aspects, such as the possibility of indirect visual feedback using the display, the effects of the different shape, different precision, and using the touchpad in an uncommon way, are not represented in either case. So even if these systems do not use additional data sources, many differences are still not accounted for. However, such studies can serve as an information source.

## 3.4 Signature Datasets

### 3.4.1 Existing Datasets

There are many publicly available datasets for signature verification, most of which were acquired on a drawing tablet [94] or a PDA. The most noteworthy are the ones used in the previously mentioned Signature Competition (SIGCOMP), on which the participants [24, 85, 89, 90] build and test their systems. Their main focus is on signatures with a pen on tablets, and they comprise X-Y, pressure and pen information. Also important in the field of pen-tablet datasets are the SVC [148] database, where users were asked to sign with a new signature they had to practice before instead of their real one, and the SIGMA [2], SUSIG [74] and BIOMET [48] databases, as well as the BioSecure databases used in [61]. The datasets consist of online signatures of 30 to more than a hundred people, and usually include skilled forgeries in the test sets.

Much fewer datasets exist for mobile phones, and not all of these have been published. The Research Group in Pattern Recognition and Digital Image Processing (RPPDI) SigData Dataset [16] contains a relatively small dataset of 15 users signatures that were recorded on a Galaxy Note device, but only with the devices pen instead of a finger, and without recording movement data. The dataset recorded in [22] is a small (11 participants) mobile dataset recorded on multiple devices with finger and stylus, but doesn't include movement data either. The one in [29], which is mainly based on 3D orientation and acceleration data for recognition, comprises of signatures from 30 participants, does not include any skilled forgeries, and is also not publicly available. The database in [104] only contains acceleration data, but no other movement data in addition to the signature image, and has samples of 20 subjects. The authors of [94] also recorded a small (25 users) database of signatures on a mobile phone which is available online, but it does neither include movement data nor skilled forgeries. In [64], it is stated that there is a lack of an extended dataset that is publicly available, with signa-

tures recorded on devices of different vendors, different screen sizes of mobile devices, acquisition over multiple sessions, and that includes soft biometrics such as touch size or phone movement captured with embedded sensors [64]. This is consistent with our own experience - to the best of our knowledge, there is no publicly available mobile dataset that contains a complete set of X-Y and movement features.

There are no databases of real signatures recorded on notebook touchpads that we know of. We are only aware of one dataset of finger-drawn paraphs that was recorded in 2008 as part of [6]. However, the users were specifically asked not to sign with their actual signature, and to instead make up and practice a paraph for the purpose of the experiment. Also, to the best of our knowledge, that database is not publicly available.

### 3.4.2 Signature Recording

The number of participants in the datasets varies widely. The smallest pen-tablet database includes signatures of 30 participants [90], but there are also multiple databases of over 100 people [74, 148], the largest one with 265 participants [2]. Mobile databases, as already described above, are smaller, with the largest ones including 25 [119], 30 [29] and 43 [23] participants. In [23], participants younger than 16 were rejected, as their signature is claimed to be less consistent. The signatures, especially in the mobile area, are usually recorded over one [29], two [104, 119] or at most three [22] sessions, whereas for pen-tablet databases, there are multiple datasets with two [74, 148] three [2, 48], four [89] and even six [137] recording sessions. The time between sessions ranges from only one day [89] to one [23, 74] or two [22] weeks, a month [104] or even more [137], and can also be variable [94]. Usually, between 10 [2, 74, 90, 94, 148] and 20 [22, 104] signatures are recorded per session, with the lowest per session being only 4 signatures [16]. In [89], the amount varied between sessions, with 12 signatures in the first and 9 in each consecutive one. Participants are given the possibility to reject and redo a signature instead of accepting it into the database if they are not satisfied with it [22]. If skilled forgeries are included, each forger usually draws about 5 forgeries per subject [24, 48, 74, 104], with four [24, 85, 104] to five [48] forgers per subject. However, there are exceptions where only one forger per subject provided between 5 [74] to 10 [2] forgeries, or where each of 4 forgers provided up to 9 forgeries per subject [85]. Forgers are participants that also provide genuine signatures [22], and they can practice each signature until they feel comfortable imitating it [2, 16, 24, 74, 148]. The forgery scenario is usually that the forger either observes someone else drawing their genuine signature multiple times [22], or that the forger is shown an image of another participants signature to imitate [2, 136], ideally of multiple different samples to practice [2].

Sampling frequencies can widely differ between datasets. In the two smartphone-based systems [104] and [29], signatures are recorded with 30 Hz. In [29] it is argued that faster recording would only introduce more noise, and that some sensors can not provide readings at faster rates at all, which is also backed by [76], where it is stated that while most acquisition devices operate between 100-200 Hz, the bio-mechanical sequence related to signing has a maximum frequency range of 20-30 Hz. In the purely movement-based system [129], the sensors are recorded at 100 Hz, while in [154], which is also purely movement-based, recording was done at only 4 Hz. In relevant pen-tablet databases, signatures are recorded with between 75 Hz [90] and 200 Hz [2, 24, 85, 89].

## 3.5 Usability

There is not much work on the usability of mobile signing systems yet. We know of four studies that have investigated it specifically and are relevant for mobile. Three of them were published within about a year by the same authors as follow-ups or extensions in a similar mobile settings [19–21], and one of them was done on a touchpad, but some of the findings are relevant to mobile [55]. The other studies discussed here have focused on certain aspects, such as screen size or visual feedback, but from a verification performance rather than a user perspective [23], or have provided interesting usability information as remarks that came up while conducting their study [29, 104].

Overall, people are comfortable using dynamic signature verification on mobile devices [20, 21]. In a study of 56 participants that signed on a smartphone with a stylus, subjects scored the signing experience a 4.13 out of 5 overall, and 87% stated that they would use the system daily [19]. Based on results obtained on mostly tablet-sized devices, users tend to feel more comfortable using a stylus rather than a finger for drawing the signature [20], although there was no significant difference in verification performance between the two modalities [23]. In 2014, about half of the users had no prior experience with handwritten signatures on mobile [19]. Errors made while signing, such as repeating strokes or signing out of bounds, were high in the first scenario of the first session doing it, but then dropped off immediately to a stable level across all later sessions and scenarios, indicating that it may be a good idea to add a few practice signatures before actual recording of the first template signatures [21]. Also in [19], the deletion rate of signatures stayed the same between phases of recording (about 8% of signatures). 75% of the deletions were made because of disliking the signature, with signing (partially) in the air in second place (12%), and the rest divided between having repeated strokes and signing out of bounds. The time spent signing decreased the more phases the user had completed, and user rated it 4/5 points in the questionnaire, showing habituation and indicating that it taking too long is not a big concern [19]. In a study done on the usability of signing on a pen-tablet, no differences between age groups in their ability to enroll in the system were found. Also, there were no differences in repeatability of signatures (using performance variation as metric) and standard signature features (using variation in feature value as metric) for different age groups within and in between sessions. The exception are time-related features such as execution time, velocity and acceleration, that show instability with age [55]. Pressure or stress while signing was shown to affect usability negatively, causing a higher number of errors in the signing process [20]. It reduced the time for signing and worsened similarity scores between signatures, but it was not a major drawback for recognition performance [19].

Visual feedback while signing is one of the most important aspects of recording signatures. In [23] and [20], it was found to be very important to users, and that not having it leads to a significant deterioration in performance and usability. This is backed by [36], where it is stated that users do not feel comfortable without it, and by [80], according to which a lack of visual feedback leads to higher variance, influences false rejection rates, and that providing it would improve signature stability [80]. Also, latency while signing causes annoyance and affects the verification performance negatively [20].

A more debated topic is the issue of screen sizes and signing on screen. One study [150] calls signing with finger difficult and uncomfortable for users, and in both [76] and [150]

it is stated that small display areas affect user interaction and lead to larger intra-class variability, which negatively affects the false rejection rate. According to [94], a lack of space affects signature complexity, which is an estimator of the difficulty for its imitation [36], may cause hesitation marks, and reduce fluency. This would also explain the lower importance of dynamic features (time, speed and acceleration) on mobile that was found in [36], and which the authors considered to be due to the unfamiliar surface and signing device. These findings are contradicted by [76], which states that while traditionally it has been thought that intra-class variability is much higher in mobile scenarios, it has always been evidenced with limited acquisition devices (as in [95]) such as PDAs, not with modern phones with good touchscreens, and that the quality of the acquired signatures can show high disparities depending on touch screen quality [76]. Also, [23] tested their system on devices with various different screen sizes, and they did not see a performance deterioration based on screen size, with small-screen devices even tending to perform slightly better. In a usability questionnaire in [29], the participants responded that they were positive and comfortable signing with their finger on screen, and there were no complaints about the screen size. In [97], the users even reported that they felt uncomfortable signing on a tablet because of its larger screen size. Also, in a study of 274 participants on pen-tablets, all signatures were less than 40mm in height [55], which fits comfortably on most modern smartphone screens.

There is not one optimal position for signing. Stylus-based experiments with mostly tablet-sized devices obtain better results if user sits on a chair and the device is resting on a table, which is also the most preferred scenario by users [20]. For such devices, users signed mostly seated (87.5%) with the device rested on the table, 7% signed standing with the phone lying on a surface, and only 5.5% held the phone themselves [19]. Finger-based devices designed to be held in hand return the best results when users handle the device without support themselves [20]. This is also backed by others. While holding the device in hand was found to cause a performance penalty in [61, 94] when using a PDA, more recently, finger-based mobile devices for signing returned the best results when the user was holding the device without support. Only pen-based signing worked better when the device was on a table and the user was sitting in a chair [36]. Also, lying the smartphone on the table instead of holding it in hand negatively affects accuracy when the accelerometer is used as input [104]. The authors of [104] state as a sidenote that all participants in their study signed holding the smartphone in their hand when recording the test set, although position was not mentioned, indicating that it is a natural behavior with modern smartphones.

## 3.6 Summary

The approaches to signature verification can be classified as either static or dynamic systems. Static systems use only the signature image as input for classification with HMMs, SVMs and neural networks, and achieve mean results of about 10% EER on skilled forgeries, with variations also due to different experimental protocols. Nowadays, dynamic systems are considered to be better for automatic verification. They have much more information available to work with, because signatures are recorded as time series, from which additional information such as signing speed can be extracted. Often, additional features are also recorded alongside the signature image, and their individual

importance is one area of active research. Traditionally, dynamic signatures are drawn on graphics tablets with a pen, which have additional pressure and pen information available that can be used to improve verification performance. With the advent of smartphones around the year 2010, automated dynamic signature verification on mobile devices has been established as a new, related field of research. In many regards, they are similar. The types of algorithms used for classification are the same, with DTW as the current standard and HMMs as a second popular option, and other classifiers such as SVMs or Neural networks only thereafter. Results on skilled forgeries range between 2% and 10% EER depending on classifier, quality of signatures and forgeries in the dataset, and experimental protocol, with pen-tablet approaches tending to perform a bit better due to the more familiar, stable writing experience and the additional data that can be recorded with a pen.

The main difference between the two is in the types and quality of features that are available, because mobile signatures are usually drawn with a finger on a touchscreen in an unstable handheld scenario, with no pen or pressure information available. Nonetheless, related work on mobile signing has been heavily focused on X-Y channel data. While this is natural considering the fields origins in stationary dynamic and even static signature verification, it neglects alternative dynamic features available on mobile, but not on pen-tablets, such as 3D movement data, that can be acquired unobtrusively while signing. Also, it neglects the potential advantage that static pen-tablet systems have over mobile devices on the X-Y channel due to the higher precision and better familiarity of subjects writing with a pen, and the additional features that a pen provides. Therefore, some consider a more holistic approach that is less reliant on the possibly weaker finger-drawn signature image compared to pen-tablets, and is instead playing to the advantages of mobile devices by exploring additional behavioral features that are potentially relevant for classification.

Buriro et al [29] and Nam et al [104] have already proposed such systems that utilize some form of 3D orientation data together with X-Y features, and both have achieved promising first results. In [122], the authors suggest that not only using the movements of the finger on the touchscreen, but also those of the device in hand could be a potential solution to the lack of resolution on mobile compared to traditional desktop applications with signing tablets. They state that movement data could be potentially helpful to improve robustness against forgeries and consider the inclusion of gyroscope and accelerometer readings to be a promising future line of research [122]. In [64], the authors also suggest that hand-movement behavior based on data provided by embedded sensors such as gyroscope or accelerometer could be taken into account [64] for verification. They further point out the lack of an extended public dataset with signatures recorded on devices of different vendors, different screen sizes of mobile devices, and acquisition over multiple sessions, that includes soft biometrics such as touch size or phone movements that would allow for investigation of these proposals.

Beyond phone movements, according to [122] there are investigations to find out if pressure information can be emulated by the number of pixels touched while signing, but it has not yet been demonstrated if such emulation is is useful or equivalent to the pressure acquired by a pen. Later works [36, 64] state that touch size and shape of the contact area have not been considered so far, but could be interesting to investigate. Finally, device interoperability also needs more research [64].

# Chapter 4

# Dataset

We recorded two datasets of real handwritten signatures drawn with a finger on mobile devices. Both the touchpad and the smartphone dataset comprise of genuine high-resolution signature data recorded as time-series, as well as skilled forgeries crafted by multiple forgers. The smartphone signatures also include movement data recorded on different embedded sensors while signing. They are designed to be easily extensible. We are publishing our participant materials in two languages, our exact settings and recording process, and all recording tools that we designed and implemented. The combined database consists of a total of 10400 signatures by 73 individual subjects. To the best of our knowledge, it is the largest mobile signature database in the world.

## 4.1 Mobile Dataset

The mobile database is a collection of real human signatures, drawn with a finger on screen while holding the device in hand. It comprises of 6300 genuine handwritten signatures, recorded over two sessions for each of the 70 participants. In each session, the signatures were captured in three different positions: sitting, standing and walking. 15 signatures were recorded per position, resulting in a total of 45 signatures per session and 90 signatures in total per subject. The database also contains four sets of five skilled forgeries per subject. Each of the sets is crafted by one of four different forgers recruited from the study participants. A fifth forger was asked to imitate the forgers own signatures, which they could not fake themselves. In sum, there are a total of 20 skilled forgeries per subject, or 1400 skilled forgeries for the whole dataset, resulting in a total of 7700 signatures in the mobile database. The statistics are also shown in table 4.1.

| Type | Amount | Subjects | Sessions | Positions | Nr./Position |
|---|---|---|---|---|---|
| Genuine | 6300 | 70 | 2 | 3 | 15 |
| Forgery | 1400 | 4 | 1 | 1 | 5 |
| Total | 7700 | 70 | 3 | 3 | – |

**Table 4.1:** The contents of the mobile dataset.

### 4.1.1 Features

The mobile database contains measurements of two biometric modalities from a total of seven different physical or software-based smartphone sensors. Every recorded signature sample consists of a set of 18 features recorded by those sensors as time series. This makes it the most comprehensive dataset that we are aware of available to date, and it allows for both movement- and touchscreen-based research on signature verification. Additionally, we store metadata about every signature, such as participant and recording device information. All data is recorded with a normalized sampling rate of 50 Hz, more on which can be found in section 4.1.2. Below, the different sensors are described, grouped by biometric modality, starting with the features acquired from the touchscreen.

- **X-Y-Coordinates** form the signature geometry (the signature image) and are recorded as time-series. They are a part of every traditional signature database that we are aware of. However, because the signatures are drawn with a finger, no pen-up coordinates are acquired, recording "null" instead in times when the subject did not touch the screen.
- **The velocity on the X- and Y-axis** (VX and VY) with which the signature is drawn. These measurements are the ones calculated and reported by Android, but it is also possible to compute the speeds as the first derivates of the X-Y-coordinate series.
- **The estimated size of the finger contact point** on screen (touchsize). This has been suggested in related work [36, 64, 122] as a possibly important feature, and we are the first database to include these measurements. Touchsize is acquired from the Android Framework using the method `MotionEvent.getSize()`, which "Returns a scaled value of the approximate size for the given pointer [157]".

The second biometric modality that is included in our database is the position and movements of the phone in hand. These measurements are recorded using the embedded sensors of the device, making it invisible to the user and requiring no additional effort. Below, we list the included features and describe our reasoning for the inclusion of each of the sensors.

- **Linear accelerometer**, which is semantically the same as high pass filtered raw accelerometer values, for examples as used in [29]. As it can be acquired directly from the Android framework, where Gyroscope readings are used to support better separation and more sophisticated filtering is applied, they are likely more precise than recording the raw accelerometer and processing them later. It represents the actual user-caused component of acceleration.
- **Gravity sensor**, which is semantically the same as low pass filtered raw accelerometer values as seen in [29], with the same benefits as for the linear accelerometer. It represents the gravity component of acceleration, indicating the position of the phone in hand by showing how gravity affects each axis.
- **Gyroscope**, which is commonly used in movement-based authentication systems such as [129] or [154], and provides the rate of rotation around the devices axes. Its readings are also used by the Android framework to separate acceleration into linear acceleration and gravity.
- **Magnetometer**, as used in [129], [154] and [29], can be relevant because of pat-

terns in the changes in the measured magnetic field while signing which can be identifying, although the values themselves are different depending on where the subject is. It has therefore been recorded and is included in the public dataset, but as it has been shown to be of no benefit at all and perform significantly worse than gyroscope and accelerometer in [129], it is currently not a part of our actual training data.

We have chosen to record gravity and linear acceleration components separately due to the likely higher precision provided by the sensor fusion used in the Android framework, compared to simple high- and low-pass filterings in preprocessing, as done in related work [29]. To get the combined measurements, semantically the same as raw accelerometer readings, the readings of these two sensors can simply be added up.

Besides the included sensors listed above, there are also two notable omissions from the set of movement sensors which occur in related work:

- **The Orientation Sensor** is used in [27] and [34] in addition to accelerometer and gyroscope. However, both these systems identify users how they take a phone call, which involves large orientation changes when moving the phone from neutral to the ear. For our purposes, this is not the case, and even the Android Documentation [157] states that it suffers from diminished accuracy. Also, it is based on the magnetic field sensor and the accelerometer, both of which we record, so the orientation sensor can safely be omitted.
- **The Compass**, as used in [83], is similar to the magnetometer in that it is not a real movement sensor, and that its potential value comes from the changes in orientation of the phone relative to the magnetic north. However, it is likely that it also suffers from the same problem as the magnetometer in terms of performance. As we already have the magnetometer, which is more commonly used, and because the gyroscope can provide us with probably better rotation information as well, the compass can safely be omitted.

### 4.1.2 Sampling

The signatures are sampled at a fixed, uniform rate of 50 Hz, and all features are aligned with each other in processing. The embedded sensors are recorded with the Android sensor delay set to `SENSOR_DELAY_GAME`, which provides measurements at 50 Hz. Touch data is recorded as reported by the phone (at least 60 Hz), as it is not possible to set a fixed recording rate there, and normalized to 50 Hz in processing.

Our sampling rate is higher than the sampling rate in [104] (30 Hz) and equal to [29] (50 Hz), both of which are smartphone-based systems similar to ours. In [29] it is argued that even faster recording would only introduce more noise, and that some sensors can not even provide readings at faster rates at all. Our choice of sampling rate is also backed by [76], where it is stated that while most (pen-tablet) acquisition devices operate between 100-200 Hz, the bio-mechanical sequence related to signing has a maximum frequency range of 20-30 Hz. In the purely movement-based system [129], the sensors are recorded at 100 Hz, while in [154], which is also purely movement-based, recording was done at only 4 Hz. In relevant pen-tablet databases, signatures are recorded with between 75 Hz [90] and 200 Hz [2, 24, 85, 89]. However, these rates are not possible on Android and likely not necessary in our case.

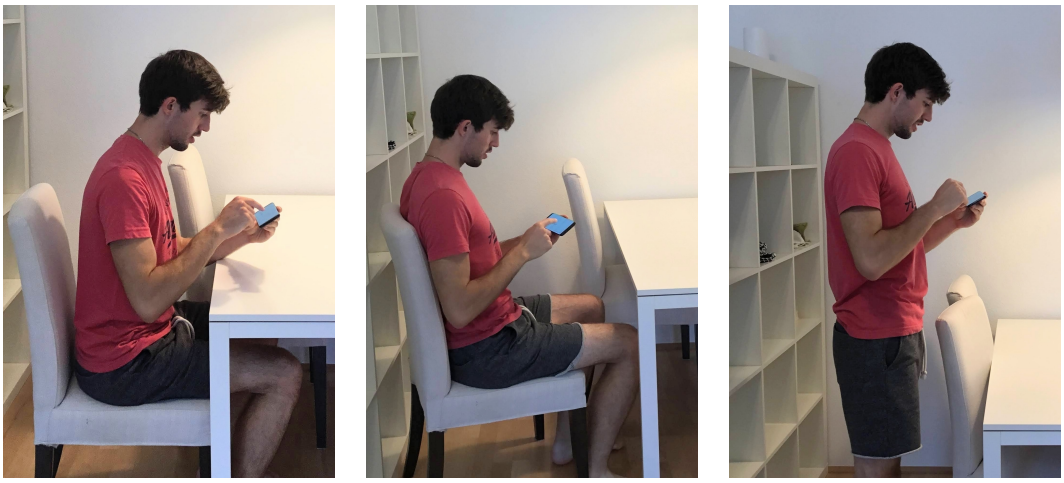
### 4.1.3 Processing

The recorded signatures go through a series of processing steps. At first, the recordings for the screen and for each sensor are interpolated using linear interpolation. Then, they are sampled to exactly 50 Hz to align the measurements of the touchscreen and each of the different movement sensors and create well-structured records with values for all features at the same time every 20ms. The process of linear interpolation and sampling to obtain an exactly fixed recording rate has already been used in related work, for example in [4]. Times while signing where the user did not touch the screen (finger in the air) are represented with null-values for the X-Y-features in each entry, as the movement features are still being recorded in the meantime. The signatures are cut to the time between the first and the last touch point, so that no further sensor readings are included before the start or after the end of the signature. Also, stroke numbers are generated for each stroke and added to the dataset.

### 4.1.4 Setting

The goal of the dataset is to provide signatures that are as close as possible to ones that would be recorded in real-world usage. Therefore, the participants were provided with detailed instructions on how to use the recording app and in which ways to record signatures, and were then asked to record the signatures at home on their own using their personal devices. A subset of the participants were visited at their homes and asked to sign on a smartphone provided by the authors instead of using their own, with everything else being equal. All signatures recorded on the SMG950F device, and only those, are part of this subset. To make sure that the same instructions were followed by all participants, general information on what to do to take part was only given in the participant instructions and never described informally to the remote participants before they read the document. However, all participants were able to ask specific questions at any time about the process if something was unclear, either in person or over the phone. We specifically asked some participants that were less adept in using their smartphones if the instructions were clear enough for them both while recording the preliminary datasets in study preparation and over the course of the main study, and all responded positively. The few questions that were asked mostly pertained to details of the recording process, such as how much more time between the two recording sessions was acceptable. The setup required everyone to read the whole document, and accidental errors (such as forgetting to record in different positions or recording to few signatures in one position) would have been clearly visible in the signature meta parameters, as users were required to leave the signing screen and select the next signing position manually. For the SMG950F device subset, we were able observe the recording process ourselves. For more information about the recording process, we included the participant materials. The study instructions are provided in the appendix for reference in both their international English and the German versions. The complete list of devices with their respective screen sizes, release dates and touchsize qualities is available online.

Recording of the signatures was done in three different positions: Sitting, standing, and walking. All participants were asked to hold the phone in their hand while signing. In figure 4.1, the different possibilities of holding the phone while recording are depicted.



(a) Sitting stabilized (b) Sitting unstabilized (c) Standing and walking

**Figure 4.1:** Recording signatures in different positions. While sitting, both recording with the arm leaning on the table (b) and recording with the arm free (a) were permitted, as long as the hand itself was not rested flat on the table. When standing or walking, the phone was always held free.

While aseat, both holding the phone in hand while leaning the arm on the table (4.1a) and holding the phone freely (4.1b) were allowed, which ever way the subject felt more comfortable. However, the subjects were asked not to rest their hand or lay the phone flat on the table in order not to impair the recording of phone movements. While standing and walking, it was only possible to hold the phone freely (4.1c). A total of 15 signatures were recorded per position per session, and it took participants between five to ten minutes to record all three sessions. After each stance, participants had to leave the signing screen in order to select the next recording position in the app settings. Other than that, no fixed breaks were required. If they felt like a signature did not look right, participants had the possibility to undo it and try again instead of saving.

All signatures were drawn with the finger on the touchscreen of the device. While we considered to provide subjects with a pen that works on capacitive touchscreens, we decided against it because it would have made signing while standing and walking much harder. Also, we wanted to create a scenario that was as close to reality as possible, and most people do not have such a pen at home. In [23] it was shown that signatures with a finger can be considered the same biometric modality as signatures with a pen. In [94], it was found that, when no additional pen features are recorded, using a pen does not result in a recognition advantage over signing with a finger. For recording the signatures, the recording app automatically switched to landscape mode in order to best use the available screen space. This is in contrast to settings in related work [19, 94] where signatures were drawn in portrait mode, not utilizing the limited space well.

The recording sessions were spaced at least seven days apart, with the mean time between being 13 days, which is the delay that is usually also used in related work. Figure 4.2 shows the distribution of time between sessions for all participants. Due to

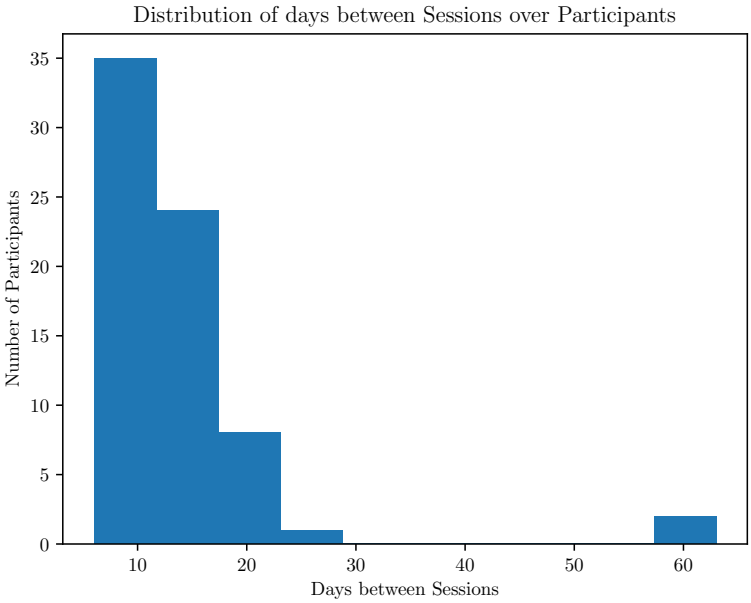



**Figure 4.2:** Time between recording sessions.

the way the study was conducted with the setting of self-recording at home, it was not possible to enforce a stricter policy with an exact time between sessions. However, that is also not the case for datasets in related work [94], because the time variance in a signature is captured no matter if it is one or three weeks apart. Also, in reality, users do not follow a fixed schedule of signing exactly every few days.

Because the participants used their own devices, we were able to record signatures on a wide range of Android smartphones. In total, 28 different devices were used, which to the best of our knowledge makes it the highest number of any database available to date. Figure 4.3 shows all the phones that were used and their respective frequencies in the dataset. Figure 4.4 provides more details about those devices. As can be seen, all of the phones are models from the recent years, with the oldest model being one released in 2015. Therefore, it is safe to assume that the recorded measurements are of the same high quality as they would be on phones released today, as there have not been any major improvements in screens or movement sensors used in phones in recent years. One exception from this is the touchsize measurement, which seems to be unreliable even on some recent phones, as shown in 4.4. Qualities 0 and 1 are certainly unusable, as the phone does not report touchsize at all or only reports one constant value throughout the whole signature. For quality 2 signatures, usefulness is unclear, as those phones do report a few different measurements per signature, but the measurements are extremely constant compared to quality 3 signatures, and it is uncertain if they reflect the real touchsize behaviour. Finally, quality 3 signatures are mostly recorded on high-end phones and exhibit a variance in measurements that would be expected for sensor readings. The assessment was done visually by looking at a few signatures

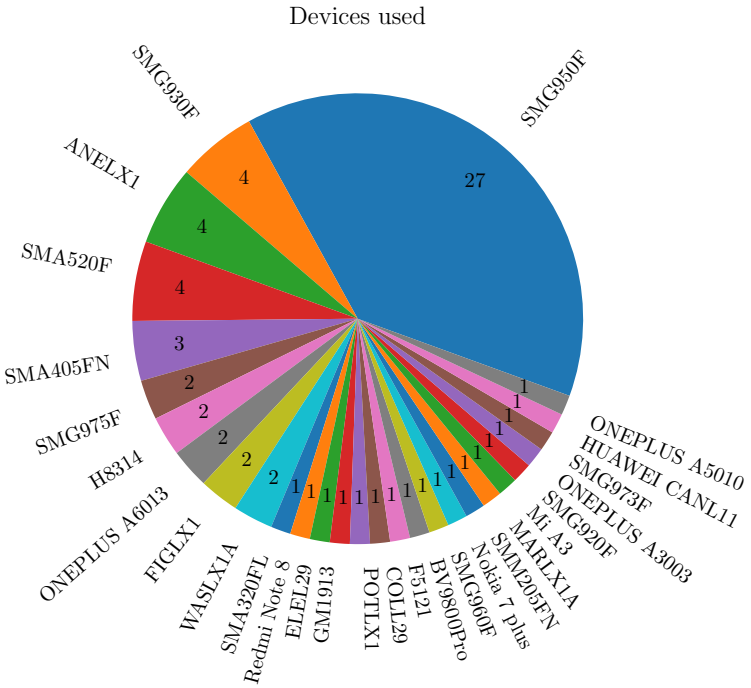



**Figure 4.3:** The devices used in the study.

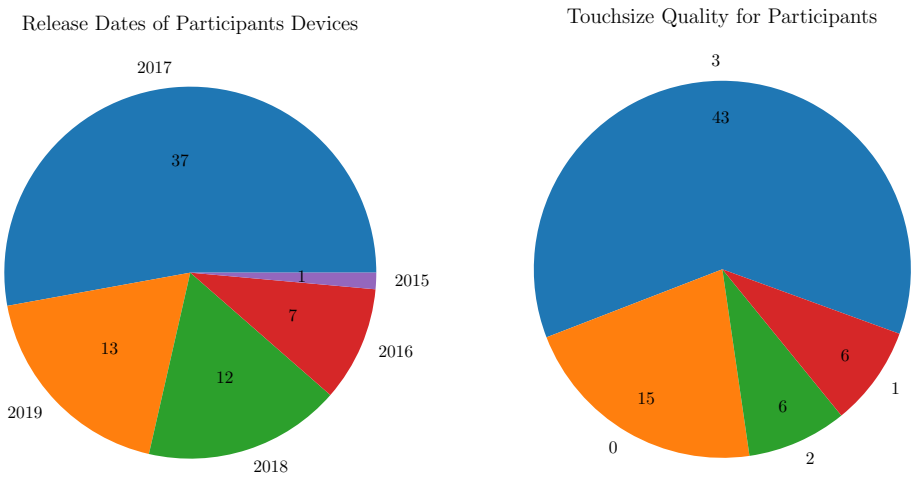



**Figure 4.4:** Phone release dates and quality of touchsize measurements. Quality 3 indicates that the measurements are likely accurate, 2 means that while it might be usable, we are not sure as the measurements are suspiciously constant, and 1 and 0 indicate that only constant values or no values at all are reported, respectively.

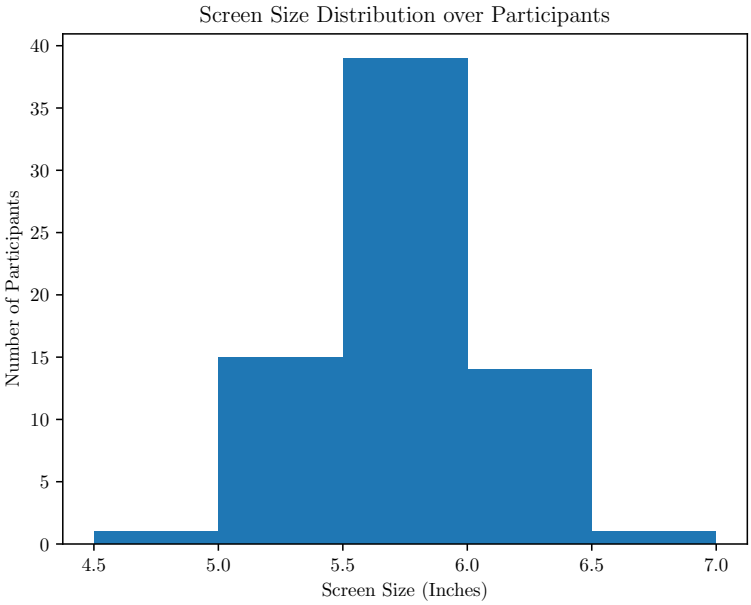



**Figure 4.5:** Display sizes on which signatures were recorded.

touchsize values for each device during quality control, and because the differences between the categories were already clearly visible, we did not implement an automated categorization process.

The display sizes of the phones used in the study range between 4.7 inches and 6.7 inches. The distribution of participants screen sizes is shown in figure 4.5. With this range, all currently common sizes of phones are well covered in the dataset. Finally, what is also interesting to note on the subject of devices is that of the 86 participants that were invited to take part in the study, four were not able to record the signatures on their own devices because their phones did not have a gyroscope built in.

For some participants, it was necessary to remove signatures from the dataset due to one of two cases. Either, the participant drew more than the minimum of 15 signatures per position and session, or, in very rare cases, an erroneous signature was saved accidentally. In the former case, we removed the last signatures of the position and session from the study dataset, so that for every subject, the first 15 signatures drawn in each session are included. Our reasoning for this is that the first signatures are important for scenarios where enrollment without practice is tested, while the signatures drawn after the 15 that were required are likely not any different from the ones before it. To rule out the latter case, we visually confirmed every signature in the database. In only 5 of the 6300 cases, recording errors (such as accidental saves after only writing one letter) were present. Those signatures were removed from the dataset and replaced with leftover signatures (where the participant drew more than 15 signatures) of the participant of the same session and stance. In a real application scenario, this would simply lead to the application asking the user for another signature, which is a minor thing to do. In

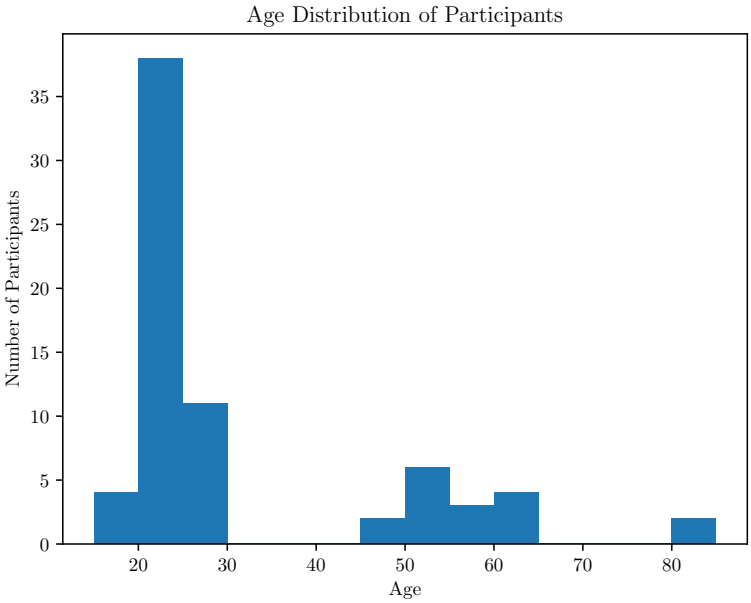



**Figure 4.6:** The age of study participants.

two cases, the respective subjects also signed with different versions of their signature within the same session. In one of these cases, we were able to re-record that session, while the other participant was removed from the study.

### 4.1.5 Participant Demographics

In total, 86 participants were invited to take part in the study, of which 70 are included in the final dataset. Three of them declined due to concerns about the publication of their signatures, and eight either did not respond or agreed to take part, but never completed the recording process. Four subjects wanted to take part but couldn't because they did not have an Android device. Four additional subjects had an Android device without a gyroscope, all of which were able to record their signatures on someone elses phone and were therefore able to participate. One subject did hand in their signatures, but had to be excluded because they used two different versions of their signature in the recording process. In the following section, we provide more details on the 70 participants that ended up in the final dataset.

More than a third of subjects in the study are female. However, this is still a slight under-representation compared to an almost equal distribution in the general population. With 11 out of 70 participants or about 15% of participants being left-handed, left-handers are slightly over-represented, compared to an overall frequency of about 10%. Most of the participants are German native speakers, the vast majority from Austria. A minority of 3 participants from The Netherlands has Dutch as their native language, and 2 participants from Greece are Greek native speakers.

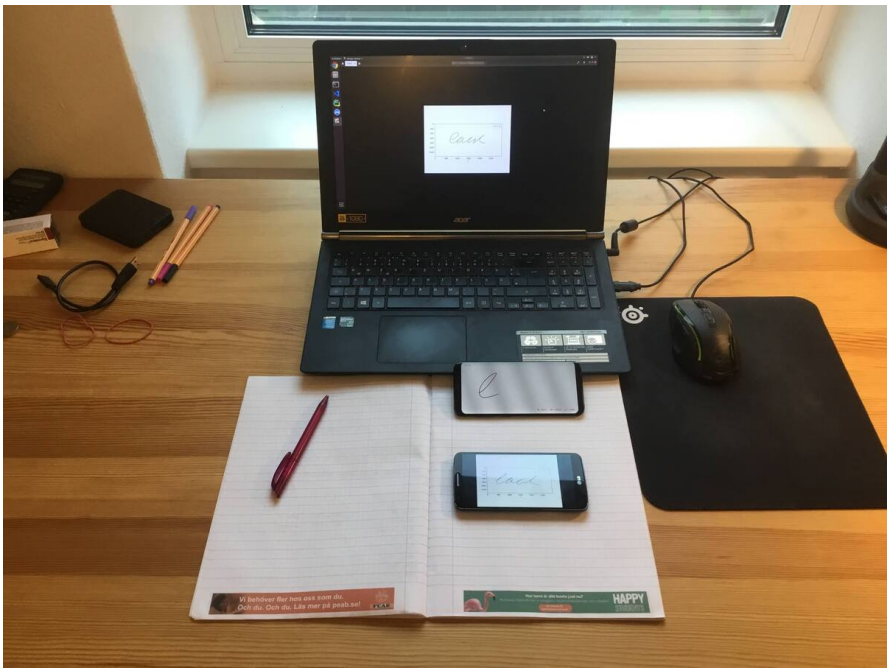


**Figure 4.7:** The forging setup the forgers had available.

In figure 4.6, the age of the study participants can be seen. While about half of the participants are people between the age of 20-25 that learned to use a smartphone at a relatively young age, there are groups of participants over 45 and even over 80 years old that did not. The youngest age at which subjects could take part is 16, as below that age the signature may be still to inconsistent [23]. There was no upper age limit to take part. The oldest participant is 83 years old, and the mean age is 31.6 years. It is important to note that the large majority of participants does not have a technical background, which would make it likely for them to be more used to digital signing than the general population. While most of the young participants are university students, they were recruited from a wide range of different fields. Of the older subjects, most are in non-technical professions.

### 4.1.6 Forgeries

For each of the 70 participants, four sets of skilled forgeries with five forgeries per set are included, resulting in a total of 20 skilled forgeries per subject. Each of a participants four forgery sets is drawn by a different forger. In total, five forgers took part in the study. All forgeries of a forger for a subject were recorded in one session, but not all forgers forged all participants signatures on the same day. Also, all forgeries were recorded in the sitting position only, with the same guidelines for hand position as for the genuine signatures. Three different devices were used in forgery recording.

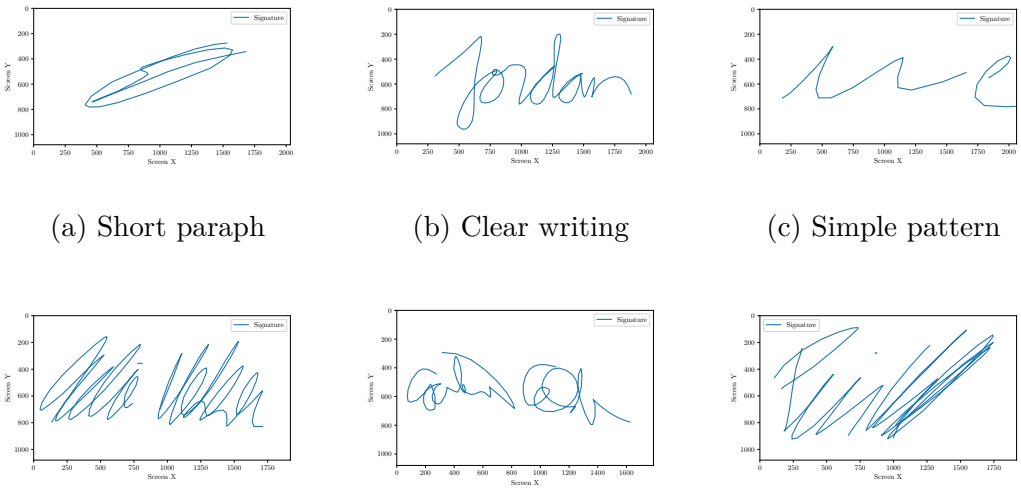
Figure 4.7 shows the setup for forgery recording that was used. Each participant used two devices for forging signatures: A phone to record the forgeries on, and a

second screen to display the signature templates, either a second phone or a notebook, which ever felt more comfortable to them. Participants were also provided with pen and paper to practice drawing the signature there. Every forger received the same amount of information about each subject. They were given the name of the subject, as well as images of all 45 first-session signatures of the subject they attempted to forge. This is a realistic attack scenario, as it is possible with some effort to come by one or multiple images of a signature of a person, as signatures are not a secret. Also, it is a harder scenario than an over-the-shoulder-attack, where the forger observes someone drawing a signature a few times and then has to imitate it from memory, as the forger has images available to practice and compare as long as needed. The hardest to detect would be an attack where the attacker is provided with a video of someone drawing their signature which can be repeated over and over again to practice, in addition to images of the signature. However, this requires more effort to obtain. The one that potential attackers could easily use would be the image scenario, which is why we test it.

The forgers were given as much time to practice each signature as they needed, and had as many attempts to sign as they wanted. Therefore, they could try without pressure and just accept five forgeries that they felt worked well into the database, while discarding attempts they were not satisfied with. They could (and did) take breaks at their own discretion. Also, in order to minimize the effects of tiredness on each subjects forgeries, the forgers started forging at different points in the dataset, so that for no subject, there are only forgeries from the beginning or end of the session. In total, it took the forgers about eight hours to complete the forgeries for all subjects, resulting in an average of about seven minutes of practice per subject to forge. However, this average can be misleading due to the varying difficulty of forging different subjects signatures. Easy-to-forge signatures often required less time with a good flow of writing, while forgers often spent more than twice the average on practicing particularly hard signatures.

In figure 4.8, examples of easy and hard to forge signatures are shown. While this is subjective for each forger, there are many signatures where all forgers agreed they were easy or difficult to imitate due to different factors. In 4.8a, the subjects signature is only a short paraph. While it is not readable like a written name, it becomes clear with the real name of the subject what letter it is, and the shape is fairly simple to draw fast. Signature 4.8b is easy to forge because it is just a short handwritten name. Therefore, by simply writing the name in the same style as the subject, a forger can craft a similar-looking imitation without much effort. Signature 4.8c is just an overall low-complexity pattern that, with a few times of practice, can be drawn alike without long practice. In contrast to the first three signatures, the second three signatures are much harder to forge. Even when knowing the name, signature 4.8d is hard to read and even harder to imitate due to the closeness and complexity with which the letters are drawn. Signature 4.8e is hard to forge because of its many sudden direction changes that are near impossible to follow at the same flow if it is not long-practiced habituation. Finally, the first part of signature 4.8f is relatively easy to imitate, but for the second part, it is almost indistinguishable which way it was drawn, due to the closeness and presumed fastness of the pattern drawing, making it difficult to imitate well.

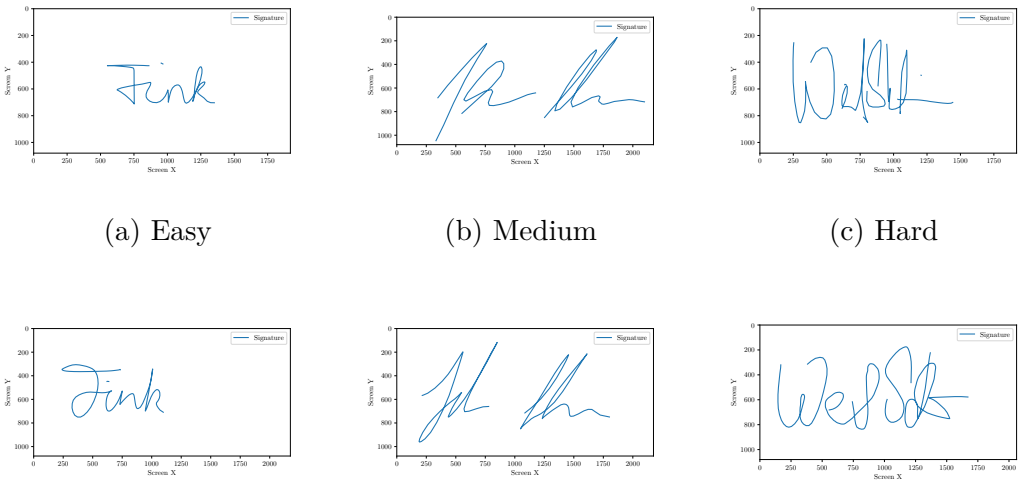
Examples of genuine signatures and their forgeries are given in figure 4.9 for the different levels of difficulty as perceived by the forgers. As can be seen, there are only minor visual differences between the easy (4.9a and 4.9d) and medium difficulty (4.9b




(a) Short paraph (b) Clear writing (c) Simple pattern

(d) Unreadable writing (e) Complex edges (f) Fast pattern

**Figure 4.8:** Different examples of easy and hard to forge signatures.




(a) Easy (b) Medium (c) Hard

(d) Easy, forged (e) Medium, forged (f) Hard, forged

**Figure 4.9:** Examples of forgery quality for different levels of difficulty. It is important to note that while the signature image is one important indicator of forgery quality, there are more parameters that are recorded, but not suitable to visualize.

and 4.9e) genuine and forged signatures. The difference in the hard case (4.9c and 4.9f) is slightly easier to see in direct comparison, mainly through the somewhat rounder execution of the forged signature. However, without having the template and forgery side by side, it would be less obvious, and it can still be considered a high-quality forgery. In general, it can be said that the older forgers focused more on getting the signature image right, while younger forgers took the signature flow more into consideration.

## 4.2 Touchpad Dataset

The touchpad database is a collection of real human signatures, drawn with a finger on the touchpad of a regular consumer laptop placed on a table in front of the participants. It contains a total of 1800 genuine handwritten signatures of 45 participants that were recorded over two sessions. Each session, 20 signatures were recorded per subject. The touchpad database also contains four sets of five skilled forgeries. As on mobile, each set is created by one of five forgers, all but one recruited from the donors of genuine signatures. As the forgers could not forge their own signature, the fifth forger had to imitate the other forgers genuine signatures. This means that there are a total of 20 forgeries per subject or 900 forgeries overall. As shown in table 4.2, this results in a total of 2700 signatures in the database. To the best of our knowledge, it is the first database of real signatures recorded on a touchpad and the second-largest mobile signature database by participants overall, second only to our mobile signature database presented in the previous section.

| Type | Amount | Subjects | Sessions | Positions | Nr./Position |
|---|---|---|---|---|---|
| Genuine | 1800 | 45 | 2 | 1 | 20 |
| Forgery | 900 | 4 | 1 | 1 | 5 |
| Total | 2700 | 45 | 3 | 1 | – |

**Table 4.2:** The contents of the touchpad dataset.

### 4.2.1 Features

Compared to the touch screens of modern smartphones, notebook touchpads are limited in resolution and capability (e.g. no direct visual feedback, no touch size). Like most touchpads built into laptops to date, our test device did not provide any additional data other than the XY-coordinates of the finger contact on the touchpad. It is important to understand that those touch coordinates are different from the coordinates of the mouse pointer on screen. Finger movements on the touchpad are transformed into movements of the mouse on screen by a driver and operating system, and they are not equivalent due to settings such as pointer speed, mouse smoothing and more. Also, the touchpad has a different resolution than the laptop screen. In the case of our test device, the touchpad has a resolution of 1236 by 898 pixels, while the devices display has a resolution of 1920 by 1080 pixels. The raw touchpad coordinates are only accessible using a special python package (the `evdev` package) on a Linux system. On Windows, it is not possible

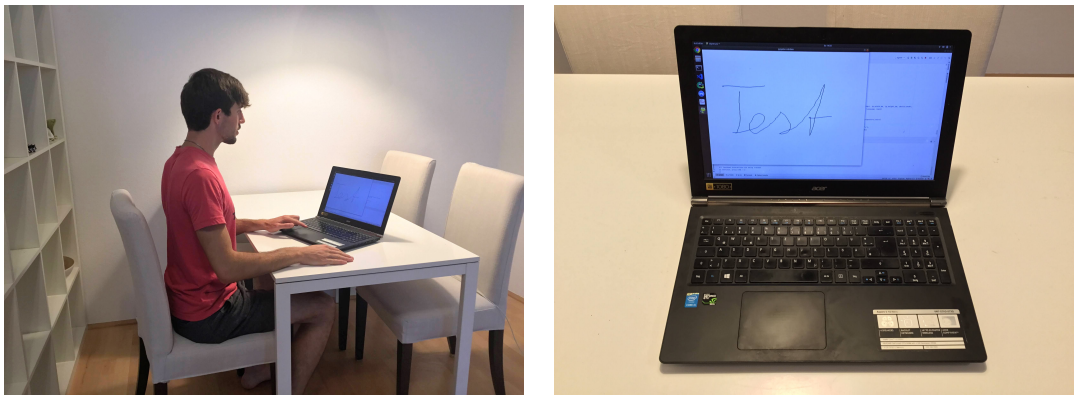
to access them at all. So, the XY-timeseries that are recorded for our dataset are the actual movements of the finger on the touchpad, not the movements of the mouse on screen. While mouse movements were also considered to be a feature in [6], we decided to not record them, as it can be irritating if the mouse moves unrelated while signing on the touchpad, and because those movements are dependant on user settings due to the reasons stated above.

The only related work on the topic of touchpad signatures [6] has tested the verification performance using a Hausdorff-distance classifier. They state that they have also recorded the pressure applied with the finger on the touchpad while drawing for their dataset of made-up paraphs in 2008. However, while they do not state the exact notebook they recorded on and only name Synaptics as touchpad manufacturer, the laptop they used must have been equipped with a resistive touchpad (hardly used anymore nowadays), as Synaptics announced the worlds first pressure-sensitive capacitive laptop touchpad only in 2012 [158, 160]. Also, while there are patents on trackpads that measure pressure [132] from as far back as 1999 [92], they are only rarely built into devices even today. The only scientific literature we could find that mentions the (non-) availability of pressure-sensing touchpads is from 2003 [33]. Nowadays, there are laptops in the premium segment, such as Apples MacBook lineup with force touch, that have some pressure-sensing capability, and there is even one application[1] that can use the pressure information for drawing when a pen is used. However, the large majority of devices still does not even have this capability, as becomes obvious when looking at the range of currently available laptops that all have clickpads. This is also exemplified by the fact that recent scientific work on using common pressure-sensitive touch surfaces resorted to external touchpads [58], and that the fact that a manufacturer[2] builds a pressure-sensitive touchpad into their flagship device still makes technology news [156]. Also, as already stated in related work, it is important to note that other than drawing tablets, these systems are still intended for clicks and not for drawing, and the usefulness for signatures is unclear [103]. Therefore, we have decided to not focus on the inclusion of pressure data into our dataset. As our own recording device was also not capable of providing pressure information, it was omitted.

### 4.2.2 Sampling and Processing

Depending on the speed of the finger, the X-Y-values are reported at widely varying frequencies on the touchpad - the faster the speed, the more values per second are provided. To allow for better processing and compatibility to the mobile dataset described in the previous section, the recordings therefore also have to be interpolated by linear interpolation and then sampled to a fixed recording rate of 50 Hz. Roughly this rate has also often been used in related work such as on smartphones or pen-tablets [29, 90, 104] and should be sufficient for accurately capturing handwritten signatures [76]. It is the same sampling rate that we have chosen for our mobile dataset. Finger-up times are represented in the timeseries through null-values for both X- and Y-axis.

[1] Inklet by Ten One Design LLC, https://tenonedesign.com/inklet.php
[2] Huawei MateBook X 2020



(a) Signing position (b) Signing setup

**Figure 4.10:** Examples of signing position and signing setup for recording the touchpad signatures.

### 4.2.3 Setting

The touchpad dataset, like the mobile dataset, aims to provide signatures that are as close to the ones that would be verified in real life as possible. Therefore, recording of the touchpad signatures was done in the participants homes or in an environment where they felt comfortable. For the participants that are in the SMG950F subset of the mobile database, recording was done in conjunction with recording the smartphone signatures, either before or after them, but always with a short break in between to avoid effects of habituation or tiredness. Unlike the mobile dataset, all recording sessions with all subjects in the touchpad database have been supervised by the authors. This decision was made because of the amount of effort and technical knowledge that is required to setup the touchpad recording application, which requires to run a Python script on Linux, something that people with non-technical background might not be able to do. For the same reason, the recording application was also operated by one of the authors at all times, only requiring the participant to sign and accept or reject each signature themselves.

Figure 4.10 shows an exemplary recording scenario. The participants were sitting at a table (as shown in 4.10a), with the laptop placed in front of them. For usability reasons, the only position in which signatures were recorded is sitting, as it it is the one that seems to have the most application scenarios. Also, because no movement data is recorded and the notebook is stationary, we consider it to be similar to scenarios where a user is standing at a desk while signing without sitting down. Recording of the signatures was done on only one device with a 5.18 inch touchpad due to the effort required for setup and to avoid interoperability problems. To record a signature, the signature Window was opened, and the subject was asked to sign with their finger on the notebooks touchpad. When they were finished, the signature window was closed either by the subject themselves or by the supervisor, and they were able to either accept or reject the signature they have just drawn. To record the next signature, the signature

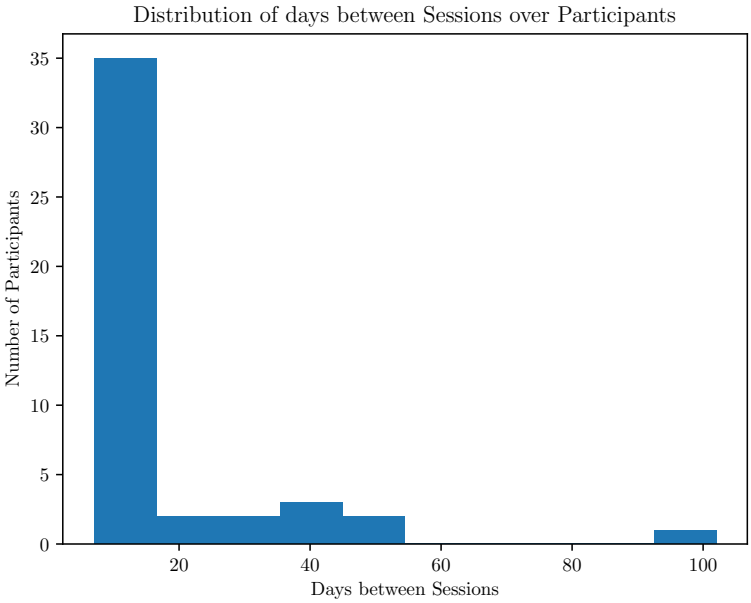



**Figure 4.11:** The time between recording sessions for the touchpad database.

window was opened again. Figure 4.10b shows the information that is displayed to a subject on screen while signing. The open signature window presents a live view of what is drawn on the touchpad. During the time the signing window is open, the mouse pointer is completely disabled and invisible to avoid confusion.

All subjects were asked to practice signing on the touchpad a few times without saving their signatures before the start of the actual recording session. This was done because many were surprised at first by the unfamiliar behavior of the touchpad and the indirect visual feedback on screen. They were also advised to try to only look down at the touchpad or only watch the screen while signing to test which way they felt more comfortable if they were unsure at first. After signing a few times, the subjects quickly got used to it, and some stated that they found signing on the touchpad more comfortable than signing on the phone. Due to this observation, we also altered our process of dealing with irregularities compared to the mobile dataset. Here, in cases when too many signatures were recorded, the first ones were removed from the dataset, as they can just be considered additional practice signatures. In total, recording of 20 signatures per subject took about 10 minutes per session to complete, and two sessions were recorded.

The time between recording sessions is shown in figure 4.11. It can be seen that the vast majority of subjects completed their first and second sessions within 14 days of each other. While the mean time between is at around 15 days due to the few outliers with a delay of up to three months, the median time between recordings is at 8 days. This greater regularity in time between sessions is due to the supervised scenario, where fixed dates have been set for first and second recordings, but in practice, it does not

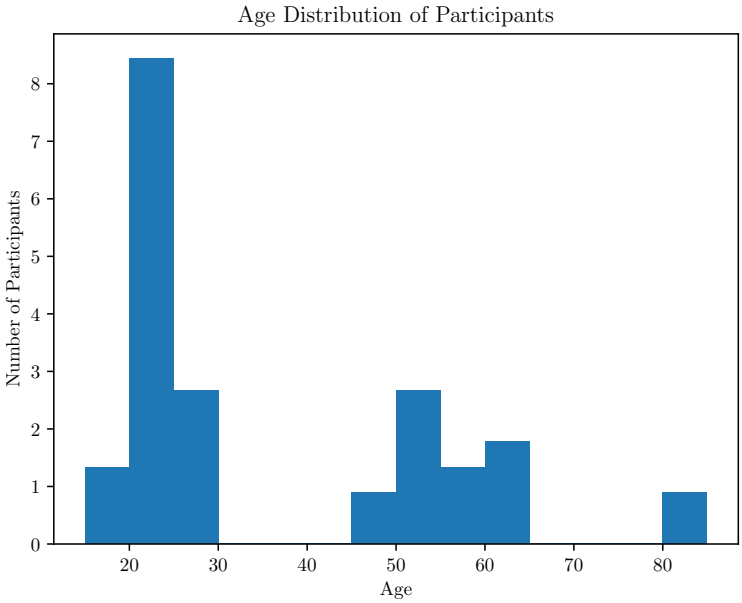



**Figure 4.12:** The age distribution among participants.

matter too much how much time is in between, as the time variance is captured either way. As for the mobile case, the minimum time between sessions has been set to seven days.

### 4.2.4 Participant Demographics

A total of 49 participants were invited to participate, of which 45 are included in the final database. One subject declined due to concerns concerns about the dataset being publicly available, and three more were not able to take part due to scheduling issues. The information in the following section describes in greater detail the 45 final participants.

The number of of male and female participants is almost equal. With 13.3% of participants signing with the left, compared to 86.7% who sign with the right hand, the amount of left-handed subjects is a bit higher than the population as a whole, where about 10% are left-handed. All participants are Austrians, with German as their native language. The overall age distribution of participants in the touchpad dataset, as shown in figure 4.12, is similar to the age distribution in the mobile dataset, but with a higher relative amount of older participants. While the younger group up to 30 years old was familiar with both smartphone and touchpad, especially the middle aged group was often less used to using a touchpad than a smartphone, as evidenced by questions about how to click or how to scroll. However, empirically they did not have more problems signing with their finger than the younger group. As in the mobile dataset, most participants were recruited from non-technical fields, and while many of

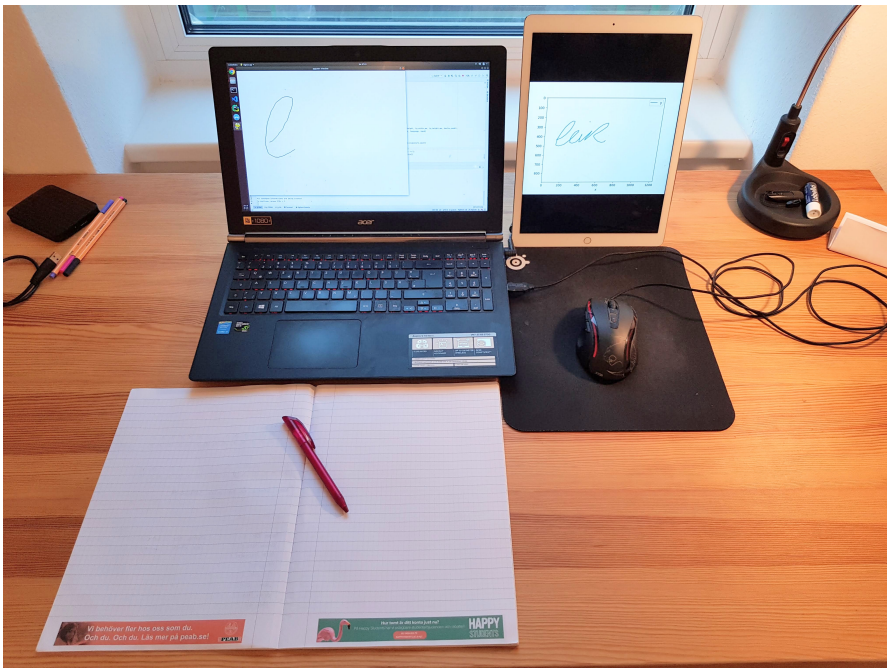


**Figure 4.13:** The setup for creating touchpad forgeries.

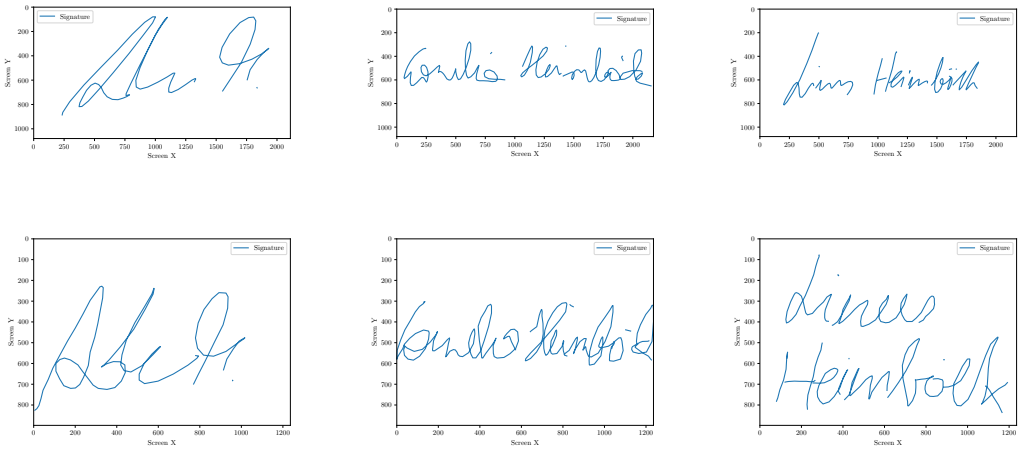
the 20 to 25 year olds are university students, many of the other participants do not have an academic background. The lower age limit for participating was also set at 16 years [23].

### 4.2.5 Forgeries

As in the mobile dataset, four sets of five skilled forgeries have been crafted for each of the participants, each set by one out of five forgers. The forgeries for each participant by a forger have been recorded in one session, but not all subjects forged by a forger had to be forged on the same day to allow for sufficient breaks in between to avoid tiring. All forgeries were made on the same device and in the same setting in which the genuine signatures were recorded as well.

Figure 4.13 displays the setup that forgers had available to create the touchpad forgeries. Two devices were used in the process: A large 12 inch iPad to display the real signatures to imitate, and the laptop on which the forgeries were recorded. The forgers also request pen and paper more often than on mobile to learn the flow of a signature in a familiar environment. Therefore, each participant was provided with enough writing material to practice. The information that they had available about their target was the same as in the mobile scenario. They were given the full name and all 20 first-session signatures as templates, which constitutes a realistic attack scenario where an impostor has obtained images of a targets signature and has ample time to practice at home.

There were no restrictions on the amount of time a forger could spend on learning each signature. Compared to forging on smartphones, more time was spent on average

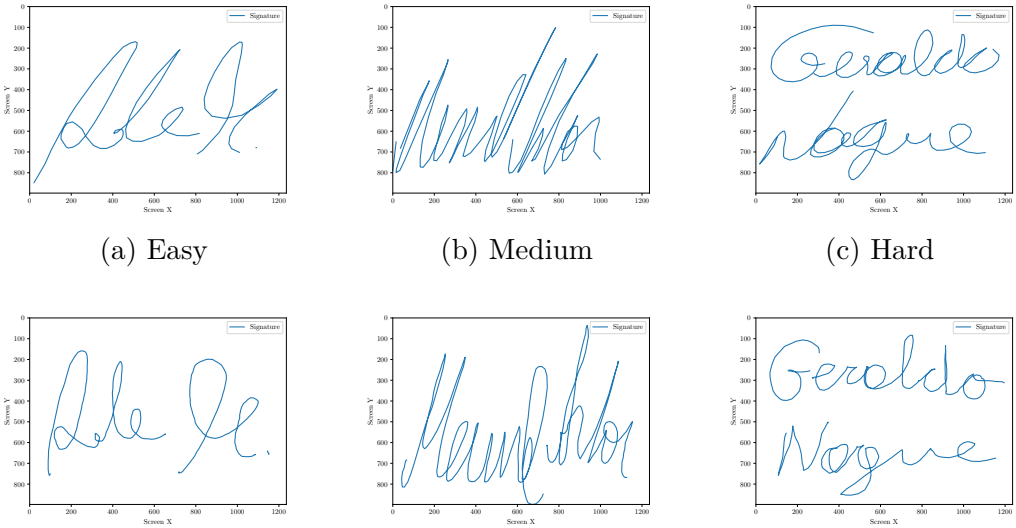



(a) Short signature (b) Same long signature (c) Split long signature

**Figure 4.14:** Comparison of signatures on phone (top) and touchpad (bottom). There were different strategies of dealing with the space constraints.

(about ten minutes per signature) due to the less familiar signing scenario. Especially for more complex signatures, more time was invested. The forgers subjective impression was that it is harder to forge signatures on a touchpad. This is largely due to the indirect visual feedback, which makes it more difficult to correctly place the finger and get a feel for the correct size of writing and space used already, as the result is not shown directly on the signing surface. Therefore, the habituation of a genuine subject places them at a greater advantage over a skilled forger than on mobile, as they do not have to think about those factors and get them right automatically. All forgers were told that not only the look of the signature image, but also the flow of the signature was important. As with mobile forgeries, it was empirically noticed that older forgers tended to focus more on perfect execution of the signature image, while younger forgers focused more on a fluent execution. All forgers could make as many attempts as they wanted, and had the possibility to accept or reject forgeries, so that only forgeries they were content with ended up in the database. As for the mobile database, to avoid effects of tiredness and habituation, forgers could take breaks as they wanted, and they started forging at different places in the dataset so that no subject would have only first-attempt or last-attempt forgeries.

Figure 4.14 shows the differences between genuine mobile and genuine touchpad signatures by the same subjects. The first point that becomes obvious are the different strategies of dealing with the constrained space on a touchpad. The subject in 4.14a did not have a problem with the signing space on either device due to the short, paraph-like signature. The subject in 4.14 has a relatively long signature, but decided to write it in one line, same as on the mobile device, resulting in a slightly squeezed signature image. Subject 4.14c also has a long signature, but decided to write it in two lines, resulting in a larger letter size than on mobile and a different image in comparison. Further, there are




(a) Easy (b) Medium (c) Hard

(d) Easy, forged (e) Medium, forged (f) Hard, forged

**Figure 4.15:** Signatures and their forgeries at different perceived difficulty levels. The signature image is one indicator of forgery quality, but fluency can be just as important.

noticable differences in the confidence with which the signatures appear to be drawn. For 4.14a, both the mobile and the touchpad signatures show a fluent execution, due to it being more a habituated drawing than actual writing. In contrast, both written signatures (4.14b and 4.14c) exhibit more insecurity on touchpad than on mobile, as empirically it is harder to draw beautiful letters on the touchpad without direct visual feedback than it is on mobile, where the resulting signature is directly visible. While the potential verification performance impacts will be discussed in the results chapter, this is also relevant to forging, as it allows for more insecurity of the forger without it being immediately noticable from the signature image.

In figure 4.15, examples of different genuine signatures and their forgeries are presented. While the overall images are similar in all three cases, there are more differences in the details. The forgery 4.15d for the genuine signature 4.15a was described as relatively easy to make. This is presumably due to it being a relatively short signature that can be written fluently without lifting the finger more than once, and because the patterns are clearly visible. Signature 4.15b and forgery 4.15 were perceived as harder probably because of two reasons: It is longer than signature 4.15a, and the letters are not as easily distinguishable. However, it is still mainly a repetitive pattern of lines up and down that has some fluency and does not require perfect exactness to look real. Finally, signature 4.15c was, somewhat surprisingly, seen as hard to forge on the touchpad, while it was not specially noticed on the phone. Its forgery, as shown in 4.15f, looks similar overall, but the line jitter is gives a sense of insecurity on closer inspection. This is likely due to multiple reasons: On the phone, it is possible to directly see what is written, while on the touchpad, it is easier to loose track. The signature is clearly written with

pen-ups in between, and a forger has to pay attention to how exactly the letters are drawn, which makes it less fluid to imitate. And finally, it is fairly long, requiring more remembering and just allowing for more possibilities to make errors.

## 4.3 Usability

In conjunction with recording our datasets, we also conducted a usability study on both signing on smartphones and signing on notebooks. To the best of our knowledge, it is the largest study on the usability of mobile signing in the world. All study participants have been asked to fill out questionnaires about their signing experience directly after completing the second signing session on each device. The study was executed using Google forms, with one set of about 15 questions for each dataset. Many items in both sets were the same, but each questionnaire also contained a few device-class-specific items. Also, most of the questions were five-point Likert-items, but there were also a few questions where more or less options were given. As we are not compiling a full Likert-scale from the responses and analyze the items individually, we consider the points to be ordinal [68]. In order to minimize the influence on the participants, the authors positioned themselves in a way where they clearly could not see the screen while participants were answering, and only came to help if something was unclear. In case of the study on mobile phones, the authors were only present for the SMG950F subset, all other questionnaires were filled out remotely. All participants were made aware that their answers would be published anonymously, but that for internal assignment of questionnaire responses to recorded signatures, we needed to know their identity. None of the subjects have shown any concern about this.

Due to the demographics of the participants, each questionnaire was available in two translations, the German original and an international English version. Most subjects were given the German version, as they were German native speakers. It was paid attention to keep the English version as close to the original as possible in both tone and meaning, and the exact same order of items and points was used in both. The results included with the dataset and reported in chapter 7 are therefore the results of the combined usability study, but the individual results on each questionnaire for the different translations are made available as well. All questionnaires in both the German and English translations can be found in the appendix.

## 4.4 Dataset Structure

The dataset is published online[3] as individual signature files in CSV format. Additionally, we also provide a second version for convenience that was pre-processed with our own processing chain as described in this and the following chapters and contains additional features that were extracted. It can be used as a quick start to signature verification. In the following section, we describe the basic structure of our primary dataset. Both datasets have the same structure, but are separated from each other.

[3] github.com/linuxh/signatureverification

```
- [ParticipantID]
    - [SessionNr]
    - [SessionNr]
    - f[ParticipantID of Forger]
    - f[ParticipantID of Forger]
    - f[ParticipantID of Forger]
    - f[ParticipantID of Forger]
- [ParticipantID]
```

**Listing 4.1:** Directory structure of the dataset.

The overall directory structure is shown in listing 4.1. On the top level, there is one directory for each participant in the dataset, named with their participant identifier. The participant id is consistent across both mobile and touchpad database, forgeries and even the usability study. If it is the same number, it is the same subject. Inside each participant's directory, there are multiple sub-directories. The ones with only a number as name contain the genuine signatures of the corresponding session, where the directory name is also the session number. Those where the number is prefixed with `f` contain all skilled forgeries by one forger for the subject, and the number is the forgers participant id.

```
sig0-[id]-[ts]-[ath]-[sw]-[sh]-[wd]-[hd]-[dev]-[pos]-[sex]-[age]-[co]-[la]-[hnd].csv

# id = Participant id
# ts = UTC Timestamp in milliseconds
# ath = Authenticity Indicator
# sw = Screen Width in Pixels (X-Axis)
# sh = Screen Height in Pixels (Y-Axis)
# wd = Width DPI (X-Axis)
# hd = Height DPI (Y-Axis)
# dev = Device Identifier
# pos = Recording Position
# sex = Signer Sex
# age = Signer Age
# co = Signer Country
# la = Signer Native Language
# hnd = Signing Hand
```

**Listing 4.2:** The structure of filename parameters.

The metadata of each signature is stored in the signature file name in human-readable form, with the structure shown in listing 4.2. The participant id is always the id of the participant whose signature it is, while the information about the Signer is always about the person who actually drew the signature. While those two are the same for genuine signatures (indicated by `ath` being set to `true`), they are not for forgeries (`ath = false`). So therefore, for forged signatures, the participant id is the id of the participant whose signature was forged, while the signer information is the information about the forger who drew the signature. The device information provides more data about the device that was used for signing, and the timestamp indicates when the signature was saved in UNIX millisecond format. The filename format is the same for both the mobile and the touchpad database. Finally, each of the databases contains one file called `usability.csv` at the root level. It contains the results of the usability study for each participant, again identified by the corresponding participant id.

## 4.5 Additional data

In addition to our primary study dataset, we are releasing four more collections of signatures. While they are not of the same scale, all four are related to the primary dataset and might be helpful in certain scenarios:

- The first preliminary dataset of over 20 participants, many of which are not in the final dataset. It contains 5 signatures per subject of one session that might be helpful for additional testing. It does not included the gravity sensor measurements included in the final dataset.
- The second preliminary dataset, consisting of 9 participants, some of which are not part of the final study dataset. It contains 35 signatures each, recorded over one session sitting, and also contains skilled forgeries. Like in the first preliminary dataset, no gravity sensor measurements are included.
- The leftover mobile signatures that were removed from the mobile database because of one of three reasons: Rare unintended, user-caused recording errors (saving too early, accidental touches of the screen), more than 15 signatures per session available, or using different versions of the signature. While these irregularities are not desired in the regular dataset, they might be of interest in certain real-world research scenarios or as additional test data.
- The leftover touchpad signatures that were removed from the touchpad database because of one of the reasons stated above.

# Chapter 5

# Approach

In this chapter we present our approach to automated signature verification on mobile devices. We outline our two lines of research on mobile phone and notebook signature verification and describe the intended usage concept and high-level structure of our common processing pipeline. We then go into the details of our approaches. Our underlying study dataset and the associated usability study are detailed in chapter 4. The results of applying our approach on our dataset and the findings of the usability study are presented in chapter 7.

In our mobile line of research, we focus on the inclusion of additional, mostly movement-based features that can be recorded unobtrusively while signing using sensors available on most smartphones, and their impact on dynamic signature verification. Most importantly, we investigate if the combination of image-based and movement-based dynamic features of a signature can improve verification performance over only image-base or only movement-based systems on mobile devices, and how strongly the individual features contribute to it. The idea is that as both consciously drawn signatures and unconscious movements while signing are biometric traits that can be used for recognition on their own, a combination of the two might perform even better than either one individually. As the phone movements that occur while signing cannot be imitated easily, they could be especially helpful to identify skilled forgeries.

Our touchpad line of research attempts to answer how well signature verification works on standard integrated notebook touchpads. This device category has received little attention so far, and offers only a limited set of available features with an unusual user experience. Therefore, both verification and usability results are compared to results on smartphones, with the goal of evaluating the viability of finger-drawn signatures on laptops. If the findings are promising, touchpads could become another relevant platform for digital signing, as they are already widespread among consumers.

To make our results comparable, we implement a shared processing and classification chain for both mobile and touchpad signatures, which is shown in figure 5.1. Apart from a few additional features that are extracted from the movement features of the smartphone signatures, it is exactly the same for both mobile and touchpad scenarios.

From a users point of view, the system is simple. At the time of first setup of the verification system, the new user signs a few times to provide signature templates, similar to setting up a face- or fingerprint based recognition system. The enrollment samples are then used to build and store a model of the users signatures which can serve as

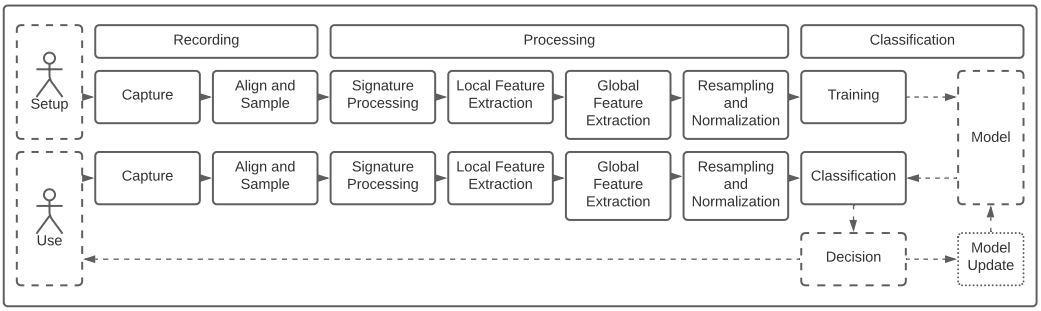



**Figure 5.1:** The overall structure of the processing and classification chain. Solid lines denote processes that are performed and the sequence in which they occur. Dashed lines are objects and results the processes create, and interactions of the processes with them. Dotted lines are optional elements that can be part of the chain, but are not required for it to work.

reference. Subsequently, when the system is used, every new signature is compared to the model for verification once it is confirmed by the user, and a decision is taken to either accept or reject the signature, allowing or preventing the user to continue.

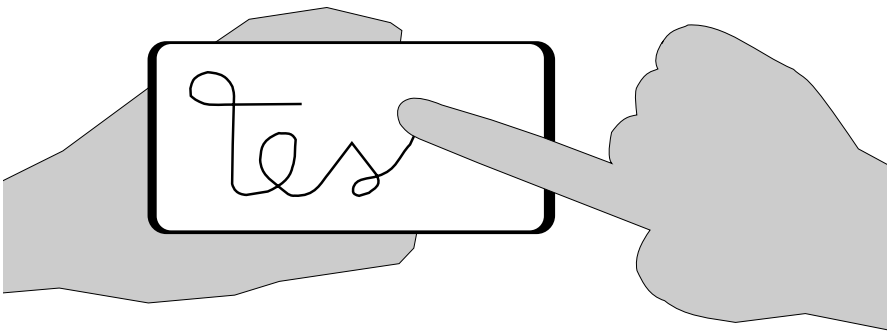


**Figure 5.2:** The user holds their phone in one hand while signing with the finger of the other hand on screen.

The process works as follows. When the signature field is shown, a signature can be drawn directly with a finger. This is done either on screen while holding the phone in hand on mobile, as shown in figure 5.2, or on the touchpad on notebooks. In the background, from the moment of first touching the signing surface, all user interactions or non-interactions are minutiously recorded by the system. Touch-coordinates, as well as finger-up times, are collected together with timestamps, in order to be able to derive behavioral information such as drawing speed and acceleration. On mobile, the system also tracks how the phone is held and moved while signing using the embedded sensors. Once the user confirms the signature, all readings are aligned and sampled to a fixed

rate to combine them into a complete signature that can be processed. For our dataset, the signatures are saved as files at this stage. From here, the actual processing of the signature starts. First, the signature time series are pre-processed, for example by applying filters. Then, local and global features are extracted, before the signature is normalized and resampled as preparation for the classification algorithm. It is important to note that while overall the steps are in the order shown in figure 5.1 and described above, some unit transformations in signature processing have to happen at different points in the chain. To make the process more clear, we have grouped them together and will point the exact order in which they appear.

For classification, special emphasis lies on the creation of a local Multi-Dimensional DTW based classification system that utilizes both X-Y features and, if available, 3D-Movement features. DTW is commonly used in signature verification and allows for a realistic real-world performance estimate, as it only requires the positive class (genuine signatures) to train. Additionally, we also test a variety of standard machine learning classifiers in different local and global scenarios. During enrollment, the verification model is created from the template signatures provided by the user. This can be different depending on the type of classifier used, from simply storing the signatures to training classifiers with large sets of negative samples. In use, the signature is verified with the model to decide whether a signature is genuine or forged. Optionally, the result can then be used to update the model with additional data.

In the following sections, the individual steps of our processing chain are described in detail, starting with the processing stage. For more details on signature recording and the steps involved to combine the individual readings to form a processable signature in the recording stage, refer to the dataset description in chapter 4.

## 5.1 Signature Processing

Signature processing, or time series processing for a more abstract terminology, is the first stage of pre-processing. Its goal is to gain uniform, noise-reduced signals independent of the recording device to prepare for feature extraction. The very first parts of this take place even earlier on device, before the processing chain begins. There, all measurements are sampled to a fixed sampling rate of 50 Hz and aligned between the different sensors. This happens transparently to the pre-processing, and more information is provided in chapter 4. The time series processing itself consists of three steps: Interpolation, filtering and transformations. While in on-device processing, interpolation of all measurement points of the whole signature was used to be able to save signatures at an artificial fixed sampling rate, its usage is different here. As all signatures are recorded without a pen, times at which the finger is in the air result in null-values for all touch-based measurements. As classifiers can not deal with null, those values need to be replaced somehow for X- and Y-coordinates on both mobile and touchpad, as well as for Android finger speed and touch size on mobile only.

For the generation of the pen-up X-Y-coordinates, two different strategies have been considered. During development, the blanks were simply filled with the mean of the respective measurement. This has the advantage of simplicity, and it does not affect the overall means of the affected features, which is why we considered it to be the least destructive strategy. In related work [94], interpolation has been used as another

strategy of supplying those missing pen-up-trajectories. In our implementation, we have therefore tested linear interpolation as a replacement strategy, and improved our DTW performance on our incomplete development dataset by roughly 2 percentage points EER compared to using means. Considering these results, we have decided to select linear interpolation for our final processing chain.

While this completes the interpolation step on the touchpad, on mobile, there are still the speed measurements and touchsize for which null-replacements have to be found. Linear interpolation is unsuitable for both. For the speed measurements, interpolating from the speed of the end of one stroke to the speed at the beginning of the second stroke would mean a steady change in speed for each point in time. This is inconsistent with the interpolation of the coordinate axes, which implies a steady change in distance per time unit and therefore constant speed. Therefore, the finger-up values for Android speed measurements have been filled with the speeds calculated as the first derivates of the (interpolated) pen-up trajectories. For touchsize, linear interpolation would mean a steady change in size of the touched area on screen while the finger is up in the air, which does not make any sense semantically. As having the finger in the air means not touching the screen at all, we have chosen to use a touchsize measurement of 0 as replacement for null.

Filtering is applied to all raw measurements except Android speed values, which are essentially derived features computed by Android itself. Using a low-pass-filter to remove slight noise variations has already been suggested in [23]. We use a fourth-order Butterworth filter, which is a popular low pass filter due to its maximally flat frequency response, meaning that it does not produce any ripples in passband of frequencies as other filters do. The order of the filter affects the filters frequency response, that is, how strongly frequencies over the cutoff frequency are damped.

$$f_{nyq} = \frac{1}{2} \cdot f_{sig} \qquad W_n = \frac{f_{cut}}{f_{nyq}} \tag{5.1}$$

We defined our cutoff frequency $f_{cut}$ at 20 Hz, which results in a normalized cutoff $W_n$ of 0.8, as shown in equation 5.1. As our signatures are recorded at a sampling rate $f_{sig}$ of 50 Hz, the nyquist frequency $f_{nyq}$ is 25 Hz. The nyquist frequency is the minimum frequency at which all information in the signal is retained, allowing for arbitrarily detailed reconstruction.

The final step in our timeseries processing is transforming the measurements in universally usable units in order to allow for interoperability between devices. While strictly, this would only be necessary for the mobile database, as all touchpad signatures have been recorded on the same notebook, it is done for both datasets to improve comparability and allow for easier expansions in the future. As Android reports the movement sensor values already in fixed units systems - acceleration and gravity measurements in $m/s^2$, gyroscope in $rad/s$ around each axis, magnetic field sensor in $\mu T$, for each axis respectively - there is no need for additional conversions there.

$$x_i' = \frac{x_i}{w_{scr}} \qquad y_i' = \frac{y_i}{h_{scr}} \tag{5.2}$$

The display based measurements, on the other hand, use the respective screen as a reference system, which is different for different devices. Therefore, the coordinates

themselves and the finger speeds, both measured by Android and derivates, need to be normalized. The touchsize is already reported as a normalized and scaled value by Android [157]. For the signature geometry (x and y coordinates), this is done by dividing through the screen dimensions to obtain relative values, as shown in equation 5.2, where $w_{scr}$ and $h_{scr}$ are the screen width and height in pixels, and $x_i$ and $y_i$ are the coordinates at the respective positions. It is applied out of order, after all local features have been derived, because due to the different aspect ratios of phone screens, drawing angles in the signature can change disproportionately, and some local feature calculations rely on them.

$$v'_{x_i} = \frac{v_{x_i}}{dpi_w} \cdot 25.4 \qquad v'_{y_i} = \frac{v_{y_i}}{dpi_h} \cdot 25.4 \tag{5.3}$$

The finger movement speeds are reported as pixels per second by Android [157]. Since different devices have different screen densities $dpi_w$ and $dpi_h$ from each other, those values also need to be converted to standard units. Therefore, the speeds $v'_{x_i}$ and $v'_{y_i}$ at each point in time on each axis are normalized by the axis' respective pixel density and multiplied by 25.4 to receive $mm/s$, as shown in equation 5.3. Those speeds are then used for further calculations in feature extraction, as described in the following section.

## 5.2 Local Feature Extraction

The local features extraction is based on features that have already been proposed in related work. The X-Y-channel features are extracted following the set described in [93], which in turn is built on [44, 81, 117, 141] and has established itself as common, for example used in [94, 135]. Features from the movement data are extracted based on the information in [29], which is one of the few systems where some form of movement data has already been used.

An overview over all X-Y-features is given in table 5.1, together with their respective definitions. All features are calculated on signature data that was preprocessed as described in the previous section. Also, for comparability between mobile and touchpad signatures, the first derivates of x and y are used for any calculations where the speed is required, not the speed values that are reported by Android. As the signature points are measured at fixed intervals, the derivate $\dot{p}_n$ at each point $p_n$ in time for a given feature can be calculated as shown in equation 5.4. To obtain the rate of change per second instead of per interval in case of the finger speed calculations $\dot{x}$ and $\dot{y}$, the results then only need to be multiplied by the sampling frequency $f_{sig}$.

$$\dot{p}_n = p_n - p_{n-1} \tag{5.4}$$

For movement data, the extraction process is less extensive. In related work [29], low- and high-pass filtered versions of the raw accelerometer readings are calculated to isolate the user-caused acceleration and gravity components. However, as we are recording both components separately as linear acceleration and gravity sensor readings, this step is already performed on device by the Android framework. For each of the sensors, the magnitude $m_n$ for each point is calculated as a fourth feature besides the x, y and z

values that are recorded, as suggested in [29]. The magnitude calculation is shown in equation 5.5. It is important to note that while gravitation magnitude is calculated, it will always result in a value of $9.81m/s^2$ and can therefore be omitted in training.

$$m_n = \sqrt{x_n^2 + y_n^2 + z_n^2} \tag{5.5}$$

| Nr. | Feature | Definition |
|---|---|---|
| 1 | x-coordinate | $x_n$ |
| 2 | y-coordinate | $y_n$ |
| 3 | Path-tangent angle | $\theta_n = arctan(\dot{y}_n/\dot{x}_n)$ |
| 4 | Path velocity magnitude | $v_n = \sqrt{\dot{y}_n + \dot{x}_n}$ |
| 5 | Log curvature radius | $\rho_n = log(1/\kappa_n) = log(v_n/\dot{\theta}_n)$, where $\kappa_n$ is the curvature of the position trajectory |
| 6 | Total acceleration magnitude | $a_n = \sqrt{t_n^2 + c_n^2} = \sqrt{\dot{v}_n^2 + v_n^2\theta_n^2}$, where $t_n$ and $c_n$ are respectively the tangential and centripetal acceleration components of the pen motion |
| 7-12 | First-order derivatives of features 1-6 | $\dot{x}_n, \dot{y}_n, \dot{\theta}_n, \dot{v}_n, \dot{\rho}_n, \dot{a}_n$ |
| 13-14 | Second-order derivatives of features 1-2 | $\ddot{x}_n, \ddot{y}_n$ |
| 15 | Ratio of the minimum over the maximum speed over a window of 5 samples | $v_n^r = \frac{min\{v_{n-4}, \ldots, v_n\}}{max\{v_{n-4}, \ldots, v_n\}}$ |
| 16 | Angle of consecutive samples | $\alpha_n = arctan\left(\frac{y_n - y_{n-1}}{x_n - x_{n-1}}\right)$ |
| 17 | First-order derivate of feature 16 | $\dot{\alpha}_n$ |
| 18 | Sine | $s_n = sin(\alpha_n)$ |
| 19 | Cosine | $c_n = cos(\alpha_n)$ |
| 20 | Stroke length to width ratio over a window of 5 samples | $r_{5_n} = \frac{\sum_{k=n-4}^{n} \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}}{max\{x_{n-4}, \ldots, x_n\} - min\{x_{n-4}, \ldots, x_n\}}$ |
| 21 | Stroke length to width ratio over a window of 7 samples | $r_{7_n} = \frac{\sum_{k=n-6}^{n} \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}}{max\{x_{n-6}, \ldots, x_n\} - min\{x_{n-6}, \ldots, x_n\}}$ |

**Table 5.1:** The set of local X-Y features that are extracted [93].

## 5.3 Global Feature Extraction

After local feature extraction and the normalization of display-based measurements as described in equation 5.2, global features of the signature are extracted. The choice

of global features is based on the ones used in [29], together with a few of our own additions. They are calculated only for the original (but filtered) time series that were recorded on the phone, not for the local features that were extracted, with the exception of the signing speed $\dot{x}$ and $\dot{y}$ and the magnitude values for the movement features. The complete set of global features that is used is displayed in table 5.2.

| Nr. | Feature | Definition |
|---|---|---|
| 1 | Signature duration | $max(timestamp)$ |
| 2 | Number of strokes | $max(strokenr) + 1$ |
| 3 | Start value of the signature geometry | $start(x), start(y)$ |
| 4 | End value of the signature geometry | $end(x), end(y)$ |
| 5 | Minimum value of the signature geometry | $min(x), min(y)$ |
| 6 | Maximum value of the signature geometry | $max(x), max(y)$ |
| 7 | Width of the signature geometry | $w = max(x) - min(x)$ |
| 8 | Height of the signature geometry | $h = max(y) - min(y)$ |
| 9 | Area of the signature geometry | $A = w \cdot h$ |
| 10 | Maximum speed while signing | $max(\dot{x}), max(\dot{y})$ |
| 11-31 | Means | $mean(ts_{sig})$ |
| 32-52 | Standard Deviations | $stdev(ts_{sig})$ |
| 53-73 | Skews | $skew(ts_{sig})$ |
| 74-94 | Kurtosis' | $kurtosis(ts_{sig})$ |

**Table 5.2:** The set of global features that are extracted. Features 11-94 are calculated for many timeseries $ts_{sig}$ of the signature: $x$, $y$, $\dot{x}$, $\dot{y}$ and touchsize of the touch-based measurements, and the x, y, z, and magnitude features of each of the four movement sensors. For the touchpad, the measurements that are not available are simply omitted.

## 5.4 Resampling and Normalization

After all features are extracted, the signatures are resampled and normalized. First, as suggested in [104], the signature geometry ($x$ and $y$) is translated to the center of the coordinate system by subtracting $start(x)$ and $start(y)$ from each element in the respective time series. While they also standardize the starting point of their accelerometer recordings, we have decided to preserve all our movement features starting points. For example for the gravity sensor, the actual values of the sensor have meaning and show how the phone was held in each axis, so not only the change of the values is of interest.

Next, the signature is resampled to normalize its length. While there are indications in literature that this might negatively affect performance due to a possible loss of timing information [73], this step is necessary in order to obtain a uniform training dataset that can be processed well by all the different types of classifiers we use to allow for comparability. Also, the same is done in many fairly recent publications such as [23], where signatures are also transformed to the same length, and also in [104], where signatures of a subject are brought to the mean length of the subject. As in [23], we use linear interpolation, but have decided to only use 50 points instead of their

256. The reason for this is two-fold. First, it is still possible to capture the overall signature geometry even of longer signatures reasonably well with 50 points, although with some loss in detail, while shorter signatures would require even less. Due to the higher complexity of longer signatures [135], one could also hypothesize that a reduction in detail might be compensated for, although there is no work on that yet. Second, mobile environments are limited in energy and processing power, and reducing the number of points in the signature by a factor of 5 also reduces the requirements on the devices hardware. To empirically test our thinking, we have used our DTW algorithm, which should be able to benefit from a more detailed time series, with both 250 and 50 point samples on preliminary test datasets, and the verification performance differences were marginal to non-existent, while computation times were reduced significantly. Therefore, we have chosen to use only 50 measurement points per signature. The result, in either scenario, are equispaced temporal sequences, where the length of the space is variable for different sequences, meaning different sampling rates for different signatures.

$$p' = \left( \frac{p - min(p)}{max(p) - min(p)} \right) \cdot (b_{upper} - b_{lower}) + b_{lower} \tag{5.6}$$

Finally, the signature size is normalized between $b_{lower} = 0$ and $b_{upper} = 1$ using min-max scaling. This is similar in principle to the size normalization applied in [104], apart from the fact that their min-max scaling is inverted. The scaling equation is shown in equation 5.6. This is applied to all time series of the signature, both recorded and extracted.

## 5.5 Classification

Classification is the final stage of our processing chain. It is done on the final, processed signatures, with features extracted and length normalized as described in the previous section. The individual classifiers are trained on labeled data - signatures for which the identity of the signer is known - to learn to distinguish between the identities of unlabeled signatures later. The task of signature verification is a binary classification task, where each signature is either labeled positive (a genuine signature of the enrolled user) or negative (a signature not by the enrolled user, no matter if is an entirely different signature or a skilled forgery). Additionally, we also test a signature recognition scenario where each signature has to be classified as belonging to a specific writer, making it a multi-class classification problem.

There are two approaches to the task of signature verification. Standard machine learning algorithms require training data that includes samples for every class they have to recognize later. For binary classification, this means that both positive signatures (signatures of the enrolled user) and negative signatures (signatures not by the enrolled user) have to be in the training set. In order to be able to classify a yet unknown illegitimate signature correctly, the negative training data has to form a good representation of the potentially infinite number of non-genuine signatures, but can include only random forgeries, as skilled forgeries are usually not available in a real-world scenario. Constructing such a negative set is challenging, as it requires a large amount of samples and can never cover all eventualities. Also, users of a system are not able to provide the negative data by themselves, leaving the task of acquiring it with the developers.

Anomaly detectors, also known as One-Class-Classifiers, only require the positive class to train for a binary classification problem. Unknown signatures are then scored on their similarity to the known enrollment samples and classified based on a threshold as positive or negative. This removes the need for a large, always imperfect set of negative samples, making this type of model ideal for verification tasks. However, both approaches are valid methods of verifying signatures and can deliver good results, which is why we consider them both in our work.

### 5.5.1 Dynamic Time Warping

Dynamic time warping is the most popular algorithm in dynamic signature verification at the moment [22, 23, 34, 76, 80, 97, 111]. It was originally proposed in [121] as a method of speech recognition, and while HMMs have mostly replaced it for that purpose, it is still highly effective for signatures as it is best suited for small amounts of training data [84]. The common implementation presented below is described in [84], and adaptations such as the use of DTW distances as features for common machine learning classifiers [73], independent DTW applications for each stroke, and the use of different distance metrics or combination methods for the individual distances of each feature are possible.

Conceptually, dynamic time warping finds the best possible non-linear alignment, and thus the smallest possible distance, between two vectors of variable length [73]. The algorithm is defined as follows. Given are two sequences $P$ and $Q$, as in equation 5.7:

$$\begin{aligned} P &= p_1, p_2, \ldots, p_i \\ Q &= q_1, q_2, \ldots, q_j \end{aligned} \tag{5.7}$$

A distance measure between points is defined according to the requirements. A common example is the simple absolute distance, as shown in equation 5.8.

$$d(p_i, q_j) = ||p_i - q_j|| \tag{5.8}$$

Both series can best be thought of as a matrix, where the points of each sequence are on an axis, and the values at each position in the matrix represent the distance between the corresponding points. A warping path $C$ (eq. 5.9) can then be defined as series of correspondences $c_k$ between sequence points $c_k = (p_i, q_j)$.

$$C = c_1, c_2, \ldots, c_k \tag{5.9}$$

A valid warping path is restricted by three requirements [125]: Monotonicity, Boundary and Step Size (commonly: Continuity) conditions. The monotonicity condition guarantees the time-ordering of points in the sequence, so that the alignment path never goes back in time. The boundary condition requires the start- and end-points of the warping path to be the start- and end-points of the aligned sequences. The step size criterion defines what kinds of movements between points are permitted through a step-pattern [50, 125]. While there are various different options [50], the most common one allows for movements of one point at a time (preventing jumps/shifts in time), either only on one axis or diagonally over both [50, 84], as shown in figure 5.3. Each movement has an associated weight $w(k)$ that is applied to the distance at the corresponding step

$k$ to be able to normalize the distance to compensate for different sequence lengths (possible depending on step pattern).

The DTW algorithm finds the warping path where the accumulated distance $g_k$ is minimal under these restrictions. It is defined recursively as in equation 5.10, with the starting condition given in equation 5.11.

$$g_k = g(i,j) = min(g_{k-1} + d(c_k) \cdot w(k)) \tag{5.10}$$

$$g_1 = g(1,1) = d(c_1) \cdot w(1) \tag{5.11}$$

With the common definition of step patterns that is depicted in figure 5.3 [84], $g_{k-1}$ becomes one of three options as shown in equation 5.12.

$$g_k = g(i,j) = min \begin{cases} g(i,j-1) + d(i,j) \cdot w(k) \\ g(i-1,j-1) + d(i,j) \cdot w(k) \\ g(i-1,j) + d(i,j) \cdot w(k) \end{cases} \tag{5.12}$$

Further restrictions can be placed on the warping path. Defining a Sakoe-Chiba band [121] or an Itakura-Parallelogram [66] as global path constraints ("warping windows") prevents the warping path to diverge too far from the diagonal and thereby from skipping different features of the sequence. Also, slope constraints can be used to limit the amount of steps that can be taken in one direction before steps in the other direction have to be made, in order to avoid matching very short parts of a sequence to very long ones [161]. In its original unconstrained form, the DTW algorithm has a runtime complexity of $O(n \cdot m)$.

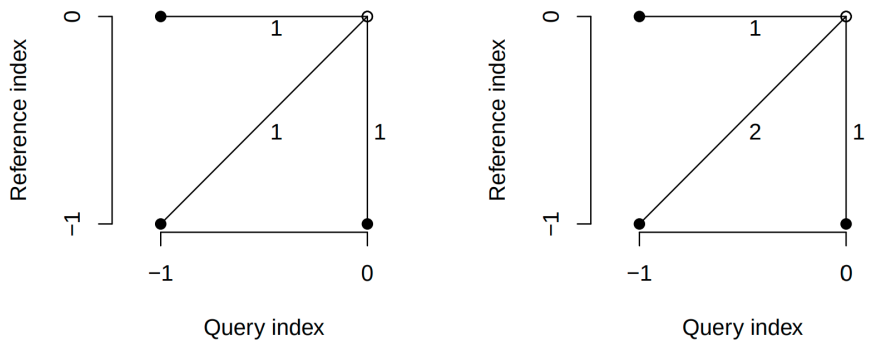



**Figure 5.3:** The most common symmetric step patterns. The quasi-symmetric (White-Neely) pattern on the left favors diagonal movements, because the diagonal movement has the same weight associated as the straight movements, but requires one less step than those to reach the same point. The standard symmetric pattern on the right is not biased [50].

### 5.5.2 Multi-Dimensional Dynamic Time Warping

Dynamic signatures are nothing else than sequences of points over time, making them perfectly suitable to be classified with dynamic time warping. For example, consider two signatures $A$ and $B$ that are given as series of x and y coordinates over time, as shown in equation 5.13.

$$\begin{aligned} A &= a_1, a_2, \ldots, a_i = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}, \ldots, \begin{pmatrix} x_i \\ y_i \end{pmatrix} \\ B &= b_1, b_2, \ldots, b_j = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}, \ldots, \begin{pmatrix} x_j \\ y_j \end{pmatrix} \end{aligned} \tag{5.13}$$

Each of the signatures points $a_i$ and $b_j$ is represented by a two-dimensional feature vector consisting of corresponding x and y coordinates of the contact point on the touch surface. Usually, to calculate the distance between the two signatures, the signatures are broken up into single series for each of their dimensions [84, 97], such as in equations 5.14 and 5.15.

$$\begin{aligned} X_A &= x_{a_1}, x_{a_2}, \ldots, x_{a_i} \\ Y_A &= y_{a_1}, y_{a_2}, \ldots, y_{a_i} \end{aligned} \tag{5.14}$$

$$\begin{aligned} X_B &= x_{b_1}, x_{b_2}, \ldots, x_{b_j} \\ Y_B &= y_{b_1}, y_{b_2}, \ldots, y_{b_j} \end{aligned} \tag{5.15}$$

Then, the distances between the individual dimensions of the signature are calculated (eq. 5.16) as described in the previous section.

$$g_X = dtw(X_A, X_B), \qquad g_Y = dtw(Y_A, Y_B) \tag{5.16}$$

$$G = g_X + g_Y \tag{5.17}$$

Finally, the individual distances per dimension are normalized and combined to obtain the resulting total distance $G$ between the two signatures, for example by simple addition as in equation 5.17.

The problem with this approach is that it results in different warping paths for each individual feature of the same signature. This means that, for example, the x-coordinate of a point $a_1$ in signature $A$ can be matched to a point $b_1$ in signature $B$, while the y-coordinate of the same point $a_1$ is matched to a totally different point $b_2$, although the coordinates belong together and get their meaning (a position on the screen) as a whole, and matching them up differently changes that meaning (a different combination of x-y-coordinates marks a different position on the screen). The same is true, albeit harder to visualize, for additional features at each point, such as acceleration or touchsize. Together, they form one coherent unit of the whole signature that describes the users actions at a point in time and should be treated as such.

Therefore, we have decided to deviate from the standard approach and apply time warping to signatures as a whole, so that the same warping path is used for all features. We title this approach Multi-Dimensional Dynamic Time Warping. In Multi-Dimensional DTW, the points of the signature are not split into their individual dimen-

sions to be aligned, but the distance between signature points is calculated including all dimensions at once. This requires a change to the distance measure that was shown in equation 5.8.

Given are again two signatures $A$ and $B$ with $n$-dimensional feature vectors, as shown in equation 5.18.

$$\begin{aligned} A = a_1, a_2, \ldots, a_i &= \begin{pmatrix} f_{1_1} \\ \ldots \\ f_{n_1} \end{pmatrix}, \begin{pmatrix} f_{1_2} \\ \ldots \\ f_{n_2} \end{pmatrix}, \ldots, \begin{pmatrix} f_{1_i} \\ \ldots \\ f_{n_i} \end{pmatrix} \\ B = b_1, b_2, \ldots, b_j &= \begin{pmatrix} f_{1_1} \\ \ldots \\ f_{n_1} \end{pmatrix}, \begin{pmatrix} f_{1_2} \\ \ldots \\ f_{n_2} \end{pmatrix}, \ldots, \begin{pmatrix} f_{1_j} \\ \ldots \\ f_{n_j} \end{pmatrix} \end{aligned} \tag{5.18}$$

To calculate the distance between two points of the signature, we now use the euclidean distance over all dimensions, as shown in equation 5.19.

$$d(a_i, b_j) = \sqrt{(f_{1_a} - f_{1_b})^2 + \cdots + (f_{n_a} - f_{n_b})^2} \tag{5.19}$$

With a one-dimensional feature vector, this is identical to the original distance measure, but it can work with an arbitrary amount of dimensions. Because there is only one distance measurement between the points, the warping path is the same for every feature, and it minimizes the path where the distance between the signature for the combination of points is minimal.

As step pattern, we have chosen the quasi-symmetric White-Neely-pattern shown in figure 5.3, which is characterized by having the same weights for all possible steps. We do not require the normalization capabilities of the symmetric alternative, as in our scenario, all signatures are already brought to the same length of points before. Finally, we have defined a Sakoe-Chiba-band of 2/5 the width of the signature length, as signatures by the same person are unlikely to exhibit larger differences, and because it has been suggested in [151] that restricting the window size might improve verification performance. The concrete implementation of our multi-dimensional dynamic time warping classifier is explained in detail in the implementation chapter. The individual test scenarios in which DTW has been tested are described in results.

### 5.5.3 Machine Learning

In addition to our main approach, anomaly detection using Multi-Dimensional Dynamic Time Warping, we also evaluate different configurations of standard machine learning classifiers in different scenarios. While the individual scenarios will be described in more detail in the results section, an overview is given here. There are three general types of tests we conduct:

- Binary classification (verification) and multi-class classification (recognition) of signatures employing standard machine learning models.
- Generalized forgery detection (user-independent) using standard machine learning models.
- One-class-classification (anomaly detection) using special one-class classifiers.

All but the multi-class recognition scenario are tested with both local and global data and on a test set that includes skilled forgeries. The multi-class scenario is treated differently because it is about recognizing many different genuine writers instead of detecting fraudulent signatures drawn by impostors.

In contrast to dynamic time warping, where we developed the classification algorithm ourselves (conceptually described in the previous section, and with algorithmic details available in the implementation chapter), we relied on publicly available versions for our machine learning classifiers, which is why we will not be going into more detail on the individual classifiers. Those can be found in the documentation of Scikit-Learn and the corresponding paper [112]. Here, we focus on the machine learning process surrounding those models that we designed.

### Standard Classification

Different machine learning algorithms can be and have been applied to a signature dataset to achieve different goals. In our research, we test the following models for their performance, most of which already occurred in related work:

- K-nearest-neighbors [29]
- SVM's (linear/rbf) [97]
- Decision tree
- Random forest [28, 29]
- Multi-layer perceptron [29, 88, 104]
- AdaBoost
- Naive Bayes [28, 29]

The decision tree classifier has been added as it is the underlying classifier for random forests, and it is interesting to see how it performs compared to an ensemble. AdaBoost, while not mentioned in related work, has been added for completeness as well.

These models are all tested in both multi-class and binary classification scenarios, and are also used as forgery detectors as described in the following section. They are trained and tested using five-fold grid-search cross validation to find the best parameters and obtain averaged results on different, previously unseen data. In $n$-fold cross-validation, the available data is split into $n$ equally sized partitions. The partitioning is stratified, so that an approximately equal distribution of classes is guaranteed for all partitions. All but one of the partitions are then used as training set, and the remaining partition is held back as test set. This is repeated $n$ times with changing hold-out partitions, so that every partition is used as test set once. The reported results are the average over all $n$ cycles. Grid-search runs this process for every possible combination of a range of pre-defined parameters for the model in order to find the optimal parameter set. The CV-result obtained with that configuration is then reported as the result of the grid search. This process is repeated for each of the classifiers mentioned above.

It is important to note that no additional separate validation set was held back, as cross validation already reports the average performance on previously unseen data. For actual production performance reporting, it would still be advisable to do so as a final safety net, but the cross-validation results are already reliable performance figures. For the purpose of this study, the benefits of having all our data available to work with

clearly outweigh the need for a final check that might be desirable in production, but requires a significant part of the dataset to be excluded.

In multi-class classification tests, the process described above is executed only once for the whole set of genuine signatures, as the goal is to recognize each of the subjects correctly. For binary (subject/others) classification, on the other hand, the whole process is repeated once for every subject in the dataset, so that each subject is the positive class once, while all others are negative, resulting in one grid-search CV performance figure per subject and classifier. The mean CV results (balanced accuracy) over all subjects are then reported for each classifier for random forgeries. To also get a performance estimate on skilled forgeries, for each subject the best CV parameters of each model are used to train that model once more and test it on skilled forgeries for the subject (skilled forgeries are not included in the training set, as that would be impossible in a real application as well). As stated before, this is a realistic scenario in practice, as it is possible to maintain a set of negative signatures for all new subjects to train with, although it is not ideal, as such a negative dataset will always be limited.

### Forgery Detection

Forgery detection uses the same models as above, but with an altered machine learning process, where the genuine signatures of all subjects are one class, and the skilled forgeries for all subjects are the other class. The goal is to create a model that is able to generally detect if a signature is genuine or forged, instead of learning one specific subject. To do so, at first, five-fold grid-search-cv is used as described above to find the best models and parameters for genuine/forgery classification. Using those model/parameter combinations, results are obtained in a process called leave-subject-out cross-validation. There, a model is trained on signatures of all but one subject and its skilled forgeries, and then tested on that held-back subjects signatures. This is repeated for every subject, so that an average result of classifying signatures of previously unseen subjects can be reported. The reason for this is the need to learn how well the model generalizes. If all subjects were included in the training set, it could be that it is only able to detect forgeries for subjects it already knows, instead of learning general characteristics of forgeries. This would defeat its point of eliminating the need to train a model for each subject, provided the dataset it was trained on includes all eventualities. Therefore, training on unseen data, as in normal CV, is not sufficient, and leave-subject-out CV is used.

### One-Class Classifiers

In anomaly detection or one-class-classification, models are trained only on one (positive) class and try to classify new data as either true (similar to positive class) or false (different from positive class). This is a desirable scenario for signature verification, as only the positive (enrolled) samples are required to train, and no set of negative samples or skilled forgeries has to be maintained. There are two anomaly detector models, or three configurations, that we tested:

- One-Class-SVM (linear/rbf) [27, 129]
- Isolation forest [27]

The models are tested in a random forgery and a skilled forgery scenario, using five-

fold grid-search-cv as described above for finding the best parameters. In the random forgery scenario, half of the subjects genuine signatures are in the training set on which the best parameters are determined, and on which the model is trained. Then, that model is tested on the test set, which includes all other subjects signatures (random forgeries) and the other half of the signatures of the subject under test. As in the binary classification before, this is done for each subject, and the mean results over all subjects of each model are reported. In the skilled forgery scenario, parameter selection and training is done on all signatures of one subject, and then it is tested on all skilled forgeries for that subject. Again, this is repeated for every subject, and mean results are reported.

## 5.6 Model Updates

As signatures are a behavioral biometric feature, they are slightly different every time they are drawn and can slowly change over time - something that has been described in related work as the template ageing effect [137]. As a means to reduce this effect or to increase the amount of templates available to build the model, model updating mechanisms can be considered. Conceptually, they are a way of using signatures drawn in operation to improve the model, for example by including them as additional templates in the positive training set under certain conditions. In our research, we currently do not test any updating strategies, as our focus lies on the impact of additional features and the additional device class of notebook touchpads. However, it is an important consideration for a real-world application, which is why the possibility is reflected in our processing chain.

# Chapter 6

# Implementation

As part of the Signature Verification Project, many artifacts have been developed in order to be able to conduct the research, all of which will be made available for reference and use online. A conceptual description of our approach has already been given in the previous section, which covers everything we did theoretically. To ease understanding and make our experiments repeatable, we additionally provide implementation details on everything that affects our outcomes. However, this is not a user documentation of our processing chain, or a source of broad source of information about the architectural principles we followed in developing our software. For that, we refer to the documentation that is provided alongside the published code repositories.

There are four important pieces of software that were developed over the course of this work:

- The Android recording application ("Signature Recording")
- The Notebook recording application
- The processing and classification chain described previously
- Utility scripts

The Android recording application is used to record the mobile phone signature database. It works on every Android device that has Android 5.0 (SDK Level 21) or newer and a built-in gyroscope, requirements that almost every phone released in recent years meets. The application offers a clean, easy to understand user interface and is published on the Google Play store to facilitate participating in the study in unsupervised scenarios. The Notebook recording application is a Python script that only runs on Linux devices and might not be compatible with touchpads from all manufacturers, as it requires the `evdev` package to access the readings provided by the touchpad driver directly. It has a console-based user interface with a Signature Window built with `PyGame`, and is intended to be used in a supervised scenario only. The processing and classification chain comprises everything that concerns data preprocessing and classification as described previously, and includes scenario files to execute the different test cases. The utility scripts are used to support the actual processing chain and contain scripts to anonymize and combine the single signatures into a database, check for recording errors, plot statistics and more. While they are important to the functioning of the process, they do not directly affect the outcome, and therefore no further information about them is provided in this section.

Each of the other components contains algorithms or processes that are important when trying to replicate our findings. Those will be described in order of their appearance in our process in the following sections, starting with data recording and ending with the classified results.

## 6.1 Recording

The following section focuses on three important aspects of signature recording: The time system conversion that has to be applied for measurements on mobile, the ways of recording and aligning the measurements on both platforms, and the interpolation algorithm used to achieve uniform constant sampling rates.

### 6.1.1 Time System Conversions

On Android, our measurements are acquired from two different sources. Touch interactions with the screen are listened for in the `onTouchEvent` method, and each change is provided as a `MotionEvent` object. Changes of the different embedded sensors are all reported as `SensorEvent` to the `onSensorChanged` listener. The issue is that those two types of events use different time systems as time stamp. `MotionEvent` reports the `SystemClock.uptimeMillis`, the time in milliseconds since the system was booted, but stops when the device enters deep sleep. `SensorEvent`, on the other hand, has a `SystemClock.elapsedRealtimeNanos` timestamp, which measures the time in nanoseconds since boot, including time where the system was in deep sleep. Both clocks are guaranteed to be monotonic, so we can rely on the measurements being in order, but Android does not provide any API or documentation on how to correctly convert between them [157].

The reason for this difference in systems is the following: `SystemClock.uptimeMillis` is the basis for most interval timing, and most methods that accept a timestamp expect it to be in `uptimeMillis` [157]. Therefore, having a `MotionEvent` reported in `uptimeMillis` is convenient for the developer, as it requires one less conversion in order to be used further. However, this is not possible for a `SensorEvent`, as embedded sensors can also report values in the background, when the screen is off and the phone is sleeping, at which time the `SystemClock.uptimeMillis` is paused and only `SystemClock.elapsedRealtime` is still active [157]. In the vast majority of applications, this difference is not a problem, because it is hardly ever required to synchronize touch events with embedded sensor measurements to the millisecond. Thus, there is no documentation available on how to convert the timestamps among each other, and to the best of our knowledge, no API is available that does so or provides the current difference between the two clocks.

There are two possible solutions to this problem that we have considered. The first is to ignore the timestamps reported with the actual `MotionEvent` and `SensorEvent` entirely, and instead obtain our own timestamps at the time the events are reported in the respective listener method with a clock of our choosing. The problem with this approach is that we do not know the possible delay between occurrence of an event and its reporting in the listener, and more importantly, there is no guarantee that the delay is constant, as it is handled in the background by the Android Framework. Therefore,

we have gone with the second option, calculating the offset between the two clocks on our own, for each signature again at the time it is saved. We know for certain that the offset does not change between starting the signature and touching save, as both are monotonic clocks and the phone is constantly awake with the screen on during the time of signing, in which case both clocks are consistently active. In order to calculate the difference as shown in equation 6.1, we have to obtain measurements of both clocks at the same time.

$$\Delta t = t_{ert} - t_{utm} \tag{6.1}$$

While measuring at the exact same time is not possible, we try to come as close as we can. The measurements are taken in two immediately consecutive instructions in the same method, as shown in listing 6.1. As both instructions are simple get requests, it is safe to assume that the execution time for each, and thus the time difference in acquiring them, is minimal. Moreover, we have tested it multiple time empirically if there is any measurable time difference between two clock readings (of the same clock) taken directly after each other, and at millisecond-accuracy, we could not find one. Therefore, we consider it safe to assume that our way of calculating the offset is accurate to a maximum difference of one millisecond, if it flips between the two instructions.

```
1 val utm = SystemClock.uptimeMillis()
2 val ert = SystemClock.elapsedRealtime()     // Returns milliseconds
3
4 val timeSystemDifference = ert - utm
```

**Listing 6.1:** The code for calculating the time system difference.

For the calculation, we can be sure that `SystemClock.elapsedRealtime` is always equal or greater than `SystemClock.uptimeMillis`, as both return the time since boot, but the latter can be paused at times, while the former is not.

$$utm_{millis} = \frac{ert_{nanos}}{1000000} - \Delta t \tag{6.2}$$

Finally, the conversion is applied to all embedded sensor measurements. Using the time system offset, the timestamps are brought from $ert_{nanos}$ to $utm_{millis}$ at the time of alignment and interpolation as shown in equation 6.2.

### 6.1.2 Alignment

On mobile, we have two types of measurements that are part of a signature: The touch interaction on screen, and the different embedded sensor readings recorded while signing. A signature starts with the first touch interaction on screen, and ends with the last time the screen was touched before the signature was confirmed - the time of actually drawing on screen. However, the Android framework does not provide a way of querying a movement sensors current reading every time a new `MotionEvent` is reported to the `onTouchEvent` listener, or to constantly poll both touchscreen and movement at fixed (synchronized) intervals. The only way to obtain measurements is to register a listener beforehand and wait for events to occur, for the movement sensors with a specifiable sampling rate (we use `SENSOR_DELAY_GAME`), and for the screen with no customization

at all. Therefore, the timestamp of the first `MotionEvent` (the first touch interaction) marks the starting time, and all other interactions have to be made to fit.

```
# Raw touch (T) and movement (M) sensor readings recorded at inconsistent rates
T:       00 -- 02 -- 04 -- 06 -- -- 09 -- 11 -- -- 14 -- 16 -- 18 -- -- 21
M: 00 -- -- -- 04 -- -- 07 -- -- -- 11 -- -- 14 -- -- 17 -- -- -- 21 -- -- -- 25

# Interpolated time series with one value every millisecond
T:       00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21
M: 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

# Set sampling frequency (F) (here: every 5 milliseconds/200Hz) and select.
F:       FF -- -- -- -- FF -- -- -- -- FF -- -- -- -- FF -- -- -- -- FF
T:       00 xx xx xx xx 05 xx xx xx xx 10 xx xx xx xx 15 xx xx xx xx 20 xx
M: xx xx 02 xx xx xx xx 07 xx xx xx xx 12 xx xx xx xx 17 xx xx xx xx 22 xx xx xx

# New, consistent signature recording with values aligned
T:       00 -- -- -- -- 05 -- -- -- -- 10 -- -- -- -- 15 -- -- -- -- 20
M:       02 -- -- -- -- 07 -- -- -- -- 12 -- -- -- -- 17 -- -- -- -- 22
```

**Listing 6.2:** The concept of interpolating and aligning signature features to combine them into one signature. Numbers are recorded values and lines denote time intervals where no value was recorded. The position of each value indicates its timestamp.

The concept of how this is done is shown in listing 6.2. For the touch events, it requires interpolation and sampling to a defined fixed rate. For embedded sensor readings, interpolation and sampling are also required to bring the events that are reported at roughly the specified rate to the exact same sampling rate as touch interactions, but additionally, the movement recordings also have to be aligned with the touch screen interactions. As we do not know when the user is going to start the signature beforehand, and because we can not access the current sensor readings directly, we have to start recording the embedded sensors before the user even touches the screen in order be able to interpolate the values at the time of first touch contact. We do this by using a buffer of 400 ms (20 values at 50 Hz) that is constantly updated with the latest reading pushing out the oldest one as long as signing has not been started, and that continues to record all subsequent readings without deleting old ones as soon as the screen has been touched. The sensor readings then continue to be recorded until the user presses the save button, even when the screen is not being touched, as a signature can consist of multiple strokes and we have no way of knowing which one is the last.

Once save has been pressed, the timestamps of the recorded movement readings are converted to the reference system the touch events use (`SystemClock.uptimeMillis`) as described in the previous section. Then, it is verified that the timing is valid, meaning that using the converted timestamps, the sensor readings still start before the first and end after the last touch recording. If so, all measurements are interpolated using the interpolation algorithm described in the following section that is used in both mobile and touchpad applications. Subsequently, the excess movement sensor readings are cut off, and the interpolated readings are sampled to a fixed rate using the (also common) sampling algorithm with the first touch event as start point. This results in an automatic alignment of touch and movement readings.

### 6.1.3 Interpolation

We use linear interpolation to calculate the values in between the measurements. A special case are all screen-related features, as there are no values available in times between strokes because the finger is in the air, and the measurement arrays contain null values. For those, the finger-up intervals - from the end of one stroke to the start of the next - are interpolated with null values, and not with interpolated in-air-trajectories to the beginning of the next stroke at this stage (however, this is done later in the processing chain).

Conceptually, the linear interpolant is the straight line between two points. The linear interpolation of a series of points is nothing more than the concatenation of interpolations of two points, where the end point of the previous pair becomes the start point of the next, and the point immediately following it in the series becomes the new end point, until the end is reached. The calculation of the interpolated value $v$ at a time $t$ between two points is shown in equation 6.3.

$$v(t) = v_0 + (t - t_0) \cdot \frac{v_1 - v_0}{t_1 - t_0} \tag{6.3}$$

In our implementation (shown in algorithm 6.1), we interpolate the raw measurements with one value every millisecond. So if the original data input into the algorithm consisted of two points recorded at $t = 0$ and $t = 3$, the result would be an array of four points, with $t = 1$ and $t = 2$ being interpolated values. This is certainly not the most efficient way to do this computationally considering that we do not need all the values and perform a selection later on. However, in terms of real-world performance impact, the differences are negligible and not noticeable by the user, and in our experimental environment, it makes experimentation with sampling techniques easier down the line. The interpolated values of screen coordinates are always rounded to the closest integer, all other values are not rounded.

**Algorithm 6.1:** Linear Interpolation Algorithm

**Require:** startTime, endTime, startVal, endVal
**Ensure:** $endTime - startTime > 1$
  $amount \leftarrow endTime - startTime$
  $result \leftarrow [amount]$
  **for** $i \leftarrow 0$ **to** $amount$ **do**
    $result[i] \leftarrow startVal + i * (endVal - startVal)/amount$
  **end for**
    **return** result

### 6.1.4 Sampling

The final recording step is sampling the interpolated signatures to a fixed rate. On mobile, this has the added benefit of aligning the sensor measurements with the touch interactions as shown above. At first, the now superfluous movement data recorded before the start and after the end of signing on screen has to be cut off. This is done by comparing the timestamps (at this point already in the same reference system) and

removing all movement events with event times lower than the start or higher than the end time of the drawing on screen. The result are multiple arrays of the same length, one for every feature of the signature, where at every index position there is an (interpolated) measurement in one millisecond steps. Sampling is now as easy as selecting every $n$-th index position. For example, to sample to 50 Hz, every 20th index position is selected. This process is shown in algorithm 6.2. Assuming that all *timeSeries* have been cut off to fit the display recordings already, the results for the different features can simply be combined to form the final signature ready to be stored and processed.

**Algorithm 6.2:** Sampling Algorithm

**Require:** timeSeries, interval
**Ensure:** $interval > 0, timeSeries.length > 0$
  $duration \leftarrow timeSeries.length$
  $result \leftarrow [duration/interval + 1]$
  $i \leftarrow 0$
  **while** $i < duration$ **do**
    $result[i] \leftarrow timeSeries[i * interval]$
    $i \leftarrow i + 1$
  **end while**
    **return** result

We have chosen this method to come as close as possible to the result we would have gotten when querying the touch and movement sensors consistently in the first place. Compared to other methods that attempt to preserve more details in the signature at the cost of less consistent sampling rate or timing information, we consider this to be the least disturbing approach, especially considering that further down the actual processing chain, the sampling resolution of the signatures is reduced to a point where those details are lost anyways. Also, it is the easiest to understand and reproduce and the least opinionated on how and which details to preserve, which is especially important as the datasets are going to be published and we could not find an agreed-upon information on how to record signatures in related work.

With our approach, we implicitly assume the events reported to the listener methods to be an exact representation of what happened on screen. In reality however, those measurements are also just obtained at a sampling rate of 50 Hz for embedded sensors and at least 60 Hz for touch events. This means that by sampling them again, we are increasing the difference to what actually happened on screen, an effect more complicated sampling methods try to avoid. Examples of some different strategies and their problems are shown in listing 6.3. The fixed sampling method S1 and the alternative sampling method A1 both use a relatively low sampling frequency for the amount of detail in the interpolated measurement IM. This causes S1 to miss one short stroke entirely. A1 circumvents this problem by adding an intermediate sampling point to capture that stroke in the sampled result, but at a cost. Either, the result is still assumed to contain equidistant measurements, which would mean stretching the time between the third and fourth measurement from 2ms to 5ms and distorting timing information. Alternatively, the result can store timestamps of each measurement for a variable sampling rate, making it hard to process and potentially impossible to use later on, for

example with DTW, which requires a constant distance between points of a time series. S2 and A2 use a higher sampling rate, and A2 also showcases another strategy of always capturing the start- and end point of each touch stroke by inserting additional sampling points. With this higher rate, S2 captures all strokes of the signature, although not perfectly, as the ends are cut off (07, 13, 25). A2 on the other hand fixes that problem, but with the same downside that was described above. The same applies to every other alternative strategy, not just the two that served as examples here.

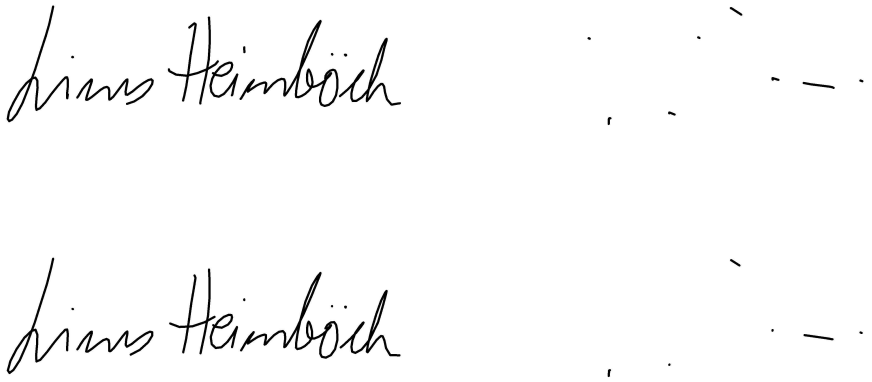
```
# Low sampling rate
S1: FF -- -- -- -- FF -- -- -- -- FF -- -- -- -- FF -- -- -- -- FF -- -- -- -- FF
A1: FF -- -- -- -- FF -- -- -- -- FF -- FF -- -- FF -- -- -- -- FF -- -- -- -- FF
IM: 00 01 02 03 04 05 06 07 nn nn nn nn 12 13 nn nn nn nn 18 19 20 21 22 23 24 25

# Low sampling rate results
S1: 00 05 nn nn 20 25
A1: 00 05 nn 12 nn 20 25

# High sampling rate
S2: FF -- -- FF -- -- FF -- -- FF -- -- FF -- -- FF -- -- FF -- -- FF -- -- FF --
A2: FF -- -- FF -- -- FF FF -- FF -- -- FF FF -- FF -- -- FF -- -- FF -- -- FF FF
IM: 00 01 02 03 04 05 06 07 nn nn nn nn 12 13 nn nn nn nn 18 19 20 21 22 23 24 25

# High sampling rate results
S2: 00 03 06 nn 12 nn 18 21 24
A2: 00 03 06 07 nn 12 13 15 18 21 24 25
```

**Listing 6.3:** Different sampling strategies and their problems.

These concerns are only relevant for touch measurements that can contain null-values and can therefore not be interpolated consistently (or only by assuming pen-up trajectories, which causes the same problem of geometry accuracy). For movement features that are recorded without interruption, these considerations are irrelevant altogether. Therefore, in systems that mainly rely on the signature geometry, making the trade-off of sacrificing some timing accuracy for a bit more accuracy in the image might be worth it. However, for a general purpose dataset and a dynamic scenario, we consider it best to not introduce our own assumptions and sample at a fixed rate.

Also, one more thing has to be noted. The measurements for embedded sensors are reported consistently by Android, with at most small millisecond differences in intervals, so interpolation never has to bridge large gaps in measurements, and are recorded at 50 Hz. The touch sampling rate for smartphone displays is even higher with at least 60 Hz, which should capture all meaningful interaction [76]. Therefore, the interactions that are reported in the listener, while not perfect, can be considered an accurate representation of what happened on screen. This means that the error of the data we are performing our sampling on is low, and our experiments in real life confirm that the additional error we introduce by sampling is not large in realistic scenarios either. Two examples are shown in figure 6.1.

It can be seen that for a normal signature, the difference between the original recording and the sampled version is minimal. All four i- and ö-dots, which are only short touches while writing, are still there, and the signature shape is almost indistinguishable. A different case is purposeful fast random tapping on screen. Two of the points present in the original pattern have been lost in interpolation, as they were too short



(d) Normal writing (e) Random fast tapping

**Figure 6.1:** Examples of a signature and an artificial pattern, before and after sampling. Top are the originals, bottom the sampled versions.

contacts to be picked up with our sampling rate of 50 Hz. Also, it can be seen that the lines have been slightly shortened. In our own tests, loss of fast fine details such as i-dots has occurred in less than 10 percent of cases when writing real signatures, significantly less frequent than cases where we forgot to dot our i's and ö's ourselves. When observing our subjects signing, the interpolations also always looked equal to what was drawn on screen to the naked eye (interpolated version is shown in-app as soon as save is pressed), although we did not pay attention to the preservation of fine details specifically. Therefore, we are confident in the accuracy of our recordings. Also, in the processing chain, the resolution is lowered anyways to improve classification computation performance.

## 6.2 Processing

Most of the processing chains implementation is already sufficiently covered by the description of the four steps in the previous sections, as the formulas are simply translated into code, and the filtering steps can be easily done with standard libraries. In this section, we discuss the resampling of the signatures to a uniform length of 50 measurements.

At the end of the processing chain, once all features are extracted, all signatures are brought to the same length, so they can be combined into a dataset all models can work with. Other than in the recording applications, we did not implement the interpolation used in this algorithm ourselves, but utilize tools from the common `numpy` library. The python code is shown in listing 6.4.

The interpolation itself is done at line 11 using `np.interp`. It returns the linear interpolant of a series of discrete data points at specifiable points in time given in `x_new`. As we have a fixed number of values we want in our outcome with equal time distance between them, we determine the positions at which to sample between the start- and

end point using `np.linspace`. For example, if the original series had 11 measurements (indices from 0 to 10) and should be resampled to 6 measurements, it would return a series of points where to "read" the interpolation that looks like $x_{new} = \{0, 2, 4, 6, 8, 10\}$, resulting in a selection of only every second point and thus a downsampling. Conversely, if the series had only 3 measurements (0 to 2) and should be resampled to 5, it would result in $x_{new} = \{0, 0.5, 1, 1.5, 2\}$ with additional reading points in between, and thus an upsampling. `x_old` specifies the positions of the input points `y_old`, which is the same as the index in our case as they are sampled at an equal rate. As opposed to the sampling during recording, we do not have issues with finger-up times in this case, because as part of the preprocessing, those missing values are linearly interpolated.

```
1  def resample_fixed(df: pd.DataFrame, n_new: int) -> pd.DataFrame:
2      n_old, m = df.values.shape
3      mat_old = df.values
4      mat_new = np.zeros((n_new, m))
5      x_old = np.linspace(df.index.min(), df.index.max(), n_old)
6      x_new = np.linspace(df.index.min(), df.index.max(), n_new)
7
8      for j in range(m):
9          y_old = mat_old[:, j]
10         y_new = np.interp(x_new, x_old, y_old)
11         mat_new[:, j] = y_new
12     return pd.DataFrame(mat_new, index=range(0, n_new), columns=df.columns)
```

**Listing 6.4:** Signature resampling to a standard length of 50 measurements.

## 6.3 Classification

For the classification part of the processing chain, we are only focusing on our implementation of dynamic time warping, as we used the scikit-learn [112] implementations for standard machine learning tasks (such as random selections, grid search etc.) and classifiers. Therefore, it should be clear from the description in approach how evaluation and classification were done, together with the documentation of scikit-learn which is available online.

### 6.3.1 Multi-Dimensional Dynamic Time Warping

Our implementation of the dynamic time warping algorithm is shown in listing 6.5. The arguments `s` and `t` are the two signatures as lists of tuples holding the features that are included in the distance computation. The `window` parameter is used to define the band within which the DTW can operate.

At first, a matrix of the dimensions $n \times m$, the lengths of the signatures, is created and filled with infinity values. Every cell is a possible match of points of the signatures and holds the cost (distance) to reach that point. The starting point, as well as all cells within the defined window, are set to 0. At line 16, the actual distance computation starts for all combination of points within the window. While the algorithm is defined recursively, we implemented an iterative version. The `cost` is the euclidean distance between the two tuples at the current position in the matrix (`i - 1` and `j - 1` because the matrix is one element larger in each dimension than the actual signatures to accommodate the

starting point at the first index). Then, the lowest cost so far from the previous step is selected, and the current cost is added to calculate the value of the cell. This results in a filled matrix (within window borders) of costs, and the lowest path cost as content of the last cell.

```
1 def multi_dim_dtw(s: tuple, t: tuple, window: int):
2     n, m = len(s), len(t)
3     w = np.max([window, abs(n - m)])
4     dtw_matrix = np.zeros((n + 1, m + 1))
5
6     for i in range(n + 1):
7         for j in range(m + 1):
8             dtw_matrix[i, j] = np.inf
9     dtw_matrix[0, 0] = 0
10
11    for i in range(1, n + 1):
12        for j in range(np.max([1, i - w]), np.min([m, i + w]) + 1):
13            dtw_matrix[i, j] = 0
14
15    for i in range(1, n + 1):
16        for j in range(np.max([1, i - w]), np.min([m, i + w]) + 1):
17            cost = distance.euclidean(s[i - 1], t[j - 1])
18            last_min = np.min([dtw_matrix[i - 1, j],
19                               dtw_matrix[i, j - 1],
20                               dtw_matrix[i - 1, j - 1]])
21            dtw_matrix[i, j] = cost + last_min
22    return dtw_matrix
```

**Listing 6.5:** Our implementation of the multi-dimensional dynamic time warping algorithm.

### 6.3.2 Classification and EER Estimation

The classification function uses the MD-DTW algorithm to calculate the distance between model (`train`) and unknown (`test`) signatures using a specified set of `features`, based on which the decision is taken if a signature is genuine or forged. The algorithm is shown in listing 6.6.

The system works as follows: First, the intra-class distances are calculated. Those are the distances between all possible combinations of signatures in the training set. Then, for each of the signatures in the test set, the distance to all of the signatures in the training set are calculated. The function `get_row_tuple()` selects all desired features of a signature and converts it to a list of tuples to be used with the multi-dimensional dynamic time warping algorithm. From those distances between each test signature and the training set, the mean is calculated, which is the reported distance of that signature to the model.

```
1 def dtw_classification(train: pd.DataFrame, test: pd.DataFrame, features: [str]):
2     intra_class_distances = intra_class_dtw(train, None, features)
3     distances = [None] * test.shape[0]
4
5     for i in range(0, test.shape[0]):
6         current_row_tuple = get_row_tuple(test.loc[i, :], features)
7         scores = [None] * train.shape[0]
8
```

```
        for j in range(0, train.shape[0]):
            comparison_row_tuple = get_row_tuple(train.loc[j, :], features)
            dtw_matrix = multi_dim_dtw(current_row_tuple, comparison_row_tuple,
    WINDOW_SIZE)
            scores[j] = dtw_matrix[dtw_matrix.shape[0] - 1, dtw_matrix.shape[1] - 1]

        distances[i] = mean(scores)
    return distances, intra_class_distances
```

**Listing 6.6:** Classification using the multi-dimensional DTW algorithm.

Listing 6.7 shows a simple exemplary prediction mechanism of how the resulting distances can be used for classification. Based on the average distance between the model signatures and their standard deviation, a threshold is defined, and all signatures that are equal to or below that threshold distance are considered to be positive. All others that are above the threshold are considered to be negative. This is not used to generate our results, it is only intended to give an example of how classification could be done.

```
def predict(model, distances):
    avg_icd = mean(model)
    stdev_icd = stdev(model)
    threshold = avg_icd + stdev_icd * 1.5

    preds = [1 if dist <= threshold else -1 for dist in distances]
    return preds
```

**Listing 6.7:** A simple exemplary predict function based on the intra-class distances. This is not actually used - all results are generated using the equal error rate calculation in listing 6.8

We obtain our own results using the equal error rate estimation shown in listing 6.8. We call it an estimation because it can not tell us the exact equal error rate, but approximates it by setting the threshold to classify the signatures in the test set so that false negative rate and false positive rate are as close as possible to being equal. This is also why there are two values for the EER, `eer` and `eer_check`. Unless FNR and FPR are exactly equal, we do not know what the best EER would be, and therefore report the FPR and FNR that are the closest together as `eer` and `eer_check`. In our results and for threshold reporting, we use `eer` as the EER value to report.

The algorithm works as follows: The distances between the model and each of the test signatures calculated in 6.6 are used together with the ground truth labeling `y_true` to calculate an ROC-curve that serves as basis for the estimation. The false negative rate is calculated, and the index where FPR and FNR are closest is selected using `np.nanargmin`. The same is done to select the threshold, and the results are reported.

```
def calculate_eer(y_true, distances):
    fpr, tpr, thresholds = roc_curve(y_true, distances, pos_label=1)
    fnr = 1 - tpr

    eer = fpr[np.nanargmin(np.absolute((fnr - fpr)))]
    eer_check = fnr[np.nanargmin(np.absolute((fnr - fpr)))]
    eer_threshold = thresholds[np.nanargmin(np.absolute((fnr - fpr)))]
    return eer, eer_check, eer_threshold
```

**Listing 6.8:** Calculation of the EER using the distances from MD-DTW classification.

# Chapter 7

# Results

In this chapter, we present the results of our research and answer the questions stated in chapter 1. We discuss the following main points:

- The performance of signature verification on notebook touchpads and how it compares to the performance on mobile phones using the same features.
- The influence of signing position and enrollment modalities on performance.
- The performance of signature verification on smartphones when using additional movement and display features that are recorded while signing, and the individual importance of those features.
- The potential interoperability effects between signatures from different devices.
- The evaluation of Multi-Dimensional DTW as a means of classification, and comparison to global machine learning approaches.
- The performance of touchpad and mobile signature verification in global classification scenarios and the impact of statistical movement features on the results.
- The usability of signing on a notebook touchpad, and how it compares to signing on mobile phones.

All results presented here have been generated using the code that is available online on our own public dataset. Due to the large amount of scenarios that we tested, we can only present a selection of important ones here. However, all of our result files are accessible online as well.

## 7.1 Scenarios

We have defined a set of different scenarios for which we generate our results. In order to make the results more readable, we describe the scenarios here, and will refer to them throughout the chapter. The scenarios in table 7.1 are grouped in groups of four, identifiable by the separating lines in between, that have an identical setting and only differ in enrollment details from each other.

The scenario definitions can be read as follows. Pos. Train states if the model was trained on signatures of only one stance (Single) or all three stances (Multi). Pos. Test describes the composition of the test set on which that model was tested. Same means that the model was tested on genuine signatures of the same stance it was trained on,

| Szenario | Pos. Train | Pos. Test | Enrollment | Session | Train size |
|---|---|---|---|---|---|
| S1 | Single | Same | First | Single | 5 |
| S2 | Single | Same | First | Multi | 5 |
| S3 | Single | Same | Random | Single | 5 |
| S4 | Single | Same | Random | Multi | 5 |
| S5 | Multi | Single | First | Single | 6 |
| S6 | Multi | Single | First | Multi | 6 |
| S7 | Multi | Single | Random | Single | 6 |
| S8 | Multi | Single | Random | Multi | 6 |
| S9 | Single | Multi | First | Single | 5 |
| S10 | Single | Multi | First | Multi | 5 |
| S11 | Single | Multi | Random | Single | 5 |
| S12 | Single | Multi | Random | Multi | 5 |
| S13 | Multi | Multi | First | Single | 6 |
| S14 | Multi | Multi | First | Multi | 6 |
| S15 | Multi | Multi | Random | Single | 6 |
| S16 | Multi | Multi | Random | Multi | 6 |
| S17 | Single | Other | First | Single | 5 |
| S18 | Single | Other | First | Multi | 5 |
| S19 | Single | Other | Random | Single | 5 |
| S20 | Single | Other | Random | Multi | 5 |

**Table 7.1:** The different scenarios that we defined. Note that scenario results S13-S16 are conceptually the same as the mean over all three positions in scenarios S5-S8. Also, scenarios S9-S12 can also be calculated by combining the results for each training position of scenarios S1-S4 and S17-S20 and taking their mean.

with one result per position. Single (for Multi-Position-Models only) means that the model was tested on each stance individually, leading again to one result per position. Multi means that the model was tested using genuine signatures from all stances. Other means that the model was tested on the stances that were not in the training set individually, leading to two results per training position. Enrollment was either done using only signatures from the first session, or signatures from both sessions, either split in 3-2 or, for multi-position models, in 3-3 fashion over the sessions. First-based enrollment means that the first signatures of each session and position were used to train. Random means that the training signatures were randomly picked within their categories, in order to test if there are performance differences when the first signatures are used for training. Train size states the total number of signatures that were used to train the model.

The results, if not stated otherwise (for subset-results), are obtained on a set of 10 random genuine signatures from the first session, 10 random genuine signatures from the second session, and all 20 skilled forgeries for the subject or 20 randomly selected random forgeries for skilled and random forgery scenarios respectively. The genuine

signatures are selected from the positions as defined in the scenarios above, the skilled forgeries are always sitting.

The different groups of scenarios have different purposes. S1-S4 test how well a model trained on signatures for one stance can predict signatures of the same stance. These are likely the most important scenarios to answer our question about the usefulness of movement data. Scenarios S5-S8 test if it is possible to build a generalized model that is able to predict all stances if it is trained on all stances, and allow us to detect eventual performance penalties suffered from doing so. These are highly desirable scenarios in practice, as they do not require multiple models to cover all stances. They also allows us to see if multi-position models can predict some stances better than others. Szenarios S9-S12 test if there is one position that is particularly good at predicting all positions. Szenarios S13-S16 test multi-class models on multi-class test data, which is very close to a realistic application. Finally, szenarios S17-S20 test cross-prediction from one position to other positions, to find combinations of positions that perform well or are interchangeable. For touchpad signatures, where only one position has been recorded, only scenarios S1-S4 are applicable, with the only position being sitting.

## 7.2 Enrollment Strategies

Before we start discussing our experiments on different platforms with our touchpad results in the next section (Sec. 7.3), we first need to provide some more information on the enrollment strategies we consider. There are multiple reasons why this is important. In our usability study, about 75% of users have reported that the look and execution of their signature on touch devices is at least partially different from their signature on paper, and even more find signing on paper more comfortable than signing digitally with their finger. Especially on touchpad, the results and our experiences during recording suggest that users might need a few practice signatures before being able to draw consistent signatures on the unfamiliar surface that are suitable for enrollment. Also, related work [137] has found that using samples from multiple sessions for enrollment helps reducing the template ageing effect and increases verification performance.

Therefore, we have defined different enrollment strategies for each of our scenarios. Using the same amount of total enrollment signatures every time, we enroll users either with the first or with random signatures from either one or both signing sessions, resulting in a total of four different scenarios. While a random enrollment is not a realistic scenario in practice (if more signatures are available, one would likely use all, or the ones with the lowest variance among each other, or discard the first ones, instead of randomly selecting), it is the most neutral way of enrolling. While any pre-defined selection (eg. first signatures, last signatures, 5 signatures after the first three) might each produce effects of its own, a random selection provides a baseline.

Our results are presented in table 7.2. There is a clear pattern, where random enrollment performs better than first-based enrollment, and multi-session enrollment performs better than single session enrollment. The effects also stack up, with the best strategy consistently being a random multi-session enrollment. The same effect, although less clear, can also be observed on random forgeries. Also, it is noticeable regardless of the data used on mobile, showing that it is relevant not only for XY-based, but also for movement-based or combined verification, and thus for features that are not generated

| **Enrollment** | $T_{XY}$S1-S4 | $M_{XY}$S1-S4 | $M_{XY}$All | $M_{Mvmt}$All | $M_{Com}$All |
|---|---|---|---|---|---|
| First, Single | 6.78% | 6.02% | 6.68% | 27.02% | 6.62% |
| First, Multi | 5.56% | 5.14% | 5.65% | 25.77% | 6.16% |
| Random, Single | 6.22% | 5.52% | 5.63% | 26.05% | 5.85% |
| Random, Multi | 4.22% | 3.90% | 4.60% | 24.50% | 4.77% |

**Table 7.2:** The results for each enrollment strategy. T for touchpad, M for mobile. The lowered letters indicate what data was used, either only XY or combined movement and XY. The numbers indicate which scenario group was used, all stands for the average over all groups. On mobile, the results are the average over all positions. All results on skilled forgeries.

consciously. It is observable in all three positions, not only sitting (which is the first one that is recorded by most users), suggesting that the short break between positions, and/or the change in position require some re-adaptation that causes the first signatures of each position to be worse. This supports the hypothesis from our usability study that users need some time to get used to signing on an unfamiliar surface. Comparing between mobile phone and touchpad, the effect is about the same. Our results also support the findings of [137], that ideally two sessions should be used for enrollment. Based on our results, we recommend to require a user to complete a few training signatures before recording the actual enrollment signatures of each position.

## 7.3 Touchpad Signature Verification

In this section, we discuss the evaluation of signature verification on notebook touchpads and report our findings. We compare the performance on random and skilled forgeries to the results on mobile phones in an equal setting, and go into depth on the similarities and differences between the two platforms. Based on these results, and parts of the results in section 7.4, we answer the corresponding research question in chapter 8.3.

| **S1-S4, Si, XY** | Mobile Sk | Touchpad Sk | Mobile Rd | Touchpad Rd |
|---|---|---|---|---|
| EER | 5.87% | 5.70% | 0.70% | 0.78% |
| STD | 8.33% | 7.98% | 3.00% | 2.88% |

**Table 7.3:** The mean results of scenarios S1-S4 for mobile and touchpad signatures using MD-DTW, with only XY-Data and only in the sitting position. Sk stands for results on skilled forgeries, Rd for results on random forgeries.

Table 7.3 displays the mean results and mean standard deviations on random and skilled forgeries for touchpad and mobile signatures over scenarios S1-S4, achieved with our MD-DTW classification, when sitting and only using XY-data for classification. This is ideal for comparisons between the two device classes, as the scenarios are identical on mobile and touchpad. Thus, any differences in performance can be attributed to issues with the respective type of signing device. The only difference in the underlying

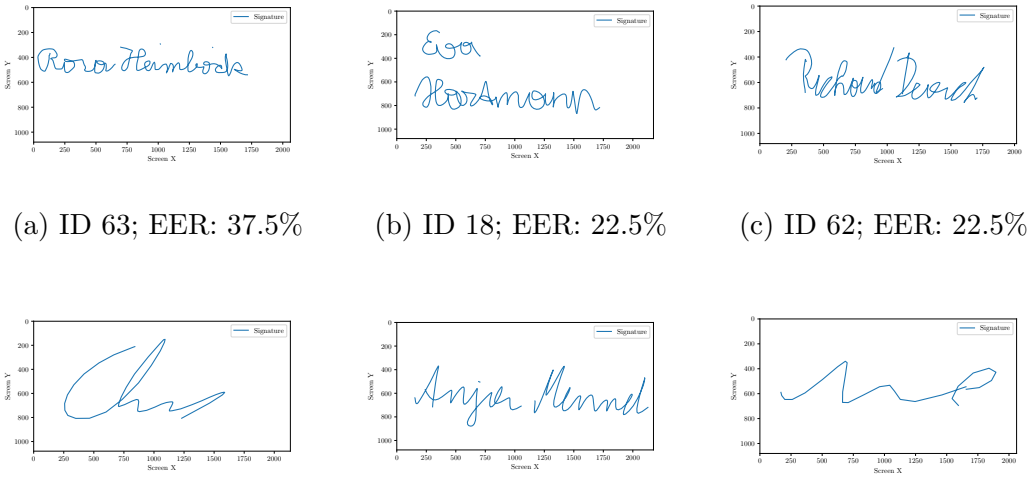
datasets is that on mobile, signatures from multiple devices are used, while on touchpad, all signatures have been recorded on the same device. We will go into depth on the effects of inter-device classification on mobile later, but can preface that the impact it has here is minimal.

| **EER, Si, XY** | Mobile Sk | Touchpad Sk | Mobile Rd | Touchpad Rd |
|---|---|---|---|---|
| S1 | 7.64% | 6.78% | 0.57% | 1.33% |
| S2 | 5.57% | 5.56% | 0.64% | 0.89% |
| S3 | 6.21% | 6.22% | 0.57% | 0.44% |
| S4 | 4.07% | 4.22% | 1.00% | 0.44% |

**Table 7.4:** The individual results of scenarios S1-S4 for mobile and touchpad signatures using MD-DTW, with only XY-Data and only in the sitting position. Sk stands for results on skilled forgeries, Rd for results on random forgeries.

Table 7.3 shows that there are no significant differences in performance between the two types of devices when the same features are used, neither for EER nor for STD. On skilled forgeries, both reach EERs of about 5.75%, and on random forgeries, both are at around 0.75% EER. The same is true when we look at the individual scenario results shown in table 7.4. While there are differences up to about 0.9 percentage points EER between individual scenarios, the results are very similar overall, and neither of the devices performs significantly better than the other. For touchpad skilled forgery results, the same pattern is noticeable as in all skilled forgery mobile results, that certain enrollment scenarios perform better than others, an effect we will also discuss later in more detail. When also considering the results achieved on the SMG950F subset of the mobile database (all signatures from the same device, as on touchpad), the touchpad tends to perform slightly better on average than mobile, with a minor increase in mobile EER to 6.30% and a standard deviation of 9.73% on skilled forgeries. On random forgeries, the EER decreases to 0.65% with a standard deviation of 2.58% on mobile, so the picture is not clear.

Some smaller differences are noticeable when comparing the performance of subjects on an individual level. To do so, we have defined a threshold of 10% EER as an acceptable level of performance, as this is the approximate average performance of similar related work. Then, we selected all signatures that performed worse in at least one of the four scenarios, for both touchpad and mobile signatures. The amount of signatures that are above the error threshold are the same for mobile and touchpad. For both device classes, about 31% of all signatures in the respective dataset fall below the threshold once, 20% twice, 10% three times, and 6% fail in all four scenarios consistently. This shows that for both notebook and phone signatures, the same amount of subjects can not be verified well consistently in similar scenarios, depending on the configuration of training and test set. The similar drop-off rate shows that both have only a few users that can never be verified well, but gradually more that appear to have some genuine signatures that make classification more difficult (the skilled forgeries in the test set always stay the same). Also, a subject being part of the inconsistent-performers on one device class does not necessarily mean that the same subject is also part of it in the other class. Only




(a) ID 63; EER: 37.5% (b) ID 18; EER: 22.5% (c) ID 62; EER: 22.5%

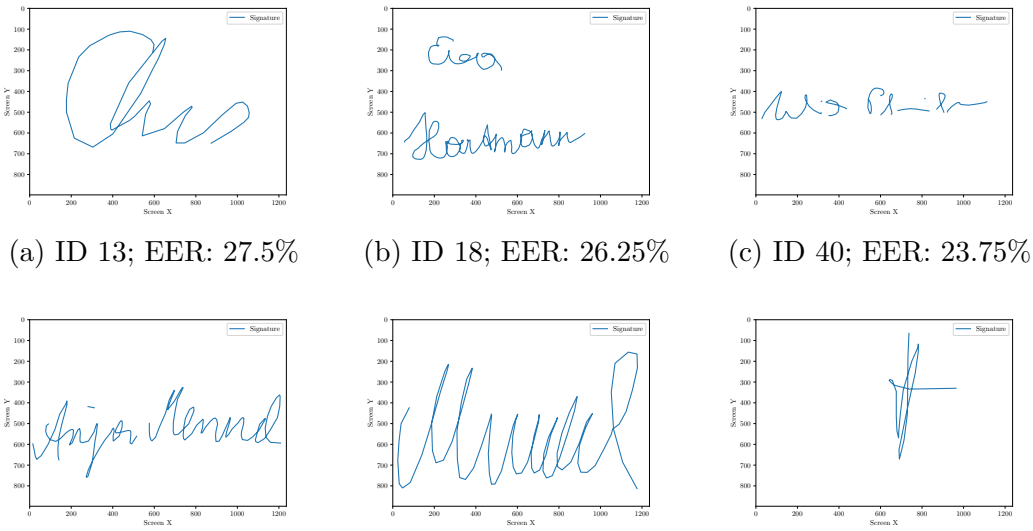
(d) ID 13; EER: 20.0% (e) ID 3; EER: 18.75% (f) ID 51; EER: 18.75%

**Figure 7.1:** Signatures of the six mobile subjects that performed worst on average over scenarios S1-S4 in skilled forgery tests using only XY-data.

40% of the subjects that are in both datasets and failed at least one scenario in one device class also failed at least once in the other device class. Only 15% of the subjects that were in our selection for at least one of the two devices used a different version of their signature on the touchpad (split the signature in two lines). As the processing and classification used is exactly the same, the overlap should be much higher if there were no difference between the two platforms.

A possible explanation can be seen when looking at the six worst performing subjects of each platform, as shown in figures 7.1 and 7.2. On mobile, four of them are well-readable written names, with few visible differences between individual signatures. Only judging by their high length and low variance, one would expect them to perform well. However, with direct visual feedback as on phone screens, they are easy to imitate both in form and drawing speed, by simply writing the name. Only two are very simple patterns one would expect to be the worst performers due to their low complexity. On the touchpad, only two of the signatures are well-readable written names (b and d), one of which (b) shows a visibly higher difference between signatures than on mobile, making classification harder. Signature c has a very high variance in how it is drawn, and is therefore hard to distinguish from any forgery with the same general shape, making it more of a pattern rather than a written name. Three of the signatures are clear simple patterns (a, e, f) that are easy to imitate without keeping track of the exact finger position or having to lift the finger from the touchpad. These results are consistent with what our forgers reported to be the easiest to forge on the respective platform, except for signature d on touchpad, where its bad performance despite being relatively long can not be explained with an unusually high variance. Overall, the worst performances

on mobile are worse than those on touchpad, with an average of 23.3% vs. 19.7% EER.




(a) ID 13; EER: 27.5% (b) ID 18; EER: 26.25% (c) ID 40; EER: 23.75%

(d) ID 3; EER: 13.75% (e) ID 7; EER: 13.75% (f) ID 57; EER: 13.75%

**Figure 7.2:** Signatures of the six touchpad subjects that performed worst on average over scenarios S1-S4 in skilled forgery tests using only XY-data.

Unfortunately, signature verification results are hard to compare, as different underlying datasets and experimental protocols are used. However, to provide some reference to our results, we compare them to the most similar related work we know of. Our results are significantly better than those in [22], where also only XY-data has been used with a traditional DTW-classification approach. On skilled forgeries, they reach 10.10% EER on a Samsung Galaxy Note Phone and 7.99% EER on an iPad when drawing with the finger. On random forgeries, their results are slightly better than ours with about 0.3% EER on their best devices, but still relatively close in absolute terms. Compared to [135], where also only XY-data was used from signatures drawn on a notebook display, both our approaches also perform better. They reach 10.2% to 5.6% EER for pen signatures and 15.7% to 10.2% EER for finger-drawn signatures using a traditional DTW (4 enrollment signatures) with complexity-specific preprocessing on skilled forgeries [135]. In [151], the authors report an EER of 9.1% on skilled forgeries for mobile phone signatures with the x/y-features alone. The only related work on touchpad signing reached an EER of about 12% on random forgeries, and about 15% on skilled forgeries made by imitating the paraph after watching the signer over the shoulder [6], using XY and pressure information with a Hausdorff-Distance based classifier.

Overall, our results show that the verification performance on touchpads is at least equivalent to the performance on mobile when the same features are used. This suggests that notebook touchpads are a suitable platform for handwritten signature verification despite the unusual way of drawing signatures, and should be considered in future work. This is somewhat surprising for multiple reasons. Related work [22, 23] has shown the

importance of good visual feedback for verification performance, an area where mobile phones have a clear advantage over notebooks. But rather than a general performance penalty for indirect visual feedback, we have only found some differences in which kind of signatures are easiest to forge and perform worst, and a generally higher difficulty to forge signatures without direct visual feedback as reported by the forgers. Finally, in our own usability study, touchpad signing tended to score lower than mobile, although some participants preferred it. This might counteract a possible effect of increased forging difficulty due to it possibly resulting in a drop in genuine signature quality.

An interesting path of future research is the effect of forging difficulty on the good touchpad results on skilled forgeries. Our random forgery results indicate that the two device categories are in the same range of performance with the same (random) forgery quality, and our skilled forgery results support that. However, the forgers in our study did state that generally, they found it more difficult to forge signatures on touchpads compared to mobile devices. Therefore it would be interesting to see if the skilled forgery results on both platforms are also similar on high effort skilled forgeries, where forgers have even more time to practice each signature to rule out any effect of possibly weaker forgeries.

## 7.4 Mobile Signature Verification

This section focuses on the use of movement sensor recordings as additional biometric features for verification. We first discuss the classification based on the signature image alone in different mobile-specific settings and compare it to the touchpad. Then, we proceed with the evaluation of movement data on its own and in combination with XY-features. We also investigate the impact of individual movement sensors on performance, and the use of the touchsize as an additional screen-based feature. Finally, we discuss potential interoperability effects between devices. Our corresponding research question is answered in chapter 8.3.

### 7.4.1 XY-Performance

On the touchpad, sitting is the only signing position we recorded because we consider it to be the most realistic usage scenario. Also, even if someone were to sign standing on a touchpad (eg. at a sales booth), contrary to mobile devices, the laptop still remains firmly placed on a table that is likely to be at a comfortable height for signing standing (eg. a standing desk), and it is therefore highly similar to signing sitting in how the signature is drawn. On mobile, on the other hand, there are far more possible positions in which a user can sign. Based on related work [29], we have chosen to consider sitting, standing and walking as the three most important ones. With multiple positions, more than the previously described four scenarios S1-S4 become possible, and there are possible differences in verification performance between individual positions. First, we focus on skilled forgery verification performance using only XY-data, as on the touchpad, and table 7.5 shows our results in different positions and scenarios.

For scenario S1-S4, we find that sitting touchpad and sitting mobile signature verification perform about equally in the same setting. However, when only relying on XY-data on skilled forgeries, sitting is the worst-performing position across all scenar-

| **Position** | Skilled | | | | Random | | | |
|---|---|---|---|---|---|---|---|---|
| | S1-S4 | S5-S8 | S9-S12 | Mean | S1-S4 | S5-S8 | S9-S12 | Mean |
| Touchpad | 5.70% | - | - | 5.70% | 0.78% | - | - | 0.78% |
| M. Sitting | 5.87% | 5.95% | 6.86% | 6.22% | 0.70% | 0.66% | 0.80% | 0.72% |
| M. Standing | 4.52% | 4.25% | 5.30% | 4.69% | 0.54% | 0.64% | 1.06% | 0.75% |
| M. Walking | 5.05% | 5.77% | 5.43% | 5.41% | 0.74% | 0.68% | 0.63% | 0.68% |

**Table 7.5:** The average results for different szenario groups in different positions using only XY-data.

ios on mobile. Standing is consistently the best, followed by walking, with a difference of approximately 0.75% EER between the stances on average. This is surprising, as we would have expected sitting as the most stable position to result in the best XY-channel data, and movement data to benefit from less stability. Also, in our usability study on mobile signing, 60% of users have preferred signing seated over signing standing on mobile phones, and 85% of participants would use the application sitting, compared to only 60% who would use it standing.

A possible explanation for this effect is that, while there is the least amount of movement while sitting, there is the most variety in how to handle the phone while signing, possibly causing differences in the signature drawings. For example, in our experimental setting, a user can hold their phone with their elbow, lower arm or wrist rested on the table, hold it freely, or rest their elbow on their knee or upper thigh while sitting forward-leaning on a chair. They can lean back in their chair or sit straight, just as they feel the most comfortable, and switch between all these options within the session and between sessions. When standing, there is only one way to hold the phone while signing with only a bit of freedom in the angle at which the phone is held, even though the position overall is less stable. For walking, it is similar, although the distraction and additional movement introduce some more inaccuracy compared to standing still.

What might also have an effect on these results is that most participants recorded their signatures in the order sitting - standing - walking, so that sitting might have suffered from an additional first-position-enrollment penalty for being the first position, similar to the first-signature enrollment penalty when using the first signatures of each position for enrollment. However, while we can not rule it out, we consider it to be unlikely to affect the overall picture. The first-signature-enrollment penalty occurs in all positions, not just in sitting signatures, suggesting that a user has to adapt again with every change of position, rather than only once in the beginning to the new signing surface. Also, users draw 15 signatures per position, giving them time to get used to the way of signing. There is no unusually high average difference between first and random enrollment scenarios for sitting signatures, which would be expected if there were a large additional penalty on the first signatures of the session. With 1.40% EER it is higher than for standing signatures (0.60% EER), but not than the walking signatures with 1.24% EER which are drawn last. The order of positions is also still the same when only considering random enrollment scenarios.

Taking into account the effects of position, it is important to revisit the comparison

between touchpad and mobile. While in the equal position, both show about the same verification performance. When comparing the verification performances on the same underlying data in their respective best position, however, mobile signature verification outperforms touchpad signing by about 1.2% EER. This might be caused by either the better usability of mobile signing reported in our usability studies that could result in better signatures, or the higher resolution at which mobile phone signatures are recorded. As both classes of devices result in accurate representations of what was drawn on the touch surface, we consider the latter cause unlikely.

But the different positions possible on mobile can not only be a benefit, as discussed previously, but also a challenge. So far, the only scenarios we considered are S1-S4, where one model is trained with signatures from one position to predict only signatures for that position. This is possible in practice, but either requires a restriction of the user to one position, a mechanism to detect which of the models to use for verification of a specific unknown signature, or a voting mechanism among models for all three positions. Therefore, it is desirable to have one model that can be used for all positions.

Scenarios S5-S8 test models that have been trained on signatures from all three positions, and report how well they perform on each of the stances individually. The only notable difference between S1-S4 and S5-S8 is in the prediction performance on walking signatures, with a difference of 0.72% EER. On average, both perform equally well, with an average of 5.15% EER compared to 5.32% EER respectively. This is great for practical application, as it eliminates the need to train multiple models for different positions at no performance penalty. Szenarios S13-S16, which also use multi-position enrollment, but test on a multi-position test set instead of individual positions as in S5-S8, result in 5.50% EER on average, which also supports these findings.

Even more desirable would it be to find one position that performs well in predicting all other positions, also eliminating the need to record enrollment signatures in multiple positions. Scenarios S9-S12 test this by training a model on signatures from only one position and testing it on a mix of all positions. Surprisingly, both standing and walking signatures perform well with 5.30% and 5.43% EER respectively. When comparing them to their correct benchmarks, which are the averages over all three positions of S1-S4 and S5-S8, there is no notable difference. As evidenced by scenarios S1-S4, standing signatures predict other standing signatures with 4.52% EER on average. Our other-position scenarios S17-S20 show that on average, standing signatures predict sitting signatures with 5.68% EER, and walking signatures with 5.88% EER. Averaging these three values results in 5.36% EER over all positions, underlining the results from scenario S9-S12. Walking signatures predict other walking signatures with 5.05% EER, standing signatures with 5.02% EER, and sitting signatures with 6.91% EER, on average 5.66% EER. Therefore, the better of the two positions is standing, both in terms of average EER and differences between individual-position results.

Neither the overall EER nor the differences between average results for different positions is higher for standing-only models than for the models in S5-S8, but at less effort for enrollment and with one less training signature. The average standard deviations are also similar, with 7.78% (mean over all positions) for S5-S8 multi-position models, and 7.14% (S9-S12) and 8.21% (S1-S4 standing, S17-S20 sitting and walking) for standing models that predict all positions. Based on these results, using a model trained with only standing signatures to predict all positions is a viable option when only XY-data

is used.

The results on random forgeries are similar to the ones on skilled forgeries presented above, although the same effects are not as strong. In S1-S4 and S5-S8, standing is the best-performing positions by very small margins. In S9-S12, standing performs the worst on predicting all positions, beaten by sitting and walking, the best position. However, the difference between the three positions is at most 0.43% EER, with very low absolute error rates. When comparing the results from combining S1-S4 and S17-S20 to the ones from S9-S12 (which is the same setting), the results are even closer, with standing performing the best at 0.72% EER, followed by sitting (0.79% EER) and walking (0.80% EER). This suggests that models from every individual position are about equally suitable to predict all positions, and that the small differences are likely due to the random selection of different test sets. On average, individual models for individual positions (S1-S4) and multi-position models for all positions (S5-S8) perform about equally well, and both perform slightly better than individual position models for all positions (S9-S12). Overall, the EERs on random forgeries are so low that any effects noticeable on skilled forgeries are generally present, but not visible as clearly.

### 7.4.2 Inclusion of Movement Data

Other than on notebooks, the XY-channel is not the only channel of information that is available on mobile. We record movement data from different embedded sensors of the phone in the background while the user is signing. The intention is to use them as additional biometric features to improve the verification performance. In our tests, we consider three possible feature sets for classification: XY-data only, movement-data only, and a combination of XY-data with one or multiple movement features. Table 7.6 shows the results for the three feature sets in different positions and scenarios. For the combined results we report here, all three movement features have been included together.

| **Skilled** | | S1-S4 | | | S5-S8 | | | S9-S12 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | XY | Cmb. | Mvmt. | XY | Cmb. | Mvmt. | XY | Cmb. | Mvmt. |
| Sit. | EER | 5.87% | 5.32% | 20.27% | 5.95% | 5.16% | 23.22% | 6.86% | 7.95% | 31.09% |
| | STD | 8.33% | 8.67% | 16.52% | 8.40% | 8.03% | 17.55% | 9.04% | 9.28% | 14.17% |
| Sta. | EER | 4.52% | 2.59% | 11.66% | 4.25% | 3.66% | 15.02% | 5.30% | 5.55% | 26.56% |
| | STD | 7.82% | 5.63% | 11.27% | 6.72% | 6.69% | 12.76% | 7.14% | 7.93% | 13.05% |
| Wal. | EER | 5.05% | 5.41% | 27.47% | 5.77% | 7.43% | 37.43% | 5.43% | 5.86% | 24.96% |
| | STD | 7.50% | 7.46% | 14.78% | 8.21% | 8.61% | 15.23% | 7.89% | 7.70% | 12.36% |
| Avg. | EER | 5.15% | 4.44% | 19.80% | 5.32% | 5.42% | 25.22% | 5.86% | 6.45% | 27.54% |
| | STD | 7.88% | 7.26% | 14.19% | 7.78% | 7.78% | 15.18% | 8.02% | 8.30% | 13.19% |

**Table 7.6:** Comparison of the average results on skilled forgeries using XY-only, XY combined with movement data, or movement data only for verification in different groups of scenarios.

What can be seen is that the combination of movement data and XY-Data does

improve detection of skilled forgeries in most important positions and scenarios compared to X/Y-Data on its own. On average over all positions, the combined signatures perform better than XY-only signatures by about 0.7% EER in scenarios S1-S4, where one model is trained per position, and about equal to XY-only signatures in scenarios S5-S8, where one model is trained with signatures from all positions. When one individual position is used to predict all positions in scenarios S9-S12, XY-only signatures outperform the combined signatues by about 0.6% EER on average. However, the averages conceal important details present in the data. In scenarios 1-8, the combined signatures outperform XY-only signatures both in sitting and standing positions. As with XY-data only, the best performance is achieved on standing signatures. There, the average performance increases by 1.93% EER in scenarios S1-S4, and by 0.59% in scenarios S5-S8. The best individual results are achieved in random multi-session enrollment using combined signatures. With only 2.00% EER and 2.14% EER on skilled forgeries, they outperform XY-only data by 1.64% EER and 1.79% EER respectively. This shows that when good movement data is available, it can have a significant positive impact on verification performance.

But even the best movement-only tests, in scenario S1-S4 standing, perform significantly worse than XY-only signatures. As evident in the results presented in table 7.6, movement data on its own is not enough to reliably detect skilled forgeries. Nonetheless, the combination of both clearly and repeatably performs better in many scenarios than either one individually. We also observe that there is a correlation between movement-only performance and combined performance. If the movement-only EER is lower (sitting/standing), there is a performance benefit, when it is higher (walking), there is a penalty. On random forgeries, we see that XY-only almost always performs better than the combined signatures by about 0.5% EER, and movement-only still mostly lies between 10% and 25% EER. The reason for this is that including movement data seems to be especially helpful for signatures that are hard to classify based on XY data alone. For signatures that can already be classified well without movement data, such as random forgeries, including it results in a performance penalty on some signatures due to its higher inaccuracy. This effect can clearly be seen when looking at the standing results of scenario 4, the best-performing setting for combined signatures. On skilled forgeries, 15 of 70 signatures show an improvement over XY-only when movement data is included. The average improvement of those signatures is by 10.3% EER, from an average 13.3% EER on XY-data only. Only four signatures performed worse with movement data. With an average of 6.25% EER, those already performed relatively well using only XY data, and worsened by 10.0% EER on average. What is notable is that the signatures that performed worse when movement data was included also performed worse than the overall result of 3.64% EER when only XY-Data was used, even though in absolute terms 6.25% EER is still a good result. Only one of the four signatures had a perfect score on XY-only data. Also, the vast majority of signatures that were not affected positively or negatively did have perfect scores. This indicates that the higher inaccuracy of movement data only affects signatures that only barely performed as well as they did without movement data. Most well-performing signatures however do so with enough stability using XY-data only that they are not negatively affected, while worse-performing signatures can benefit.

| **Random** | | S1-S4 | | | S5-S8 | | | S9-S12 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | XY | Cmb. | Mvmt. | XY | Cmb. | Mvmt. | XY | Cmb. | Mvmt. |
| Sit. | EER | 0.70% | 1.57% | 17.16% | 0.66% | 1.56% | 19.57% | 0.80% | 1.45% | 25.82% |
| | STD | 3.00% | 4.58% | 14.86% | 2.34% | 4.39% | 16.61% | 2.92% | 3.74% | 12.19% |
| Sta. | EER | 0.54% | 0.82% | 9.84% | 0.64% | 0.82% | 14.06% | 1.06% | 1.31% | 23.23% |
| | STD | 2.35% | 2.84% | 10.99% | 2.60% | 2.74% | 13.07% | 3.80% | 3.51% | 12.32% |
| Wal. | EER | 0.74% | 1.15% | 19.91% | 0.68% | 0.93% | 24.34% | 0.63% | 1.20% | 21.95% |
| | STD | 2.47% | 3.21% | 12.69% | 2.39% | 2.65% | 13.61% | 2.47% | 3.33% | 11.19% |
| Avg. | EER | 0.66% | 1.18% | 15.64% | 0.66% | 1.10% | 19.32% | 0.83% | 1.32% | 23.67% |
| | STD | 2.60% | 3.54% | 12.84% | 2.44% | 3.26% | 14.43% | 3.07% | 3.53% | 11.90% |

**Table 7.7:** Comparison of the average results on random forgeries using XY-only, XY combined with movement data, or movement data only for verification in different groups of scenarios.

The same scenario on random forgeries exhibits the same pattern, but with much better overall results of 0.71% on XY and 0.43% EER on combined data. Three signatures benefited from the addition of movement data and improved from 11.6% EER to a perfect score of 0% EER. Three signatures, all of which had a non-perfect score of 0.95% EER on XY-data only, worsened by 5% EER. On random forgeries, this is the exception, as can be seen in table 7.7. In most cases, the combined signatures perform worse than XY-only signatures. Because of the already very low EERs, there are very few signatures that can still benefit from movement data, while more can be negatively affected as described. The result on random forgeries is therefore usually a slight performance penalty when movement data is included. However, here in scenario 4, the movement-only performance is so good with 8.29% EER that the combination still outperforms XY-only. This shows that good movement data can result in improvements even on a very high level. What is also noteworthy about the movement data on random forgeries is that it is not sufficient to reliably detect random forgeries in most cases on its own, with EERs of above 10%. However, it does perform better than on skilled forgeries on average by about 4-5% EER.

| Enrollment | S13 | S14 | S15 | S16 |
|---|---|---|---|---|
| XY | 6.71% | 4.71% | 6.36% | 4.86% |
| Combined | 4.43% | 3.36% | 4.64% | 3.29% |
| Movement | 18.86% | 19.50% | 18.57% | 15.36% |

**Table 7.8:** The results for multi-position models (S13-S16) when walking is not in the training or test set.

What has to be considered is that in scenarios S5-S8, walking signatures are still included in the training set when testing on sitting or standing signatures. Only training a model on sitting and standing signatures and thus restricting the user to those two

positions would result in even better performance on combined signatures, but only in equal or worse performance on XY-only signatures, and therefore increase the combined signatures performance advantage. Our reasoning is that walking is very different from standing or sitting when considering movement data. When considering only XY-data, the difference is much smaller, and as we have discussed previously, walking is even better than sitting. This is also evidenced by our results. The advantage of combined signatures over XY-only signatures in S5-S8 when sitting and standing remains relatively constant at 0.8% EER and 0.6.% EER, respectively. On walking signatures, however, the XY-only signatures have an advantage of 1.66% EER, and there is only about a 0.2% EER difference to sitting signatures for XY-only, compared to a 2.3% EER difference for combined signatures. Also, when a model for combined signatures is trained only on walking to predict walking as in S1-S4, it performs only 0.3% EER worse than XY-only signatures, and about equal to combined sitting signatures. This shows that there is no problem with predicting walking signatures per se, although movement data does not seem to be beneficial, but that there is a large difference to the other two positions that affects combined prediction performance.

Because of the difference we found between walking and sitting/standing positions, we have tested how combined models perform when walking data is not included in training or testing using modified versions of our scenarios S13-S16. This is realistic, as the vast majority of users does not want to sign walking anyways. Also, it is still secure. If a forger were to forge while walking and there are no genuine walking samples in the training set, the difference only becomes larger and the signature is less likely to be accepted. Our results are presented in table 7.8, and the differences are well visible. When walking data is part of the training set, XY-only signatures reach 5.32% EER on average over all positions, and 5.1% EER over sitting and standing only. Combined signatures are predicted with 5.42% EER and 4.41% EER respectively, already showing a 1.0% EER performance improvement when walking signatures do not have to be predicted. Now, when walking signatures are also not part of the training set, this further improves by 0.5% EER to an average 3.93% EER for combined signatures, while XY-only performance drops to a mean EER of 5.66%, a difference of 1.73%. This clearly shows the negative impact of having walking signatures in the model for combined signatures. XY-signatures profit from them being a part of the model in our experiments, as walking is better than sitting when no movement data is included.

Based on our findings, we would therefore recommend different enrollment strategies for XY-only and combined signatures. With XY-only signatures, we have found that using a model built from all three positions performs well in predicting all three positions (5.32% EER). When restricting the usage to sitting and standing because walking is very rare anyways, a model built on all three positions still outperforms a model built only on those two positions, even though walking is not even present in the test set anymore (5.1% EER vs. 5.66% EER). As this is caused by the worse sitting signatures, they can also be removed from the training set. As we have shown in scenarios S9-S12, a model trained only on standing signatures can predict all three stances (5.3% EER) and only sitting and standing (5.1% EER) well, on the same level of performance as if one model were trained for each stance individually (5.15% EER). It has the additional advantage of being the easiest to enroll for the user. For combined signatures, it is better to either train one model for each stance individually (4.44% average EER for all three

stances, 3.96% EER for sitting/standing only), or to train one multi-position model both sitting and standing signatures to predict those two stances (3.93% EER). Those two positions also seem to be predictable well using standing signatures only for training (3.95% EER), but due to the possible differences in phone movement patterns, we would not advocate for doing so. If walking is absolutely required, an additional model can be trained on walking signatures for walking signatures only (5.41% EER) and used when increased movement is detected. Even with optimized enrollment strategies however, combined signatures perform significantly better than XY-only signatures in sitting and standing positions (3.93% EER compared to 5.1% EER), although at a somewhat higher complexity.

Our results only partially support the global scenario findings of [29] that it is not possible to use only one position to predict all positions well, but that it is possible to train one model on all positions to predict all positions well. What we can say from our results is that while the performance with only one training position on all positions is still acceptable on its own, the inclusion of movement data results in a penalty compared to only using XY-data overall. When looking at individual position pairs in the results of scenarios S17-S16, we can see that between sitting and standing signatures, other-stance prediction including movement data outperforms XY-only prediction by 0.4% to 0.6% EER, evidencing the similarity of the two positions. When using walking signatures to predict standing signatures, combined data is better than XY by even 1.2% EER. However, the other way round, XY-only beats combined signatures by 1.5% EER. The latter is likely because a walking model accepts more signatures as genuine that would otherwise have been false rejections, while a standing model rejects more genuine walking signatures, and not due to better forgery detection of the walking model. Walking/sitting and sitting/walking pairs are always worse when movement data is included, but again much more so when the model is built from sitting signatures to predict walking signatures due to the same reasons. We also find that it is not feasible to have one model to predict all three stances when movement data is included because of the large difference between walking and sitting/standing positions. However, it is possible to have only one model trained on sitting and standing signatures to predict both positions well.

In our tests using movement data, walking is the worst-performing position, and in none of the scenarios did including movement data result in any benefit over XY-only signatures in this position. This is surprising, as in related work [29, 129], walking was found to be the best-performing stance, albeit on smaller datasets than ours. It was reasoned that this is due to the increased amount of biometric movement data the embedded sensors can pick up while walking compared to more stable positions such as standing or sitting. Based on our results, we think the additional movement introduced by a users steps has the opposite effect. While standing, our system performs best both on XY-data alone, which we discussed in the previous section, but also on movement data only. The second-best position using movement data alone is sitting. Therefore, we think what is beneficial is that the phone is held without support while standing, similar to the results of related work. This allows for more phone-movement to occur while signing that can be used for identification compared to sitting, where the phone is usually more stable. When walking, the phone is also held without support, similar to standing. However, the steps introduce a significant amount of noise in form of movement

that is caused not by signing, but by walking. This movement can maybe be used to train models for additional gait-pattern identification, which would require the user to walk while signing, or it has to be filtered out. Without either, it hinders identification based on the movement patterns of signing even when a model is trained specifically for walking signatures (S1-S4). It also makes training one model for multiple positions more difficult due to its large difference to the other two positions.

Overall, the inclusion of movement data does result in improved verification performance against skilled forgeries in sitting and standing positions. On average, the performance benefit is between 0.6% and 2% EER, depending on the position and scenario. The best average results are at 2.59% EER against skilled forgeries over standing S1-S4. Against random forgeries, the combination of both channel results in a slight performance penalty of about 0.5% EER on average. Due to the already very high verification performance of XY alone, additional movement data can not provide its benefits, but can affect signatures of some subjects negatively. Compared to the touchpad with a verification performance of 5.70% EER, combined standing signatures (2.59% EER) perform significantly better with a difference of 3.11% EER, while with the same features and position, both are about equal. This shows that mobile phones can rely on other positions and additional features to improve verification, and are therefore able to outperform the touchpad overall.

### 7.4.3 Individual Movement Features

So far, we have only considered movement data as a whole. However, our movement data feature set consists of three channels of information, the accelerometer, the gyroscope, and the gravity sensor. Now, we are going to focus on their individual contribution to verification performance. In table 7.9, it is shown how each of them performs on their own in scenarios S1-S4 on the whole dataset.

| S1-S4 | Skilled | | | | Random | | | |
|---|---|---|---|---|---|---|---|---|
| All Sig. | Acc | Grav | Gyro | Ts | Acc | Grav | Gyro | Ts |
| Sitting | 28.04% | 27.95% | 26.34% | —% | 22.52% | 27.59% | 20.11% | —% |
| Standing | 21.04% | 24.61% | 20.36% | —% | 16.46% | 23.95% | 17.21% | —% |
| Walking | 27.13% | 32.50% | 36.43% | —% | 24.15% | 30.95% | 31.88% | —% |
| Average | 25.40% | 28.35% | 27.71% | —% | 21.04% | 27.50% | 23.07% | —% |

**Table 7.9:** The results when only features from one movement sensor are used for prediction at a time on all data. Because the touchsize measurements of some devices are not of sufficient quality, no touchsize measurements are reported.

The average results of all three sensors are within a range of about 3% EER. None of them performs remarkably well or remarkably poorly overall. The average results on random forgeries are similar. Accelerometer and gyroscope show a slight improvement of 4% EER, while the gravity sensor does not perform significantly better than on skilled forgeries. This is interesting because it shows us that it is not harder to detect skilled forgeries than random forgeries based on the way the phone is held in hand, and

that the feature is thus not affected by the conscious effort of the forger to imitate the signature based on the signature image. For our forgeries, where the attack scenario was an impostor that had signature images, but no possibility to observe the subject drawing the genuine signature, there was no way of knowing for the impostor how the subject held the phone. Further work is required to see if forgers who have this possibility pay attention to and are able to imitate how the phone is held. The performance difference for accelerometer and gyroscope between random and skilled forgeries can be explained with the attempt of the forgers to imitate the signature image well. Similar tapping and drawing patterns on the screen (when imitating a signature) likely cause similar acceleration and rotation patterns, so it is expected that the according movement sensor results reflect the better visual imitation to some degree. Unsurprisingly, standing is the best-performing position for all three movement sensors. What is interesting to note is that the gyroscope is more negatively affected by walking conditions than the other two sensors. The gravitation sensor also shows a performance penalty compared to sitting and standing, while the accelerometer performs even slightly better than sitting when walking. Based on these results, it should be considered to remove the gravity sensor and gyroscope from models trained for walking, while the accelerometer is suitable for all three positions.

| S1-S4 | Skilled | | | | Random | | | |
|---|---|---|---|---|---|---|---|---|
| SMG950F | Acc | Grav* | Gyro | Ts | Acc | Grav | Gyro | Ts |
| Sitting | 30.56% | 18.24% | 31.30% | 33.81% | 25.10% | 25.84% | 22.87% | 6.49% |
| Standing | 24.54% | 13.98% | 25.37% | 34.91% | 17.04% | 21.20% | 20.00% | 7.78% |
| Walking | 31.67% | 21.67% | 36.76% | 38.24% | 25.00% | 26.76% | 29.44% | 9.54% |
| Average | 28.92% | 17.96% | 31.14% | 35.65% | 22.38% | 24.60% | 24.10% | 7.93% |

**Table 7.10:** The results when only features from one movement sensor are used for prediction at a time on the SMG950F subset. Note that because only two of the four sets of skilled forgeries were drawn on an SMG950F device, the test set only contains 20 signatures (10 forgeries, 10 genuine), as opposed to 40 when the whole dataset is used.

To rule out any inter-device issues and have a reference to compare the touchsize results (discussed later) to, we have computed the same scenarios on the SMG950F subset of our database. In it, there are only signatures drawn on the equally-named device by 27 users, as well as two sets of five skilled forgeries per user that were also drawn on the SMG950F smartphone. Because of the smaller test set (due to the fewer skilled forgeries) and possible interoperability effects we will discuss later, some smaller deviations between the results can be expected. However, there should not be any large differences in patterns. This is true for both accelerometer and gyroscope results on skilled forgeries, and for all three sensors on random forgeries. The absolute EERs are a bit higher than on the whole dataset, but the patterns are the same. The gravity sensor on skilled forgeries however performs significantly better than both other sensors. Also, while it shows the same changes between positions as before, it is much better on the SMG950F subset, rather than slightly worse like the other two. This is likely not indicative of a real-world performance benefit or a large interoperability issue (such as

much better/worse data quality) though, which is why it is marked with an asterisk. In the both cases, we would expect to see the same effect at least on skilled and random forgeries of a same-device dataset. Also, if there were a significant data quality difference between devices, we would expect it to result in larger differences between genuine signatures of one device and skilled forgeries of another one. This would mean better separability of forgeries in a multi-device database compared to a one-device database, and thus in the end a better performance, rather than a worse one, on the whole dataset.

We suspect that this is due to the skilled forgeries being drawn by only two, rather than four, skilled forgers. Both of them likely held the phone in a way that is less frequent in the dataset, and their forgeries were thus easier to differentiate from more subjects and their genuine signatures. As the gravity sensor has a fixed range between 0 $m/s^2$ and 9.81 $m/s^2$ on each axis, and the magnitude of all three axes combined always has to be 9.81 $m/s^2$, its values also represent the absolute position (how the phone is held on each axis compared to the ground) within the reference. This is not the case for the other two sensors (both essentially show just a history of movement around 0 without minimum or maximum limits) and makes it easier for such an effect to occur. Even when genuine signer and forger perform the exact same relative changes, but hold the phone at a different absolute angle in one axis, that difference results in a constant distance between each point of the two time series for the gravity sensor (position sensor), while it does not on movement sensors such as accelerometer and gyroscope that reset to 0 when nothing changes.

This case is an interesting example of how only one dimension has to be different between genuine signature and forgery to detect a fake signature, and how having multiple dimensions could possibly increase the chance of detection. What also suggests that there is a benefit in combination is the comparison to the movement-only results in table 7.6 of szenario S1-S4. On average over all three positions, movement-only verification using all three channels at once results in an EER of 19.80% on skilled forgeries. The average results for the individual features in table 7.9 are significantly worse, on average by 7.35% EER at 27.15% EER. The combination also significantly outperforms individual features in all positions except walking, where accelerometer-only and the combined movement features perform about equally. The same effect of the combination performing better than the individual features can also be seen on the SMG950F subset.

Based on these movement-only results, we would expect that XY-data plus any one movement sensor would not outperform the combination of XY-data with all three movement sensors at once on average. To test this, we have computed szenarios S1-S4 with XY data plus only one movement feature at a time, both on the whole dataset and the SMG950F subset. The results are shown in table 7.11 and table 7.12.

On the whole dataset (table 7.11), all scenarios where some form of movement data was included clearly outperform XY-only signatures on average, proving its value for verification. However, the inclusion of all three features is barely better than the inclusion of the accelerometer only on average over all three positions, and both are equal on average when only sitting and standing signatures are considered, at 3.96% EER. In both sitting and walking positions, two of the three individual features outperform the combination of all three sensors in that position. The gyroscope is the best sensor when sitting, the accelerometer when standing, and the gravity sensor when walking.

| S1-S4 | XY | Comb. | XY/Acc | XY/Grav | XY/Gyro | XY/Ts |
|---|---|---|---|---|---|---|
| Sitting | 5.87% | 5.32% | 5.11% | 5.97% | 4.86% | —% |
| Standing | 4.52% | 2.59% | 2.82% | 3.79% | 3.63% | —% |
| Walking | 5.05% | 5.41% | 5.61% | 4.88% | 5.38% | —% |
| Average | 5.15% | 4.44% | 4.51% | 4.88% | 4.62% | —% |

**Table 7.11:** The results when the movement features are added to XY-features individually on the whole dataset on skilled forgeries.

If only the best sensor would be used for each position (which would require a way to determine a users position when signing in a real application), the average EER over all three positions could be dropped to 4.19% compared to the current 4.44% when all are combined. This also shows that each of the sensors is a valuable and worth it to be recorded. Not visible in table 7.11, but also interesting, is that with including the accelerometer only, the best skilled-forgery result on the whole dataset is reached in szenario 4, standing, at 1.79% EER, compared to 2.00% EER in the same setting when including the combination of movement features.

| S1-S4 | XY | Comb.* | XY/Acc | XY/Grav* | XY/Gyro | XY/Ts |
|---|---|---|---|---|---|---|
| Sitting | 6.30% | 4.16% | 5.09% | 4.26% | 4.81% | 6.11% |
| Standing | 6.39% | 2.96% | 4.63% | 3.15% | 4.35% | 5.28% |
| Walking | 6.30% | 6.48% | 7.41% | 4.91% | 7.97% | 6.12% |
| Average | 6.33% | 4.54% | 5.71% | 4.10% | 5.71% | 5.84% |

**Table 7.12:** The results when the movement features are added to XY-features individually on the SMG950F subset on skilled forgeries.

To make sure that potential device interoperability issues do not affect our results, we have also performed our tests on the SMG950F database subset, the results of which are shown in table 7.12. As on the whole dataset, the inclusion of individual movement features outperforms XY-data on its own. The combination of XY and gravity sensor performs particularly well, but as discussed previously, this is likely not a real performance benefit, which is why it is marked with an asterisk. Because the combined signatures include gravity sensor data, those results are also affected and marked accordingly. To be able to compare the combined performance with the performance of the individual features, we therefore need to correct the combined results. Our best reference are the results on the whole dataset. The unaffected average results (XY, XY/Acc, XY/Gyro) are about 1.2% EER worse on the SMG950F subset. When adding that to the combined result, there is again almost no difference between including one individual feature and including the combination of all three, as on the whole dataset. Without correction, a significant improvement for the combined signatures can be observed.

Based on our results, one possible explanation is the following. Without XY-data, all three individual movement sensors perform about equally, with a relatively weak

absolute level of performance, as can be seen in tables 7.9 and 7.10. None of the features on its own is capable of separating genuine signatures and forgeries of all subjects well, but each of the features performs well for some subjects, where the forgers way of handling the phone was different in that dimension. This is a pattern that is noticeable in the individual per-subject results, that even though the average performance level is lower, some subjects are below 10% EER or even show perfect scores. We assume that the different movement features complement each other to some degree, so that different features perform well on different subjects, and the combination thus results in a greater number of well-identifiable subjects. For those subjects where one of the three features works very well, that feature can compensate for the others and make forgeries identifiable on its own, while for subjects where none perform particularly well, not much changes.

When XY-data is included, however, the overall performance level is much higher in that dimension. The genuine signatures of the majority of subjects are already well discernible from their skilled forgeries based on the signature image, and the more (lower-performing) movement dimensions are included, the more weight is given to noise that introduces uncertainty. For some of the few signatures that are not identifiable well based on XY-data, adding one of the movement features can already be the factor that makes them identifiable, while having a lower impact than all three together on the majority of subjects. Adding more movement features possibly helps lift a diminishing number of additional low-performers, but adds an increased uncertainty to the majority of high-performers, and thus potentially limits the benefit of adding additional features. To be sure, further research is required into why the combination of movement features results in significant improvements without XY-data, but not with XY-data. Also, exploring different combinations of movement features in order to increase combined verification performance is an interesting path for future work. Also, improved processing of movement data could result in an overall better performance level in future systems.

| S1-S4 | XY | Comb. | XY/Acc | XY/Grav | XY/Gyro | XY/Ts |
|---|---|---|---|---|---|---|
| Sitting | 0.70% | 1.57% | 0.79% | 1.27% | 0.82% | —% |
| Standing | 0.54% | 0.82% | 0.54% | 0.93% | 0.38% | —% |
| Walking | 0.74% | 1.15% | 0.84% | 0.61% | 0.79% | —% |
| Average | 0.66% | 1.18% | 0.72% | 0.93% | 0.66% | —% |

**Table 7.13:** The results when the movement features are added to XY-features individually on the whole dataset on random forgeries.

On random forgeries, we have already described that XY-only performs stronger than the combination of XY with all three movement sensors. When only one movement sensor at a time is included in tests on the whole dataset (table 7.13), the results are significantly better than with the combination of all three, but never better than XY on its own on average. This aligns with how movement data affects signature verification performance. In individual positions (XY/Grav, walking; XY/Gyro, standing), a minimal outperformance is observable, although the EERs of the movement sensors

in those positions on their own are not unusually low. On the SMG950F subset (table 7.14), the results are less conclusive, as the inclusion of only the accelerometer performs slightly worse than the inclusion of all three sensors, while the individual addition of gravity sensor or gyroscope leads to a slight outperformance of the XY-only signatures on average. However, the effects in both directions are still relatively small, and the reduced test set size and number of subjects might also lead to a slightly higher amount of variation in results.

| S1-S4 | XY | Comb. | XY/Acc | XY/Grav | XY/Gyro | XY/Ts |
|---|---|---|---|---|---|---|
| Sitting | 0.65% | 0.56% | 1.11% | 0.65% | 0.37% | 0.19% |
| Standing | 0.74% | 0.65% | 0.93% | 0.56% | 0.83% | 0.46% |
| Walking | 0.65% | 1.20% | 1.02% | 0.09% | 0.65% | 0.74% |
| Average | 0.68% | 0.80% | 1.02% | 0.43% | 0.62% | 0.46% |

**Table 7.14:** The results when the movement features are added to XY-features individually on the SMG950F subset on random forgeries.

In summary, we can say that when used on their own, none of the movement features significantly outperforms the others on skilled or random forgeries. When they are combined in movement-only tests, that combination leads to significant improvements in performance. Also, the use of movement data in verification together with the signature image considerably improves the verification results on skilled forgeries. However, the benefit of combining multiple movement features does only very slightly, if at all, carry over to tests where XY-data is available. Including the accelerometer on its own performs about equally well as including all three together, and gyroscope and gravity sensor are not far behind. All three perform better on average than XY-data on its own. The best on all data on average is the use of different individual sensors for each position. With the gyroscope when sitting, the accelerometer when standing, and the gravity sensor when walking, an average of 4.19% EER over all three positions could be achieved, 0.25% better than when all three sensors are included at once, and 0.32% better than the best (same) individual sensor in all positions. On random forgeries, the penalty of including movement data for verification is lower if only one instead of three sensors are included, but XY-only performs still about equal or better in most cases and is recommended.

### 7.4.4 Touchsize

Using the touchsize as an additional display-based feature for verification has been suggested in related work [36, 64, 122]. It is considered as a possible substitute for missing pressure data, but has not been tested yet. In this section, we evaluate its contribution to the verification performance on skilled and random forgeries. All our results have been obtained on the SMG950F subset of our mobile database, as not all the devices in the database report accurate touchsize information.

The results of touchsize as an individual feature can be seen in table 7.10 on random and skilled forgeries, alongside the individual movement features for comparison. The

most important observation about its performance is the difference between skilled and random forgeries. While the movement features perform significantly better on random than on skilled forgeries, they are still within the same, relatively high, area of absolute EERs. The touchsize however is different. On skilled forgeries, it is clearly the worst-performing individual features at about 35% EER. On random forgeries, it performs the best on average by a margin of almost 15%, and more than 27% points EER better than on skilled forgeries. With an average EER of 7.93%, it is even suitable to detect random forgeries by itself.

A possible reason for this large difference between random and skilled forgeries is the touchsizes relation to other screen-based features, such as XY-data. Our results suggest that if two signature images look alike, the touchsize time series is also relatively similar automatically. On skilled forgeries, this is the case, and the touchsize is not detailed enough to be able to separate genuine signature from forgery on its own. Because all signatures therefore look the same in terms of touchsize, it performs poorly. On random forgeries, however, similarity is already enough to identify genuine signatures, because the random forgeries are completely different signatures with equally different touchsize series. Compared to the movement sensors, the touchsize is thus easier to forge, because it just happens by drawing. The movement sensors are less affected by a concious effort to imitate something, which explains their better performance on skilled forgeries. But the touchsize is (likely for the same reason) also much more consistent among signatures, while movement sensors have much more variation even for the same subject. This is shown by the much larger difference between random and skilled forgeries for the touchsize compared to movement sensors, where there seems to be a significant amount of confusion with different-looking signatures.

Another effect that makes the touchsize different from the movement sensors is that it performs best when sitting, and worse the less stable the position is. This could be because in a more stable position where the phone can not move much, the pressure of the finger is better applied to the screen. In less stable positions where there is no possibility of supporting the arm, more of the pressure can result in a movement of the phone, instead of in an increase in contact size when the soft fingertip is pressed against the fixed screen surface.

The results when the touchsize is used together with XY data of the signature are presented in table 7.12 for skilled and in table 7.14 for random forgeries. Again, for comparison, the results for the individual movement sensors are also displayed. On skilled forgeries, the performance where the touchsize was included as an additional feature is better on average than the XY-only signatures, with 5.84% EER compared to 6.33% EER overall. Most of the difference is caused by the strong performance when standing, but a small benefit is observable in all three positions. The accelerometer and gyroscope experiments perform a little bit better than the touchsize, as was to be expected from the movement-only tests on skilled forgeries. On random forgeries, the combination of XY and touchsize slightly outperforms XY-only signatures by 0.22% EER, with the same pattern as touchsize-only of sitting being the best and walking the worst position.

Overall, we consider touchsize to be a valuable feature for verification that should be recorded when available. We have found performance improvements on both skilled and random forgeries, and especially its discriminative power against random forgeries

looks promising. Based on our tests and its likely close relation to other screen-based features, we recommend including it as an additional feature. Further work is required to test it on a larger set of skilled forgeries than is available in the SMG950F subset of our database, and investigate possible improvements by including combinations of one or multiple movement features and the touchsize in XY-based verification.

### 7.4.5 Interoperability

We have already mentioned potential interoperability effects between signatures from different smart phones a few times. In our whole processing chain, we pay attention to remove any differences we can and normalize all signatures to a uniform standard. However, different screen aspect ratios, resolutions, levels of sensor accuracy on different smartphones, and many other factors could potentially still have an effect on the results we obtain on our whole dataset. To be sure, we use the SMG950F subset of our data for comparison, where all subjects have signed on the exact same device. We have chosen scenarios S1-S4 because they are both realistic and the least complicated, with no interposition effects we have to account for. Both random and skilled forgeries are good scenarios to test this. Due to the randomness of random forgeries, the skill of forgers has no influence. However, because the random forgeries and genuine signatures are already relatively different, interoperability effects might not be visible in the results if they are only relatively small. On skilled forgeries, small differences would be clearly visible, because skilled forgeries and genuine signatures are very similar per definition, but the quality of forgeries by different attackers can also have an influence. Therefore, if there is only a small difference, we can not say with certainty that interoperability issues caused it, because there are two more sets of skilled forgeries available for the whole dataset than for the SMG950F subset.

If there are differences between devices, we expect the SMG950F subset to perform worse. Such differences would not affect the training of a model in either case, because all signatures of one subject are always from the same device in our database. It would only affect the negative (forgery) class of the test set, which includes skilled or random forgeries from multiple phones, and the larger the difference between devices, the easier it is to identify signatures that have not been drawn on the same device as the genuine signatures. This would lead to an artificial improvement in performance on the whole dataset, but to no change on the same-device subset. The larger the difference, the less interoperability is there between different phones. In practice, this would mean that for a successful forgery attempt, an attacker would need to obtain the same phone as the genuine user is signing on. But in reverse, it also means that a new signature model would have to be trained for each device the user is signing on, and that it is not sufficient to have one model per user. This can be no problem at all, for example if signing is used as a means of biometric unlocking of a device, where the signature is always drawn on that device. For other, less controlled use cases, it can be problematic.

Starting with random forgeries, there is no significant difference to be seen (tables 7.13 and 7.14). There is no difference between the whole dataset and the SMG950F subset when only XY-information is considered, and on different forms of combined signatures, there are only performance differences of at most 0.5% EER on the average results that are not all in favor of the same subset. The largest difference where

the whole database outperforms the SMG950F subset, as we would expect in case of interoperability effects, is only 0.3% EER. On individual movement features of random forgeries (tables 7.9 and 7.10), the picture is similar. Accelerometer and gyroscope are worse on the SMG950F subset by about 1% EER. The gravity sensor performs about 3% better on the subset than on the whole dataset. However, the gravity sensor is a software sensor based on accelerometer and gyroscope, so if interoperability issues were the cause of the performance difference on either of the three sensors, we would expect it to perform worse, not slightly better, on the SMG950F subset.

On skilled forgeries, there is a difference of about 1.2% EER by which the SMG950F subset performs worse than the whole dataset (tables 7.11 and 7.12), both on XY-only and on XY/Acc and XY/Gyro combinations. The amount varies between positions, but EERs are consistently better on the whole dataset. We do not consider the combination with all three movement features and the gravity sensor due to its likely unrealistic outperformance on the SMG950F subset we have discussed previously. The accelerometer and gyroscope on their own both perform about 4% EER worse on the SMG950F subset than on the whole dataset (tables 7.9 and 7.10). This is not only the case on average over all positions, but also on average for each position individually. Also, the movement sensors are likely not affected as much by the skill of a forger than the XY-dimension. Therefore, we think this is likely caused by interoperability.

To conclude, we are confident in saying that there is likely a small interoperability effect between different smartphones. It is not large, because it does not show up in random forgery results, but our skilled forgery results suggest that it is there. Both the XY-only signatures and the physical movement sensors accelerometer and gyroscope perform consistently slightly worse on the SMG950F subset than on the whole dataset. However, the effect is small in relation to the absolute levels of performance and should not be a problem for signature verification across devices. It also does not change any of our other findings. Nonetheless, it is an interesting path for future work to investigate ways of eliminating this effect entirely by improved universal preprocessing of XY and movement data.

## 7.5 Multi-Dimensional Dynamic Time Warping

All the local results we have discussed so far have been calculated using Multi-Dimensional Dynamic Time Warping. We introduced it in chapter 5 as our own classification approach based on the commonly used dynamic time warping algorithm. To end the part on local classification, we now evaluate it as a means of signature verification. The datasets and experimental procedures used in related work are too different from each other and our own to be able to meaningfully compare MD-DTW to other methods of classification. Direct assessment against other systems will only be possible once they have been tested on our dataset as well. However, the results of MD-DTW in isolation look promising.

MD-DTW is a one-class classification approach, which is ideal for signature verification because no set of negative samples has to be created and maintained. In all our scenarios we have used only 5 or 6 genuine signatures for enrollment, and the algorithm still performed well. Our usability study has shown that only 3-5% of users are not willing to provide at least five signatures for one-time enrollment, which shows that it is

realistic in practice. It also requires almost no time for training, and far less than most of our machine learning models we tested in global scenarios, making it ideal for the mobile scenario we tested it in where battery and processing capability is limited.

We have discussed different performance aspects at length in the previous sections. On random forgeries, we reach 0% EER in multiple scenarios and positions using MD-DTW for classification, and an overall level of EERs that is significantly below 1% on XY-only data. On skilled forgeries, our best result is 1.79% EER in the best-performing configuration with XY and the accelerometer, and multiple results around 2% EER when movement data is included. The overall level of performance of around 5% EER shows that it is capable of separating similar-looking samples of genuine signatures and skilled forgeries well with different sets and types of features. Therefore, we can recommend MD-DTW as a method of classification in future work. However, there are interesting topics for future work. By choosing a different distance measure, it is possible to penalize distance between two time series more or less harshly. Research on the impact of such changes would be highly interesting. Also, the combination of MD-DTW with pre-selection methods based on additional global data would be interesting. For example, requiring a similar number of strokes or signing time as a pre-requisite for a signature to even be considered genuine could filter out forgeries that are hard for any DTW-based algorithm to identify, because just by their signature image they look similar. A more advanced version of this, and an interesting line of research, would be the combination of a local MD-DTW classifier with global (pre-) classifiers such as a general forgery detector like the one described in the following sections.

## 7.6 Global Models

In addition to local one-class classification with our MD-DTW algorithm, we have also investigated some global classification scenarios. For those, we use traditional machine learning classifiers available in the open Python machine learning library `sklearn`. Many of them have also been evaluated in related work.

We consider three types of settings for our global tests. N-Class classification is a traditional signature recognition task. Using samples from all genuine signers to train at once, the goal is to assign the remaining genuine test signatures correctly to the respective subjects. These tests, while not necessary for verification, show us how well genuine signatures of the same subject can be identified, and if they can be distinguished from one another. The subject-others classification is a two-class verification scenario. The enrollment signatures of a user are combined with a set of 1500 random forgeries (randomly selected from the dataset). They serve as the negative class to train with. No skilled forgeries are included in the negative set, because in a real-world enrollment, no such forgeries would be available either. The different models are tested on random and skilled forgeries. Finally, our forgery detection tasks are a enrollment-free classification approach using traditional machine learning models. The classifiers are trained on a set of genuine signatures and skilled forgeries from multiple users. Our goal is to learn the general (global) characteristics of a forged signature and to be able to identify forged signatures without having to train a model for a specific subject at all. When someone signs, one model that can be deployed to everyone should be able to tell whether their signature is genuine or forged. To make sure that the model generalizes well, we obtain

our results in a leave-subject-out cross-validation. In the following sections, we will discuss our results in these three scenarios and their implications in detail. During development, we have also tested a fourth setting with one-class classifiers based on traditional machine learning algorithms. Both one-class SVMs and isolation forests did not perform well in our tests, which is why they are not included here. However, the scripts are still included in our project files that are available online for further reference.

### 7.6.1 N-Class Classification

All N-Class classification results we report are balanced accuracy (BA) scores obtained in five-fold stratified crossvalidation. For both mobile and touchpad tests, the whole datasets of 6300 and 1800 genuine signatures have been used respectively. The best parameters for each model are obtained using the grid search crossvalidation implementation of `sklearn`, and the best result for each model is reported. Different sets of global features have been used. Similar to our MD-DTW experiments, we call them XY-only, movement-only and combined data. The touchpad results are marked as touchpad and only rely on XY data. Our results are presented in table 7.15.

| N-Class | XY | | Combined | | Movement | | Touchpad | |
|---|---|---|---|---|---|---|---|---|
| | BA | STD | BA | STD | BA | STD | BA | STD |
| KNN | 56.22% | 5.80% | 57.20% | 5.70% | 26.90% | 4.90% | 56.80% | 5.70% |
| LSVM | 92.60% | 4.30% | 91.60% | 5.90% | 80.40% | 10.90% | 88.20% | 8.00% |
| RSVM | 49.80% | 7.70% | 46.20% | 8.50% | 23.50% | 4.50% | 41.00% | 10.50% |
| DT | 88.00% | 3.30% | 88.50% | 6.60% | 79.20% | 5.00% | 89.00% | 7.30% |
| RF | 98.70% | 1.00% | 99.60% | 0.60% | 94.70% | 4.60% | 98.60% | 1.50% |
| MLP | 90.40% | 3.70% | 93.80% | 3.00% | 80.30% | 8.70% | 87.00% | 4.10% |
| AdaB | 49.80% | 6.30% | 56.40% | 12.00% | 28.30% | 5.90% | 78.00% | 5.80% |
| NB | 97.10% | 2.00% | 96.20% | 3.20% | 73.90% | 9.60% | 97.40% | 2.20% |

**Table 7.15:** The balanced accuracy results of a classical signature recognition task, where each machine learning model performs n-class classification, with n being the number of genuine subjects.

There are clear performance differences between different models, but five consistently perform at close to 90% BA or above in all but the movement-only tests. The results are not directly comparable to our MD-DTW results, because it is a multi-class classification, not a subject-others classification on random forgeries, but the amounts of error are similarly low. Random forests perform the best, with about 99.6% balanced accuracy on combined data. This clearly shows that global the global information we extract is valuable for classification.

It is interesting how the different sets of global information compare in terms of how well signatures of different subjects can be recognized. The combination of global XY-data with global movement data does result in a performance improvement for five of the eight machine learning algorithms we use. On random forests, the best model, this improvement is 0.9% BA. Of the other well-performing models, MLPs and decision trees also improve, while linear SVMs and naive bayes show slightly worse performance

compared to XY-only. What is remarkable is how well global movement-only data can be classified on its own. With random forests, we reach 94.7% balanced accuracy at a standard deviation of only 4.6%. This is better than most models on XY-only data and highlights how well signatures can be separated with only their global movement information in a suitable setting.

In comparison between touchpad and mobile signatures, there is no significant difference in performance to be found on the better-performing models. Random forests reach 98.6% balanced accuracy, which is only 0.1% lower than on mobile XY-only data, where they also perform best. Naive bayes and decision tree classifiers show slightly better results on the touchpad, while MLP and linear SVM perform worse by about 3-4% BA. Overall, the differences are dependent on the type of model used, and the best ones perform about equally well. Therefore, we argue that global touchpad and global phone signature data is of equal quality and equally suitable for recognizing individual subjects in machine learning classification. In the same setting, where no element of skill and forging difficulty is involved, the classification works well on both, and mobile signatures only gain a slight advantage by including additional movement features.

### 7.6.2 Subject-Others Classification

While the multi-class test performed in section 7.6.1 tests the separability between subjects in different settings and the quality of the global information, two-class classification with random and skilled forgeries is closer to signature verification applications. With our subject-others classification tasks, we test how well the same machine learning models are able to distinguish random and skilled forgeries from genuine signatures, when only given random forgeries and genuine signatures to train. This is a possible setup for real-world enrollment, and our results are presented in table 7.16.

| Subject | XY | | Combined | | Movement | | Touchpad | |
|---|---|---|---|---|---|---|---|---|
| Others | RF | SK | RF | SK | RF | SK | RF | SK |
| KNN | 76.54% | 96.86% | 77.05% | 96.64% | 61.86% | 98.43% | 75.69% | 98.33% |
| LSVM | 85.58% | 93.00% | 92.51% | 96.00% | 69.43% | 98.36% | 84.21% | 96.89% |
| RSVM | 59.16% | 99.86% | 57.45% | 99.86% | 53.17% | 99.86% | 52.02% | 100.0% |
| DT | 91.34% | 87.43% | 90.22% | 93.07% | 82.65% | 97.71% | 91.13% | 87.11% |
| RF | 91.21% | 92.71% | 89.80% | 95.86% | 79.68% | 98.43% | 90.94% | 91.56% |
| MLP | 89.43% | 94.50% | 92.70% | 95.93% | 73.32% | 98.64% | 86.87% | 95.89% |
| AdaB | 97.02% | 90.71% | 97.92% | 93.79% | 91.95% | 96.79% | 96.83% | 90.56% |
| NB | 98.13% | 85.57% | 97.27% | 89.21% | 90.56% | 90.86% | 98.19% | 88.00% |

**Table 7.16:** The balanced accuracy results of subject-others signature verification. The random forgery (RF) results are the results of the best parameters from five-fold grid-search CV and include genuine signatures and random forgeries, the skilled forgery (SK) results are from tests of the model with those parameters on skilled forgeries only.

The overall performance of the different classifiers looks promising. Conceptually, our subject-others scenario is similar to scenarios S5-S8 for MD-DTW, where one model is trained on all stances to predict all stances. This is not only less computationally

intensive than training multiple individual models for different positions on the mobile device or a server, but also results in having the most signatures available to train with. Because training machine learning models generally requires more enrollment samples than our MD-DTW approach and usually also requires more processing power, we consider this to be the most realistic setting. On average over all subjects CV results on random forgeries, AdaBoost and Naive Bayes perform the best. On mobile XY-only data, they reach 97.02% and 98.13% balanced accuracy respectively. All other models also perform at around 90% balanced accuracy on random forgeries, except for RBF SVMs and the KNN classifier, which are significantly worse. On the touchpad, the performance is about equal to the performance on mobile signatures. Most well-performing models are well within 0.3% BA of one another. Only the linear SVMs and the MLP score significantly lower on the touchpad than on mobile, by about 1.3% and 2.5% BA respectively. Those are also the two well-performing models with the largest performance drop on touchpad-signatures in N-class classification. We can not say for certain what the cause of this difference is, but one possible reason could be that they are negatively affected by the lower amount of signatures that are available to train with on the touchpad.

Linear SVMs and MLPs also show the most improvement when movement-data is included in subject-others classification on random forgeries, with a respective 7% and 3.3% gain in balanced accuracy. Of the other models, both random forests and decision trees respond with slightly worse performance (1.4% and 1.1% BA, respectively). The best-performing models, AdaBoost and Naive Bayes, also show inconclusive results, with a 0.9% increase for the former, and a 0.9% decrease for the latter. What is remarkable is how well the global classification works on only movement data with both of them, with 91.95% and 90.56% BA. While we can not say conclusively over all models if the addition of movement data brings a noticeable benefit, we see that it can result in significant benefits for individual models and even allows for promising classification results on its own. Therefore, we think it should be considered and tested more extensively in future work.

The results on skilled forgeries are presented in the neighboring columns to the random forgery results on each feature set. They are obtained by using the best-performing parameter set for each subject of each classifier to train a new model on all 90 genuine signatures and all random forgeries for that subject. That model is then used to predict a test set of all 20 skilled forgeries for the subject to find the TRR (and thus, the FAR) under ideal training conditions. There are no genuine signatures in the test set, because all of them are already used in training. We chose this approach because this is what attackers would achieve on such a system. Models that already performed well on sets of genuine signatures and random forgeries, such as AdaBoost or MLPs, are also able to reliably detect significantly more than 90% of skilled forgeries. The recognition performance on XY-only mobile and touchpad signatures is about equal on both, and the results using combined XY and movement data are even significantly better across the board.

There might be global features that indicate whether a signature is genuine or forged, and by that could allow the models to detect them easily. This is not necessarily good or bad. Depending on the feature(s) that identify a forgery and how easy they can be controlled by the forger, it could either make a strong case for global signature

| Rank | Feature | Perc. | Rank | Feature | Perc. | Rank | Feature | Perc. |
|---|---|---|---|---|---|---|---|---|
| 1 | maccZ | 3.43% | 6 | minX | 1.22% | 11 | mgravMag | 0.98% |
| 2 | mgravY | 3.14% | 7 | my | 1.17% | 12 | startX | 0.91% |
| 3 | mdY | 2.96% | 8 | nrStrokes | 1.11% | 13 | ktdX | 0.90% |
| 4 | mdX | 1.96% | 9 | mgravZ | 1.11% | 14 | mgravX | 0.57% |
| 5 | sddY | 1.24% | 10 | sddX | 1.06% | 15 | ktx | 0.55% |

**Table 7.17:** The average permutation feature importance over all subjects of the random forest classifier when trained on genuine signatures and random forgeries and tested on a set of genuine signatures and skilled forgeries in a subject-others scenario on combined mobile data. The average balanced accuracy of the random forest over all subjects is 87.86% at a standard deviation of 8.37%. Permutation feature importance was chosen over impurity-based metrics due to its better suitability for high-cardinality features. It tests how much the performance is affected if one particular feature is replaced by random values over multiple repetitions.

verification or indicate that our skilled forgeries are too easy to identify. Therefore, we have determined the individual feature importance of a random forest model when classifying skilled forgeries using combined data, and our results are shown in table 7.17.

As can be seen, there is no single dominant feature in our decision trees that would explain how well skilled forgeries are recognized. Most important are the means of two movement features, followed by the means of the first derivates (speed) of the signature coordinates on X- and Y-axis. It is important to note that no skilled forgeries are involved in the training of those trees, because they also would not be available in a real application, but they are evaluated on a set of 20 genuine signatures and 20 skilled forgeries. The training set contains the remaining set of 70 genuine signatures and all available random forgeries, nearly identical to the setting in 7.19. Initially, we suspected that skilled forgeries might be identifiable by the possibly longer time forgers might have taken to draw them, which would have been a signal of low forgery quality. However, duration is only at position 18 with 0.54% change in performance, and is thus clearly not the determining factor. Generally, there are many dynamic speed-related and movement-related features on the higher ranks, which is consistent with our observation that forgers focused more on and were better able to imitate the signature image, rather than the way of signing. It also shows that global movement features can be a useful addition. However, none of the features stand out individually, so the reason is likely not an issue with forgery quality.

An open question is the number of signatures that are wrongly classified as forgeries, although they are genuine (FRR). For the results in table 7.16, we could not obtain this measure, because all genuine signatures were already used in training to test how well forgeries could be detected with the maximum amount of information available. However, the near-perfect or even perfect performance of models such as KNN or RBF SVM that performed very poorly on genuine signatures and random forgeries suggests that the FRR is high, and that those models just classify all signatures as forgeries. Therefore, given the promising results on some other classifiers that also performed well on random forgeries, we have tested additional skilled forgery scenarios with less genuine

signatures to train, so we would be able to determine both FRR and FAR. Our results are shown in table 7.18.

| Subject | XY | | Combined | | Movement | | Touchpad | |
|---|---|---|---|---|---|---|---|---|
| Others | FAR | FRR | FAR | FRR | FAR | FRR | FAR | FRR |
| KNN | 4.36% | 49.71% | 4.79% | 48.93% | 1.71% | 79.43% | 2.44% | 52.11% |
| LSVM | 13.29% | 32.36% | 7.00% | 16.36% | 4.21% | 57.79% | 8.00% | 36.33% |
| RSVM | 1.00% | 69.86% | 1.00% | 68.00% | 0.36% | 95.07% | 0.56% | 72.78% |
| DT | 18.36% | 31.50% | 12.14% | 31.93% | 6.79% | 45.79% | 21.89% | 28.56% |
| RF | 11.14% | 30.07% | 8.29% | 36.71% | 1.29% | 53.79% | 11.67% | 29.78% |
| MLP | 6.71% | 61.07% | 4.86% | 47.14% | 3.71% | 67.50% | 6.11% | 37.89% |
| AdaB | 11.07% | 12.14% | 9.07% | 6.79% | 3.71% | 22.79% | 14.78% | 8.78% |
| NB | 12.64% | 3.07% | 8.79% | 7.07% | 7.57% | 13.21% | 12.67% | 3.78% |

**Table 7.18:** The FAR and FRR of different models trained on 20 genuine signatures and 1500 random forgeries, and tested on a set of 20 skilled forgeries and 20 unseen genuine signatures.

The models that performed the best against skilled forgeries on XY-data (KNN, RBF SVM, and MLP) also have the highest false rejection rates, both in the results using 20 genuine signatures (table 7.18) and those using 70 genuine signatures (7.19) to train. This was expected from the results in table 7.16. However, even with only 20 signatures to train, there are models that perform well overall. On XY-data, AdaBoost reaches 88.4% balanced accuracy, and the Naive Bayes classifier reaches 92.15% balanced accuracy. It also only has a false rejection rate of 3.07%, which is important for practical use where it would be frustrating for a genuine user to have their signatures constantly rejected with higher FRRs. On combined signatures, AdaBoost performs significantly better than on XY-only with 92.07% balanced accuracy and an improvement in both FAR and FRR, and Naive Bayes holds its level of XY-only at also 92.07% balanced accuracy, but with more balanced FAR and FRR. Due to their significantly worse results, we do not consider the other models we trained on only 20 signatures, but our results suggest that global movement data is a valuable addition to global XY-features for verification. Movement data on its own consistently leads to lower FARs, but at considerably higher FRR compared to XY-only results. Touchpad and mobile XY-only signatures also perform about equal on global features in terms of balanced accuracy. There is barely any difference between the results of the Naive Bayes classifier, while AdaBoost shows an increased FAR and decreased FRR on the touchpad compared to the mobile phones. All other classifiers, with the exception of MLPs, also deliver similar results for both at a generally lower performance. Therefore, we can say that both platforms seem to be equally suitable for signature verification on global features.

In comparison to our MD-DTW classifier, all global models perform significantly worse, although they have more enrollment signatures available to them. AdaBoost and Naive Bayes show the best results on combined global data at 92.07% BA, while MD-DTW in scenarios S5-S8 reaches 5.42% EER or 94.58% BA on average, a difference of 2.51% BA. This would be a problem for a real-world application of such classifiers, as

our usability study shows that only about half of the participants would be willing to provide more than 20 enrollment signatures. However, as most of our machine learning models require many samples to train with by design, we increase the number of genuine enrollment signatures to 70, and our results are presented in table 7.19.

| Subject | XY | | Combined | | Movement | | Touchpad | |
|---|---|---|---|---|---|---|---|---|
| Others | FAR | FRR | FAR | FRR | FAR | FRR | FAR | FRR |
| KNN | 7.36% | 26.50% | 6.86% | 25.14% | 4.21% | 56.71% | – | – |
| LSVM | 17.35% | 20.07% | 12.00% | 7.57% | 4.21% | 40.43% | – | – |
| RSVM | 3.00% | 41.36% | 3.15% | 38.43% | 1.57% | 85.93% | – | – |
| DT | 22.64% | 12.14% | 12.07% | 14.00% | 8.29% | 19.21% | – | – |
| RF | 16.07% | 9.50% | 10.21% | 9.14% | 4.07% | 19.14% | – | – |
| MLP | 14.79% | 24.07% | 8.79% | 19.79% | 6.00% | 37.93% | – | – |
| AdaB | 19.64% | 3.07% | 10.36% | 2.00% | 5.14% | 7.21% | – | – |
| NB | 16.93% | 1.36% | 12.00% | 2.00% | 11.14% | 8.14% | – | – |

**Table 7.19:** The FAR and FRR of different models trained on 70 genuine signatures and 1500 random forgeries, and tested on a set of 20 skilled forgeries and 20 unseen genuine signatures. Because there are only 40 genuine signatures available per subject on the touchpad, we could not test this scenario and have left the column blank.

What we observe is a significant improvement in false rejection rates compared to our results with only 20 genuine signatures. This can be explained by the higher number of genuine samples that are available to train with, on which models can better learn to recognize them. However, it also results in a worsening of false acceptance rates, suggesting that the better recognition of genuine signatures comes at the cost of increased erroneous recognition of skilled forgeries as genuine signatures, as there are no skilled forgery samples available in the training set for the classifiers to learn the difference. Random Forests on XY-only, and Random Forests and Linear SVMs on combined data, improve enough to also become viable options besides AdaBoost and Naive Bayes. The latter two only show a small improvement and even a slight performance drop respectively on XY-only data, but an improvement of 2% and 1% on combined data in terms of balanced accuracy. These results also show more clearly than before that the combination of global XY and movement data results in better classification performance.

On random forgeries, where the problem of not having skilled forgeries to train with is nonexistent, we can also see the same pattern. There is a benefit of between 0.3% and 1.3% balanced accuracy for including movement data as additional features to the signature image on mobile. Also, touchpad and mobile are at a similar level of performance, but the touchpad is better by about 1% BA at about 96% BA for AdaBoost and Naive Bayes overall. With 70 genuine signatures to train with (on mobile phone data only), both classifiers reach an average of over 98.5% balanced accuracy, and are thereby at a similar level of performance as our MD-DTW classification against random forgeries.

To summarize, our experiments on global subject-others classification show that the combination of XY and movement data does result in a performance benefit on both

random and skilled forgeries. We also find that both mobile and touchpad signatures are similar in performance and suitable as platforms for signature verification on global data. With enough signatures available to train with, some models are able to come close to the performance of MD-DTW on random forgeries, and show good, although not equally good performance as MD-DTW on skilled forgeries. This is promising for the potential of global data, but also shows the limitations of our models and training configuration. However, there are other possible setups for global classification that require less samples, and based on our results, we can recommend combined global movement and XY information as feature set and both mobile phones and touchpads as possible platforms for future work.

### 7.6.3 Forgery Detection

The final global scenario we have tested is forgery detection. In it, we train models that learn general characteristics of forged signatures and do not require any additional enrollment signatures. Provided sufficient performance, this would make it possible to deploy one general model to distinguish genuine signatures and forgeries for everyone. It could also be used as an additional step in traditional enrollment-based verification, where new signatures not only have to pass a local DTW-based verification, but also a global forgery detector that sorts out suspicious signatures even earlier. Such a setup could benefit from advantages of using both local and global data, which is a topic further research should investigate. Our results from leave-subject out cross validation are presented in table 7.20.

| Forg. | XY | | Combined | | Movement | | Touchpad | |
|---|---|---|---|---|---|---|---|---|
| Det. | FAR | FRR | FAR | FRR | FAR | FRR | FAR | FRR |
| KNN | 38.43% | 26.02% | 47.14% | 15.59% | 54.21% | 14.14% | 41.00% | 24.11% |
| LSVM | 38.43% | 6.19% | 23.36% | 6.29% | 28.64% | 6.89% | 21.89% | 13.89% |
| RSVM | 94.64% | 1.87% | 99.86% | 0.06% | 84.79% | 3.84% | 90.78% | 4.61% |
| DT | 35.79% | 17.29% | 20.79% | 9.05% | 18.36% | 10.02% | 30.33% | 17.78% |
| RF | 26.93% | 8.33% | 16.07% | 2.57% | 18.21% | 3.70% | 15.56% | 16.67% |
| MLP | 45.36% | 4.02% | 28.14% | 4.76% | 27.71% | 8.78% | 35.11% | 12.94% |
| AdaB | 28.00% | 7.97% | 10.86% | 4.00% | 13.79% | 4.25% | 19.44% | 8.39% |
| NB | 29.43% | 15.38% | 8.93% | 20.43% | 5.71% | 24.33% | 16.44% | 27.33% |

**Table 7.20:** The FAR and FRR of different models in a forgery detection scenario. Parameters were selected using grid search, the average with the best parameters over all subjects in leave-subject-out CV is reported.

The results show that most models are not suitable for forgery detection due to their high false acceptance rates. On their own, they would not be reliable enough to detect targeted attacks, and together with an MD-DTW classifier, they would likely not bring any benefit in detection performance. However, there are two exceptions that show a low-enough false rejection rate to not be frustrating for potential users, while still having an acceptable level of forgery detection: Random Forests at 16.07% FAR, and AdaBoost at 10.86% FAR, both on combined data. Naive Bayes, which has the

lowest FAR of all models on combined data with 8.93%, is not suitable because of its high false rejection rate.

Overall, we can see that combined signatures perform the best, clearly better than XY-only signatures. Those are also outperformed by touchpad signatures, albeit less significantly. Whether this is due to better signature data or because of the aforementioned potentially higher forging difficulty on touchpads as a platform is unclear and is a possible path for further research. What is surprising is that also global movement data outperforms XY-data in the forgery detection setting, and is close to combined signatures in terms of balanced accuracy. Both Random Forests at 90.68% BA and 89.05% BA and AdaBoost at 92.57% BA and 90.98% BA show promising results on combined and movement-only data respectively. This is particularly interesting because it is the only setting in which movement-only signatures consistently outperform XY-only signatures. To be able to say whether this is because global movement data is better for distinguishing genuine signatures from forgeries than global XY-only data, or if the effect is there because skilled forgeries have been recorded sitting only, while genuine signatures have been recorded in other positions as well, we have conducted additional tests. We have run the forgery detector on movement-only signatures with the same settings as before with both AdaBoost and Random Forests as classifiers, but only included signatures that have been recorded sitting. What we found is that while FAR and FRR rate are closer together now (8.64% FAR, 10.38% FRR for AdaBoost and 10.64% FAR, 10.71% FRR for Random Forests), both are within 0.5% balanced accuracy of their previous results (90.49% and 89.33% respectively). This strongly suggests that this remarkably high performance is not caused by the differences between positions. Rather, our results demonstrate the potential of movement data for identifying differences, and warrants further investigation in future work.

Overall, the results using AdaBoost or Random Forests as classifiers on combined and movement data seem to be promising to detect skilled forgeries without training for individual subjects. XY-only signatures are outperformed by touchpad signatures, possibly due to the higher difficulty of forging without direct visual feedback. The level of performance is at around 90% balanced accuracy, and based on our results we think that a more in-depth investigation into general forgery detection and its inclusion as part of a combined local/global verification with MD-DTW could be interesting.

## 7.7 Usability

The usability studies are the final part of our research on mobile digital signing, and the topic of our third research question that is answered in chapter 8.3. In this section, we discuss our findings in detail. For both the mobile and the touchpad database, every participant has filled in a usability questionnaire immediately after completing the second signing session. There are some questions that are equal for both the touchpad and the mobile questionnaires and allow for direct comparison between the results for both devices. The other questions are specific to either touchpad or mobile phone and concern topics such as preferred signing position or behavior while signing on that device. Two questions are not directly about any specific device, but about digital signature verification overall. The results are presented in tables 7.21-7.24. The full questions and answers can be found in the appendix.

| Nr. | Touchpad Questions | Mean | STD |
|---|---|---|---|
| 13 | **Q:** What do you find more comfortable for digital signing?<br>**A:** "Touchpad much more comfortable" (1) to "Smartphone much more comfortable" (5) | 3.62 | 1.22 |
| 14 | **Q:** How did you feel about the width of the touchpad while signing?<br>**A:** "Much too narrow" (1) to "Much too wide" (5) | 2.36 | 0.60 |
| 15 | **Q:** Did you look on the touchpad or the display while signing?<br>**A:** "Always touchpad" (1) to "Always display" (5) | 2.38 | 1.57 |

**Table 7.21:** The results for the questions in our touchpad usability study.

Table 7.21 shows the results to the touchpad-specific questions. Most notable is question 13, where participants were asked for a direct comparison between the two device classes directly after completing the recording process on both devices. We can see that overall, there is a clear tendency of preferring signing on a smartphone over signing on the touchpad. However, the relatively high standard deviation shows the wide range of opinions we see when looking at the underlying data. About a quarter of respondents found signing on a touchpad "somewhat more comfortable" or even "much more comfortable". When including the 13.3% of users who did not have a preference, a total of 37.7% of participants are neutral or leaning towards the touchpad. These results are much better than we initially expected, and show that the touchpad is a real alternative to the smartphone for mobile digital signing.

What is interesting is how the users dealt with the different forms of visual feedback on mobile phones and the touchpad. Question 4 in table 7.22 shows us that the direct visual feedback on smartphone displays was clearly seen as helpful, with a relatively low standard deviation. Only one single participant, or 1.4%, found it to be "somewhat irritating", all others were neutral or mostly positive about it. On the touchpad, the mean is almost balanced at 3.16, but with a large standard deviation showing the mixed opinions. 35.5% responded that the indirect visual feedback was "very irritating" or "somewhat irritating" for them, 37.7% found it "very helpful" or "somewhat helpful". This suggests that there should be an option to turn live visual feedback while signing on the touchpad off for those who do not like it. They could be shown a confirmation screen instead where they can accept or reject the signature once they are done signing. What is also interesting about the indirect visual feedback is where participants looked while signing, as shown in question 15. The results are leaning towards looking on the touchpad, although with a very high standard deviation that shows the heterogeneity of opinions. Almost 45% always, and an additional 20% mostly, looked on the touchpad. Of the others, 20% always looked on the screen, and 6.7% more mostly did so, leaving barely any middle ground. These numbers align well with the percentages of participants that found indirect visual feedback irritating or were indifferent to it. It is confirmed when looking at the individual responses that almost all participants that found the indirect visual feedback irritating looked at the touchpad. However, there are some

| Nr. | Comparative Questions (Mobile Top, Touchpad Bottom) | Mean | STD |
|---|---|---|---|
| 1 | **Q:** What way is signing more comfortable for you?<br>**A:** "Much more comfortable on [smartphone/touchpad]" (1) to "Much more comfortable on paper" (5) | 4.08<br>4.27 | 1.07<br>0.74 |
| 2 | **Q:** How are look and execution of your signature on the [display/touchpad] compared to your signature on paper?<br>**A:** "Completely different" (1) to "Completely the same" (5) | 2.86<br>2.64 | 0.92<br>0.92 |
| 3 | **Q:** How did you feel about the size of the [display/touchpad] while signing?<br>**A:** "Far too small" (1) to "Far too big" (5) | 2.80<br>2.44 | 0.46<br>0.58 |
| 4 | **Q:** How was it for you to be able to see your signature on the display while signing?<br>**A:** "Very irritating" (1) to "Very helpful" (5) | 4.24<br>3.16 | 0.86<br>1.33 |
| 5* | **Q:** Would you prefer paperless signing on a [smartphone/touchpad] or* signing on paper if both options were possible?<br>**A:** "Paperless much preferred"(1) to "Paper much preferred"(5) | 3.42<br>3.31 | 1.30<br>1.47 |
| 6 | **Q:** How likely is it that you would use such a solution for signing digital documents in your daily life?<br>**A:** "Very unlikely" (1) to "Very likely" (5) | 3.49<br>3.38 | 1.20<br>1.32 |
| 7 | **Q:** You have just signed 45 times for our research data recording. If necessary, how many times at most would you be willing to sign for the one-time-setup of a real system of this kind?<br>**A:** "Less than 5" to "More than 50", with steps "5, 10, 15, 20, 30, 40, 50 times maximum" between. Reported is the *mean maximum*, counting "Less than 5" as 1 and "More than 50" as 51. | 28.17<br>26.58 | 17.59<br>17.36 |
| 8* | **Q:** How is signing on a [smartphone/touchpad] compared to other digital solutions that are currently in use, for example at banks or post offices?<br>**A:** "Much less comfortable*" (1) to "Much more comfortable" (5), with an option "I have never signed digitally yet". Three participants used it on both and are excluded in Q8 results. | 3.51<br>3.14 | 0.99<br>1.08 |

**Table 7.22:** The results of the comparative questions of our usability studies. The top results in each cell are the mobile results, the bottom results the the touchpad results. In question 5, five respondents to the international English questionnaire responded to the question saying "...smartphone *over* signing on paper..." while the German version with 66 respondents said "*or (oder)*". In question 8, the English questionnaire contained a literal translation of the German version saying "*more uncomfortable*". In German, this phrasing sounds much clearer than in English, where "*less comfortable*" is easier to understand. Therefore, we corrected both instances here and marked the question with an asterisk. The original versions of the questionnaire can be found in the appendix, and the raw data is provided as CSV-file with the database.

users that found the indirect visual feedback helpful, but still looked at the touchpad while signing. In conversation, many of the participants that reported looking on the touchpad said that they used the smudges the finger left on the surface as a sort of direct visual feedback. This indicates that for about two third of participants, direct visual feedback can not be replaced simply by using indirect visual feedback, although for some having it is still helpful. The 37.7% that preferred touchpad over smartphone for digital signing or were indifferent predominantly looked at the touchpad.

There is not much difference overall between the two device classes when it comes to digital signing compared to signing on paper. Questions 1 and 2 test this, and paper is preferred over both signing on a smartphone and signing on a touchpad, although a bit more strongly in case of the touchpad. It is noteworthy that on the phone, about 10% of subjects rated digital signing to be more comfortable than paper, while only 2.2% did on the touchpad. There does not seem to be much difference in terms of look and execution of the signature between device classes either. The results are around the middle ground of "partly different, partly the same" on paper and digital for both, slightly leaning towards different. However, as we can see in questions 5 and 6, there is still a high user acceptance for digital signing. Despite finding paper more comfortable, the responses to question 6 are leaning towards likely using such a system in daily life. Overall, paper is still the more popular option if given the choice, but on both device classes, there are about 40% of users that do not have a preference or would rather use digital signing. Also, although users generally preferred signing on a smartphone, there is almost no difference between the device classes here, suggesting that comfort is not the most deciding factor for adoption. In fact, with 33% of touchpad users that actually prefer digital signing, that percentage is even slightly higher than on mobile with 25%. It is also important to reiterate that most of our respondents are young, but do not have a technical background.

The size of the signing surface was no problem for most touchpad participants. With a mean of 2.80 and a low standard deviation in question 3, it is close to "Exactly right" (3). The touchpad was generally perceived as being too small. Especially its narrowness was a problem (question 14), while users generally did not have any issues with its height, as we also observed while recording and in conversation. In question 8, the subjects thus clearly rated signing on a smartphone as more comfortable than other solutions they have used so far. Despite the aforementioned issues with its size, signing on a touchpad was still perceived as equally comfortable as existing solutions, even leaning towards slightly better. What is also encouraging is the willingness of participants to provide enrollment signatures. Immediately after signing many times for dataset recording, they reported that they were still willing to provide a maximum of at least over 26 signatures for enrollment on average. This is much more than we actually need for one-class classification with MD-DTW. Only two participants on mobile and touchpad would not be willing to provide the 5 signatures we used in our experiments in a real system ("Less than 5"). Only about a quarter selected a maximum of 10 or fewer, so even with a few practice signatures before enrollment, most users would still set up the system.

For mobile (table 7.23), we have queried the position preferences of users. Sitting was clearly favored over standing in question 10, with only about 14% that find standing more comfortable, and about an additional quarter that liked or disliked both equally.

| Nr. | Mobile Questions | Mean | STD |
|---|---|---|---|
| 9* | **Q:** How would it be for you to lay the smartphone on the table to sign instead of holding it in your hand?<br>**A:** "Much less comfortable*"(1) to "Much more comfortable"(5) | 3.03 | 0.98 |
| 10 | **Q:** How was it more comfortable for you to sign: Sitting or standing?<br>**A:** "Much more comfortable sitting" (1) to "Much more comfortable standing" (5) | 2.27 | 1.05 |
| 11 | **Q:** How was it more comfortable for you to sign: Standing or walking?<br>**A:** "Much more comfortable standing" (1) to "Much more comfortable walking" (5) | 1.52 | 0.84 |
| 12 | **Q:** In which positions would you use the system in your daily life?<br>**A:** "Sitting", "Standing", "Walking", multiple selections possible | Si: 84.5%<br>St: 59.2%<br>Wa: 9.9% | |

**Table 7.23:** The results of the questions in our mobile usability study. In question 8, the English questionnaire contained a literal translation of the German version saying "*more uncomfortable*". In German, this phrasing sounds much clearer than in English, where "*less comfortable*" is easier to understand. Therefore, we corrected it here and marked the question with an asterisk.

Users on average do not have a preference towards holding the phone in hand or laying it on the table while signing when seated. Only 4.2% responded with "much more comfortable" to question 9, and also 4.2% selected "much less comfortable". Three about equal parts of 30% stated that both felt equal to them or leaned somewhat in one or the other direction. Standing however was much preferred over walking, as can be seen in question 11. Only 4.2% found walking to be somewhat more comfortable. This also clearly shows in the results of question 12. Only 9.9% users would use the system when walking, while 59.2% would use it standing, and 84.5% would use it sitting. Not a single user would use the system exclusively when walking, while there are some users that would only use the system standing. These results emphasize sitting and standing as the two most important positions, while walking could also be left out without much concern.

Finally, we have asked our participants about two general issues of digital signing in question 16 and 17 (table 7.24). About 80% of respondents selected that they would be more likely to use it if it made forging signatures more difficult, showing that the security of signatures is a factor of concern. Somewhat surprisingly due to discussions about identity theft and signature privacy we had with a few individual participants, about a third of all subjects reported that they would not have concerns about the protection of their signature data. Also, only between 5% and 10% have "many" concerns. These results are much more in favor of digital signature verification than we expected, and look promising for real-world introduction of such systems.

| Nr. | Overall questions (Mobile Top, Touchpad Bottom) | Yes | No |
|---|---|---|---|
| 16 | **Q:** Would you be more likely to use digital signing if it would make forging your signature harder? | 78.9% | 21.1% |
| | **A:** Selection of either "Yes" or "No" | 84.4% | 15.6% |
| 17 | **Q:** Do you have concerns regarding data protection if such a system gets your signatures? | 70.4% | 29.6% |
| | **A:** Selection of "Yes, many", "Yes, some", or "No". Both positive answers have been grouped together here. | 62.2% | 37.8% |

**Table 7.24:** The results for the questions in our touchpad usability study.

To sum up, we can say that the the touchpad is a viable platform for digital signing not only from a performance, but also from a usability perspective. While most users prefer signing on a smartphone, 25% of our participants found signing on the touchpad more comfortable, and an additional 13% did not have a preference. Users tend to like the feel of signing on paper more than signing digitally, but would use a paperless solution in their daily lives nonetheless, and some would even prefer it when given the choice. The 106 mm by 78 mm touchpad on our test device was considered by many to be too narrow to be comfortable, while the size of the smartphone screens was usually not an issue. Opinions on the indirect visual feedback on the notebook display were divided, with some that found it very helpful and others that were irritated by it. Many users also did not look on the screen while signing, but rather looked on the touchpad where the smudges caused by the finger provided some sort of direct visual feedback. Overall, users felt positively about digital signing on both devices, and the results are promising for the introduction of real-world applications.

# Chapter 8

# Conclusion

The main objective of this thesis was to investigate new platforms and additional features for dynamic signature verification from both a performance and a usability perspective in order to facilitate the secure and widespread use of handwritten digital signing. Of special importance was the protection of signatures against targeted imitation attacks on smartphones and touchpads, and using unconscious phone movements as behavioral biometric alternatives to pressure and pen information that require special hardware to be acquired.

## 8.1 Summary

In chapter 1, the significance of signatures as a cultural technique of authentication, their transition and unchanged importance in a mobile digital world as context of our objective, and the research questions and contributions made by this work within this context have been summarized. Chapter 2 discussed different methods of personal identification and the role of biometrics as an effortless way of confirming identity. It then introduced signatures as a specific example of behavioral biometric authentication, and highlighted the new possibilities in verification opened by the dynamic signatures that can be recorded on digital devices. The current state of the art in the field was presented in chapter 3. It includes a very diverse set of time-series based or statistical classification algorithms, preprocessing chains, and enrollment scenarios. On mobile devices, it also encompasses the consideration of different methods to derive additional features or to compensate for the lack of ones that are only available in a static setting. There is only little research so far that included additional sensor recordings in signature verification, and almost none that has systematically investigated the effect of their use together with traditional image-based features. We further find that there is the need of a comprehensive public mobile dataset to facilitate future work in the area, and discover that the possibility of signing on notebook touchpads has hardly been considered so far.

In chapter 4, the study dataset was introduced. It comprises of a total of 10400 signatures by 73 participants and covers two biometric modalities with data from seven different physical or software-based sensors. To the best of our knowledge, this makes it the largest and most comprehensive mobile signature database in the world. Chapter 5 presented the three stages of the processing chain, recording, preprocessing and clas-

sification. Special emphasis was on Multi-Dimensional Dynamic Time Warping (MD-DTW), a DTW-based classification algorithm and part of the contribution of this thesis that computes the minimal distance between two signatures coherently over all feature dimensions at once, rather than on each one individually. Algorithmic details and information on the implementation of performance-critical parts of the approach were provided in chapter 6. It also assessed the influence of different sampling and alignment strategies on the accuracy of the recorded signatures and explained the choice of linear interpolation and sampling as the least distorting option. Lastly, the evaluation results were outlined in chapter 7. In section

## 8.2 Contributions

This thesis contributes to signature verification on mobile devices with five interconnected parts that can be summarized as follows:

- *The creation of a multi-modal extensible mobile signature database:* To the best of our knowledge, there is currently no other publicly available signature database that comprises both time-series signature information and time-series movement information from different embedded sensors acquired while signing. The lack of such a database on which to conduct research has also been pointed out in related work [64], and its recording and publishing is part of this thesis.
- *The development and open-source publication of recording tools and classification algorithm:* We designed a) the first publicly available scientific signature recording applications for phones and notebook touchpads that we are aware of, and b) the Multi-Dimensional Dynamic Time Warping (MD-DTW) algorithm as a method of distance calculation between signatures. Both our implementations and our database are made available open source.
- *The evaluation of integrated notebook touchpads as a platform for signature verification:* This thesis is the first to explore the authentication of finger-drawn handwritten signatures on notebook touchpads. The only similar work we know of has investigated the classification of made-up paraphs in 2008 [6] on a resistive touchpad. We compare our results to the verification performance on handheld mobile devices and evaluate touchpads as a platform for digital signing.
- *The execution of comparative usability studies on handwritten mobile signing:* We conducted the first usability study on touchpad signatures, and with 71 respondents the largest one on mobile phones. We demonstrate the viability of touchpads a possible platform for dynamic signature verification, point out differences between the device categories, and discuss both their potential benefits and problems.
- *The evaluation of movement data as additional biometric information for signature verification:* We systematically evaluate movement data as a biometric feature in the context of handwritten signatures. The data is recorded unobtrusively from the phones embedded sensors while the user is drawing their signature on the device screen as time series. It is then used on its own and in different combinations of signature image and individual sensor features in local and global scenarios for classification and the difference in performance is measured.

## 8.3 Answers to the Research Questions

From the evaluation results and the corresponding findings, we are able to answer the research questions stated at the beginning of this thesis:

*How does the verification performance of signatures drawn with a finger on an integrated notebook touchpad compare to signatures drawn on a smartphone screen?*
In the same position and using the same features, the verification performance on touchpad and smartphone signatures is about equal. The average results for both are at about 5.75% EER on skilled and 0.75% EER on random forgeries using MD-DTW for classification, which demonstrates that notebook touchpads are a viable platform for handwritten signature verification. When other positions and additional features are included that are only available on handheld mobile devices, smartphone signatures can deliver noticeably better results. In ideal conditions (standing, combined data), the average verification performance reaches 2.59% EER, an improvement of 3.11% EER over the touchpad.

*Does the combination of image-based and movement-based dynamic features of a signature drawn on a mobile device improve verification performance over only image-based and only movement-based systems, and how much do the individual features contribute to the final performance?*
The combination of image-based and movement-based features for signature verification on smartphones results in a clear increase of verification performance in sitting and standing positions against skilled forgeries. The performance benefit over only using dynamic image-based features lies between 0.6% and 2% EER, depending on the individual scenario. Average results in local classification using MD-DTW are at 4.44% EER (combined) and 5.15% EER (image-only) over all stances, and at 3.96% EER (combined) and 5.2% EER (image-only) for sitting and standing signatures only. The best average result on combined data for a single position is 2.59% EER when standing. Against random forgeries, the inclusion of movement information results in average performance levels of about 1.2% EER, a penalty of about 0.5% EER compared to only image features.

Movement features improve the overall result by supporting the classification of signatures that can not be verified well using image-based features alone, while not affecting the outcomes for most already well-performing subjects. Thereby, they can contribute even though the results using movement features only are significantly worse than the ones obtained when only relying on image-based features. It also explains the negative effect against random forgeries, where due to the already very high verification performance of most subjects, movement data can not provide its benefits, but can affect signatures of some subjects negatively with its higher uncertainty.

None of the individual movement features (linear acceleration, gravity sensor, gyroscope) notably outperforms the others on skilled or random forgeries on their own, but when they are combined in movement-only tests, that combination leads to significantly better results. However, that benefit only barely carries over to tests where image features are available. Including the accelerometer on its own performs about equally well as including all three together, and also lead to the best individual result against skilled forgeries over all our experiments at 1.79% EER. The best approach over all positions

is to use the gyroscope when sitting, the accelerometer when standing, and the gravity sensor when walking, which results in an average of 4.19% EER, about 0.3% EER better than the combination of all three sensors at all times.

*What is the usability of signing with a finger on an integrated notebook touchpad, especially with regard to the indirect visual feedback and the narrow shape of the touchpad, and how does it compare to signing on a mobile phone screen?*
The touchpad is a viable platform for digital signing from a usability perspective. While most users prefer signing on a smartphone, 25% of our participants found signing on the touchpad more comfortable, and an additional 13% did not have a preference. Users prefer signing on paper over signing digitally on either platform, but tend to report they would be likely to use such a solution in their daily life, with almost equal results for both devices. The 106 mm by 78 mm touchpad of our test device was considered by many to be too narrow, and some users resorted to writing their signature in two lines to make better use of the available vertical space. The size of smartphone screens was usually not an issue. Opinions on the indirect visual feedback on the notebook display were divided, with some that found it very helpful and others that were irritated by it. Many users also did not look on the screen while signing, but rather looked on the touchpad where the smudges caused by the finger provided some sort of direct visual feedback. The lack of real direct visual feedback also increased the perceived forging difficulty reported by our forgers, and is a possible reason for different kinds of signatures being especially vulnerable to attacks on the two device classes.

Furthermore, we are able to contribute to open questions that have been asked in related work as introduced in chapter 1:

- *Touchsize:* The inclusion of the size of the finger contact area on screen as an additional feature has resulted in performance improvements of 0.5% EER against skilled and 0.2% EER against random forgeries on average. Especially its discriminative power against random forgeries looks promising, where it reaches an average EER of 7.93% on its own in our experiments, probably due to a close relation to other screen-based features. Only about 40% of devices in our database did report high-quality measurements, many of which are in the premium segment, but when it is available, we consider touchsize to be a valuable possible feature for verification that should be recorded.
- *Interoperability:* The use of different touchscreens and movement sensors in different device models can result in differences in the data that is recorded. Those differences make forgeries from other devices easier to identify, but also negatively affect the common use of signature templates across devices. There is likely a small interoperability effect between different smartphones. Based on our results, we consider it to be at about 1.2% EER for image-based features, and about 4% EER for individual movement features. The effect is rather minor in relation to the absolute levels in performance, and it should not be a problem for signature verification across devices. It also does not change any of our other findings.
- *Position and enrollment:* In almost all our experiments, standing is the best-performing position. A possible reason is that signatures are more uniform because it allows for fewer variations in how the device is held than sitting, while delivering more good movement information without the noise caused by steps when walking.

The exception is touchsize, where the increased stability when sitting possibly leads to better application of the finger pressure on the screen, and thus better results. Classification of all positions using one model trained on signatures from all positions, or even one model trained on only standing signatures, is possible for image-based signatures at no significant performance penalty. For combined signatures, this can only be done when walking signatures are excluded due to the their dissimilarity to other positions. However, training an individual model for walking signatures only is possible and delivers good results. Overall, enrollment with randomly selected signatures performs better than enrollment with the first signatures of a session, suggesting that some practice signatures are required before recording samples in real-world applications. Enrollment with samples from both sessions outperforms single-session enrollment, which confirms the template-ageing effect and multi-session enrollment as possible mitigation.

## 8.4 Future Work

While our work has made some contributions to the field of signature verification on mobile devices, there remain other aspects that are open for further investigation in the future. For touchpads, an evaluation of the verification performance on other devices with different touchpads could bring valuable insights and help establish them as a platform for handwritten signatures. Using a recording setup where smudges are hard to see would also allow for an investigation into their importance to users as orientation while signing, as some have reported using them as substitute for the missing direct visual feedback. What also requires more work is the potentially increased forging difficulty on touchpads due to the lack of direct visual feedback. Tests with high effort skilled forgeries could show if there is a difference between both device classes.

Another potential path for future research is to evaluate the effect of the combination of different movement features, possibly even on the level of individual axes, and of combinations of movement features with the touchsize. The touchsize as a feature on its own also warrants further investigation. Tests on skilled forgeries by more different forgers could help quantify the benefit of its inclusion, and a comparison to pressure information recorded at the same time could show a possible correlation, and help determine if it is a potential substitute. On the level of preprocessing, future work could investigate ways to further reduce the effect of interoperability to achieve universal signature samples that can be used on all devices. Also, the processing of movement features and the possible extraction of additional features is an open path for improvement.

To contribute to signature verification more generally, improved enrollment strategies could be evaluated that incorporate practice phases and strategies to select the most similar signatures from a set of enrollment samples to build the best possible model. MD-DTW as a classification algorithm should be evaluated against other means of classification on the same data and setting, and improved as necessary. Also, an investigation into the use of other distance metrics for skilled forgery detection could deliver interesting results. Finally, based on our good results on global information, and especially in the forgery detection scenario on combined data, we also recommend further work into the combination of local and global classifiers to improve verification performance.

# Appendix A

# Participant Instructions (English)

Thank you for your willingness to help us build a dataset of real handwritten signatures on mobile devices. The collected data will be used in the project for training machine learning models to classify signatures and detect forgeries. The results will be part of my masters thesis, and the dataset will be published so that other research project can benefit from it as well. You can find more information about the project and the use of your signatures in the Recording App under the points "Disclaimer" and "Info".

## A.1 Participation

Taking part in the study is easily possible from your home by installing the Recording App for Android (Recording App) on your smartphone. The study consists of two parts, with a break of at least one week in between:

1. Installation and first signature recording
2. Second signature recording (at least 7 days later) and answering the questionnaire

Every part only takes about 10-15 minutes.

### First Signature Recording

When you start the app, you can see the screen on the right side. The **preparation** works step by step as follows:

1. Click *Disclaimer* under *My Signatures* and read how we use your signatures.
2. Click the three dots in the top right corner to view the menu.
3. Open the settings.
4. Fill in the parts *Personal* and *Statistical* fully and correctly.
5. Click *Disclaimer Agreed* to consent to the terms of participation and to be able to record signatures.

Signature Recording consists of three parts:

1. Record sitting (15 signatures)
2. Record standing (15 signatures)
3. Record walking (15 signatures)

In the top left corner of the recording screen, you can see the amount of signatures that have been saved so far. On the bottom right, you can leave the recording screen with *Exit*, undo and repeat a signature that didn't look right with *Undo*, and confirm and save a signature with *Save*. Once you have saved a signature, a button saying *Next* appears on the top right corner, with which you can continue and start recording the next signature.

- **Record sitting:** Sit down and hold your smartphone in one hand. You can support your *elbow* or *forearm* somewhere if it is more comfortable for you. Please **do not** support your *hand* somewhere and **do not** lay down your smartphone on a table to sign. Sign with one finger of the free hand with your normal signature on the display.
- **Record standing:** Stand up. Hold your smartphone in one hand, and sign with one finger of the other hand with your normal signature on the display.
- **Record walking:** Slowly walk around in your room. Hold the smartphone in one hand and, while you walk, sign with one finger of the other hand with your normal signature on the display.

Sign naturally with your real signature. It is not necessary for all signatures to look completely identical, but please stick with one version of your signature. If a signature did not really work or does not look right, click *Undo* and try it again.

To start **recording**, take the following steps:

1. Open *Settings* and select *Sitting* as the first recording position.
2. Make sure that you have selected the correct hand under *Statistical*. If you sign with your right hand on the display, select right, else select left.
3. Click *Record* in the main menu to start recording signatures.
4. Record **15 signatures sitting** as described above, and close the recording view with *Exit* when you are done.
5. Open *Settings* and select *Standing* as second recording position.
6. Record **15 signatures standing** as described above, and close the recording view with *Exit* when you are done.
7. Open *Settings* and select *Walking* as third recording position.
8. Record **15 signatures walking** as described above, and close the recording view with *Exit* when you are done.

Once you are done recording, click the three dots in the top right corner and select *Send Signatures*. After a short delay, the Share-Screen of your smartphone will pop up and you can send the signatures to me via WhatsApp or E-Mail: *[Phone Number],[E-Mail Address]*

### Second Recording Session

**At least one week later:** The second recording session works exactly like the first one. Again, what you have to do:

1. Record sitting (15 signatures)
2. Record standing (15 signatures)
3. Record walking (15 signatures)

Please make sure that you use the same device that you used for the first recording session, and check if all settings are correct. **Take the same steps as for the first recording session.**

Send the signatures of the second recording session to me using *Send Signatures*.

Fill in the questionnaire *immediately after the second recording session*. You can find it at the following link: *[Link to Usability Questionnaire]*

### Record Signatures of Others (Optional)

If you have friends or family members who would like to take part, the easiest way for them to do so is to download the Recording App on their own Smartphone and to follow the instructions given above.

If that is not possible, you can also record their signatures on your own smartphone. In that case, please make sure to do the following:

- Please make sure before every recording that all settings are set to the right person.
- Both recording sessions of a person have to be done on the same device.
- Please strictly follow the instructions above for each person you record. Don't forget to change the position (sitting, standing, walking) setting.
- Every participant has to fill in the questionnaire after the second recording session.

**Thank you very much for your participation!**

## A.2 Illustrations

The illustrations that have been used throughout the participant information sheet are presented in figure A.1. They show possible arm and hand positions for recording sitting and and standing signatures, the main menu of the recording application with the options menu open, and the signing screen with a signature that is not yet saved.

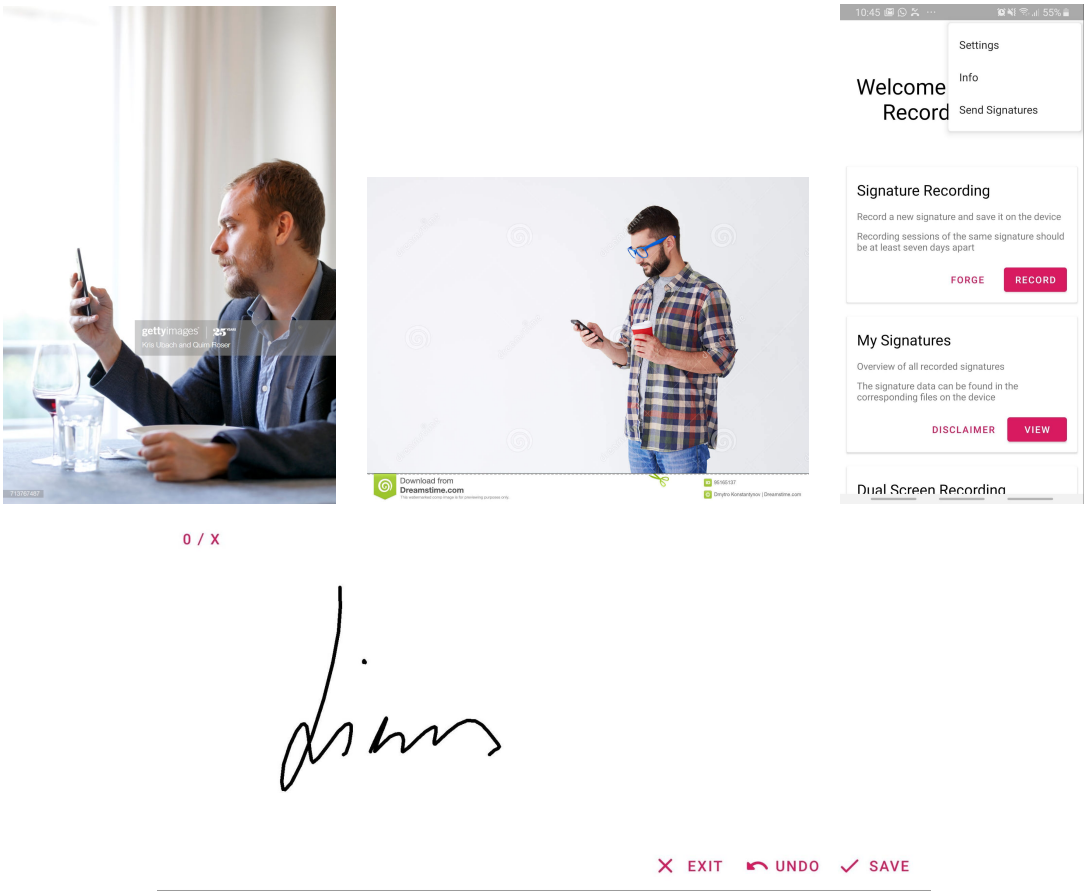



**Figure A.1:** These illustrations have been used in the participant instructions.

# Appendix B

# Participant Instructions (German)

Vielen Dank für Ihre Bereitschaft, uns beim Aufbau eines Datensets von handschriftlichen Unterschriften auf mobilen Geräten zu helfen. Die gesammelten Daten werden im Rahmen des Projekts verwendet, um Machine Learning Models zur Zuordnung und Fälschungserkennung von Unterschriften zu trainieren. Die Ergebnisse werden Teil meiner Masterarbeit, und das Datenset wird veröffentlicht, um auch anderen Forschungsprojekten die Möglichkeit zu bieten, davon zu profitieren. Mehr Informationen über das Projekt und die Verwendung der Unterschriften finden Sie unter den Punkten "Disclaimeründ Ïnfoïn der Aufnahme-App.

## B.1 Teilnahme

Die Teilnahme ist einfach von Zuhause aus möglich, indem Sie sich die Aufnahme-App für Android (Recording App) auf Ihrem Smartphone installieren. Die Studie besteht aus zwei Teilen, mit einer Woche Pause dazwischen:

- Installation und erste Unterschriften-Aufnahme
- Zweite Unterschriften-Aufnahme (Mindestens 7 Tage später) und Beantwortung des Fragebogens

Jeder Teil dauert nur etwa 10-15 Minuten.

### Erste Unterschriften-Aufnahme

Wenn Sie die App starten, sehen Sie den Bildschirm auf der rechten Seite. Die *Vorbereitung* funktioniert Schritt für Schritt so:

1. Klicken Sie unter *My Signatures* auf *Disclaimer* und lesen Sie sich durch, wie wir Ihre Unterschriften verwenden.
2. Klicken Sie auf die drei Punkte oben rechts, um das Menü anzuzeigen.
3. Klicken Sie auf *Settings*, um die Einstellungen zu öffnen.
4. Füllen Sie die Bereiche *Personal* und *Statistical* vollständig und richtig aus.
5. Klicken Sie auf *Disclaimer Agreed*, um den Teilnahmebedingungen zuzustimmen und Unterschriften aufnehmen zu können.

Die Unterschriften-Aufnahme besteht aus drei Teilen:

1. Aufnahme sitzend (15 Unterschriften)
2. Aufnahme stehend (15 Unterschriften)
3. Aufnahme gehend (15 Unterschriften)

Oben links sehen Sie die Anzahl der bisher gespeicherten Unterschriften. Unten rechts können sie mit *Exit* die Unterschriften-Ansicht schließen, mit *Undo* eine Unterschrift noch einmal neu machen, wenn Sie sich vertan haben, und mit *Save* eine Unterschrift speichern. Wenn sie eine Unterschrift gespeichert haben, erscheint oben rechts ein Feld *Next*, mit dem Sie die nächste Unterschrift starten können.

- **Sitzende Aufnahme:** Setzen Sie sich hin und halten Sie Ihr Smartphone in einer Hand. Sie können Ihren *Ellenbogen* oder *Unterarm* abstützen, wenn es für Sie angenehm ist. Bitte stützen Sie Ihre *Hand* **nicht** ab und legen das Smartphone zum Unterschreiben **nicht** auf den Tisch. Unterschreiben Sie mit einem Finger der anderen Hand mit Ihrer normalen Unterschrift auf dem Display.
- **Stehende Aufnahme:** Stehen Sie auf. Halten Sie das Smartphone in einer Hand, und unterschreiben Sie mit einem Finger der anderen Hand mit Ihrer normalen Unterschrift auf dem Display.
- **Gehende Aufnahme:** Gehen Sie langsam im Raum umher. Halten Sie das Smartphone in einer Hand, und während Sie gehen, unterschreiben Sie mit einem Finger der anderen Hand mit Ihrer normalen Unterschrift auf dem Display.

Unterschreiben Sie natürlich mit Ihrer echten Unterschrift. Es ist nicht notwendig, dass alle Unterschriften völlig identisch aussehen, aber bitte bleiben Sie bei einer Version Ihrer Unterschrift. Wenn Ihnen eine Unterschrift nicht gelungen ist oder nicht richtig aussieht, klicken Sie auf *Undo* und machen Sie sie erneut.

Um die **Aufnahme** zu beginnen, befolgen Sie folgende Schritte:

1. Öffnen Sie die *Einstellungen* und wählen Sie *Sitting* als die erste Aufnahme-Position aus.
2. Vergewissern Sie sich, dass Sie unter *Statistical* die richtige Hand ausgewählt haben. Wenn Sie mit der rechten Hand auf dem Bildschirm unterschreiben, wählen sie rechts, ansonsten links.
3. Klicken Sie im Hauptmenü auf *Record* um zu beginnen Unterschriften aufzunehmen.
4. Nehmen Sie **15 Unterschriften im Sitzen** auf wie oben beschrieben, und schließen Sie die Unterschriften-Ansicht danach mit *Exit*.
5. Öffnen Sie die *Einstellungen* und wählen sie *Standing* als zweite Aufnahme-Position aus.
6. Nehmen Sie **15 Unterschriften im Stehen** auf wie oben beschrieben, und schließen Sie die Unterschriften-Ansicht danach mit *Exit*.
7. Öffnen Sie die *Einstellungen* und wählen Sie *Walking* als dritte Aufnahme-Position aus.
8. Nehmen Sie **15 Unterschriften im Gehen** auf wie oben beschrieben, und schließen Sie die Unterschriften-Ansicht danach mit *Exit*.

Wenn Sie mit der Unterschriften-Aufnahme fertig sind, klicken Sie auf die drei Punkte und wählen Sie *Send Signatures* aus. Ihnen wird die Teilen-Ansicht Ihres Smartphones

angezeigt, und Sie können die Unterschriften per WhatsApp oder E-Mail an mich schicken: *[Telefonnummer],[E-Mail-Adresse]*

### Zweite Unterschriften-Aufnahme

**Mindestens eine Woche später:** Die zweite Aufnahme funktioniert gleich wie die erste Aufnahme. Wir benötigen wieder:

1. Aufnahme sitzend (15 Unterschriften)
2. Aufnahme stehend (15 Unterschriften)
3. Aufnahme gehend (15 Unterschriften)

Bitte verwenden Sie dasselbe Gerät wie bei der ersten Aufnahme, und vergewissern Sie sich, dass alle Einstellungen richtig sind. **Folgen Sie denselben Schritten wie bei der ersten Aufnahme.**

Schicken Sie mir die Unterschriften der zweiten Aufnahme-Runde wieder mit *Send Signatures* zu.

Füllen Sie **gleich nach der zweiten Aufnahme** den Fragebogen aus. Sie finden ihn unter dem folgenden Link: *[Link zum Usability-Fragebogen]*

### Unterschriften Anderer Personen Aufnehmen (Optional)

Wenn Sie Familie oder Freunde haben, die ebenfalls teilnehmen möchten, ist der einfachste Weg, wenn diese sich die App auf deren eigenem Smartphone installieren und die oben angegebenen Anweisungen befolgen.

Falls das nicht möglich ist, können Sie deren Unterschriften auch auf Ihrem eigenen Smartphone aufnehmen. Bitte beachten Sie dabei folgende Punkte.

- Bitte überprüfen Sie vor jeder Aufnahme, dass alle Daten in den Einstellungen für die richtige Person eingestellt sind!
- Beide Aufnahme-Runden müssen auf demselben Gerät erfolgen!
- Folgen Sie für jede Person strikt der Anleitung oben. Vergessen Sie nicht, für jede Position (Sitzend, Stehend, Gehend) die Einstellungen zu ändern.
- Jede Teilnehmerin und jeder Teilnehmer muss nach der zweiten Aufnahme den Fragebogen beantworten.

**Vielen Dank für Ihre Teilnahme!**

## B.2 Illustrationen

Die Illustrationen, die in der Anleitung verwendet wurden, werden in Grafik A.1 gezeigt. Sie zeigen mögliche Positionen für Arm und Hand, um Unterschriften sitzend und stehend aufzunehmen, das Hauptmenü der Aufnahme-App mit offenem Options-Menü, und das Unterschriften-Fenster mit einer Unterschrift die noch nicht gespeichert ist.

# Appendix C

# Usability Questionnaires (English)

In this appendix, we include the full international English versions of the usability questionnaires. The questions, and the possible answers to each question, are presented in order of their appearance on the questionnaire. The questions are printed bold. Below each question is the description that was shown alongside (if applicable), and then the answering options separated with a slash. Each option has a number in brackets at the end that shows how many of the participants selected it. The numbers are combined across both language versions of the questionnaire. For the individual results of a specific language version, refer to the usability CSV-files we provide with the database. Note that the English version of the touchpad questionnaire is included for easier understanding, but was not used because all participants were German-speaking.

## C.1 Mobile Questionnaire

Please fill in this questionnaire directly after the second signature recording session. Your answers help us to understand the results of our study and the practicality of signing on smartphones. Your answers will of course be anonymised, but we need your name for internal assignment. The terms and conditions stated in the disclaimer also apply here.

**Agreement to Disclaimer**
We need your agreement to the terms in the disclaimer of our study. You can read it by clicking the button "Disclaimer" in the Recording App where you have recorded your signatures.
Yes, I agree (71) / No, I do not agree (0)

**Please state your full name**
We need your full name for internal assignment. It will of course not be published.
*[Free-text textfield]* (71)

**What way is signing more comfortable for you?**
Much more comfortable on smartphone (4) / Somewhat more comfortable on smartphone (3) / Both are the same (5) / Somewhat more comfortable on paper (30) / Much more comfortable on paper (29)

**How are look and execution of your signature on the display compared to**

**your signature on paper?**
Completely different (7) / Mostly different (14) / Partly different, partly the same (33) / Mostly the same (16) / Completely the same (1)

**How did you feel about the size of the display while signing?**
Far too small (0) / Somewhat too small (16) / Exactly right (53) / Somewhat too big (2) / Far too big (0)

**How would it be for you to lay the smartphone on the table to sign instead of holding it in your hand?**
Much more uncomfortable (3) / Somewhat more uncomfortable (21) / Equally comfortable or uncomfortable (21) / Somewhat more comfortable (23) / Much more comfortable (3)

**How was it more comfortable for you to sign: Sitting or standing?**
Much more comfortable sitting (20) / Somewhat more comfortable sitting (23) / Equally comfortable or uncomfortable (18) / Somewhat more comfortable standing (9) / Much more comfortable standing (1)

**How was it more comfortable for you to sign: Standing or walking?**
Much more comfortable standing (47) / Somewhat more comfortable standing (14) / Equally comfortable or uncomfortable (7) / Somewhat more comfortable walking (3) / Much more comfortable walking (0)

**How was it for you to be able to see your signature on the display while signing?**
Very irritating (0) / Somewhat irritating (1) / Neither irritating nor helpful (17) / Somewhat helpful (17) / Very helpful (36)

**Would you prefer paperless signing on a smartphone over signing on paper if both options were possible?**
Paperless much preferred (9) / Paperless somewhat preferred (9) / Both the same, no preference (11) / Paper somewhat preferred (27) / Paper much preferred (15)

**How likely is it that you would use such a solution for signing digital documents in your daily life?**
Very unlikely (6) / Somewhat unlikely (8) / I can't say (18) / Somewhat likely (23) / Very likely (16)

**Would you be more likely to use digital signing if it would make forging your signature harder?**
Yes (56) / No (15)

**You have just signed 45 times for our research data recording. If necessary, how many times at most would you be willing to sign for the one-time-setup of a real system of this kind?**
Less than 5 (2) / 5 times maximum (8) / 10 times maximum (7) / 15 times maximum (8) / 20 times maximum (11) / 30 times maximum (9) / 40 times maximum (3) / 50 times maximum (15) / More than 50 (8)

**In which positions would you use the system in your daily life?**
*Muliple selections possible:* Sitting (60) / Standing (42) / Walking (7)

**Do you have concerns regarding data protection if such a system gets your signatures?**
Yes, many (8) / Yes, some (42) / No, none (21)

**How is signing on a smartphone compared to other digital solutions that are currently in use, for example at banks or post offices?**
Much more uncomfortable (3) / Somewhat more uncomfortable (6) / Equally comfortable or uncomfortable (22) / Somewhat more comfortable (27) / Much more comfortable (10) / I have never signed digitally yet (3)

## C.2 Touchpad Questionnaire

Please fill in this questionnaire directly after the second signature recording session. Your answers help us to understand the results of our study and the practicality of signing on notebook touchpads. Your answers will of course be anonymised, but we need your name for internal assignment. The terms and conditions stated in the disclaimer also apply here.

**Agreement to Disclaimer**
We need your agreement to the terms in the disclaimer of our study. You can read it by clicking the button "Disclaimer" in the Recording App, or we will provide it to you in printed form.
Yes, I agree (45) / No, I do not agree (0)

**Please state your full name**
We need your full name for internal assignment. It will of course not be published.
*[Free-text textfield]* (45)

**What way is signing more comfortable for you?**
Much more comfortable on touchpad (0) / Somewhat more comfortable on touchpad (1) / Both are the same (5) / Somewhat more comfortable on paper (20) / Much more comfortable on paper (19)

**What do you find more comfortable for digital signing?**
Touchpad much more comfortable (2) / Touchpad somewhat more comfortable (9) / Both are the same (6) / Smartphone somewhat more comfortable (15) / Smartphone much more comfortable (13)

**How are look and execution of your signature on the touchpad compared to your signature on paper?**
Completely different (5) / Mostly different (15) / Partly different, partly the same (16) / Mostly the same (9) / Completely the same (0)

**How did you feel about the size of the touchpad while signing?**
Far too small (2) / Somewhat too small (21) / Exactly right (22) / Somewhat too big (0) / Far too big (0)

**How did you feel about the width of the touchpad while signing?**
Much too narrow (3) / Somewhat too narrow (23) / Exactly right (19) / Somewhat too wide (0) / Much too wide (0)

**Did you look on the touchpad or the display while signing?**
Always touchpad (20) / Mostly touchpad (9) / Both equally often (4) / Mostly display (3) / Always display (9)

**How was it for you to be able to see your signature on the display while signing?**
Very irritating (5) / Somewhat irritating (11) / Neither irritating nor helpful (12) / Somewhat helpful (6) / Very helpful (11)

**Would you prefer paperless signing on a touchpad over signing on paper if both options were possible?**
Paperless much preferred (9) / Paperless somewhat preferred (6) / Both the same, no preference (3) / Paper somewhat preferred (16) / Paper much preferred (11)

**How likely is it that you would use such a solution for signing digital documents in your daily life?**
Very unlikely (4) / Somewhat unlikely (11) / I can't say (5) / Somewhat likely (14) / Very likely (11)

**Would you be more likely to use digital signing if it would make forging your signature harder?**
Yes (38) / No (7)

**You have just signed 45 times for our research data recording. If necessary, how many times at most would you be willing to sign for the one-time-setup of a real system of this kind?**
Less than 5 (2) / 5 times maximum (4) / 10 times maximum (8) / 15 times maximum (2) / 20 times maximum (8) / 30 times maximum (7) / 40 times maximum (1) / 50 times maximum (9) / More than 50 (4)

**In which positions would you use the system in your daily life?**
*Muliple selections possible:* Sitting (60) / Standing (42) / Walking (7)

**Do you have concerns regarding data protection if such a system gets your signatures?**
Yes, many (2) / Yes, some (26) / No, none (17)

**How is signing on a touchpad compared to other digital solutions that are currently in use, for example at banks or post offices?**
Much more uncomfortable (3) / Somewhat more uncomfortable (8) / Equally comfortable or uncomfortable (16) / Somewhat more comfortable (10) / Much more comfortable (5) / I have never signed digitally yet (3)

# Appendix D

# Usability Questionnaires (German)

In this appendix, we include the full German versions of the usability questionnaires. The questions, and the possible answers to each question, are presented in order of their appearance on the questionnaire. The questions are printed bold. Below each question is the description that was shown alongside (if applicable), and then the answering options separated by a slash. Each option has a number in brackets at the end that shows how many of the participants selected it. The numbers are combined across both language versions of the questionnaire. For the individual results of a specific language version, refer to the usability CSV-files we provide with the database.

## D.1 Fragebogen Smartphone

Bitte beantworten Sie diesen Fragebogen direkt nach der zweiten Unterschriften-Aufnahme. Er hilft uns dabei, die Ergebnisse unserer Studie sowie die praktische Nutzbarkeit des Unterschreibens auf Smartphones einzuschätzen. Ihre Antworten werden selbstverständlich anonymisiert, wir benötigen jedoch Ihren Namen zur internen Zuordnung. Die im Disclaimer beschriebenen Teilnahmebedingungen gelten auch hier.

**Zustimmung zum Disclaimer**
Wir benötigen Ihre Zustimmung zu den Bedingungen unserer Studie. Diese finden Sie unter dem Punkt "Disclaimerïn der Recording App, in der Sie Ihre Unterschriften aufgenommen haben.
Ja, ich stimme zu (71) / Nein, ich stimme nicht zu (0)

**Bitte geben Sie Ihren vollen Namen an**
Wir benötigen Ihren vollen Namen zur internen Zuordnung. Er wird selbstverständlich nicht mit veröffentlicht.
*[Freitext-Textfeld]* (71)

**Worauf ist Unterschreiben für Sie angenehmer?**
Sehr viel angenehmer auf Smartphone (4) / Etwas angenehmer auf Smartphone (3) / Beides gleich angenehm (5) / Etwas angenehmer auf Papier (30) / Sehr viel angenehmer auf Papier (29)

**Wie sind Aussehen und Ausführung Ihrer Unterschrift auf dem Display im Vergleich zu Ihrer Unterschrift auf Papier?**

Völlig anders (7) / Vorwiegend anders (14) / Teilweise anders, teilweise gleich (33) / Vorwiegend gleich (16) / Völlig gleich (1)

**Wie haben Sie die Größe des Bildschirms beim Unterschreiben empfunden?**
Viel zu klein (0) / Etwas zu klein (22) / Genau richtig (53) / Etwas zu groß (2) / Viel zu groß (0)

**Wie wäre es für Sie, das Smartphone zum Unterschreiben auf dem Tisch abzulegen, anstatt es in der Hand zu halten?**
Sehr viel unangenehmer (3) / Etwas unangenehmer (21) / Gleich angenehm oder unangenehm (21) / Etwas angenehmer (23) / Sehr viel angenehmer (3)

**Wie war es angenehmer für Sie zu unterschreiben: Sitzend oder stehend?**
Sehr viel angenehmer sitzend (20) / Etwas angenehmer sitzend (23) / Gleich angenehm oder unangenehm (18) / Etwas angenehmer stehend (9) / Sehr viel angenehmer stehend (1)

**Wie war es angenehmer für Sie zu unterschreiben: Stehend oder gehend?**
Sehr viel angenehmer stehend (47) / Etwas angenehmer stehend (14) / Gleich angenehm oder unangenehm (7) / Etwas angenehmer gehend (3) / Sehr viel angenehmer gehend (0)

**Wie war es für Sie, Ihre Unterschrift während des Unterschreibens auf dem Display sehen zu können?**
Sehr irritierend (0) / Etwas irritierend (1) / Weder irritierend noch hilfreich (17) / Etwas hilfreich (17) / Sehr hilfreich (36)

**Würden Sie das Papierlose Unterschreiben auf dem Smartphone oder das Unterschreiben auf Papier bevorzugen, wenn beides möglich ist?**
Papierlos stark bevorzugt (9) / Papierlos etwas bevorzugt (9) / Beides gleich, keine Präferenz (11) / Papier etwas bevorzugt (27) / Papier stark bevorzugt (15)

**Wie wahrscheinlich ist es, dass Sie eine solche Lösung zum Unterschreiben von digitalen Dokumenten im Alltag verwenden würden?**
Sehr unwahrscheinlich (6) / Eher unwahrscheinlich (8) / Kann es nicht sagen (18) / Eher wahrscheinlich (23) / Sehr wahrscheinlich (16)

**Würden Sie das digitale Unterzeichnen eher verwenden, wenn es das Fälschen Ihrer Unterschrift erschwert?**
Ja (56) / Nein (15)

**Sie haben für die Forschungsdatenaufnahme gerade 45 Mal unterschrieben. Wie viele Unterschriften wären Sie wenn notwendig bereit, bei der einmaligen Einrichtung eines echten solchen Systems maximal abzugeben?**
Weniger als 5 (2) / Maximal 5 (8) / Maximal 10 (7) / Maximal 15 (8) / Maximal 20 (11) / Maximal 30 (9) / Maximal 40 (3) / Maximal 50 (15) / Mehr als 50 (8)

**In welchen Positionen würden Sie das System im Alltag verwenden?**
*Mehrfache Auswahl möglich:* Sitzend (60) / Stehend (42) / Gehend (7)

**Haben Sie Bedenken bezüglich des Datenschutzes, wenn ein solches System Unterschriften von Ihnen erhält?**
Ja, sehr (8) / Ja, etwas (42) / Nein, keine (21)

**Wie ist das Unterschreiben auf dem Smartphone im Vergleich zu bisherigen digitalen Lösungen, die bereits im Einsatz sind, etwa bei Banken oder bei der Post?**
Sehr viel unangenehmer (3) / Etwas unangenehmer (6) / Gleich angenehm oder unangenehm (22) / Etwas angenehmer (27) / Sehr viel angenehmer (10) / Ich habe noch nie digital unterschrieben (3)

## D.2 Fragebogen Touchpad

Bitte beantworten Sie diesen Fragebogen direkt nach der zweiten Unterschriften-Aufnahme. Er hilft uns dabei, die Ergebnisse unserer Studie sowie die praktische Nutzbarkeit des Unterschreibens auf Notebook-Touchpads einzuschätzen. Ihre Antworten werden selbstverständlich anonymisiert, wir benötigen jedoch Ihren Namen zur internen Zuordnung. Die im Disclaimer beschriebenen Teilnahmebedingungen gelten auch hier.

**Zustimmung zum Disclaimer**
Wir benötigen Ihre Zustimmung zu den Bedingungen unserer Studie. Diese finden Sie unter dem Punkt "Disclaimerïn der Recording App, oder Sie bekommen sie von uns schriftlich vorgelegt.
Ja, ich stimme zu (45) / Nein, ich stimme nicht zu (0)

**Bitte geben Sie Ihren vollen Namen an**
Wir benötigen Ihren vollen Namen zur internen Zuordnung. Er wird selbstverständlich nicht mit veröffentlicht.
*[Freitext-Textfeld]* (45)

**Worauf ist Unterschreiben für Sie angenehmer?**
Sehr viel angenehmer auf Touchpad (0) / Etwas angenehmer auf Touchpad (1) / Beides gleich angenehm (5) / Etwas angenehmer auf Papier (20) / Sehr viel angenehmer auf Papier (19)

**Was finden Sie zum digitalen Unterschreiben angenehmer?**
Touchpad viel angenehmer (2) / Touchpad etwas angenehmer (9) / Beides gleich angenehm oder unangenehm (6) / Smartphone etwas angenehmer (15) / Smartphone viel angenehmer (13)

**Wie sind Aussehen und Ausführung Ihrer Unterschrift auf dem Touchpad im Vergleich zu Ihrer Unterschrift auf Papier?**
Völlig anders (5) / Vorwiegend anders (15) / Teilweise anders, teilweise gleich (16) / Vorwiegend gleich (9) / Völlig gleich (0)

**Wie haben Sie die Größe des Touchpads beim Unterschreiben empfunden?**
Viel zu klein (2) / Etwas zu klein (21) / Genau richtig (22) / Etwas zu groß (0) / Viel zu groß (0)

**Wie haben Sie die Breite des Touchpads beim Unterschreiben empfunden?**
Viel zu schmal (3) / Etwas zu schmal (23) / Genau richtig (19) / Etwas zu breit (0) / Viel zu breit (0)

**Haben Sie beim Unterschreiben auf das Touchpad oder auf das Display geschaut?**

Immer Touchpad (20) / Vorwiegend Touchpad (9) / Beides etwa gleich viel (4) / Vorwiegend Display (3) / Immer Display (9)

**Wie war es für Sie, Ihre Unterschrift während des Unterschreibens auf dem Display sehen zu können?**
Sehr irritierend (5) / Etwas irritierend (11) / Weder irritierend noch hilfreich (12) / Etwas hilfreich (6) / Sehr hilfreich (11)

**Würden Sie das Papierlose Unterschreiben auf dem Touchpad oder das Unterschreiben auf Papier bevorzugen, wenn beides möglich ist?**
Papierlos stark bevorzugt (9) / Papierlos etwas bevorzugt (6) / Beides gleich, keine Präferenz (3) / Papier etwas bevorzugt (16) / Papier stark bevorzugt (11)

**Wie wahrscheinlich ist es, dass Sie eine solche Lösung zum Unterschreiben von digitalen Dokumenten im Alltag verwenden würden?**
Sehr unwahrscheinlich (4) / Eher unwahrscheinlich (11) / Kann es nicht sagen (5) / Eher wahrscheinlich (14) / Sehr wahrscheinlich (11)

**Würden Sie das digitale Unterzeichnen eher verwenden, wenn es das Fälschen Ihrer Unterschrift erschwert?**
Ja (38) / Nein (7)

**Sie haben für die Forschungsdatenaufnahme gerade 45 Mal unterschrieben. Wie viele Unterschriften wären Sie wenn notwendig bereit, bei der einmaligen Einrichtung eines echten solchen Systems maximal abzugeben?**
Weniger als 5 (2) / Maximal 5 (4) / Maximal 10 (8) / Maximal 15 (2) / Maximal 20 (8) / Maximal 30 (7) / Maximal 40 (1) / Maximal 50 (9) / Mehr als 50 (4)

**Haben Sie Bedenken bezüglich des Datenschutzes, wenn ein solches System Unterschriften von Ihnen erhält?**
Ja, sehr (2) / Ja, etwas (26) / Nein, keine (17)

**Wie ist das Unterschreiben auf dem Touchpad im Vergleich zu bisherigen digitalen Lösungen, die bereits im Einsatz sind, etwa bei Banken oder bei der Post?**
Sehr viel unangenehmer (3) / Etwas unangenehmer (8) / Gleich angenehm oder unangenehm (16) / Etwas angenehmer (10) / Sehr viel angenehmer (5) / Ich habe noch nie digital unterschrieben (3)

# References

## Literature


[1] Majd Abazid, Nesma Houmani, and Sonia Garcia-Salicetti. "Enhancing Security on Touch-Screen Sensors with Augmented Handwritten Signatures". *Sensors* 20.3 (2020), p. 933 (cit. on pp. 26, 27).

[2] S. M. S. Ahmad et al. "SIGMA - A Malaysian signatures' database". In: *2008 IEEE/ACS International Conference on Computer Systems and Applications*. Mar. 2008, pp. 919–920 (cit. on pp. 15–17, 29, 30, 36).

[3] Nora Alkaldi and Karen Renaud. "Why do people adopt, or reject, smartphone password managers?" (2016) (cit. on p. 6).

[4] Fernando Alonso-Fernandez, Julian Fierrez, and Javier Ortega-Garcia. "Sensor Interoperability and Fusion in Signature Verification: A Case Study Using Tablet PC". In: vol. 3781. Jan. 2005, pp. 180–187 (cit. on pp. 19, 37).

[5] Fadi Aloul, Syed Zahidi, and Wassim El-Hajj. "Two factor authentication using mobile phones". In: *2009 IEEE/ACS International Conference on Computer Systems and Applications*. IEEE. 2009, pp. 641–644 (cit. on pp. 4, 8).

[6] Tansu Alpcan et al. "A lightweight biometric signature scheme for user authentication over networks". In: *Proceedings of the 4th international conference on Security and privacy in communication netowrks*. 2008, pp. 1–6 (cit. on pp. 28, 30, 47, 89, 122).

[7] Salvatore Aurigemma, Thomas Mattson, and Lori Leonard. "So much promise, so little use: What is stopping home end-users from using password manager applications?" (2017) (cit. on p. 6).

[8] Adam J Aviv et al. "Practicality of accelerometer side channels on smartphones". In: *Proceedings of the 28th Annual Computer Security Applications Conference*. 2012, pp. 41–50 (cit. on p. 6).

[9] Adam J Aviv et al. "Smudge attacks on smartphone touch screens." *Woot* 10 (2010), pp. 1–7 (cit. on p. 6).

[10] Gonzalo Bailador et al. "Analysis of pattern recognition techniques for in-air signature biometrics". *Pattern Recognition* 44.10 (2011). Semi-Supervised Learning for Visual Content Analysis and Understanding, pp. 2468–2478. URL: http://www.sciencedirect.com/science/article/pii/S0031320311001683 (cit. on pp. 16, 18).

[11] Mohammadreza Hazhirpasand Barkadehi et al. "Authentication systems: A literature review and classification". *Telematics and Informatics* 35.5 (2018), pp. 1491–1511 (cit. on p. 8).

[12] Luana Batista, Eric Granger, and Robert Sabourin. "Dynamic selection of generative–discriminative ensembles for off-line signature verification". *Pattern Recognition* 45.4 (2012), pp. 1326–1340 (cit. on p. 14).

[13] Mustafa Berkay Yilmaz and Kagan Ozturk. "Hybrid user-independent and user-dependent offline signature verification with a two-channel CNN". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 526–534 (cit. on p. 14).

[14] Diego Bertolini et al. "Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers". *Pattern Recognition* 43.1 (2010), pp. 387–396 (cit. on p. 13).

[15] Diego Bertolini et al. "Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers". *Pattern Recognition* 43.1 (2010), pp. 387–396 (cit. on p. 14).

[16] Byron Bezerra et al. *Handwriting: Recognition, development and analysis*. Jan. 2017, pp. 1–394 (cit. on pp. 16, 17, 29, 30).

[17] Debnath Bhattacharyya et al. "Biometric authentication: A review". *International Journal of u-and e-Service, Science and Technology* 2.3 (2009), pp. 13–28 (cit. on p. 5).

[18] Andrea Bianchi and Ian Oakley. "Wearable authentication: Trends and opportunities". *it-Information Technology* 58.5 (2016), pp. 255–262 (cit. on p. 8).

[19] Ramon Blanco-Gonzalo et al. "Automatic usability and stress analysis in mobile biometrics". *Image and Vision Computing* 32.12 (2014), pp. 1173–1180 (cit. on pp. 31, 32, 38).

[20] Ramon Blanco-Gonzalo et al. "Usability analysis of dynamic signature verification in mobile environments". In: *2013 International Conference of the BIOSIG Special Interest Group (BIOSIG)*. IEEE. 2013, pp. 1–9 (cit. on pp. 31, 32).

[21] Ramon Blanco-Gonzalo et al. "Usability evaluation of biometrics in mobile environments". In: *2013 6th International Conference on Human System Interactions (HSI)*. IEEE. 2013, pp. 123–128 (cit. on p. 31).

[22] R Blanco-Gonzalo et al. "Handwritten signature recognition in mobile scenarios: Performance evaluation". In: Oct. 2012 (cit. on pp. 17, 18, 20, 28–30, 65, 89).

[23] R Blanco-Gonzalo et al. "Performance evaluation of handwritten signature recognition in mobile environments". *IET Biometrics* 3 (Sept. 2013) (cit. on pp. 10–12, 15–17, 20, 23, 28–32, 38, 43, 51, 60, 63, 65, 89).

[24] V. L. Blankers et al. "ICDAR 2009 Signature Verification Competition". In: *2009 10th International Conference on Document Analysis and Recognition*. July 2009, pp. 1403–1407 (cit. on pp. 16, 17, 19, 29, 30, 36).

[25] Joseph Bonneau. "The science of guessing: analyzing an anonymized corpus of 70 million passwords". In: *2012 IEEE Symposium on Security and Privacy*. IEEE. 2012, pp. 538–552 (cit. on p. 6).

[26] John Brainard et al. "Fourth-factor authentication: somebody you know". In: *Proceedings of the 13th ACM conference on Computer and communications security*. 2006, pp. 168–178 (cit. on p. 5).

[27] Attaullah Buriro et al. "Itsme: Multi-modal and unobtrusive behavioural user authentication for smartphones". In: *International Conference on Passwords*. Springer. 2015, pp. 45–61 (cit. on pp. 8, 17, 36, 70).

[28] Attaullah Buriro et al. "Touchstroke: Smartphone user authentication based on touch-typing biometrics". In: *International Conference on Image Analysis and Processing*. Springer. 2015, pp. 27–34 (cit. on pp. 17, 18, 69).

[29] A. Buriro et al. "Hold and Sign: A Novel Behavioral Biometrics for Smartphone User Authentication". In: *2016 IEEE Security and Privacy Workshops (SPW)*. May 2016, pp. 276–285 (cit. on pp. 9, 11, 15, 16, 18, 22, 24, 25, 29–33, 35, 36, 47, 61–63, 69, 90, 97).

[30] Karoline Busse, Julia Schäfer, and Matthew Smith. "Replication: no one can hack my mind revisiting a study on expert and non-expert security practices and advice". In: *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. 2019 (cit. on p. 6).

[31] Liang Cai and Hao Chen. "On the practicality of motion based keystroke inference attack". In: *International Conference on Trust and Trustworthy Computing*. Springer. 2012, pp. 273–290 (cit. on p. 6).

[32] Liang Cai and Hao Chen. "TouchLogger: Inferring Keystrokes on Touch Screen from Smartphone Motion." *HotSec* 11.2011 (2011), p. 9 (cit. on p. 6).

[33] Timothy TH Chen, S Pels, and Saehee Sarah Min. "Flowfield and beyond: applying pressure-sensitive multi-point touchpad interaction". In: *2003 International Conference on Multimedia and Expo. ICME'03. Proceedings (Cat. No. 03TH8698)*. Vol. 1. IEEE. 2003, pp. I–49 (cit. on p. 47).

[34] Mauro Conti, Irina Zachia-Zlatea, and Bruno Crispo. "Mind how you answer me! Transparently authenticating the user of a smartphone when answering or placing a call". In: *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. 2011, pp. 249–259 (cit. on pp. 17, 36, 65).

[35] Sounak Dey et al. "Signet: Convolutional siamese network for writer independent offline signature verification". *arXiv preprint arXiv:1707.02131* (2017) (cit. on pp. 12–14).

[36] Moises Diaz et al. "A perspective analysis of handwritten signature technology". *ACM Computing Surveys (CSUR)* 51.6 (2019), pp. 1–39 (cit. on pp. 3, 31–33, 35, 103).

[37] Moises Diaz et al. "Dynamic signature verification system based on one real signature". *IEEE transactions on cybernetics* 48.1 (2016), pp. 228–239 (cit. on p. 19).

[38] Bruno Dumas et al. "MyIDea-multimodal biometrics database, description of acquisition protocols". *Proc. Third COST* 275 (2005), pp. 59–62 (cit. on pp. 11, 16, 17).

[39] Mounim El Yacoubi, Robert Sabourin, and Flavio Bortolozzi. "An off-line Signature Verification System using HMM and Graphometric Features". In: Dec. 2000 (cit. on p. 14).

[40] Asghar Fallah, Mahdi Jamaati, and Ali Soleamani. "A new online signature verification system based on combining Mellin transform, MFCC and neural network". *Digital Signal Processing* 21.2 (2011), pp. 404–416 (cit. on p. 19).

[41] Tao Feng et al. "Continuous mobile authentication using virtual key typing biometrics". In: *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE. 2013, pp. 1547–1552 (cit. on p. 8).

[42] M. A. Ferrer, M. Diaz-Cabrera, and A. Morales. "Synthetic off-line signature image generation". In: *2013 International Conference on Biometrics (ICB)*. 2013, pp. 1–7 (cit. on p. 14).

[43] Miguel A Ferrer, Jesus B Alonso, and Carlos M Travieso. "Offline geometric parameters for automatic signature verification using fixed-point arithmetic". *IEEE transactions on pattern analysis and machine intelligence* 27.6 (2005), pp. 993–997 (cit. on p. 14).

[44] Julian Fierrez et al. "HMM-based on-line signature verification: Feature extraction and signature modeling". *Pattern recognition letters* 28.16 (2007), pp. 2325–2334 (cit. on p. 61).

[45] Julian Fierrez-Aguilar et al. "An off-line signature verification system based on fusion of local and global information". In: *International workshop on biometric authentication*. Springer. 2004, pp. 295–306 (cit. on p. 14).

[46] Julian Fierrez-Aguilar et al. "An on-line signature verification system based on fusion of local and global information". In: *International Conference on Audio- and Video-Based Biometric Person Authentication*. Springer. 2005, pp. 523–532 (cit. on pp. 12, 23).

[47] Javier Galbally, Marta Gomez-Barrero, and Arun Ross. "Accuracy evaluation of handwritten signature verification: Rethinking the random-skilled forgeries dichotomy". In: *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE. 2017, pp. 302–310 (cit. on p. 13).

[48] Sonia Garcia-Salicetti et al. "BIOMET: A Multimodal Person Authentication Database Including Face, Voice, Fingerprint, Hand and Signature Modalities". In: June 2003, pp. 845–853 (cit. on pp. 11, 17, 29, 30).

[49] Hugo Gascon et al. "Continuous authentication on mobile devices by analysis of typing motion behavior". *Sicherheit 2014–Sicherheit, Schutz und Zuverlässigkeit* (2014) (cit. on p. 8).

[50] Toni Giorgino et al. "Computing and visualizing dynamic time warping alignments in R: the dtw package". *Journal of statistical Software* 31.7 (2009), pp. 1–24 (cit. on pp. 65, 66).

[51] Mayank Goel, Jacob Wobbrock, and Shwetak Patel. "GripSense: using built-in sensors to detect hand posture and pressure on commodity mobile phones". In: *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 2012, pp. 545–554 (cit. on p. 16).

[52] Marta Gomez-Barrero, Julian Fierrez, and Javier Galbally. "Variable-length template protection based on homomorphic encryption with application to signature biometrics". In: *2016 4th International Conference on Biometrics and Forensics (IWBF)*. IEEE. 2016, pp. 1–6 (cit. on p. 21).

[53] Marta Gomez-Barrero et al. "Enhanced on-line signature verification based on skilled forgery detection using sigma-lognormal features". In: *2015 international conference on biometrics (ICB)*. IEEE. 2015, pp. 501–506 (cit. on p. 20).

[54] Yasmine Guerbai, Youcef Chibani, and Bilal Hadjadji. "The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters". *Pattern Recognition* 48.1 (2015), pp. 103–113 (cit. on p. 14).

[55] Richard Guest. "Age dependency in handwritten dynamic signature verification systems". *Pattern Recognition Letters* 27.10 (2006), pp. 1098–1104 (cit. on pp. 31, 32).

[56] Luiz G Hafemann, Robert Sabourin, and Luiz S Oliveira. "Learning features for offline handwritten signature verification using deep convolutional neural networks". *Pattern Recognition* 70 (2017), pp. 163–176 (cit. on p. 14).

[57] Luiz G Hafemann, Robert Sabourin, and Luiz S Oliveira. "Offline handwritten signature verification—Literature review". In: *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE. 2017, pp. 1–8 (cit. on p. 14).

[58] Changyo Han, Katsufumi Matsui, and Takeshi Naemura. "ForceStamps: Fiducial Markers for Pressure-sensitive Touch Surfaces to Support Rapid Prototyping of Physical Control Interfaces". In: *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 2020, pp. 273–285 (cit. on p. 47).

[59] Seongkook Heo and Geehyuk Lee. "Forcetap: extending the input vocabulary of mobile touch screens by adding tap gestures". In: *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*. 2011, pp. 113–122 (cit. on p. 16).

[60] Rami Al-Hmouz et al. "Quantifying dynamic time warping distance using probabilistic model in verification of dynamic signatures". *Soft Computing* 23.2 (2019), pp. 407–418 (cit. on p. 19).

[61] Nesma Houmani et al. "BioSecure signature evaluation campaign (BSEC'2009): Evaluating online signature algorithms depending on the quality of signatures". *Pattern Recognition* 45.3 (2012), pp. 993–1003 (cit. on pp. 19, 29, 32).

[62] Kai Huang and Hong Yan. "Off-line signature verification based on geometric feature extraction and neural network classification". *Pattern Recognition* 30.1 (1997), pp. 9–17 (cit. on p. 14).

[63] Xinyi Huang et al. “A generic framework for three-factor authentication: Preserving security and privacy in distributed systems”. *IEEE Transactions on Parallel and Distributed Systems* 22.8 (2010), pp. 1390–1397 (cit. on pp. 4, 8).

[64] Donato Impedovo and Giuseppe Pirlo. “Automatic signature verification in the mobile cloud scenario: survey and way ahead”. *IEEE Transactions on Emerging Topics in Computing* (2018) (cit. on pp. 3, 29, 30, 33, 35, 103, 122).

[65] Iulia Ion, Rob Reeder, and Sunny Consolvo. “"... no one can hack my mind": Comparing Expert and Non-Expert Security Practices”. In: *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*. 2015, pp. 327–346 (cit. on p. 6).

[66] Fumitada Itakura. “Minimum prediction residual principle applied to speech recognition”. *IEEE Transactions on acoustics, speech, and signal processing* 23.1 (1975), pp. 67–72 (cit. on p. 66).

[67] Jeff Johnson et al. *Picture gesture authentication*. US Patent 8,650,636. Feb. 2014 (cit. on pp. 5, 6).

[68] Ankur Joshi et al. “Likert scale: Explored and explained”. *Current Journal of Applied Science and Technology* (2015), pp. 396–403 (cit. on p. 54).

[69] Edson JR Justino, Flávio Bortolozzi, and Robert Sabourin. “Off-line signature verification using HMM for random, simple and skilled forgeries”. In: *Proceedings of Sixth International Conference on Document Analysis and Recognition*. IEEE. 2001, pp. 1031–1034 (cit. on p. 13).

[70] Edson Justino, Flavio Bortolozzi, and Robert Sabourin. “A comparison of SVM and HMM classifiers in the off-line signature verification”. *Pattern Recognition Letters* 26 (July 2005), pp. 1377–1385 (cit. on p. 14).

[71] Meenakshi K. Kalera, Sargur N. Srihari, and Aihua Xu. “Offline Signature Verification And Identification Using Distance Statistics”. *Int. J. Pattern Recognit. Artif. Intell.* 18 (2004), pp. 1339–1360 (cit. on p. 14).

[72] Christina Katsini et al. “Security and usability in knowledge-based user authentication: A review”. In: *Proceedings of the 20th Pan-Hellenic Conference on Informatics*. 2016, pp. 1–6 (cit. on p. 4).

[73] Alisher Kholmatov and Berrin Yanikoglu. “Identity authentication using improved online signature verification method”. *Pattern recognition letters* 26.15 (2005), pp. 2400–2408 (cit. on pp. 17, 63, 65).

[74] Alisher Kholmatov and Berrin Yanikoglu. “SUSIG: An on-line signature database, associated protocols and benchmark results”. *Formal Pattern Analysis & Applications* 12 (Sept. 2009), pp. 227–236 (cit. on pp. 16, 17, 26, 29, 30).

[75] “Knock, Knock. Whos There? On the Security of LGs Knock Codes”. In: *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*. USENIX Association, Aug. 2020, pp. 37–59. URL: https://www.usenix.org/conference/soups2020/presentation/samuel (cit. on pp. 5, 6).

[76] Ram P. Krish et al. “Dynamic Signature Verification on Smart Phones”. In: vol. 365. May 2013 (cit. on pp. 10, 12, 15, 23, 24, 30–32, 36, 47, 65, 78).

[77] Rajesh Kumar, JD Sharma, and Bhabatosh Chanda. "Writer-independent off-line signature verification using surroundedness feature". *Pattern recognition letters* 33.3 (2012), pp. 301–308 (cit. on p. 14).

[78] Imane Lamiche et al. "A continuous smartphone authentication method based on gait patterns and keystroke dynamics". *Journal of Ambient Intelligence and Humanized Computing* 10.11 (2019), pp. 4417–4430 (cit. on p. 8).

[79] Andreas Lanitis. "A survey of the effects of aging on biometric identity verification". *International Journal of Biometrics* 2.1 (2010), pp. 34–52 (cit. on p. 9).

[80] M. Lech and A. Czyzewski. "A handwritten signature verification method employing a tablet". In: *2016 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*. Sept. 2016, pp. 45–50 (cit. on pp. 19, 31, 65).

[81] Hansheng Lei and Venu Govindaraju. "A comparative study on the consistency of features in on-line signature verification". *Pattern Recognition Letters* 26.15 (2005), pp. 2483–2489 (cit. on p. 61).

[82] Hansheng Lei, Srinivas Palla, and Venu Govindaraju. "Mouse-based signature verification for secure Internet transactions". In: *Applications of Neural Networks and Machine Learning in Image Processing IX*. Vol. 5673. International Society for Optics and Photonics. 2005, pp. 153–160 (cit. on p. 28).

[83] Lingjun Li, Xinxin Zhao, and Guoliang Xue. "Unobservable re-authentication for smartphones." In: *NDSS*. Vol. 56. Citeseer. 2013, pp. 57–59 (cit. on p. 36).

[84] Stan Z Li and Anil Jain. *Encyclopedia of biometrics*. Springer Publishing Company, Incorporated, 2015, pp. 1382–1387 (cit. on pp. 65–67).

[85] M. Liwicki et al. "Signature Verification Competition for Online and Offline Skilled Forgeries (SigComp2011)". In: *2011 International Conference on Document Analysis and Recognition*. Sept. 2011, pp. 1480–1484 (cit. on pp. 17, 19, 24, 29, 30, 36).

[86] Marte Dybevik Loge. "Tell me who you are and i will tell you your unlock pattern". MA thesis. NTNU, 2015 (cit. on pp. 5, 6).

[87] Paul Maergner et al. "Combining graph edit distance and triplet networks for offline signature verification". *Pattern Recognition Letters* 125 (2019), pp. 527–533 (cit. on p. 14).

[88] Fahad Malallah et al. "Online Handwritten Signature Recognition by Length Normalization using Up- Sampling and Down-Sampling". *International Journal of Cyber-Security and Digital Forensics (IJCSDF)* 4 (Dec. 2014), pp. 302–313 (cit. on pp. 19, 69).

[89] M. I. Malik et al. "ICDAR 2013 Competitions on Signature Verification and Writer Identification for On- and Offline Skilled Forgeries (SigWiComp 2013)". In: *2013 12th International Conference on Document Analysis and Recognition*. Aug. 2013, pp. 1477–1483 (cit. on pp. 16, 19, 24, 29, 30, 36).

[90] Muhammad Imran Malik et al. "ICDAR2015 Competition on Signature Verification and Writer Identification for On- and Off-line Skilled Forgeries ( SigWIcomp2015 )". In: Jan. 2015, pp. 1186–1190 (cit. on pp. 16, 17, 19, 29, 30, 36, 47).

[91] Jani 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. IEEE. 2005, pp. ii–973 (cit. on pp. 8, 17).

[92] John K Martinelli et al. *Force sensing semiconductive touchpad.* US Patent 5,943,044. Aug. 1999 (cit. on p. 47).

[93] Marcos Martinez Diaz et al. "Dynamic signature verification for portable devices". MA thesis. 2008 (cit. on pp. 12, 15, 20, 22, 23, 27, 61, 62).

[94] Marcos Martinez-Diaz et al. "Mobile Signature Verification: Feature Robustness and Performance Comparison". *IET Biometrics* 3 (May 2014) (cit. on pp. 11, 12, 15–18, 20, 22, 23, 27, 29, 30, 32, 38, 39, 59, 61).

[95] Marcos Martinez-Diaz et al. "Towards mobile authentication using dynamic signature verification: useful features and performance evaluation". In: *2008 19th International Conference on Pattern Recognition.* IEEE. 2008, pp. 1–5 (cit. on p. 32).

[96] Mark Mattingley-Scott and Michael Pyschny. *Method to validate the identity of a user of a mobile computer and mobile computer.* US Patent App. 11/280,112. June 2006 (cit. on p. 28).

[97] Aitor Mendaza-Ormaza et al. "Analysis of handwritten signature performances using mobile devices". In: *2011 Carnahan Conference on Security Technology.* IEEE. 2011, pp. 1–6 (cit. on pp. 19, 20, 32, 65, 67, 69).

[98] Weizhi Meng et al. "TMGuard: a touch movement-based security mechanism for screen unlock patterns on smartphones". In: *International Conference on Applied Cryptography and Network Security.* Springer. 2016, pp. 629–647 (cit. on p. 8).

[99] Thomas S Messerges, Ezzy A Dabbish, and Robert H Sloan. "Investigations of Power Analysis Attacks on Smartcards." *Smartcard* 99 (1999), pp. 151–161 (cit. on p. 8).

[100] Muhammad Muaaz and Rene Mayrhofer. "Smartphone-based gait recognition: From authentication to imitation". *IEEE Transactions on Mobile Computing* 16.11 (2017), pp. 3209–3221 (cit. on p. 9).

[101] Collin Mulliner et al. "SMS-based one-time passwords: attacks and defense". In: *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment.* Springer. 2013, pp. 150–159 (cit. on p. 8).

[102] Kiran P Nagbhidkar and PV Bagdi. "Online signature verification on smart phone using discrete wavelet transforms". *IORD Journal of Science & Technology* 2.2 (2015), pp. 1–6 (cit. on p. 12).

[103] Choonsung Nam and DongRyeol Shin. "Force-touch measurement methodology based on user experience". *International Journal of Distributed Sensor Networks* 14.4 (2018), p. 1550147718767794. eprint: https://doi.org/10.1177/1550147718767794. URL: https://doi.org/10.1177/1550147718767794 (cit. on pp. 16, 47).

[104] Seungsoo Nam et al. "Forged Signature Distinction Using Convolutional Neural Network for Feature Extraction". *Applied Sciences* 8.2 (2018), p. 153 (cit. on pp. 11, 15, 18, 21, 22, 29–33, 36, 47, 63, 64, 69).

[105] Shu Nishiguchi et al. "Reliability and validity of gait analysis by android-based smartphone". *Telemedicine and e-Health* 18.4 (2012), pp. 292–296 (cit. on p. 8).

[106] Luiz S. Oliveira et al. "The graphology applied to signature verification". In: 2005 (cit. on p. 14).

[107] Javier Ortega-Garcia et al. "MCYT baseline corpus: a bimodal biometric database". *IEE Proceedings-Vision, Image and Signal Processing* 150.6 (2003), pp. 395–401 (cit. on p. 26).

[108] M Erdal Özbek, Mohamed Amine Haytom, and Estelle Cherrier. "Recognition of biometric unlock pattern by GMM-UBM". In: *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE. 2018, pp. 1–4 (cit. on p. 8).

[109] Emre Özgündüz, Tülin Şentürk, and M Elif Karslıgil. "Off-line signature verification and recognition by support vector machine". In: *2005 13th European Signal Processing Conference*. IEEE. 2005, pp. 1–4 (cit. on p. 14).

[110] Pascal Paillier. "Public-key cryptosystems based on composite degree residuosity classes". In: *International conference on the theory and applications of cryptographic techniques*. Springer. 1999, pp. 223–238 (cit. on p. 21).

[111] Nilakantha Paudel, Marco Querini, and Giuseppe Italiano. "Handwritten Signature Verification for Mobile Phones". In: Jan. 2016, pp. 46–52 (cit. on pp. 24, 65).

[112] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on pp. 69, 80).

[113] Philip Quinn. "Estimating Touch Force with Barometric Pressure Sensors". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. New York, NY, 2019, 689:1–689:7. URL: https://dx.doi.org/10.1145/3290605.3300919 (cit. on p. 16).

[114] Hannes Rantzsch, Haojin Yang, and Christoph Meinel. "Signature embedding: Writer independent offline signature verification with deep metric learning". In: *International symposium on visual computing*. Springer. 2016, pp. 616–625 (cit. on p. 14).

[115] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. "Enhancing security and privacy in biometrics-based authentication systems". *IBM systems Journal* 40.3 (2001), pp. 614–634 (cit. on pp. 7–9).

[116] Yanzhi Ren et al. "User verification leveraging gait recognition for smartphone enabled mobile healthcare systems". *IEEE Transactions on Mobile Computing* 14.9 (2014), pp. 1961–1974 (cit. on p. 8).

[117] Jonas Richiardi, Hamed Ketabdar, and Andrzej Drygajlo. "Local and global feature selection for on-line signature verification". In: *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. IEEE. 2005, pp. 625–629 (cit. on p. 61).

[118] Aditi Roy, Nasir Memon, and Arun Ross. "Masterprint: Exploring the vulnerability of partial fingerprint-based authentication systems". *IEEE Transactions on Information Forensics and Security* 12.9 (2017), pp. 2013–2025 (cit. on p. 7).

[119] N. Sae-Bae and N. Memon. "Online Signature Verification on Mobile Devices". *IEEE Transactions on Information Forensics and Security* 9.6 (June 2014), pp. 933–947 (cit. on pp. 10, 15, 25, 26, 30).

[120] Kei Sakamoto and Yojiro Tonouchi. *Information processing apparatus and signature data input programs.* US Patent 7,426,288. Sept. 2008 (cit. on p. 29).

[121] Hiroaki Sakoe and Seibi Chiba. "Dynamic programming algorithm optimization for spoken word recognition". *IEEE transactions on acoustics, speech, and signal processing* 26.1 (1978), pp. 43–49 (cit. on pp. 65, 66).

[122] Raul Sanchez-Reillo. "Signature analysis in the context of mobile devices". *Image and Vision Computing* 55 (2016). Recognizing future hot topics and hard problems in biometrics research, pp. 34–37. URL: http://www.sciencedirect.com/science/article/pii/S0262885616300476 (cit. on pp. 3, 10–12, 15, 33, 35, 103).

[123] 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.* 2012, pp. 1–10 (cit. on p. 6).

[124] Stuart Schechter, Cormac Herley, and Michael Mitzenmacher. "Popularity is everything: A new approach to protecting passwords from statistical-guessing attacks". In: *Proceedings of the 5th USENIX conference on Hot topics in security.* 2010, pp. 1–8 (cit. on p. 6).

[125] Pavel Senin. "Dynamic time warping algorithm review". *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA* 855.1-23 (2008), p. 40 (cit. on p. 65).

[126] Muhammad Sharif et al. "A framework for offline signature verification system: Best features selection approach". *Pattern Recognition Letters* (2018) (cit. on p. 14).

[127] B. H. Shekar et al. "Grid structured morphological pattern spectrum for off-line signature verification". In: *2015 International Conference on Biometrics (ICB).* 2015, pp. 430–435 (cit. on p. 14).

[128] D. Simsons, R. Spencer, and S. Auer. "The Effects of Constraining Signatures". *Journal of the American Society of Questioned Document Examiners* 14 (June 2011), pp. 39–50 (cit. on p. 23).

[129] Zdeňka Sitová et al. "HMOG: New behavioral biometric features for continuous authentication of smartphone users". *IEEE Transactions on Information Forensics and Security* 11.5 (2015), pp. 877–892 (cit. on pp. 8, 17, 18, 30, 35, 36, 70, 97).

[130] Amir Soleimani, Babak N Araabi, and Kazim Fouladi. "Deep multitask metric learning for offline signature verification". *Pattern Recognition Letters* 80 (2016), pp. 84–90 (cit. on p. 14).

[131] Furkan Tari, A Ant Ozok, and Stephen H Holden. "A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords". In: *Proceedings of the second symposium on Usable privacy and security.* 2006, pp. 56–66 (cit. on p. 6).

[132] Matthew Dominic Tenuta. *Capacitive and resistive-pressure touch-sensitive touchpad*. US Patent 9,619,044. Apr. 2017 (cit. on p. 47).

[133] Ruben Tolosana et al. "Benchmarking desktop and mobile handwriting across COTS devices: The e-BioSign biometric database". *PloS one* 12.5 (2017), e0176792 (cit. on p. 20).

[134] Ruben Tolosana et al. "DeepSign: Deep on-line signature verification". *arXiv preprint arXiv:2002.10119* (2020) (cit. on pp. 22, 27).

[135] Ruben Tolosana et al. "Exploiting complexity in pen-and touch-based signature biometrics". *International Journal on Document Analysis and Recognition (IJDAR)* (2020), pp. 1–13 (cit. on pp. 20–22, 27, 61, 64, 89).

[136] Ruben Tolosana et al. "Exploring recurrent neural networks for on-line handwritten signature biometrics". *Ieee Access* 6 (2018), pp. 5128–5138 (cit. on pp. 22, 30).

[137] Ruben Tolosana et al. "Reducing the template ageing effect in on-line signature biometrics". *IET Biometrics* 8.6 (2019), pp. 422–430 (cit. on pp. 9, 10, 27, 28, 30, 71, 85, 86).

[138] Sebastian Uellenbeck et al. "Quantifying the security of graphical passwords: the case of android unlock patterns". In: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 2013, pp. 161–172 (cit. on p. 5).

[139] Umut Uludag and Anil K Jain. "Attacks on biometric systems: a case study in fingerprints". In: *Security, steganography, and watermarking of multimedia contents VI*. Vol. 5306. International Society for Optics and Photonics. 2004, pp. 622–633 (cit. on p. 7).

[140] Blase Ur et al. "Do users' perceptions of password security match reality?" In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016, pp. 3748–3760 (cit. on p. 6).

[141] Bao Ly Van, Sonia Garcia-Salicetti, and Bernadette Dorizzi. "On using the Viterbi path along with HMM likelihood information for online signature verification". *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37.5 (2007), pp. 1237–1247 (cit. on p. 61).

[142] Jesús Francisco Vargas-Bonilla et al. "Off-line Handwritten Signature GPDS-960 Corpus". *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)* 2 (2007), pp. 764–768 (cit. on p. 14).

[143] Ignacio Velasquez, Angelica Caro, and Alfonso Rodriguez. "Authentication schemes and methods: A systematic literature review". *Information and Software Technology* 94 (2018), pp. 30–37 (cit. on pp. 4, 8).

[144] R. Vera-Rodriguez et al. "E-biosign: stylus- and finger-input multi-device database for dynamic signature recognition". In: *3rd International Workshop on Biometrics and Forensics (IWBF 2015)*. Mar. 2015, pp. 1–6 (cit. on p. 20).

[145] Emanuel Von Zezschwitz et al. "Making graphic-based authentication secure against smudge attacks". In: *Proceedings of the 2013 international conference on Intelligent user interfaces*. 2013, pp. 277–286 (cit. on p. 6).

[146] Ding Wang and Ping Wang. "On the implications of Zipf's law in passwords". In: *European Symposium on Research in Computer Security*. Springer. 2016, pp. 111–131 (cit. on p. 6).

[147] Cong Wu et al. "Liveness is Not Enough: Enhancing Fingerprint Authentication with Behavioral Biometrics to Defeat Puppet Attacks". In: *29th {USENIX} Security Symposium ({USENIX} Security 20)*. 2020 (cit. on p. 7).

[148] Dit-Yan Yeung et al. "SVC2004: First international signature verification competition". In: *International conference on biometric authentication*. Springer. 2004, pp. 16–22 (cit. on pp. 16, 17, 19, 29, 30).

[149] Mustafa Berkay Yılmaz and Berrin Yanıkoğlu. "Score level fusion of classifiers in off-line signature verification". *Information Fusion* 32 (2016), pp. 109–119 (cit. on p. 14).

[150] F. J. Zareen and S. Jabin. "Authentic mobile-biometric signature verification system". *IET Biometrics* 5.1 (2016), pp. 13–19 (cit. on pp. 17, 31).

[151] H. Zhang, X. Liu, and C. Chen. "An online mobile signature verification system based on homomorphic encryption". *International Journal of Innovative Computing, Information and Control* 13 (Oct. 2017), pp. 1623–1636 (cit. on pp. 12, 21, 68, 89).

[152] Ziming Zhao, Gail-Joon Ahn, and Hongxin Hu. "Picture gesture authentication: Empirical analysis, automated attacks, and scheme evaluation". *ACM Transactions on Information and System Security (TISSEC)* 17.4 (2015), pp. 1–37 (cit. on p. 5).

[153] Ziming Zhao et al. "On the security of picture gesture authentication". In: *22nd USENIX Security Symposium (USENIX Security 13)*. 2013, pp. 383–398 (cit. on pp. 5, 6).

[154] Jiang Zhu et al. "Sensec: Mobile security through passive sensing". In: *2013 International Conference on Computing, Networking and Communications (ICNC)*. IEEE. 2013, pp. 1128–1133 (cit. on pp. 8, 17, 30, 35, 36).

[155] Moshe Zviran and Zippy Erlich. "Identification and authentication: technology and implementation issues". *Communications of the Association for Information Systems* 17.1 (2006), p. 4 (cit. on p. 5).

## Online sources

[156] Alex Alderson. *Huawei MateBook X: August 19-bound premium ultrabook to feature an Apple-inspired FreeTouch trackpad*. Aug. 2020. URL: https://www.notebookcheck.net/Huawei-MateBook-X-August-19-bound-premium-ultrabook-to-feature-an-Apple-inspired-FreeTouch-trackpad.486587.0.html (visited on 08/15/2020) (cit. on p. 47).

[157] Google Developers. *API Reference*. 2020. URL: https://developer.android.com/reference (visited on 08/13/2020) (cit. on pp. 35, 36, 61, 73).

[158] Synaptics Incorporated. *Who We Are*. 2020. URL: https://www.synaptics.com/company/overview (visited on 08/15/2020) (cit. on p. 47).

[159] Zach Pace. *Signing in with a picture password*. Dec. 2011. URL: https://docs.microsoft.com/en-us/archive/blogs/b8/signing-in-with-a-picture-password (visited on 09/23/2020) (cit. on p. 6).

[160] Anand Lal Shimpi. *Synaptics ForcePad: A Thinner, Pressure Sensitive Trackpad for Notebooks*. Aug. 2020. URL: https://www.anandtech.com/show/6171/synaptics-forcepad-a-thinner-pressure-sensitive-trackpad-for-notebooks (visited on 08/15/2020) (cit. on p. 47).

[161] Elena Tsiporkova. *Dynamic Time Warping Algorithm*. 2013. URL: http://www.mathcs.emory.edu/~lxiong/cs730_s13/share/slides/searching_sigkdd2012_DTW.pdf (visited on 09/08/2020) (cit. on p. 66).

# Check Final Print Size

— Check final print size! —

width = 100mm
height = 50mm

— Remove this page after printing! —