Authentication in Systems with Limited Input Capabilities



Jack Sturgess

Kellogg College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2023

For my parents, Andrew and Christine, who supported me every step of the way, and in loving memory of my nan, Kathleen.

With special thanks to Klaudia Krawiecka and Sebastian Köhler, with whom it was a pleasure to collaborate.

Abstract

The popularity of smart devices—whether hand-held, wearable, or deployed in an environment—has risen sharply in recent years. Users want convenient, remote access on multiple devices to a multitude of services, from sending emails, to administering their smart home, to making mobile payments. These devices typically lack traditional input capabilities, such as a mouse and a keyboard, and their small form factor limits the ways in which users can interact with them. This presents new challenges for user authentication.

Smart devices do, however, offer other sensors that can be leveraged to authenticate the user in non-standard ways. In particular, behavioural biometrics, such as gait and keystroke dynamics, which had once been rendered impractical before ubiquitous sensors became readily available, now present promising opportunities to authenticate users continuously without requiring any effort from the user. Smart devices also facilitate new forms of automatic de-authentication and mutual authentication in scenarios that were previously implausible.

In this thesis, we focus primarily on the use of inertial sensors and the context of mobile payments. First, as mobile payment applications have become available on smartwatches, we show that inertial sensors on a smartwatch can be used to authenticate the user as he makes a payment. Furthermore, we perform gesture recognition on wrist motion data to infer whether such a payment was intentional (as opposed to accidental or as part of a relay attack). Second, motivated by a device with even fewer input capabilities, we explore the use of inertial sensors on a smart ring for authentication purposes. We demonstrate that finger (or wrist) motion data can be used to authenticate payments made with either a smartwatch or a smart ring and we show that the data can also be used to authenticate the user when knocking on a door for access control purposes. Third, we show that inertial sensors embedded in the handle of an electric vehicle charger can be used to authenticate the wielder and reduce the number of times that he is required to authorise a charging session manually on his smartphone. Last, motivated by the threats posed by malicious payment terminals and the growing need for specialised hardware, we design and evaluate a novel mobile payment system that uses a visual channel to achieve mutual authentication between a user and a generic terminal, where the terminal authenticates itself to the user before the user is asked to reveal any secret information.

Contents

List of Figures **xi**

List of Tables **xiii**

List of Abbreviations **xv**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Contribution . . . 3
1.3 Scope . . . 5
1.4 Ethical Considerations . . . 5
1.5 Thesis Outline . . . 5

2 Background **7**
2.1 Authentication Systems . . . 7
2.2 Password-based Authentication . . . 8
2.3 Biometric-based Authentication . . . 10
2.3.1 Attacks against Biometric Systems . . . 11
2.4 Authentication in Systems with Limited Input Capabilities . . . 12
2.4.1 Continuous Authentication . . . 12
2.4.2 Browser-based Systems . . . 12
2.4.3 Smartphone-based Systems . . . 13
2.4.4 Wearable-based Systems . . . 15
2.4.5 Smart Home-based Systems . . . 17
2.5 Authentication in Mobile Payment Systems . . . 18
2.5.1 Attacks against Mobile Payment Systems . . . 19

3 WatchAuth: User Authentication and Intent Recognition with a Smartwatch **21**
3.1 Motivation . . . 21
3.2 System Design . . . 23
3.2.1 Design Goals . . . 23
3.2.2 Design Considerations . . . 23
3.2.3 System Model . . . 26
3.2.4 Threat Model . . . 27
3.3 Experiment Design . . . 29

3.3.1 Experiment Overview . . . 29
3.3.2 Point-of-sale Terminals . . . 29
3.3.3 Sensor Module . . . 30
3.3.4 User Study . . . 31
3.4 Methods . . . 33
3.4.1 Data Processing . . . 33
3.4.2 Feature Extraction . . . 34
3.4.3 Classification . . . 35
3.4.4 Performance Metrics . . . 37
3.5 Results . . . 40
3.5.1 Anatomy of a Tap Gesture . . . 40
3.5.2 Zero-effort Attack . . . 41
3.5.3 Informed Attack . . . 43
3.5.4 Observation Attack . . . 44
3.5.5 Feature Informativeness . . . 45
3.5.6 Sensor Selection . . . 47
3.5.7 Terminal Positions . . . 49
3.5.8 Misclassification by Activity Type . . . 50
3.5.9 Enrolment Parameters . . . 51
3.5.10 Stability over Time . . . 52
3.5.11 Feasibility as an Additional Factor . . . 53
3.5.12 Privacy Implications . . . 55
3.6 Discussion . . . 57
3.7 Related Work . . . 59
3.8 Conclusion . . . 61

4 RingAuth: User Authentication with a Smart Ring 63
4.1 Motivation . . . 63
4.2 System Design . . . 65
4.2.1 Design Goals . . . 65
4.2.2 Design Considerations . . . 66
4.2.3 System Model . . . 66
4.2.4 Threat Model . . . 67
4.3 Experiment Design . . . 67
4.3.1 Experiment Overview . . . 67
4.3.2 Point-of-sale Terminals . . . 69
4.3.3 Wearable Devices and Sensor Modules . . . 69
4.3.4 Gesture Types . . . 70
4.3.5 User Study . . . 70
4.4 Methods . . . 71

4.4.1 Data Processing . . . 71
4.4.2 Feature Extraction . . . 76
4.4.3 Classification . . . 76
4.4.4 Performance Metrics . . . 77
4.5 Results . . . 77
4.5.1 Zero-effort Attack . . . 77
4.5.2 Observation Attack . . . 80
4.5.3 Feature Informativeness . . . 83
4.5.4 Sensor Selection . . . 86
4.5.5 Terminal Positions . . . 87
4.5.6 Enrolment Parameters . . . 88
4.5.7 Privacy Implications . . . 89
4.6 Discussion . . . 91
4.7 Related Work . . . 92
4.8 Conclusion . . . 94

5 CableAuth: User Authentication over a Physical Channel 95
5.1 Motivation . . . 95
5.2 System Design . . . 97
5.2.1 Design Goals . . . 97
5.2.2 Design Considerations . . . 97
5.2.3 System Model . . . 98
5.2.4 Threat Model . . . 99
5.3 Experiment Design . . . 100
5.3.1 Experiment Overview . . . 100
5.3.2 Cable Enclosures . . . 101
5.3.3 Devices and Sensor Modules . . . 101
5.3.4 User Study . . . 102
5.4 Methods . . . 103
5.4.1 Data Processing . . . 103
5.4.2 Feature Extraction . . . 103
5.4.3 Classification . . . 105
5.4.4 Performance Metrics . . . 105
5.5 Results . . . 106
5.5.1 Zero-effort Attack . . . 106
5.5.2 Feature Informativeness . . . 107
5.5.3 Composite Model . . . 108
5.5.4 Feasibility as an Additional Factor . . . 109
5.6 Discussion . . . 110
5.7 Related Work . . . 111
5.8 Conclusion . . . 111

6 VisAuth: Mutual Authentication over a Visual Channel 113
6.1 Motivation . . . 113
6.2 System Design . . . 115
6.2.1 Design Goals . . . 115
6.2.2 System Requirements . . . 115
6.2.3 System Model . . . 116
6.2.4 Threat Model . . . 117
6.3 System Architecture . . . 118
6.3.1 Cryptographic Materials . . . 118
6.3.2 Biometric-based Identification . . . 120
6.3.3 Authentication Protocol . . . 121
6.4 Security Analysis . . . 123
6.5 Discussion . . . 125
6.6 Related Work . . . 126
6.7 Conclusion . . . 128

7 Summary 131

Bibliography 133

List of Figures

3.1 The equipment used in our WatchAuth experiment. 28
3.2 Two examples of point-of-sale terminals. 29
3.3 The demographics of participants in our user study. 32
3.4 Plots of a tap gesture. 34
3.5 Plot showing EER. 39
3.6 Average EERs for our authentication and intent recognition models. 41
3.7 Average ROC curves for our authentication and intent recognition models. 42
3.8 Results of our observation attack against our *position-known* authentication model. 44
3.9 Average EERs for our authentication and intent recognition models by sensors. 48
3.10 Average EERs for our authentication model if trained on different numbers of enrolment samples. 51
3.11 Average EERs for our authentication model for users who participated in both rounds of our user study. 52
3.12 Average FARs and FRRs when optimised for our authentication and intent recognition models. 53
3.13 Average accuracy scores for our sex inference models. 56
3.14 Average R^2 scores for our height inference models. 56

4.1 The equipment used in our RingAuth experiment. 68
4.2 The attacker's restricted view of the victim. 68
4.3 Plots of a ring tap gesture 73
4.4 Plots of a watch tap gesture 74
4.5 Plots of a secret knock gesture 75
4.6 Average EERs for our payment models. 78
4.7 Average ROC curves for our payment models. 78
4.8 Average ROC curves for our access control models. 80
4.9 Results of our observation attack against our *position-known* payment models. 81
4.10 Results of our observation attack against our access control models. 82
4.11 Average EERs for our payment models by sensors. 85
4.12 Average EERs for our payment and access control models if trained on different numbers of enrolment samples. 88

4.13 Average accuracy scores for our sex inference models. 89
4.14 Average R^2 scores for our height inference models. 90

5.1 The equipment used in our CableAuth experiment. 100
5.2 Plots of an unhook gesture. 104
5.3 Average EERs for our unhook and plug-in models. 106
5.4 Average ROC curves for our unhook and plug-in models. 107
5.5 Average FARs and FRRs when optimised for our unhook model. . . 109

6.1 The system model of our scheme during the authentication phase. . 117
6.2 The authentication protocol of our scheme. 122

List of Tables

3.1 Details of the terminals used in our WatchAuth experiment. 28
3.2 Average FARs by height similarity for our authentication model. . . 43
3.3 Modal top-five features for our authentication and intent recognition models. 46
3.4 Average EERs for our *position-specific* authentication and intent recognition models. 49
3.5 Average FARs (tuned to the EER) by non-tap gesture type in our intent recognition model. 50
3.6 Average R^2 scores for our height inference models by gesture type. . 57

4.1 Average EERs for our access control models. 79
4.2 Modal top-five features for our payment and access control models. 84
4.3 Average EERs for our access control models by sensors. 86
4.4 Average EERs for our *position-specific* payment models. 87

5.1 Average EERs for our composite model. 109

6.1 Summary of the cryptographic materials used in our scheme. 119

List of Abbreviations

2FA . . . . . Two-factor Authentication

3D . . . . . Three-dimensional

AUC . . . . . Area Under the (ROC) Curve

Bluetooth LE Bluetooth Low Energy

DC . . . . . Direct Current

ECG . . . . . Electrocardiography

EER . . . . . Equal Error Rate

EMV . . . . . Europay, Mastercard, and Visa

EV . . . . . Electric Vehicle

FAR . . . . . False Acceptance Rate

FIDO . . . . . Fast Identity Online

FRR . . . . . False Rejection Rate

GPS . . . . . Global Positioning System

GRV . . . . . Gyroscope Rotation Vector

IDE . . . . . Integrated Development Environment

IMU . . . . . Inertial Measurement Unit

LED . . . . . Light-emitting Diode

LSB . . . . . Least Significant Bit

MAC . . . . . Media Access Control

MEMS . . . . Microelectomechanical System

MFA . . . . . Multi-factor Authentication

MIDI . . . . . Musical Instrument Digital Interface

MITM . . . . . Man-in-the-middle

NFC . . . . . Near-field Communication

PIN . . . . . Personal Identification Number

PPG . . . . . Photoplethysmography

PSD2 . . . . . Updated Payment Services Directive

QR Code . . . Quick Response Code

ROC . . . . . Receiver Operating Characteristic

SMS . . . . . Short Message Service

SSO . . . . . Single Sign-on

TLS . . . . . Transport Layer Security

U2F . . . . . Universal Second Factor

UAF . . . . . Universal Authentication Framework

1 Introduction

1.1 Motivation

Passwords are the primary mechanism by which users authenticate themselves to digital services. However, passwords suffer from many shortcomings in terms of security and usability. To be resistant against brute force guessing attacks, a password must be sufficiently long and random, which makes it slower to enter and harder to remember [151]. Password re-use across multiple accounts is common, which means the compromise of one can lead to the compromise of many [27, 135]. Password managers can mitigate some of these problems, but present a single point of failure, require the user to trust the service, and are not immune to faults [136]. The legitimacy of passwords as a means to guarantee the authenticity of a user is further undermined by poor implementations, such as service providers storing passwords in plaintext [43], the availability of leaked credentials that bolster future brute force attacks, social engineering, phishing, and keylogging malware.

Biometric recognition is a promising alternative that offers solutions to many of the problems posed by the use of passwords. The initial barriers to adoption that biometric systems faced, such as high deployment costs and error rates, have evaporated in recent years as smartphones that support various biometric sensors have become readily available. Biometrics can be split into two categories: physiological and behavioural. Physiological biometrics measure physical characteristics, such as fingerprints and face geometry, and are the most commonly found in modern

smartphones. These characteristics can be captured quickly as the user interacts with the sensor. However, they can also be captured by an attacker and, unlike passwords, when compromised they cannot be changed or revoked. Attention was drawn to this problem in 2014, when attackers famously captured a German politician's fingerprints using only high-definition photographs as she delivered a speech from a lectern [53]. With a stolen physical biometric, an attacker can execute a presentation attack, such as using a fake, 3D-printed finger or a thin silicon mould placed over the tip of a real finger [40]. Behavioural biometrics measure patterns of movement, such as gait and keystroke dynamics. These characteristics are measured over time and without any effort required from the user; they are also significantly more difficult for an attacker to capture. The user's behaviour will naturally change over time, which means that the system will need to be updated, but this is also advantageous, as any samples that an attacker may have captured will gradually become out-of-date. Behavioural biometrics were impractical before embedded sensors became pervasive. The use of wearable devices, such as smartwatches, has provided even more opportunity to measure user behaviours over time. The unobtrusive collection of these characteristics facilitates continuous authentication, where the user's identity is checked frequently. This opens up new possibilities, such as automatic de-authentication whenever a user stops interacting with the system or leaves the room [1, 74, 88, 89].

As the number and heterogeneity of devices increases and new digital services emerge, so the nature of authentication must adapt: new contexts that require authentication, such as mobile payments and remote administration of a smart home, and new devices that require access to services, such as smartwatches and smart rings, with greatly varying sensor offerings and input capabilities, mandate the pursuit of new, strong forms of authentication.

In this thesis, we investigate how a number of devices with limited input capabilities can be used for authentication purposes with a particular focus on mobile payment systems. We explore the challenges that payment systems face in supporting a wide range of heterogeneous devices, both in terms of hardware and in terms of establishing trust between users and nondescript payment terminals.

1.2 Contribution

Here, we summarise the peer-reviewed publications that collectively form the foundation of this thesis. We have included only those published works for which I was the lead author and the main contributor (except for the third, as detailed below).

- As mobile payment systems have become available on smartwatches, a number of challenges have arisen in terms of user authentication and intent recognition. We hypothesised that the inertial sensors on the watch would be able to provide a solution in both cases. We conducted a user study to collect inertial sensor data as users wore a watch and made payments at real-world point-of-sale terminals. The results of this work were published at the *IEEE 7th European Symposium on Security and Privacy (EuroS&P)* in 2022 [139]. An ancillary paper considering the privacy implications of the work was published at the *IEEE 8th WristSense Workshop* at the *IEEE 20th International Conference on Pervasive Computing and Communication (PerCom Workshops)* in 2022 [138].

  We returned to this work over a year after the initial user study and performed another, using the same equipment. This enabled us to test our models on a larger user base, to invite back some of the original users to investigate the stability over time of the *tap gesture* biometric, and to expand our threat model. The results of this work were combined with the previous works into an extended version that is currently under review to be published. We made the datasets for this work publicly available [141, 145]. We also published an article on this work in the *Inspired Research* magazine [140].

- As hardware technology continues to miniaturise, smart rings have started to emerge as viable devices for a number of applications. We compare the effectiveness of using inertial sensors for user authentication purposes on a ring with those on a smartwatch worn on the same arm when performing various gestures using each of the devices. We conducted a user study to collect inertial sensor data in both payment and access control contexts.

The results of this work are currently available in preprint [142] and are under review to be published. We made the dataset for this work publicly available [146]. We also published an article on this work in the *Inspired Research* magazine to share our research with a general readership [144].

- As electric vehicles have grown in popularity, a number of challenges have arisen in terms of user authentication. We hypothesised that inertial sensors, if they were to be embedded into the handle of the charging cable, could be used to authenticate the user during a charging session. We conducted a user study to collect inertial sensor data as users unhooked and plugged in a real-world charging cable to 3D-printed enclosures fabricated to match the charging socket of an electric vehicle. The results of this work were published at the *ISOC Symposium on Vehicle Security and Privacy (VehicleSec)* in 2023 [143]. We made the dataset for this work publicly available [147].

  In this work, Sebastian Köhler and I shared the title of lead author and we contributed equally. Sebastian provided valuable domain expertise in electric vehicles that helped to shape the design of the experiment and created the 3D-printed enclosures. I provided the data collection infrastructure and performed the data analysis using similar principles to those learned and applied in my previous works. We both designed and conducted the user study and prepared the paper for publication.

- As mobile payment systems have become prevalent, often being integrated into other systems, a number of challenges have arisen in terms of their attack surface and their dependence on certain technologies. We considered the common weaknesses of these systems and derived a set of requirements that future mobile payment systems should satisfy. We then presented a novel scheme that satisfies these requirements by using a visual channel. A short paper based on this work was published at the *16th International Conference on Cryptology and Network Security (CANS)* in 2017 [137]; a full version will be published shortly.

1.3 Scope

We consider a system to have *limited input capabilities* if it lacks for authentication purposes traditional input capabilities, such as a mouse or a keyboard, and those that have recently become commonplace, such as a fingerprint reader. We focus primarily on the use of inertial sensors due to their small size and low cost, which makes them a feasible addition to a wide range of heterogeneous devices.

We consider a system to be *optimised for usability* if it uses for authentication purposes implicit factors that do not require any effort from the user in such a way as to render unnecessary, and therefore reduce the number of, authentication actions required of the user. While we collect some general feedback from the participants in our user studies, and present it where relevant, it is not the aim of this work to explore qualitative usability studies that attempt to measure usability by any other metric.

1.4 Ethical Considerations

The collection of biometric data raises an ethical concern as the data collected from a participant in a user study could be used to impersonate that user at any system that uses that biometric modality. We address this concern by anonymising the data to make it difficult to map any biometric data to any particular user.

All of our user studies were reviewed by our Computer Science Departmental Research Ethics Committee and approvals were granted under reference numbers CS_C1A_19_043 to CS_C1A_19_043_005.

1.5 Thesis Outline

In Chapter 2, we present an overview of authentication topics and related works to place our work in context, with a particular emphasis on user authentication in systems with uncommon or limited input capabilities.

In Chapter 3, we outline the challenges of authenticating a user using a smartwatch as compared to smartphones in the context of mobile payments. We design

and conduct a user study to collect inertial sensor data as users make payments with a watch, we show that this data can be used to authenticate the user and recognise his intent to pay, both implicitly and simultaneously, and we evaluate the system against different attacks.

In Chapter 4, we investigate whether inertial sensors can be used to authenticate the user of a smart ring in the contexts of mobile payments and access control. We design and conduct a user study to collect inertial sensor data from both a ring and a smartwatch worn on the same arm as users make payments with each device and as they knock on a door. We compare the suitability of each device as its own primary authentication factor and as a secondary factor for the other and we evaluate the system against different attacks.

In Chapter 5, we identify that common payment schemes used in the context of electric vehicle charging authenticate the vehicle but not the user. We design and conduct a user study to collect inertial sensor data as users handle a charging cable, we show that this data can be used to authenticate the user implicitly, which could also prevent a thief from charging a stolen electric vehicle, and we evaluate the system as a second factor based on its ability to improve either the security or the usability of an existing authentication system.

In Chapter 6, we identify a number of common weaknesses in mobile payment systems and we derive a list of system requirements that encapsulate them. We present a novel mobile payment scheme that satisfies these requirements, including mutual authentication, by using biometric identification over a visual channel and we evaluate it against different attacks.

In Chapter 7, we summarise our findings and conclude the thesis.

2
Background

2.1 Authentication Systems

The purpose of user authentication is to verify the user's identity with as much confidence as possible. An exemplar framework for modern user authentication is given by the Updated Payment Services Directive (PSD2), introduced by the European Union in 2020 to overhaul payment regulations in its banking sector. One of its core principles, entitled Strong Customer Authentication, mandates the use of multi-factor authentication (MFA) for all payment transactions. To fulfil this, a system must verify the user's identity with at least two independent factors that are based on either *knowledge* (something only the user knows), *possession* (something only the user possesses), or *inherence* (something only the user is).

Each type of factor has well understood strengths and weaknesses. Knowledge-based factors, such as passwords, require the user to reveal to the system some secret information that he has memorised; these factors scale well, are cheap to implement and maintain in terms of resources, and can be changed easily to reset defences, but they must be remembered, can be forgotten or re-used by the user, and can be guessed or observed by an attacker. Possession-based factors, such as tokens, require the user to interact with or present to the system a device that he carries; these factors do not require any mental effort, but they must be carried around, can be lost or damaged by the user, and can be stolen, cloned, or forged by an attacker. Inherence-based factors, such as fingerprints, do not require any

mental effort nor anything to be carried around, but they can be captured and replayed or imitated by an attacker in various ways depending on the characteristic used. An ideal combination of factors complement one another so as to maximise the effort required of an attacker to compromise them all.

Authentication systems must balance *security* with *usability*. If a factor focuses too heavily on security and neglects usability, such that it requires a lot of effort of the user, then the user will strive to make it more usable with a possible cost to security—*e.g.*, if a password is too long and complicated for a user to remember, he might write it down and risk it being seen by an attacker. Factors that require some effort of the user are called *explicit factors*; this includes knowledge- and possession-based factors and those inherence-based factors that require the user to interact with a sensor, such as a fingerprint scanner, solely to have the characteristic captured. Factors that require no effort of the user are called *implicit factors*; this includes those inherence-based factors where the characteristic can be captured innately. The use of implicit factors can greatly improve the usability of a system by reducing the amount of effort the user is required to expend.

2.2 Password-based Authentication

For decades, passwords have been the primary mechanism by which users authenticate themselves to digital services. Efforts to harden password-based systems have resulted in the following four predominant strategies: user-centric design, password managers, single sign-on (SSO) systems, and two-factor authentication (2FA).

User-centric design, proposed by Adams & Sasse [2], advocates for usability via design principles that accommodate, rather than vilipend, human limitations—*e.g.*, to encourage stronger passwords, one should reduce the frequency of mandated password changes, otherwise users tend to view their choice as transitory and choose weaker passwords that are easier to memorise. Zhang-Kennedy *et al.* [165] surveyed the literature on passwords and suggested a set of revised guidelines to reflect realistic usage patterns. One such guideline was to categorise accounts by importance and to re-use passwords only for low-value accounts.

Password re-use has been observed persistently in user behaviour [12, 110, 135]. Password guessability is another common problem [34, 94]; Stobert & Biddle [135] found that users create passwords that have some form of significance to them and Wang & Wang [154] found that password strength meters, designed to nudge users towards stronger passwords, had inconsistent rules and failed to produce passwords that could withstand brute force guessing attacks. Password managers federate user credentials into a single store (often on a cloud server or integrated into a Web browser) such that a user need only remember a single password to access the rest. Lyastani *et al.* [85] found that the use of password managers encourages the use of password generators, which in turn reduces re-use and yields stronger passwords. The drawback of password managers is that they present a single point of failure and the user must trust the service provider.

SSO systems act as identity providers, managing user identity by issuing authentication tokens to service providers. The user authenticates once to the identity provider and that then authenticates him to other services. In principle, centralised authentication strategies can leverage strong tools, such as real-time threat and anomaly detection, with minimal user effort; however, attacks have been shown against the most common SSO standards, including SAML [133], OpenID [87, 157], and OAuth [148]. SSO systems also present a single point of failure; Mainka *et al.* [86] considered the impact of a malicious identity provider in OpenID and showed that all accounts on a service provider could be put at risk due to identity spoofing and key swapping attacks. Ghasemisharif *et al.* [47] proposed a 'single sign-off' extension to OpenID that would allow users to revoke access to all accounts associated with a compromised identity provider, to aid in recovery.

2FA systems strengthen an existing authentication system with a second factor, typically a number-generating token, that increases the difficulty for an attacker to impersonate the user to cloud services. Early variants used SMS messaging (however, this was shown to be vulnerable to spoofing attack [60]). 2FA systems are convenient to deploy because they can be added to existing systems without upheaval. Quermann *et al.* [113] provided a recent survey of 2FA systems used by popular websites and found that many lacked recovery mechanisms.

2.3 Biometric-based Authentication

A biometric is a measured characteristic of an individual that can be used to authenticate (or identify) that individual. Jain *et al.* [58] identified the following seven desiderata that a characteristic should have in order to be suitable as a biometric:

- *universality*: every person should have the characteristic;
- *uniqueness*: every person should be different in terms of the characteristic;
- *permanence*: the characteristic should not change over time;
- *collectability*: the characteristic should be measurable;
- *performance*: the characteristic should be able to distinguish between persons with sufficient accuracy, speed, and robustness;
- *acceptability*: the capture and use of the characteristic should be acceptable to every person; and
- *circumvention*: the characteristic should be difficult to imitate.

A biometric can be loosely categorised as either *physiological* or *behavioural*. Physiological biometrics are physical characteristics of the user, such as a fingerprint, and are measured in discrete actions. Behavioural biometrics are patterns of movements exhibited by the user, such as gait and voice, and are measured over time. If a biometric can be measured while the user engages in other tasks then it can be employed as an implicit factor.

In the initial *enrolment phase*, the system takes a number of biometric measurements, called *samples*, from the user. Features are extracted from the user's samples and encapsulated in a mathematical object called a *template*. In each subsequent *authentication phase*, in which the user attempts to authenticate himself to the system, a fresh sample is taken and features are extracted anew and matched against the template. The similarity between these features and the template is quantified into a confidence score; if this score exceeds a given threshold then the user is authenticated. Biometric systems can also be used to identify the user; here, the fresh sample is compared against all templates until a sufficiently near match is found.

In terms of PSD2, all biometrics are inherence-based factors. Unlike knowledge- and possession-based factors, biometrics cannot be shared readily between users due to the uniqueness desideratum; however, mimicry attacks can be feasible given sufficient resources and natural collisions can occur in large sets of users. Also unlike other factors, biometrics cannot be changed or revoked, so it is important for biometric systems to handle users' biometric data securely and to process it locally to protect against theft. Despite the permanence desideratum, biometrics tend to *drift* (*i.e.*, change naturally) over time. This can be advantageous, since any samples that an attacker may have captured will gradually become out-of-date. Lee *et al.* [76] demonstrated a mitigation technique for drift using an update mechanism to replace the user's old template with a new one by averaging it with the latest sample.

Behavioural biometrics tend to need longer authentication phases and a much longer enrolment phase than their physiological counterparts. This had rendered them impractical until recently, but now the availability of embedded sensors in smartphones—and, increasingly, wearable devices too—has overcome this challenge and provided new opportunities to measure the user over time.

2.3.1 Attacks against Biometric Systems

Presentation Attack. A presentation attack is where the attacker presents to the system a fake sample—*e.g.*, the attacker might capture the victim's characteristic and replay it or he might observe it and imitate him. This attack can be done to impersonate the victim. It can be countered by verifying the liveness of the sample and by using a well-trained classifier and a strongly unique characteristic.

Poisoning Attack. A poisoning attack is where the attacker exploits an unprotected update mechanism by presenting to it a series of fake samples that gradually update the template to include the attacker's likeness. This attack can be done to corrupt the victim's template so that the system wrongly accepts the attacker as the legitimate user [14, 84]. It can be countered by limiting the frequency of updates or by gating the update mechanism behind another factor.

2.4 Authentication in Systems with Limited Input Capabilities

2.4.1 Continuous Authentication

Implicit factors facilitate *continuous* authentication, where the unobtrusive nature of the factors enables the system to authenticate the user frequently to maximise its confidence in his identity at all times.

Continuous authentication provides resistance against presentation attacks by increasing the amount of effort required of the attacker (to impersonate the victim continuously). It also facilitates automatic de-authentication, where the user is logged out of the system as soon as he moves away or stops interacting with it. This addresses the so-called *lunchtime attack*, where the attacker takes advantage of the idle time before a timed log-out mechanism activates to access an unmanned workstation when the legitimate user is away temporarily.

Continuous authentication can operate in the background, without the user having to initiate it, which makes it ideal for systems with limited input capabilities. In smart homes, users may wish to switch between devices fleetingly, making a reduction in the number of authentication actions required of the user desirable.

Fuelled by the availability of multifarious embedded sensors in hand-held devices, wearable devices, and the smart home, recent works have demonstrated the use of a variety of factors, such as GPS location, usage habits, environmental conditions, proximity to other devices, and behavioural biometrics to authenticate the user continuously on Web browsers, smartphones, and wearable devices.

2.4.2 Browser-based Systems

Web browsers store and expose information that is capable of identifying the user. Browser fingerprinting makes use of Web-related elements, such as cookies and IP addresses, to identify users via their device. Laperdrix *et al.* [72] provided a survey of these techniques, highlighting their prolific usage in (unconsented) user tracking. Such techniques can also be used to support user authentication; this was evidenced by Facebook in a developer video [108].

Freeman *et al.* [45] proposed a system to strengthen password-based authentication in Web browsers by using location, browser information, and time of day to create a user profile. They provide strong adversarial analysis against evasion and poisoning attacks and validate their prototype against a large dataset.

2.4.3 Smartphone-based Systems

Smartphones offer a plethora of sensors that can be used to authenticate the user continuously to the device. Jakobsson *et al.* [59] proposed the first fully implicit system, which maintains a continuous trust score that decays over time and increases based on recognised activity, such as call history and battery usage. Kayacik *et al.* [64] proposed a system that scales with available factors and updates its user profile to account for drift; they used three public datasets containing different sensor data from 7, 35, and 100 subjects and achieved an average accuracy score of 0.95. Li *et al.* [78] used application usage to profile users and achieved an accuracy score of 0.89 in a user study of 76 participants. Fridman *et al.* [46] fused together four separate factors (stylometry, application usage, browser usage, and GPS location) and found that location was the most significant contributing factor in a user study of 200 participants over 30 days. Yao *et al.* [161] used usage patterns, such as browser and application usage, and showed that their system provides resistance against an informed attacker who knows the user's routines; they achieved an accuracy score of 0.95 in a user study of 5 participants. Some systems have used ambient sound as a factor [63, 156]; some have used touchscreen interaction [29, 44, 65].

Riva *et al.* [121] used the accelerometer and ambient contextual information, such as light and sound, to reduce the number of authentication and de-authentication actions required of the user by gauging the necessity of those actions based on the availability of other factors in the environment—they were able to reduce the number of unnecessary authentication actions required of the user by 42% in a user study of 9 participants. Hayashi *et al.* [52] used GPS location to select which (if any) authentication actions should be required of the user, reducing the number by 68% in the home (and having a stricter security setting for unfamiliar places) in a

user study of 36 participants. Gupta *et al.* [50] used contextual information to infer when the system was in a safe environment so as to adjust automatic log-out times accordingly. Miettine *et al.* [95] built on this further; their context-aware access control system reduced the number of unnecessary authentication actions by 70% in a user study of 15 participants. Hintze *et al.* [54] used public datasets to build a simulation model in which gait, voice, face recognition, and keystroke dynamics were used to authenticate users continuously; within their model, they simulated 4,494 subjects based on 807,195 days of real-world device usage data and reduced the number of unnecessary authentication actions by 99%.

The inertial measurement unit (IMU) of a device contains inertial sensors, such as an accelerometer and a gyroscope, that are commonly used in continuous authentication. Some works have used the IMU on a smartphone to authenticate the user. Conti *et al.* [22] proposed a system in which the user is authenticated as he answers or places a phone call; they had each user make 50 movements and used dynamic time warping to measure the similarity between the expected data and the received data; they achieved a FAR of 0.04 in a user study of 10 participants. Lee *et al.* [76] proposed an improved version with an adaptive learning component: the system focused on reducing actions whenever the user picks up the device and, if the implicit check fails but the explicit check passes, the implicit profile gets updated; they were able to reduce the number of unnecessary authentication actions required of the user by 33% in a user study of 24 participants over 7 days. Noor *et al.* [103] incorporated back-of-device capacitive sensors in a similar system—concluding, however, that little is gained from this versus using only the accelerometer. Shrestha *et al.* [131] used a number of sensors, but primarily the inertial sensors, to authenticate users as they made mobile payments with a smartphone; they achieved an F-measure score of 0.93 in a user study of 20 participants. Muaaz & Mayrhofer [97] used inertial sensors on a smartphone to authenticate the user by inferring gait.

As network connectivity becomes less of a barrier, some systems are making use of centralised, cloud-based services to process data gathered on a smartphone. CASTRA [130] uses location and inertial sensors to authenticate users based on

gait, proximity between the user and the device, and location history; the system dynamically changes the user's access rights based on real-time context awareness and risk assessment, unlocking access to critical tasks only when confidence is high enough. UnifyID [159] is a commercial cloud-based system claiming to use over 100 behavioural factors to replace passwords by using continuous authentication.

2.4.4 Wearable-based Systems

Some smart devices can be attached to the user's clothing, implanted inside the body, or worn as accessories. Zeng *et al.* [164] mounted five accelerometers at fixed positions on users' bodies and fused the data to authenticate them; they achieved an accuracy score of 0.97 in a user study of 30 participants. Inertial sensors have been deployed in head-mounted [79] and shoe-mounted [66] systems with similar success.

Smartwatches and fitness trackers can also be used for authentication and tracking purposes [7]. A number of systems have used data from the inertial sensors on a smartwatch to authenticate users that perform explicit gestures, such as hand movements [80, 160, 163] and finger-clicking [17], and by inferring gait [21, 61, 101].

Some works have used inertial sensors on a smartwatch to authenticate the user while hand-writing. Levy *et al.* [77] authenticated the user by hand-written signatures against a mimicry attacker who traced over them with a stylus on a screen; taking 15 signature samples from each user, they achieved an EER of 0.03 in a user study of 66 participants. Griswold-Steiner *et al.* [49] authenticated the user by general hand-writing against a mimicry attacker who watched a video of each victim before impersonating him; using 5 seconds of data from (only) the accelerometer, they achieved EERs of between 0.06 and 0.16 across various scenarios. Other works were able to identify users as they wrote [20, 48].

Instead of verifying the identity of the user, some systems verify the continued presence of a user in a user-agnostic manner. These systems assume that the user has been authenticated by some other factor and they perpetuate that state of authentication until given a reason to stop. Mare *et al.* [88] inferred a sequence of interactions from inertial sensors on a smartwatch and correlated it with a sequence

of workstation inputs to verify continuously that the wearer of the watch was interacting with the workstation, such that he could be de-authenticated automatically whenever the correlation broke (however, this was found to be vulnerable to mimicry because the attacker could control both streams [57]). Acar *et al.* [1] presented a more sophisticated system that considers keystroke dynamics, not just workstation inputs. Liang *et al.* [81] showed that inertial sensor data from a smart ring can be correlated with mouse movements at a workstation. Mare *et al.* [89] and Lee *et al.* [74] correlated inertial sensor data from a smartwatch and a smartphone to verify the continued interaction of the wearer with the phone.

Other systems operating on a smartwatch have adapted medical sensors for authentication purposes, such as infra-red to authenticate users by wrist veins [115], photoplethysmography for blood flow (since hand gestures compress arteries) [38, 127, 166], and electrocardiogram for heart rate [19, 116]. These systems have shown promise and some achieved EERs of as low as 0.02 in user studies of between 28 and 89 participants. Cornelius *et al.* [26] and Rasmussen *et al.* [118] investigated the use of bioimpedance (*i.e.*, the body's response to an electric pulse) for authentication purposes; the former used a wrist-worn device and included wrist circumference among their classification features and the latter used hand-held electrodes for data collection but envisaged that their system would be integrated into a secure payment terminal to provide a second factor to PIN entry by passing a current through the user every time a (metal) key on the keypad was pressed.

Headsets that support augmented, virtual, or mixed reality systems have used iris recognition for authentication purposes. Some works [35, 132, 167] have shown that eye movement tracking, whether captured naïvely or reflexively (as a response to a stimulus), can be used effectively to authenticate users continuously. Other headset-based modalities that have produced promising results include head-nodding (*e.g.*, to music) [79], passive body motion [102, 111], gait [128], and brain signal patterns using functional near-infra-red spectroscopy [126].

2.4.5 Smart Home-based Systems

Environment-wide systems that use connected smart devices to authenticate the user to a central system have been proposed over the years and are becoming less impractical to implement as smart technology evolves. Desmedt *et al.* [31] proposed a system based on threshold cryptography [30] where token fragments are shared among a number of factors (such as devices in the environment or chips embedded in clothing) such that a minimum number of these factors must be present simultaneously to authenticate the user. Pico [71, 134] built on this principle, proposing that one device contain encrypted secrets, such as credentials, and that the key fragments required to decrypt these secrets be spread across other nearby devices. In the proposed implementation, the decrypted secret is a password—this is cumbersome if used as a replacement for a single password, but may have promise if used as a password manager [3] or incorporated into a wider system. Hocking *et al.* [55] proposed an authentication aura, in which confidence in the identity of the user is continuously scored using information from surrounding tokens and devices, each individually weighted by the system based on its capability and independence to promote factor heterogeneity and increase the effort required of an impersonator; this score degrades over time, necessitating fresh readings and providing resistance to replay attacks, and scales well with the number of factors in the user's environment. Dasgupta *et al.* [28] proposed an adaptive system that chooses a subset of available factors based on environmental conditions to authenticate the user. One notable weakness of this system is that the attacker can influence factor selection by modifying the environment—*e.g.*, if the attacker has difficulty in impersonating the user's voice, he could create noise in the environment to cause the system to reject voice recognition from consideration.

FIDO Alliance [41] presented two families of protocols, Universal Second Factor (U2F) and Universal Authentication Framework (UAF), that enable client-side devices to support user authentication during a transaction with a Web server. U2F operates with a token, such as YubiKey, whereas UAF is a more general framework, supporting a multitude of devices from tokens to biometric readers.

Typically, each external factor stores its own user template(s) and independently authenticates the user with whatever capabilities it has. In both Hocking's and Dasgupta's works [28, 55], the integrity of factors (before and after enrolment) is assumed, leading to a risk of tampering if an adversary has access to any of the devices. In comparison, UAF requires devices to meet certification requirements and to support remote attestation for frequent integrity-checking, at the cost of limiting the number of devices that are compliant.

Agadakos *et al.* [4] used smart devices to locate and track users to determine if they could feasibly be where they claim to be in order to reject spurious authentication attempts. This system can be hosted in the cloud or on a smart hub and its range can be extended using beacons (third party devices) that report back whenever a user is detected. Similarly, Ashibani *et al.* [9] proposed a context-aware system that runs on a smart hub and to which the user submits his calendar during enrolment; the system uses location and time to disqualify unexpected authentication attempts.

Some works [23, 67, 129] have used the physical properties of wireless signals, and the effect that the human body has on their propagation, either to infer behaviours and profile the user or for de-authentication purposes.

Krawiecka *et al.* [69, 70] used inertial sensors and microphones attached to household fixtures to identify and authenticate users by interaction. Kaczmarek *et al.* [62] measured seated posture and weight distribution using sensors attached to a chair to verify the presence of the user continuously.

2.5 Authentication in Mobile Payment Systems

Mobile payment systems are especially relevant to our work, because authentication is of key importance in these systems and the devices used—whether they be payment cards, smartphones, *etc.*—are typically small, portable, and have limited input capabilities.

In recent years, the popularity of 2FA cashless payment systems has risen sharply. Consumers moved from cash to cashless payment cards primarily for

convenience, not security—the first generation of magnetic strip payment cards were authenticated with an easily-forged, hand-written signature. These systems were widely replaced with Europay, Mastercard, and Visa (EMV) payment card systems (also known as *Chip and PIN*), protected by a chip embedded into the card and a short numerical password called the personal identification number (PIN). Here, the card is inserted into a specialised point-of-sale terminal with an in-built keypad and the PIN is entered by the user and verified by the chip to authenticate the user and authorise the payment. The latest generation of cards support contactless payments over NFC and only require the PIN to be entered after a fixed number of uses or when a cumulative spending limit is reached.

Just as the payment card unburdened users from having to carry cash, so the smartphone now unburdens users from having to carry dedicated payment cards and other tokens. Tap-and-pay systems, such as Google Pay, are applications on a smartphone that enable the user to provision payment cards to a virtual wallet such that they can be used to make payments via the smartphone. Here, the user authenticates to the smartphone (which commonly entails using either a PIN, a fingerprint, or a face scan, depending on its input capabilities) and then the smartphone communicates with the terminal, typically implementing EMV protocols, over NFC, to facilitate the payment.

2.5.1 Attacks against Mobile Payment Systems

Man-in-the-middle (MITM) Attack. A MITM attack is where the attacker intercepts and continues to pass on messages that are being sent between parties that believe that they are communicating directly. This attack can be done to read information, such as keys and other cryptographic materials, or to modify information, such as the payment amount and the destination account. It can be countered by using a secure channel to authenticate messages between parties.

Phishing Attack. A phishing attack is a particular case of MITM attack where the attacker tricks the victim into divulging secret information—*e.g.*, the attacker might deploy a rogue point-of-sale terminal to collect card details and PINs for future attacks.

Replay Attack. A replay attack is where the attacker duplicates and later re-sends a message (without necessarily being able to read it). This attack can be done to replay a previous action, such as repeating a payment. It can be countered by including a unique session token or nonce inside of each encrypted component in the message to verify freshness.

Relay Attack. A relay attack is where one attacker at a terminal initiates a transaction and passes the challenge message over some channel to a second attacker who uses it (via some rogue terminal) to trigger a response from a victim's device, then the second attacker passes the response message back to the first who uses it to respond to the terminal. This attack can be done to make a payment at the expense of a distant victim. It can be countered by using distance bounding, where a timer is included in the challenge-response protocol and if the round-trip time exceeds a given bound then no physical proximity is assumed and the response is rejected. In Chapter 6, we argue that it can also be countered by using a visual channel between the user and the terminal.

EMV Vulnerabilities. A number of serious flaws in the EMV protocols have been discovered (and fixed) over the last 20 years. In 2010, Murdoch *et al.* [100] reported that one of the PIN verification methods was vulnerable to a MITM attack that could enable an attacker to trick the terminal into believing that any PIN entered had been verified. Whenever the terminal sent a PIN verification request to the card, the response from the card was not authenticated, so the dummy PIN could be blocked from reaching the card and a success message could be passed as a response. The card would continue to participate in the transaction despite not receiving a PIN as there existed valid modes of operation in which a PIN was not required. In 2014, Bond *et al.* [15] reported that the unpredictable numbers generated by some terminals and used as nonces could be replaced or predicted to enable an attacker to replay (or *pre-play*) transactions. In 2014, Emms *et al.* [39] reported that PIN verification could be avoided by switching to a foreign currency. For a comprehensive overview of the vulnerabilities of, and steps towards the formal analysis of, the EMV protocols, which is outside of the scope of this thesis, we direct the reader to the works of Basin *et al.* [13] and Radu *et al.* [114].

3

WatchAuth: User Authentication and Intent Recognition with a Smartwatch

3.1 Motivation

In recent years, tap-and-pay systems have had their functionality extended from smartphones onto smartwatches. When paired with a phone, a watch can access the virtual wallet and store tokens to allow it to make payments over NFC even when the phone is not present. Unlike smartphones, current smartwatches do not offer fingerprint readers or cameras for face recognition, but instead use short numerical passwords for user authentication. For convenience, a smartwatch can be configured to remain unlocked (*i.e.*, in an authenticated state) after a single password entry until it is restarted or removed from the wrist.

Payment systems are becoming optimised towards convenience. New 'transit modes', such as Apple Pay Express Travel mode [6], enable payments to be made at busy transport barriers without any user interaction either with the payment device or with the terminal, but limit the scope of misuse by working only for certain merchant codes (however, the implementation was shown to be vulnerable to relay attack [114]). Here, explicit authentication actions are obviated to improve usability at a cost to security. Furthermore, tap-on-phone applications enable smartphones to accept NFC payments, providing convenient new terminals for

vendors but introducing a new risk of relay attacks and accidental payments in everyday situations. These optimisations towards convenience cause an uncertainty to arise for the payment provider as to whether the user intended to make a given payment.

For tap-and-pay systems on a smartwatch, we are presented with two problems: there is a need for new authentication factors and for a mechanism to ascertain the user's intent to pay. Given the trend towards convenience, implicit factors that do not require any effort from the user are not just desirable, but increasingly necessary for a payment system to remain viable in a growing and competitive market. This means that we need to resolve both problems implicitly and with minimal response delay so as not to inconvenience the user.

For tap-and-pay systems on a smartphone, Shrestha *et al.* [131] showed that users making mobile payments can be authenticated by their movements using various sensors on the smartphone—including inertial sensors, such as accelerometers and gyroscopes, and environmental sensors, such as air pressure sensors. Authentication decisions based on this data could then be used on-device to enable or disable the NFC module or by a payment provider to authorise a payment. While not all of these sensors are available on commercial smartwatches, the inertial sensors typically are (since smartwatches were developed primarily for health and fitness monitoring applications, such as pedometry). We take inspiration from this work and develop a smartwatch-based system using a similar underlying principle.

In this chapter, we investigate the extent to which inertial sensors can be used to authenticate the user and recognise his intent to pay, both implicitly and simultaneously, when making a payment at a point-of-sale terminal using a smartwatch. We conduct two user studies—18 months apart, using the same apparatus—to collect tap gesture data. Using this data, we evaluate the effectiveness of using tap gestures as a behavioural biometric to authenticate users and as a recognisable activity from which to infer intent to pay.

3.2 System Design

3.2.1 Design Goals

The objective of this chapter is to investigate the use of inertial sensors on a smartwatch for user authentication and intent recognition purposes. We select the context of mobile payments and consider tap gestures, as measured by the inertial sensors on a smartwatch, made by the user when making a mobile payment at a point-of-sale terminal. We pose the following questions.

- To what extent can tap gestures made with a smartwatch be used to authenticate the user implicitly during a transaction?
- To what extent can tap gestures made with a smartwatch be used to infer the user's intent to pay implicitly during a transaction?

3.2.2 Design Considerations

Gesture Boundaries. Compared to tap-and-pay systems implemented on a smartphone, the starting point of a tap gesture is more difficult to determine on a smartwatch than on a phone. A phone is picked up with an explicit gesture, providing an indicator, whereas a watch is worn continuously. While the continuous collection of motion data has only a negligible impact on the battery life of our watch, gesture segmentation and classification tasks have a greater impact and cannot practically be done continuously with current hardware. At present, this problem is avoided because an explicit action is required in all tap-and-pay systems to initiate a payment (*e.g.*, clicking a side button on the device); however, in convenience-optimised, zero-interaction payment schemes, this parity breaks. To ensure that our data reflects the most difficult scenario, we preclude participants in our user studies from interacting with the watch between tap gestures. We address the challenge of finding starting points by representing tap gestures with sliding time windows of sensor data extended backwards from the NFC contact point.

Change in Orientation. A smartwatch undergoes a much greater change in orientation to perform a tap gesture than a smartphone. A phone can be tapped on the terminal without any change in orientation, owing to its sensor placement and

the user's arm not being in the way; whereas, a watch is constrained to the wrist of the user and so must follow the physiology of the arm as it is moved until the watch face rests against the terminal. Moreover, the sensor axes are set relative to the orientation of the device (see Section 3.3.3); thus, although the watch travels from the user to the terminal along a single axis in the external reference frame, its movement is measured in the reference frame of the watch along multiple sensor axes as it changes orientation during travel. One way to stabilise this behaviour would be to transform the sensor data into the external co-ordinate system (*e.g.*, as done by Ardüser *et al.* [8]). Unfortunately, this would require some ground truth from both reference frames to calibrate the transformation, which is not practical in our case—we would need either (i) to prompt the user to hold the watch in a known position, which would impose an inconvenience, or (ii) to infer backwards from when the watch face is flush against the terminal, which would require us to know the position of the terminal. We instead address this challenge by extracting a number of axis-invariant features to inform our classifiers.

Dominant Hand. One drawback of using wrist-worn sensors is that users tend to perform pertinent gestures with their dominant hand and wear a smartwatch on the wrist of the other. In our case, this problem is avoided as the user must move the watch itself to the terminal when making a payment.

Antenna Size. The size of an NFC field is determined by the size of the antenna that generates it. Communication between devices over NFC requires their NFC fields to overlap, so when a payment device is moved towards a terminal to make a payment there is a delay before the NFC contact point is found that is dependent on the sizes and positions of their antennae. The antenna in a contactless payment card is spread across its surface, forming a broad field that is tolerant of variations in the size and position of the antennae in different models of terminal. However, the antenna in a smartwatch is constrained due to its smaller form factor, making communication susceptible to longer and variable delays across different models of terminal. This imposes a dilemma: to avoid uncontrolled variability, we would need to use the same model of terminal throughout, but doing so would prevent our results from being fully terminal-agnostic. To address this challenge,

we use a single NFC reader for consistent data collection, which is our priority, and we repeat the data collection exercise with it set at multiple positions—informed by terminals observed in the wild and generalised by using other models of terminal as fixtures—to achieve results that are, at least, position-agnostic.

Privacy Risk. Several works have shown that inertial sensor data from a smartwatch can be used to infer alphanumeric inputs, potentially leaking secret information. Wang *et al.* [155] combined wrist motion data with a linguistic model that matched common short letter sequences to infer contextual inputs (English words) being typed on a keyboard. Liu *et al.* [83] and Wang *et al.* [153] inferred PIN inputs on the keypad of a terminal. Sen *et al.* [124] showed that the location of touchscreen taps on a smartphone screen (and therefore the keys tapped on its virtual keyboard) held in the same hand as the watch could be inferred using wrist motion data. Ardüser *et al.* [8] combined wrist motion data with audio recordings (for segmentation) to infer text characters written on a whiteboard with the same hand. Other works have shown that wrist motion data can be used to recognise object interactions, such as operating a household tap or light switch [117, 149], playing table football [123], smoking [109], and various other hand-related activities [73]. Riaz *et al.* [120] showed that personal characteristics, such as age range, sex, and height range, could be inferred from the inertial sensor data of four measuring devices worn simultaneously on the ankle, wrist, back, and chest as users walked back and forth. These works demonstrate the risk posed to the user when yielding inertial sensor data to an application. To measure the risk to privacy posed by using gesture biometrics, we investigate the extent to which our tap gesture data can be misused to infer the personal characteristics of the user. Moreover, we hypothesise that if there exists a connexion between users' tap gestures and one or more of their personal characteristics, such that one can infer the latter from the former, then this connexion might be exploitable by an adversary and so presents a potential attack vector. We include this in our threat model as the informed attacker (see Section 3.2.4).

3.2.3 System Model

We consider a system model in which a user is wearing a smartwatch on his wrist and is using it to make an NFC-enabled payment at a point-of-sale terminal in a typical setting—namely, in a shop or at the entry barriers to a public transportation system. To make a payment, we assume that the user performs a *tap gesture* by moving his wrist towards the terminal until the watch is near enough to exchange data via NFC. The NFC contact point is when the payment provider would decide whether to approve the payment, so we assume that this marks the end of the tap gesture for real-time purposes. We assume that the watch has an accelerometer and a gyroscope and that we have access to the data generated by these sensors. We use data from only the inertial sensors; this enables us to compare our results fairly with those of related works (in Section 3.7 and in subsequent chapters).

While the current generation of smartwatch is dependent upon a paired smartphone for administration purposes (*e.g.*, installing applications), they may operate independently once set up. As such, we do not assume or require that a phone be present and we do not make use of any additional data, such as location and proximity data, that one might provide.

Using the wrist motion data collected from the smartwatch, we create two separate models: an *authentication model*, in which we verify the identity of the user, and an *intent recognition model*, in which we infer the intention of the user to make the payment. For the latter, we assume that a tap gesture is composed of a sufficiently deliberate sequence of movements as to be unlikely to be performed unintentionally, such that if we identify a tap gesture during a transaction then we infer that the payment is intentional.

The principal goal of this work is to show that the tap gesture is a biometric capable of authenticating the user and recognising intent to pay, implicitly and simultaneously. To demonstrate the applicability of our approach, we evaluate our models in three use-cases. First, for *in-store usage*, we restrict ourselves to using sensor data that is available in real-time at the terminal, collected before the

NFC contact point, pursuant to the assumption given above. Second, for *retrospective fraud detection*, we assume that a payment provider has access to historic sensor data, collected before and after the NFC contact point, and we extend our models accordingly. Third, for use as an *additional factor*, we optimise our models (i) to minimise the occurrence of false negatives so that implementing our models alongside an existing authentication system provides a strict improvement to security with negligible cost to usability and (ii) to minimise the occurrence of false positives so as to measure the positive impact on usability that our system provides by removing unnecessary authentication actions from the user.

3.2.4 Threat Model

For the authentication model, we consider an adversary that has possession of a legitimate user's smartwatch, has unlocked it, is wearing it, and is attempting to use it to make a payment at a terminal. The adversary may have (maliciously) stolen the watch or (benignly) borrowed it; we include the latter as we seek to prevent the user from sharing the watch for payment purposes. Our goal here is to authenticate the legitimate user and to reject the adversary by using only tap gestures. We consider the following three types of presentation attack:

- *zero-effort attack*: for any given victim, all other users are considered to be passive, zero-effort attackers;
- *informed attack*: an attacker who chooses victims of similar height, motivated by a hypothesis that users of a similar height would be more difficult to tell apart using wrist motion data when interacting with terminals; and
- *observation attack*: an active attacker who sees the victim perform gestures (*e.g.*, via a hidden camera) and then attempts to mimic him.

For the intent recognition model, we consider that the user has unlocked the smartwatch and that, while it remains on the wrist and in an unlocked state, an unintentional payment has been initiated. This could be malicious, where an adversary may have moved a terminal or other NFC-enabled device to the watch unbeknownst to the user, such as a relay attack, or it could be accidental. We



Figure 3.1: The equipment used in our experiment: six fixed terminals (labelled, see Table 3.1 for details), an NFC reader, a Raspberry Pi for timestamp collection, and a smartwatch. Inset: our smartwatch worn on the left wrist with the sensor axes shown.

Terminal	Height (cm)	Tilt (°)	Distance (cm)
1	100	0	5
2	120	60	25
3	95	45	-10
4	105	30	15
5	110	15	10
6	115	90	30
F	picked up from centre of platform		

Table 3.1: Details of the terminals used in our experiment; the indices match those labelled in Figure 3.1 and 'F' is the freestyle terminal. *Height* is measured from the floor to the lowest point of the terminal; *Tilt* is the angle of inclination at the lowest point of the terminal; and *Distance* is measured from the front of the stand to the point of the terminal that is nearest to the user. Terminals 2 and 6 match terminals on self-service checkouts at supermarket chains, roughly an arm's length from the user.

assume that the user is wearing the watch and performing nondescript activities in any of three public settings: while (i) *walking* or (ii) commuting *on a bus or train*, where an adversary would have ample access to the watch, or (iii) *in a shop*, where an accidental payment may be mistaken for intentional by other factors because of its location. Our goal here is to recognise that the user did not perform a tap gesture at the time of the payment and therefore did not intend to make the payment.

In this work, we concentrate on the extent to which gesture biometrics can be used to defend against these attacks. We do not consider threats to other components in the payment system, tampering of devices, poisoning of biometric templates, malware, or denial of service attacks.



Figure 3.2: Two examples of point-of-sale terminals commonly found in the UK: the terminal on a self-service checkout at a supermarket (left) and the terminals on a train station barrier (right, in yellow). The positions of these terminals are replicated by our Terminals 2 and 3, respectively.

3.3 Experiment Design

3.3.1 Experiment Overview

To evaluate the extent to which wrist motion data can be used to achieve our goals, we designed a user study to collect data. Our experiment consisted of six point-of-sale terminals on an adjustable stand fixed at a height of 100 cm, an ACR122U NFC reader connected to a Raspberry Pi for timestamp collection, and a Samsung Galaxy Watch running the Tizen 4.0 operating system. Figure 3.1 shows our apparatus. We built a data collection application with the Tizen Studio IDE and installed it on the smartwatch to collect timestamped wrist motion data as the user wore it. To collect data for a single tap gesture, the NFC reader was first affixed to the front of the terminal being used and the user performed the tap gesture on it as if making an NFC-enabled payment. Each NFC contact point timestamp was captured by the Raspberry Pi and a short spacing delay was then initiated before the user was instructed to perform the next tap gesture. Our application also functioned in a standalone manner, enabling participants to collect data outside of the laboratory during various non-payment activities.

3.3.2 Point-of-sale Terminals

To emulate a real-world mobile payment setting, we captured tap gestures using *seven* terminals: six in fixed positions (as shown in Figure 3.1 and detailed in

Table 3.1) and one 'freestyle'. For five of the six fixed terminals, we surveyed prominent supermarket and restaurant chains to find popular or standardised terminal positions (in terms of height, tilt angle, and distance from the user) and set our terminals to match common configurations. We set the other fixed terminal to match the position of the terminal on a train station barrier (Terminal 3, as shown in Figure 3.2). For the freestyle terminal, the user picked up the NFC reader directly with his other hand and performed a tap gesture on it, returning it after each interaction, just as if a shopkeeper had handed an unmounted terminal to a customer. We chose these scenarios to represent a broad cross-section of the real-world instances in which a smartwatch user may be required to perform a tap gesture as part of a payment transaction.

The six fixed terminals remained deactivated throughout the experiment because their functionality was not required. For consistent data collection, we affixed the NFC reader to each terminal when using it. As such, the terminals should be regarded only as fixtures that enforced positions, as well as a tool for immersing the user in a payment scenario.

3.3.3 Sensor Module

The Tizen platform exposed access to the IMU of the smartwatch in the form of four data streams from its MEMS accelerometer, gyroscope, and magnetometer. The *accelerometer* measured change in velocity and the *gyroscope* measured angular velocity. The *linear accelerometer* was derived from the accelerometer with the effects of gravity excluded. The *gyroscope rotation vector* (GRV) was a fusion of sensor readings to compute the orientation of the device. Our application collected timestamped data from all four at a sampling rate of 50 Hz.

The inertial sensors measured wrist motion along three axes that were relative to the frame of the watch (as shown in Figure 3.1). Motion along the x-axis corresponded with arm extension or withdrawal; the y-axis, with side-to-side arm waving; and the z-axis, directly up- and downwards through the watch face.

3.3.4 User Study

To collect our data, we conducted two user studies that were reviewed and approved by our Computer Science Departmental Research Ethics Committee. We recruited 31 participants, including students, staff, and members of the public. Each participant attended three data collection sessions, performing 10 tap gestures on each of the seven terminals in each session. The first and second sessions occurred together, separated by a short break lasting roughly 5 minutes; the third session occurred on a different day. We collected 210 tap gestures from each user.

The participant was instructed to stand in front of the terminal stand while performing the tap gestures. Aside from this, we did not prescribe any constraints on positioning as we wanted the user to stand and interact with the terminals naturally as though making payments in a real-world setting.

We deployed our user study twice, 18 months apart; the second round was conducted in a different location but used the exact same movable terminal stand and apparatus. The first round had 16 participants, the second had 21. A subset of 6 participants attended both rounds, which enabled us to investigate the stability over time of tap gesture biometrics.

Impersonation. A subset of 21 participants consented to having their gestures recorded. The first 3 were recorded as victims and the latter 18 watched the video recordings and attempted to impersonate them; then, at the end of the study, the first 3 were invited back to attempt to impersonate the other 18. The camera was fixed in position, as if hidden, and so the amount of observable information was controlled. For each attack, the attacking participant watched a short video of the victim performing a tap gesture three times and then made three attempts to mimic him. The attacker wore the smartwatch on the same wrist as the victim during this exercise. This design enabled us to investigate the robustness of our system against impersonation and to compare attacker skill and victim vulnerability (also known as *wolf and lamb analysis*).

Intent Recognition. A subset of 9 participants also collected and labelled data outside of the laboratory while walking, commuting on a bus or train, or

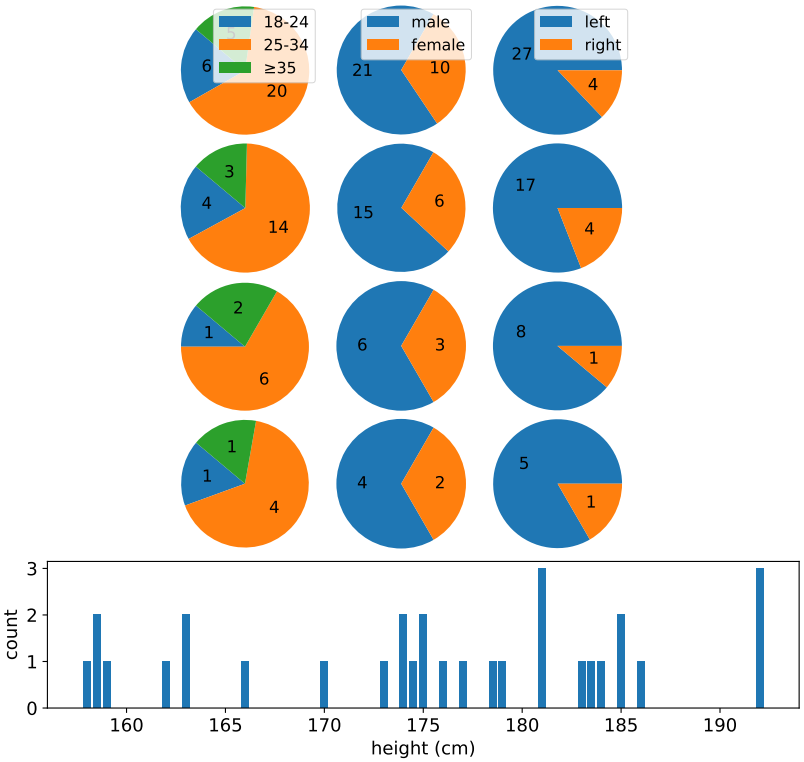


Figure 3.3: The distribution of age (left), sex (centre), and on which wrist the smartwatch was worn (right) of participants in our user study (top, n=31), the subset that participated in the impersonation exercise (upper middle, n=21), the subset that collected out-of-lab data (lower middle, n=9), and the subset that participated in both rounds (bottom, n=6). Underneath: the heights of participants in our user study (n=31).

in a shop—daily activities identified in our threat model as likely settings where unintentional payments might be made. In total, we collected 1,088 minutes of out-of-lab activity data across all users (601 minutes of walking, 317 minutes on a bus or train, 148 minutes in-store, and 22 minutes of combined activities[1]). This data formed the negative class for our intent recognition model.

User Statistics. Of our 31 participants: 20 regularly wore a watch of some kind (11 of which wore a smartwatch), 22 had paid with a smartphone before, and 5 had paid with a smartwatch (*i.e.*, 45% of those who regularly wore one). The tap gesture is intuitive to perform; we observed no difference, either during the user study or in subsequent analysis, between those who regularly wore any watch and those who did not, nor between those who had paid with a smart device and those who had not. A breakdown of participant demographics is shown in Figure 3.3.

[1]We refer to an activity as *combined* if the user performed multiple of these activities, each exclusively, but did not label them individually.

3.4 Methods

3.4.1 Data Processing

We collected time-series data from four inertial sensors. At time t, each accelerometer, gyroscope, or linear accelerometer datum is given in the form (t, x, y, z) and each GRV datum is given as a quaternion in the form (t, x, y, z, w).

We express a tap gesture using inertial sensor data within a time window. To retrieve the tap gestures for each user, we segment 4-second blocks of sensor data by using the NFC contact point timestamps as the endpoint of each window. In our preliminary testing, we found that a 4-second window size was sufficient to encapsulate the entirety of each gesture. We also retrieve an additional 2 seconds of data after the NFC contact point for when we consider retrospective fraud detection.

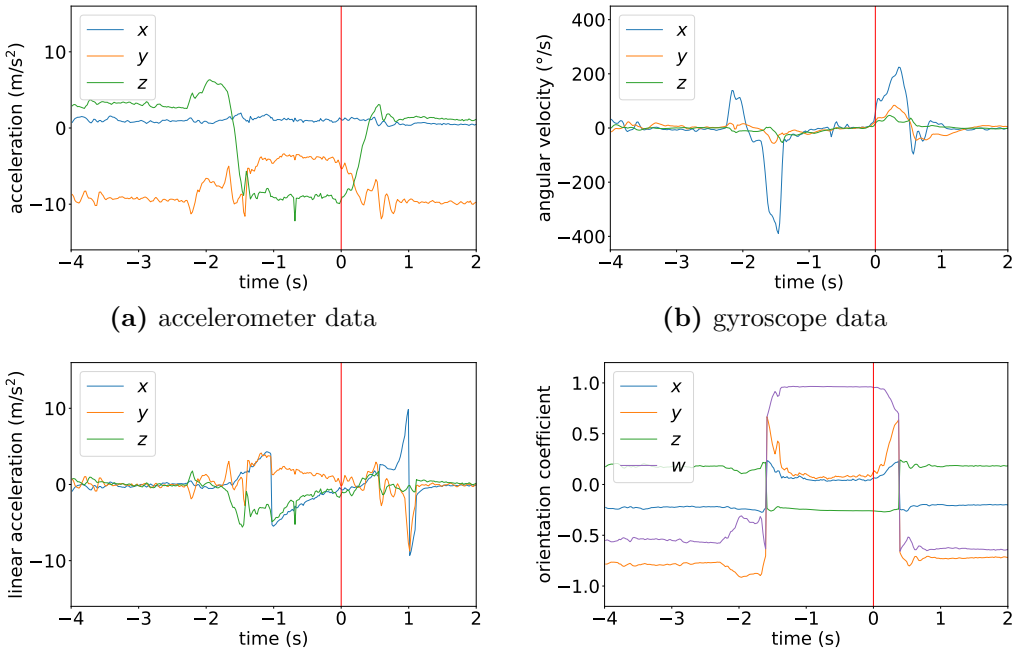
Figure 3.4 shows the sensor data for an exemplar tap gesture. We infer from the accelerometer data that the smartwatch reached the terminal at approximately 1.5 seconds before the NFC contact point and then, from the gyroscope and GRV data, that the user adjusted the orientation of the watch face to align it with the terminal to find the NFC connexion. The data after the NFC contact point shows the user's arm withdrawing.

To find the optimum parameters for a tap gesture, we compare the performances of gestures bounded by different window sizes and offsets, where the offset is the time between the NFC contact point and the end of the window. For an NFC contact point timestamp T_0, a window size s, and an offset o, we retrieve a tap gesture with start time T_S and end time T_E, where

$$T_E = T_0 - o$$

$$T_S = T_E - s$$

We segment the inertial sensor data collected outside of the laboratory into comparable 4-second blocks. In total, we obtain 32,441 4-second, non-tap gestures by segmenting the data every 2 seconds to ensure a full overlap so as not to eliminate any interesting regions.



(a) accelerometer data **(b)** gyroscope data

(c) linear accelerometer data **(d)** GRV data

Figure 3.4: Plots of the same tap gesture (made at Terminal 1), showing data 4 seconds before the NFC contact point and 2 seconds after from the four inertial sensors used to collect our data. The NFC contact point is set at time 0 (indicated with a red line).

3.4.2 Feature Extraction

Whenever a gesture is retrieved, we apply a Butterworth low-pass filter with cut-off frequency normalised by the Nyquist frequency, which is half the sampling rate, to smooth out the noise. We tried several filters and found this to work best for our data—*i.e.*, it smoothed out the noise effectively while being the least destructive to the patterns in the data. This process results in samples that are binned in time, retaining a sampling rate of 50 Hz. We then process the following five dimensions for each sample: the filtered x-, y-, and z-values, the energy of those filtered values, and the energy of the raw (unfiltered) values, where energy is given by

$$\text{energy of } (x, y, z) = \sqrt{x^2 + y^2 + z^2}$$

As GRV data are expressed as quaternions, for those we process only the four filtered values (since the Euclidean norm of a quaternion is always 1). In total, we process each gesture in 19 dimensions.

For each gesture, we extract the following ten statistical features in each dimension: *minimum*, *maximum*, *mean*, *median*, *standard deviation*, *variance*, *interquartile range*, *kurtosis*, *skewness*, and *peak count*. We also calculate the *mean* and *maximum velocities* along each axis, the *displacement* along each axis, and the *Euclidean displacement* from each of its accelerometer, gyroscope, and linear accelerometer vectors, adding another 30 features. (Note that we calculate these latter features by integrating or double-integrating with respect to the post-filter sampling interval.) We reduce each gesture to a feature vector containing 220 members.

We began with a larger set of features that had been used successfully by other authors in similar cases with time-series data [1, 25, 88, 119] and we pruned it down by using normalised Gini importances to reject the least informative features during our preliminary testing. The Gini importance of a feature is a measure of its effect to decrease the impurity of the model and thus its positive impact on classification [16]. We chose to include kurtosis and skewness due to an observation that in several dimensions there was one prominent peak whose shape or position correlated well per user. We chose to extract features from the energy of our samples and to calculate the Euclidean displacement due to the variability in orientation of the smartwatch in transit between the user and the terminal.

3.4.3 Classification

We employ supervised learning approaches to train separate models for authentication and intent recognition. We use random forest classifiers in each of our models. A random forest is an ensemble learning method that combines the efforts of multiple decision trees, each constructed from a randomly-selected, bootstrapped sample of training data, and outputs the modal class. Random forests have been shown to be efficient, able to estimate the importance of features, and robust against noise [18, 92]. To balance relevance with learning time, we include 100 decision trees in each forest [106]. To reduce the impact of random generation on our results, and to avoid the deceptive practice of selecting results from only the most performant forest, we train and test each classifier ten times with different forest randomisation seeds and average the outcomes.

Position-agnostic Model. For our authentication model, we train a set of classifiers that are user-dependent. This means that a separate template (and decision threshold) is generated for each user. In each classifier, we take a given user's tap gestures as the positive class and other users' tap gestures as the negative class. As this is an authentication use-case, we ensure that the training data temporally precedes the testing data by taking the tap gestures collected in users' first and second data collection sessions as training data (analogous to the enrolment phase, where the user's template is created) and those collected in the third session as testing data (analogous to the authentication phase). For each user, we train multiple classifiers, each one excluding a different fixed terminal in a *leave-one-out* manner; we train the classifier on users' tap gestures made at the other terminal positions and then test it on those made at the excluded terminal, to ensure that the model is agnostic to the position of the terminal and generalised. To evaluate the zero-effort attack, we include data from all other users in the negative class; with 31 users and 6 fixed terminals, this gives $31 \times 6 \times 10 = 1,860$ classifiers. To evaluate the informed attack, we exhaustively pair each user as a victim with each other user as an attacker and exclude the attacker's samples from the training data; with 930 pairs, this gives $930 \times 6 \times 10 = 55,800$ classifiers.

Position-known Model. To evaluate the observation attack, we do not exclude tap gestures based on terminal—we assume that, since the attacker has already observed the victim making a tap gesture at that terminal, the victim's template will have been updated to include knowledge of it. We pair each user as a victim with each other user as an attacker and exclude the attacker's samples from both the training and testing data, then generate a model with no knowledge of the attacker to find the victim's decision threshold in that pairing, tuned to the EER (which we take as the baseline FAR, or *base-FAR*). Then, we test the attacker's impersonation samples against that tuned classifier to find his attack success rate (the *observation-FAR*). We compare these two FARs to measure the success of the observation attack. (See Section 3.4.4 for an explanation of these terms.)

Position-specific Model. For investigative purposes, we train a set of user-dependent classifiers in which each is trained and tested on tap gestures made only

at a single terminal. This model enables us to measure the effectiveness of our approach if applied to systems with standardised terminals that occur fixed at the same position in every deployment, such as public transport systems.

Intent Recognition Model. For our intent recognition model, we train a set of activity recognition classifiers that are user-agnostic. This means that all users share a common template, such that the model can be deployed standalone, alongside our authentication model, or added to an existing system, in all cases as a pre-trained model that does not require an enrolment phase for each new user. We take all users' tap gestures as the positive class and all users' non-tap gestures as the negative class—*i.e.*, we treat tap gestures made at a terminal as intentional and other gestures as unintentional. We want our results to reflect the most difficult scenario: that of a new user whose data is not yet included in the common template. So, for each user, we train classifiers using only the data of other users to ensure that they are user-agnostic and generalised.

Age, Sex, and Height Inference Models. To measure the risk to privacy posed by the misuse of tap gesture data, we train a set of classifiers and regressors to infer the various personal characteristics that we collected from participants in our user study (as summarised in Figure 3.3). For age, we have a multi-class classification problem with three age classes: 18 to 24, 25 to 34, and 35 and over. For sex, we have a binary classification problem. For height, we have a regression problem with height values ranging between 158 cm and 192 cm (rounded to the nearest 0.5 cm). In each case, for each user, we train random forest classifiers and regressors using the labelled tap gesture data from all other users.

3.4.4 Performance Metrics

In our models, the *true positives*, TP, is the number of times that the positive class (*i.e.*, the legitimate user or intentional gesture) is correctly accepted; the *true negatives*, TN, is the number of times that the negative class (*i.e.*, the adversary or unintentional gesture) is correctly rejected; the *false positives*, FP, is the number of times that the negative class is wrongly accepted; and the *false negatives*, FN, is the number of times that the positive class is wrongly rejected.

To compare our results with those of our closest related work [131] using the same metrics, we calculate precision, recall, and F-measure. Precision indicates *security*, by measuring how well a model rejects the negative class, and is given by

$$\text{precision} = \frac{TP}{TP + FP}$$

Recall indicates *usability*, inasmuch as it measures how well a model avoids misclassifying the positive class and causing inconvenience to the user, and is given by

$$\text{recall} = \frac{TP}{TP + FN}$$

F-measure is the harmonic mean of the precision and recall (equally weighted), offering a rough fusion of the two, and is given by

$$\text{F-measure} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

The decision threshold, θ, is the score at which the classifier chooses to assign to a sample the positive class rather than the negative. To finely tune a classifier, we adjust θ to modify the trade-off between security and usability; a larger θ is more resilient to false positives and thus favours security, a smaller θ favours usability.

To quantify the performance of our models, we find the optimum decision threshold such that the false acceptance rate (FAR) equals the false rejection rate (FRR); this point is called the equal error rate (EER). The FAR and FRR are antagonistic insofar as setting θ to favour one will disfavour the other. The FAR inversely indicates security, by measuring the likelihood that the negative class will be wrongly accepted, and is given by

$$\text{FAR} = \frac{FP}{FP + TN}$$

The FRR inversely indicates usability, by measuring the likelihood that the positive class will be wrongly rejected, and is given by

$$\text{FRR} = \frac{FN}{FN + TP} = 1 - \text{recall}$$

The EER is a widely used metric in authentication systems that gives a measure of system performance when consideration is balanced evenly between security and usability.

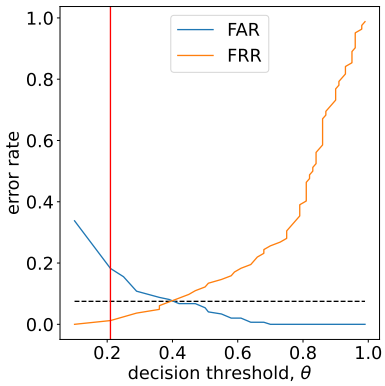


Figure 3.5: An example plot showing an EER of 0.07 at $\theta = 0.42$. At $\theta = 0.21$ (indicated with a red line), the model is optimised to minimise the FRR and has an FAR of 0.18.

To understand in greater detail how well a classifier can distinguish between classes, we plot a receiver operating characteristic (ROC) curve by comparing its true positive rate against its false positive rate (*i.e.*, its recall against its FAR) for each decision threshold. The area under this curve (AUC) gives a measure of separability, where a score of 1 indicates perfect class separation, a score of 0.5 indicates none, and a score of 0 indicates inversion.

To quantify the performance of our models when used as an additional factor in an existing authentication system, we want to measure the security and usability gains that we can provide without incurring any cost in terms of having a negative impact on the usability or security, respectively, of the existing system.

For security, we tune each model to minimise the occurrence of false negatives (so as not to degrade usability, since each false negative requires the user to re-authenticate) by selecting the optimum decision threshold such that the FRR is less than 0.001 and the corresponding FAR is as low as possible (an example of which is shown in Figure 3.5). Then, to measure the strict security gains given by the model, we take the FAR under this condition. (Note that we choose a threshold of 0.001 as it has an appropriate level of precision for our number of samples.)

For usability, we tune each model to minimise the occurrence of false positives by selecting θ such that the FAR is less than 0.001 and the corresponding FRR is as low as possible. The FRR under this condition is a measure of the strict usability gains given by the model in the following way: if we assume that our

model is implemented as an outer shield around the existing system, such that if it authenticates the user implicitly with high confidence then it enables him to bypass the other authentication factor(s), then our model effectively reduces the number of times that the user needs to perform some explicit authentication action by 1−FRR, or the recall.

To quantify the performance of our age and sex inference models, we calculate the accuracy. This is a widely used metric in supervised learning tasks that measures how closely the output of a model matches the true values and is given by

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

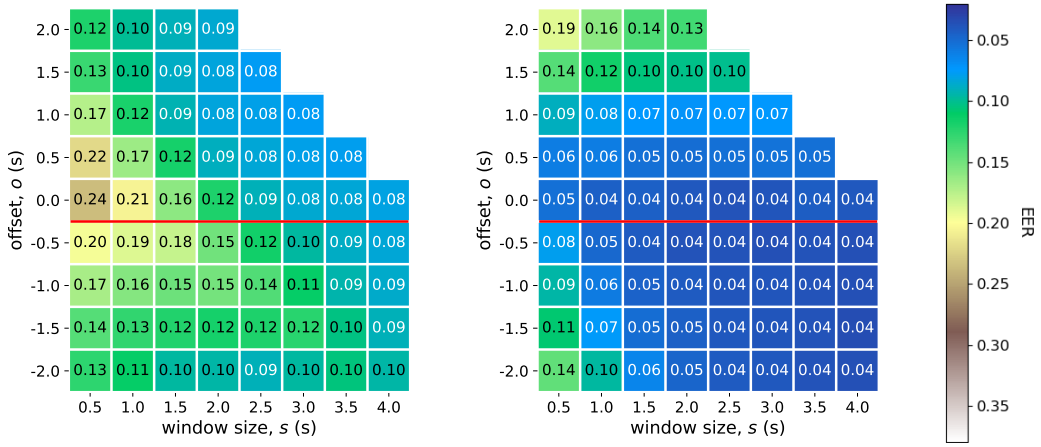
However, accuracy is not an appropriate metric for our other models because it does not differentiate between whether a failure is a false negative or a false positive, which is a consequential distinction in authentication use-cases [36].

To quantify the performance of our height inference models, we calculate the coefficient of determination, denoted by R^2. This is a widely used metric in regression tasks that measures how well the line of regression fits the data.

3.5 Results

3.5.1 Anatomy of a Tap Gesture

We observe that a tap gesture can be demarcated into the following three phases: *reaching*, *alignment*, and *withdrawal*. In the reaching phase, the user extends his arm to move the smartwatch towards the terminal. Once the watch is touching or very close to touching the terminal, the user stops reaching and enters the alignment phase. In the alignment phase, the user aligns the watch face with the terminal and tentatively moves it around to find an NFC connexion, owing to the short-ranged nature of NFC technology. Once a connexion is established and the payment is approved, the terminal notifies the user with a sound or message and the user withdraws. We find that the alignment phase for a typical tap gesture in our study begins roughly 0.5 to 1.5 seconds before the NFC contact point, depending on how quickly the NFC connexion is established, and ends roughly 0 to 1 seconds after, depending on how quickly the user reacts to the notification.



(a) authentication model **(b)** intent recognition model

Figure 3.6: Average EERs for our authentication and intent recognition models by window size and offset. Tap gestures that end at or before the NFC contact point, which are therefore compatible with in-store usage, are above the red line.

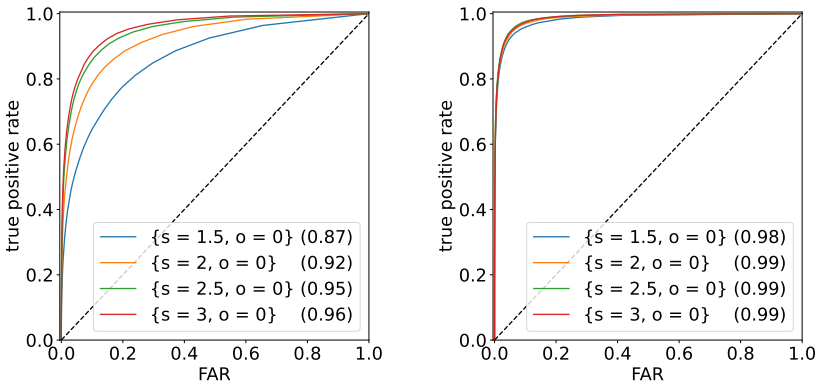
3.5.2 Zero-effort Attack

Figure 3.6 shows the average EERs for our authentication and intent recognition models by window size and offset. Each score in Figure 3.6a is the average over 1,860 classifiers and in Figure 3.6b, over 90 (see Section 3.4.3 for details). Figure 3.7 shows the average ROC curves for the strongest performing window parameters.

In-store Usage. For real-time usage, we focus only on offsets $o \geq 0$ (above the red line), where the tap gestures end at or before the NFC contact point, so the system can use the result in deciding whether to approve the payment.

For authentication, we see in Figure 3.6a that our model achieves better EERs the larger the window. Favouring a smaller window size to reduce memory usage, we achieve optimal performance at $\{s = 2, o = 1\}$ with an EER of 0.08. To ease implementation, an offset of 0 is preferable and we can see that this comes at little cost to EER. We see in Figure 3.7a that although $\{s = 3, o = 0\}$ performs better than $\{s = 2.5, o = 0\}$ (with AUCs of 0.96 and 0.95, respectively), the improvement is not enough to out-trade the smaller window size, so our optimum window parameters are $\{s = 2.5, o = 0\}$. We find that even a window of 0.5 seconds of wrist motion data provides a reasonable level of authentication when $o \geq 1.5$.

For intent recognition, we find in Figure 3.6b that offset is the determining



(a) authentication model **(b)** intent recognition model

Figure 3.7: Average ROC curves for our authentication and intent recognition models with different window sizes and offsets. The AUCs are given in parentheses.

factor and that smaller offsets yield stronger results; the model achieves its best in-store results when $o = 0$, with an average EER of 0.04 and AUC of 0.99.

Considering authentication and intent recognition together, we find that the optimum window parameters for in-store usage are $\{s = 2.5, o = 0\}$ (*i.e.*, a window of 2.5 seconds of sensor data taken immediately before the NFC contact point). Using a single tap gesture so windowed, our models can simultaneously and implicitly authenticate the user with an average EER of 0.09 (precision 0.95, recall 0.87, F-measure 0.88, AUC 0.95) and recognise intent to pay with an average EER of 0.04 (precision 0.93, recall 0.82, F-measure 0.86, AUC 0.99).

Retrospective Fraud Detection. For historic analysis, we consider all of the results windows.

For authentication, we see a continuation of the previous pattern. Near to the red line, we see poor results from windows that contain data collected predominantly in the alignment phase (towards the left) and, below it, we see little value in data collected after the NFC contact point; this suggests that neither aligning with nor withdrawing from the terminal elicit the movements that are most discriminative between users. The reaching phase is key for authentication.

For intent recognition, we find the opposite. We see a strong correlation between the preponderance of alignment phase data in a window and the strength of its results. This is most evident when viewed in the inverse, as we see that the less

Similarity Radius (cm)	Number of Pairs	FAR		
		Similar Height	Dissimilar Height	All
5	260	0.14	0.12	0.12
10	454	0.14	0.07	0.12
15	614	0.13	0.07	0.12
20	764	0.12	**0.06**	0.12
25	850	0.12	**0.04**	0.12

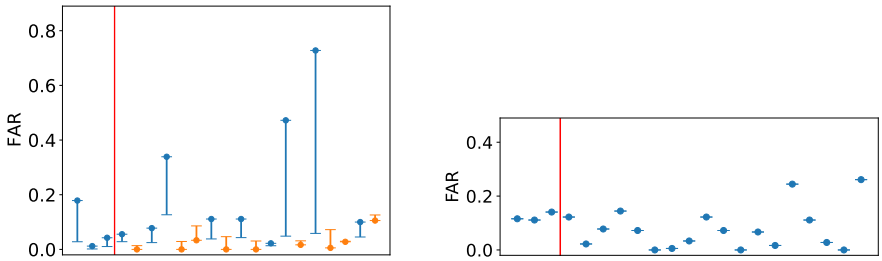
Table 3.2: Average FARs by height similarity in optimum window $\{s = 2.5, o = 0\}$ for our authentication model. The second column shows the number of pairs of victims and attackers whose height difference is less than or equal to the similarity radius.

alignment phase data a window has, the weaker its results: at the top of Figure 3.6b, the larger the positive offset, the less alignment phase data it is likely to contain, and at the bottom-left, the larger the negative offset and the smaller the window, the less alignment phase data it is likely to contain. The constricted movement as the watch moves in conformity to the surface of the terminal and the manner in which the user reacts to finding the NFC connexion are peculiar to the tap gesture and so act to distinguish it from other gestures.

3.5.3 Informed Attack

Table 3.2 shows the results of our informed attack against our authentication model. We pair each user as a victim against each other user as an attacker, exclude the attacker's samples from the training data, and train a *position-agnostic* classifier for each pair. We define a pair of users to be of a similar height if their difference in heights is less than or equal to a given value, which we call the *similarity radius*. For each similarity radius in Table 3.2, we calculate the average FAR of classifiers for pairs that are similar in height, the average FAR of those that are not similar, and the average FAR of all classifiers.

We see that pairs of a similar height yield consistently higher FARs than those of dissimilar heights, suggesting that users of a similar height are more difficult for the classifiers to tell apart, and therefore an informed attacker will gain a passive advantage when impersonating a victim of a similar height in making a payment. Moreover, we see that as the similarity radius is set larger, we get lower FARs for



(a) victims **(b)** attackers

Figure 3.8: Results of our observation attack against our *position-known* authentication model in optimum window $\{s = 2.5, o = 0\}$ on Terminals 2 and 3. The plot on the left shows for each user as a victim the average FAR of the user-specific base model (base-FAR, flat line) and the average FAR when attacked (observation-FAR, circle); if the latter is greater, then the victim's line is coloured blue, otherwise it is orange. The plot on the right shows for each user as an attacker the average FAR achieved when attempting to impersonate other users. The red line separates the first 3 users from the other 18, indicating the two groups of users who attacked each other.

dissimilar pairs. This shows that, on average, the greater the disparity in height between the victim and the attacker, the better the classifiers are at distinguishing between them and so the lower the passive attack success rate.

3.5.4 Observation Attack

Figure 3.8 shows the results of our observation attack against our *position-known* authentication model. Each of the first 3 users was the victim of 1,080 impersonation attempts on Terminals 2 and 3 (18 attackers $\times$ 2 terminals $\times$ 3 attempts at each gesture $\times$ 10); each of the other 18 users, separated by the red line, was the victim of 180 attempts (3 attackers, same terminals).

The base model achieves average base-FARs of 0.04; when attacked, the average success rate (observation-FAR) is 0.05. Figure 3.8a shows that two of our users are *lambs* (*i.e.*, users who are especially susceptible to impersonation in this characteristic), where their observation-FAR is significantly greater than their base-FAR, and two more are borderline. Figure 3.8b shows that none of our attackers is able to maintain a consistent attack success rate, with the highest rates being 0.26 and 0.24 (owing mostly to the lambs). This is an important result, as it suggests that our system provides resistance against impersonation, presenting a low likelihood that an attacker in the wild could predictably mimic a given victim.

Our attackers, while enthusiastic and attentive, were not necessarily skilled at impersonation nor were they trained—*i.e.*, they were not given feedback after each impersonation attempt. Any victim-specific training is hard to justify in a real-world mobile payment setting, where the system may lock out or throttle the attacker after a number of failed attempts. It is plausible that experienced or trained attackers could have achieved better attack success rates.

3.5.5 Feature Informativeness

To see which features are most informative to our models, we sort the features that contributed to each classifier by Gini importance and sum the number of times that each feature ranks among the top five. Table 3.3 shows the modal top-five features summed over classifiers with optimum window parameters $\{s = 2.5, o = 0\}$ and across all 62 window size and offset combinations. (Note that, with respect to the counts, there are 20 times more classifiers for authentication.)

For authentication, we see in Table 3.3a that features derived from the y- and z-axes of the gyroscope are common among the most informative; this suggests that the forward roll of the wrist is a key discriminator between users. The extrema in acceleration along the x-axis, representing the rapidity of the extension and withdrawal of the arm, are shown to be by far the most important features.

For intent recognition, we see in Table 3.3b that the number of peaks in the energy of linear accelerometer data is of particular importance, far exceeding any other feature in the count across all windows. This feature represents the frequency with which the watch starts and stops moving during the tap gesture and is prominent here likely owing to the significance of the alignment phase data to this model and the abrupt movements performed during that phase. It is notable that there are no GRV-derived features among the commonest (indeed, we also tallied the top-twenty features and found no GRV-derived features present there either). Together, these findings suggest that the distinctiveness of the alignment phase does not come from the orientation of the watch face, but from the *changes* in orientation detected across sensors. Features that are derived from the x-axis of the gyroscope, which measures the tilt of the wrist from side to side (see

$\{s = 2.5, o = 0\}$		**All Windows**	
Feature	#	**Feature**	#
`Acc-x-min`	406	`Acc-x-min`	17310
`Acc-z-max`	311	`Acc-x-max`	16446
`Acc-x-max`	293	`Acc-x-velomean`	10412
`Gyr-y-med`	231	`Gyr-y-mean`	9853
`Gyr-z-velomean`	227	`GRV-x-mean`	9841
`Gyr-z-mean`	225	`Gyr-y-med`	9657
`GRV-x-min`	215	`Acc-z-max`	9244
`Gyr-y-mean`	210	`Acc-x-med`	9223
`Acc-y-max`	209	`Gyr-y-max`	8682
`GRV-x-max`	188	`Gyr-z-velomean`	8502

(a) authentication model

$\{s = 2.5, o = 0\}$		**All Windows**	
Feature	#	**Feature**	#
`Acc-z-iqr`	79	`LAc-unf-pkcount`	3397
`Acc-z-kurt`	65	`Gyr-x-mean`	1985
`Gyr-x-mean`	57	`Acc-z-var`	1789
`LAc-unf-pkcount`	55	`Acc-z-stdev`	1706
`Acc-disptotal`	44	`Acc-z-iqr`	1696
`Acc-unf-iqr`	43	`Acc-z-med`	1316
`Gyr-unf-min`	43	`Acc-unf-iqr`	1118
`Acc-z-var`	33	`Gyr-x-velomean`	1011
`Acc-z-stdev`	20	`Gyr-unf-min`	971
`Acc-x-pkcount`	11	`Acc-disptotal`	907

(b) intent recognition model

$\{s = 2.5, o = 0\}$		**All Windows**	
Feature	#	**Feature**	#
`Acc-z-kurt`	73	`Acc-z-var`	2109
`Gyr-unf-min`	73	`Acc-unf-iqr`	2049
`Acc-z-iqr`	70	`Gyr-x-mean`	1841
`Acc-unf-iqr`	65	`Acc-z-stdev`	1716
`Gyr-x-mean`	52	`Acc-z-iqr`	1485
`Acc-z-var`	47	`Acc-unf-pkcount`	1412
`Acc-disptotal`	35	`Acc-z-med`	1368
`Acc-z-stdev`	30	`Gyr-unf-min`	1239
`Acc-x-pkcount`	3	`Gyr-x-disp`	1117
`Gyr-x-disp`	2	`Gyr-x-velomean`	1043

(c) intent recognition model; accelerometer & gyroscope data

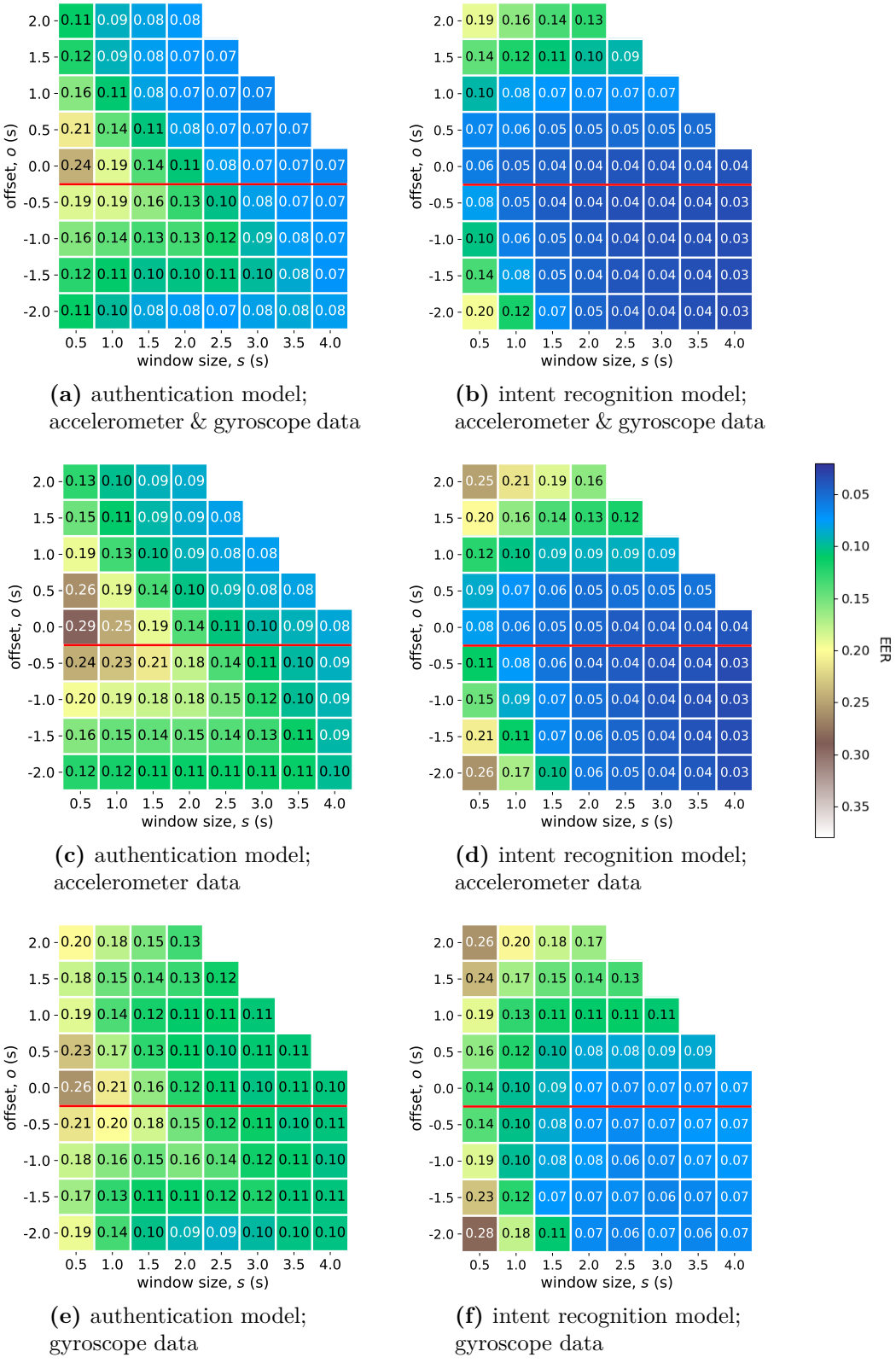
Table 3.3: Modal top-five features by Gini importance summed over all classifiers in optimum window $\{s = 2.5, o = 0\}$ and across all windows for our authentication and intent recognition models (for the latter, once trained and tested on all sensor data and once on a subset). Features are given in the format sensor-axis-statistic; `unf` is the energy of the unfiltered $\{x, y, z\}$ values and `disptotal` is the Euclidean displacement.

Figure 3.1), likely express this most profoundly (in particular, `Gyr-x-velomean` gives a running approximation of the sideways orientation of the device). We also see that features derived from the z-axis of the accelerometer are frequently among the most important in distinguishing between gestures; this is likely to be because sustained movement in the direction of the watch face is peculiar to the tap gesture.

3.5.6 Sensor Selection

We collected wrist motion data from all four of the inertial sensors available on our smartwatch. Some devices are more limited in their offering—the accelerometer is the commonest sensor, as it is the smallest and cheapest, followed by the gyroscope. To assess the feasibility of our approach on devices with fewer sensors, we train a set of sensor-specific models in which each classifier is trained and tested on data from a subset of sensors. Figure 3.9 shows the average EERs for our models when using data from (i) only the accelerometer and gyroscope, (ii) only the accelerometer, and (iii) only the gyroscope (*cf.* Figure 3.6, which uses data from all four sensors).

When using data from only the accelerometer and gyroscope, we see improved results in both models (EERs of 0.07 for authentication and 0.03 for intent recognition). This suggests that the inclusion of the linear accelerometer and GRV is unnecessary and pollutes the classifiers. These sensors derive their values from the other two, so the information that they provide is likely to be redundant and the calculations that generate them may introduce noise. Table 3.3c shows the modal top-five features for the intent recognition model with these sensors omitted; compared with Table 3.3b, we see that the frequency of starts and stops remains important, with `Acc-unf-pkcount` rising in prominence in the absence of `LAc-unf-pkcount` (although not to the same extent, per the count); the rest of the list is reordered but largely unchanged. When using data from only the accelerometer, we see that the results in poorly performing windows become worse proportional to how poor they were when using data from all four sensors, with little change in the best performing windows. When using data from only the gyroscope, we see weaker results.



(a) authentication model; accelerometer & gyroscope data

(b) intent recognition model; accelerometer & gyroscope data

(c) authentication model; accelerometer data

(d) intent recognition model; accelerometer data

(e) authentication model; gyroscope data

(f) intent recognition model; gyroscope data

Figure 3.9: Average EERs for our authentication and intent recognition models by window size, offset, and sensors. Tap gestures that end at or before the NFC contact point are above the red line.

	EER			
Terminal	$s = 1.5$	$s = 2$	$s = 2.5$	$s = 3$
1	0.10	0.10	0.09	0.10
2	0.17	0.13	0.11	0.10
3	0.14	0.11	0.10	0.11
4	0.12	0.08	0.07	0.08
5	0.11	0.09	0.08	0.09
6	0.12	0.09	0.08	0.08
F	0.09	**0.07**	**0.07**	0.08
agnostic	0.16	0.12	0.09	0.08

(a) authentication model

	EER			
Terminal	$s = 1.5$	$s = 2$	$s = 2.5$	$s = 3$
1	**0.01**	**0.01**	**0.01**	**0.01**
2	0.03	0.03	0.03	0.03
3	0.04	0.04	0.04	0.04
4	**0.01**	**0.01**	**0.01**	**0.01**
5	**0.01**	**0.01**	**0.02**	**0.02**
6	0.04	0.04	0.03	0.03
F	0.07	0.07	0.07	0.08
agnostic	0.04	0.04	0.04	0.04

(b) intent recognition model

Table 3.4: Average EERs for our *position-specific* authentication and intent recognition models with different window sizes ($o = 0$ in each case) by terminal. Our *position-agnostic* results are included for comparison.

3.5.7 Terminal Positions

We collected tap gestures performed on a range of terminals. Some systems use highly standardised terminals that are set in the same position in many locations. To evaluate our approach in a single-terminal scenario, we train a set of position-specific models in which each classifier is trained and tested on tap gestures for a single terminal. Table 3.4 shows the average EERs for our *position-specific* authentication and intent recognition models.

For authentication, we see in Table 3.4a that, with the exception of Terminal 2, the single-terminal models generally outperform the position-agnostic model. We see that the results for the freestyle terminal are particularly strong; the freedom to pick up the terminal elicited smoother and more comfortable movements from users in the reaching phase, allowing for user-distinctive traits to be presented.

For intent recognition, we find in Table 3.4b that the distance between the user and the terminal appears to correlate well with our results. As shown in Table 3.1, Terminals 2 and 6 are an arm's length away from the user (and yield weak results), Terminals 1, 4, and 5 are near, and Terminal 3 protrudes. The protrusiveness of Terminal 3 elicited a change in pose of the user and the smoother gesture that resulted from this is less distinctive; whereas, for Terminals 1, 4, and 5, no such change in pose was prompted, so users interacted with these

Activity Type	Number of Samples	Proportion of Samples (%)	FAR
walking	17890	55.15	0.03
bus or train	9463	29.17	0.05
in-store	4417	13.62	**0.06**
all	32441	100	0.04

Table 3.5: Average FARs (tuned to the EER) by non-tap gesture type in optimum window $\{s = 2.5, o = 0\}$ for our intent recognition model. This excludes combined activity data, which account for 2% of non-tap gesture samples.

terminals with a contorted arm twist, causing a more conspicuous and distinct gesture. Terminal 1 proved to be particularly awkward for shorter users, causing the most conspicuous gestures. Conversely, the convenience afforded to the user by picking up the freestyle terminal minimised the time spent in the alignment phase, depriving resultant gestures of discriminative movements. We find that user comfort is beneficial in authentication, but detrimental in intent recognition.

The strongest results in Table 3.4b are given by the three terminals that have the least tilt angles (see Table 3.1 for details). Terminal 1 is flat on the surface of the platform and so demands the greatest wrist rotation from the user; this echoes our findings in Section 3.5.5 and suggests that wrist rotation is a key discriminator between tap gestures and other gestures.

3.5.8 Misclassification by Activity Type

To see where misclassifications in our intent recognition model are most likely to occur, we sort them by activity type. Table 3.5 shows the number, proportion, and FAR of non-tap gestures of each type. We see that in-store gestures account for the greatest proportion of false positives; it is likely that typical in-store actions, such as reaching for a product on a shelf and rotating the wrist to read its label or to place it in a basket, exhibit some similar movement pattern fragments to those found in a tap gesture.

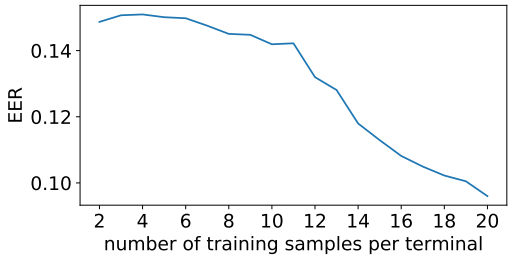


Figure 3.10: Average EERs for our authentication model if trained on different numbers of enrolment samples in optimum window $\{s = 2.5, o = 0\}$. Each classifier is trained on six terminals.

3.5.9 Enrolment Parameters

Behavioural biometric systems typically entail a burdensome enrolment phase, where the user must perform the measured characteristic repeatedly to create the initial template. The burden placed on the user is weighed against the scarcity of training data and its effect on the performance of the model. To evaluate the extent to which we can expedite the enrolment phase, we compare the average EERs of our authentication model when the classifiers are trained on smaller positive classes (*i.e.*, fewer user samples). Figure 3.10 shows that our model can authenticate the user with an average EER of 0.14 when it is trained on just 12 of the user's tap gestures (spread evenly over six terminals), which can be performed in less than a minute. We see that the EER improves as more samples are included in the training set. This suggests that an update mechanism would benefit the model over time, relaxing upfront requirements and incorporating subsequent tap gestures to improve the classifier as the system is used. (Note that our intent recognition model is user-agnostic, so does not require the user to undergo an enrolment phase. Nonetheless, we posit that it could be improved over time in the same manner.)

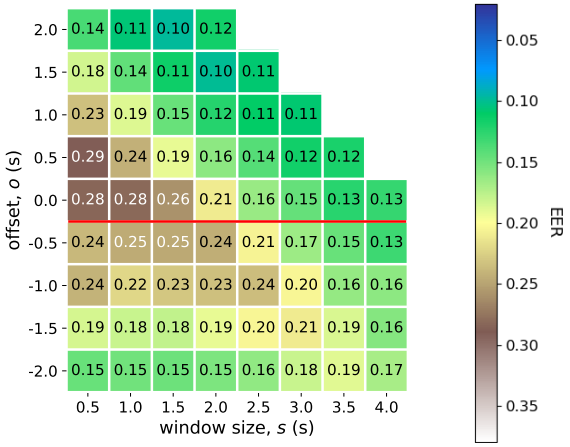
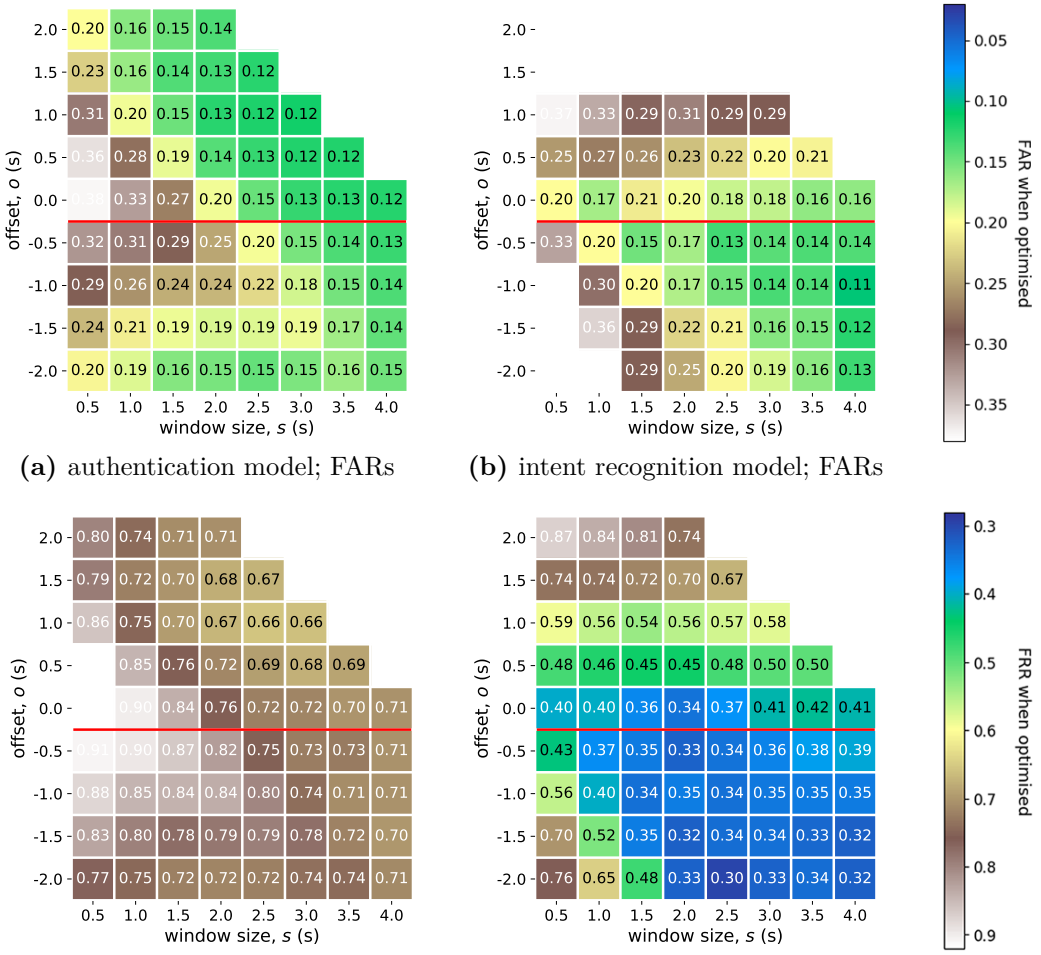


Figure 3.11: Average EERs for our authentication model for users who participated in both rounds of our user study, trained on tap gestures from the first round and tested on tap gestures from the second, 18 months later, indicating the impact of drift when compared with Figure 3.6a, by window size and offset. Tap gestures that end at or before the NFC contact point are above the red line.

3.5.10 Stability over Time

Behavioural biometric systems are prone to characteristic drift, where the user's template becomes outdated over time. To evaluate the stability over time of tap gestures, we compare the average EERs of our authentication model (as shown in Figure 3.6a) with the average EERs of those users who participated in both rounds of our user study when the model is trained on their tap gestures from the first round and tested on their tap gestures from the second round, which is shown in Figure 3.11. (Note that all other users' tap gestures were also included in the latter model to ensure that the model is trained comparably, but their EERs were excluded from the results.) We see an additive increase in EERs of 0.03 in the larger windows and of 0.07 in optimum window $\{s = 2.5, o = 0\}$. These are promising results, indicating that tap gesture biometrics drift only a little, even over an inactive period of 18 months. In practice, payment systems are expected to be used regularly and such a small amount of drift can be remedied with a template update after each successful payment.



(a) authentication model; FARs **(b)** intent recognition model; FARs

(c) authentication model; FRRs **(d)** intent recognition model; FRRs

Figure 3.12: Average FARs when optimised to minimise false negatives and FRRs when optimised to minimise false positives for our authentication and intent recognition models by window size and offset. Tap gestures that end at or before the NFC contact point are above the red line.

3.5.11 Feasibility as an Additional Factor

To evaluate the strict gains that our models can provide to an existing authentication system, we adjust the decision thresholds to make any negative impact negligible and measure the FARs and FRRs under those conditions. Figure 3.12 shows the average FARs and FRRs when optimised for our authentication and intent recognition models by window size and offset.

Added Security. For our models to add a strict improvement to security, we tune our classifiers to minimise the occurrence of false negatives.

For authentication, we see in Figure 3.12a that our model can add to an existing

system a layer of false acceptance detection with a FAR of 0.12 without imposing a burden on usability. This means that we can reject 88% of the adversary's tap gestures at no cost to usability.

For intent recognition, we see in Figure 3.12b that our model can add a layer of unintentional payment detection with a FAR of 0.16 for in-store usage and 0.11 for retrospective fraud detection without incorrectly rejecting intentional payments.

Added Usability. For our models to add a strict improvement to usability, we tune our classifiers to minimise the occurrence of false positives.

For authentication, we see in Figure 3.12c that our model can add to an existing system a strong layer of security at the cost of a FRR of 0.66. This means that we can reduce the number of unnecessary authentication actions required of the user by 34% at no cost to security.

For intent recognition, we see in Figure 3.12d that our model can identify all unintentional payments at a cost of wrongly rejecting 34% of intentional payments. This means that we can reduce the number of unnecessary intent verification requests that may be required of the user by 66% without the payment provider having to worry about unintentional payments.

Unsharable Factor. Our approach adds two components to the security of an existing system. First, each of our models adds an implicit layer of false acceptance detection, tuned either for security, for usability, or for a balance of both, as quantified by the error rates above. Second, our authentication model also introduces to the system an unsharable factor that ensures that only the legitimate user can make payments with the smartwatch. Unsharability is an oft overlooked yet advantageous property of biometrics (as a result of the uniqueness desideratum) and one that cannot otherwise be achieved by knowledge- or possession-based factors. We have shown that we are able to provide these gains without negatively impacting the existing system.

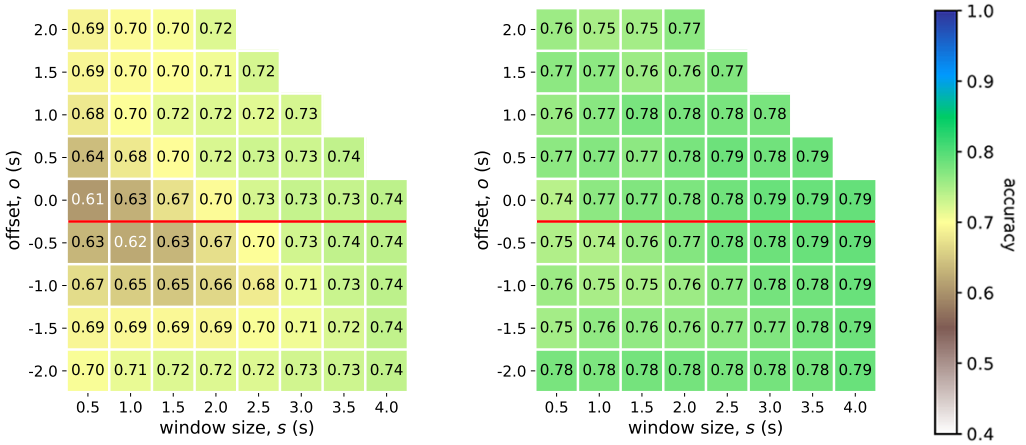
3.5.12 Privacy Implications

We have shown that even a small window of inertial sensor data can be used to authenticate the user and to identify intent to pay. To investigate whether any private information, such as personal characteristics, is exposed via inertial sensor data while performing a tap gesture, we train a set of models to infer various traits. Figures 3.13 and 3.14 show the average accuracy and R^2 scores for our sex and height inference models, respectively.

For age range, our model achieves average accuracy scores of between 0.56 and 0.61 in all windows, indicating that we cannot infer the age range of users.

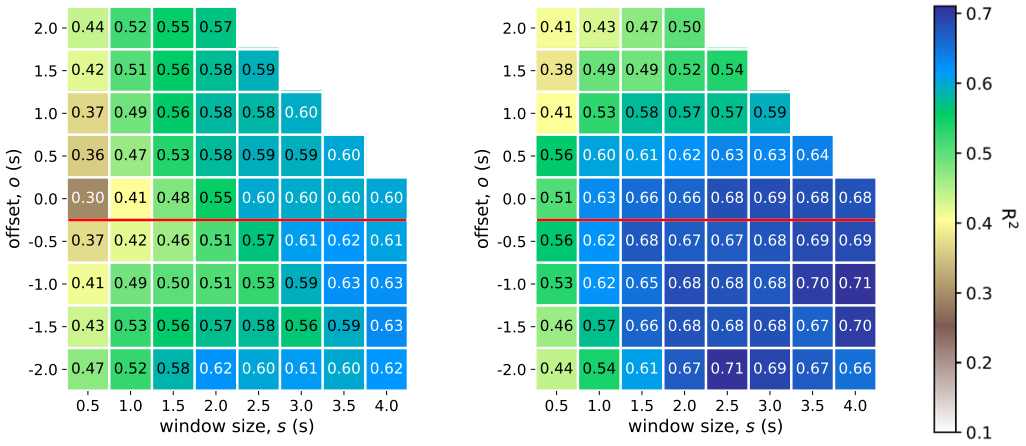
For sex, we see in Figure 3.13a that our model performs poorly in smaller windows, but achieves average accuracy scores of up to 0.74 in larger windows. Noting that all 4 of the participants in the user study who wore the smartwatch on their right wrist were female, we train another sex inference model on data from only those users who wore it on their left wrist. We see in Figure 3.13b that this model scores significantly higher, with average accuracy scores of between 0.74 and 0.79 in all windows. This suggests that the tap gesture data from the right-wrist users were blurring the decision boundaries rather than providing any advantage. We can infer the sex of users reasonably well, likely due to its correlation with height.

For height, we see in Figure 3.14a that our model achieves R^2 scores of up to 0.63 in larger windows. Above the red line, we find that our classifiers fail to agree on the most important features; `Gyr-unf-mean` is common among the modal top-five features in the better performing windows, suggesting that overall rotational speed may be distinctive, and to a lesser extent so are `Acc-y-max` and `Gyr-z-pkcount`. When we apply a *position-specific* approach, we see in Figure 3.14b that a single-terminal model can achieve R^2 scores of up to 0.71 on tap gestures for Terminal 3. In both models, we see the strongest results in those windows that contain more withdrawal phase data and the most influential features in those windows are derived from the x-axis of the accelerometer, which corresponds to up- and downwards movement when the dorsum of the hand is angled towards the terminal.



(a) all users **(b)** left-wrist users

Figure 3.13: Average accuracy scores for our sex inference models by window size and offset. Tap gestures that end at or before the NFC contact point are above the red line.



(a) all terminals **(b)** Terminal 3

Figure 3.14: Average R^2 scores for our height inference models by window size and offset. Tap gestures that end at or before the NFC contact point are above the red line.

Together, these findings suggest that it is the straightening of the user's back as he pulls away from the terminal that betrays his height. These results demonstrate a relationship between height and tap gestures; we can infer the height of users reasonably well and an informed attacker can select victims of a similar height to confuse an authentication classifier (as shown in Section 3.5.3).

All of our terminals are set at heights between 95 cm and 120 cm from the floor (as detailed in Table 3.1). To evaluate whether this restriction impacts our height inference models, we train a set of models in which each classifier is trained and

	EER			
Gesture Type	$s = 1.5$	$s = 2$	$s = 2.5$	$s = 3$
tap gestures; all terminals	0.54	0.58	0.62	0.62
tap gestures; Terminal 3	0.60	0.63	0.71	0.75
non-tap gestures	**0.63**	**0.65**	**0.66**	**0.68**
all users; tap gestures; all terminals	0.48	0.55	0.60	0.60
all users; tap gestures; Terminal 3	0.66	0.66	0.68	0.69

Table 3.6: Average R^2 scores for our height inference models with different window sizes ($o = 0$ in each case) by gesture type, using data from only those users who collected data both inside and outside of the laboratory. Our tap gesture results for all users are included for comparison.

tested on data from only those users who collected tap gesture data inside and non-tap gesture data outside of the laboratory. Table 3.6 shows the average R^2 scores for these models by gesture type. We see that we can infer the height of users comparably well regardless of whether the user is interacting with a terminal when the data is collected. For the model using non-tap gesture data, we find that the most important features are `Acc-x-min`, as it is for our authentication models, and `Gyr-unf-pkcount`. We find similar features for the models using tap gesture data. This suggests that other biometrics that are measured by inertial sensors may also be vulnerable to an informed attacker who selects victims of a similar height.

3.6 Discussion

Power Consumption. Smartwatches are designed to facilitate always-on sensing (*e.g.*, for health and fitness monitoring). To measure the impact of our data collection application in practical terms, we wore two Samsung Galaxy Watches in an identical state, but with one running our application. Without any effort put into performance optimisation, our application caused the watch running it to consume an additional 1.5% of battery capacity per hour. While we do not implement the random forest classifier on the smartwatch, we argue that its energy consumption would be negligible due to the limited number of inferences that would be required per day (only when the user makes a payment).

Response Time. We calculated the computation time for classifying a single tap gesture, averaged over 10,000, to be 7.11 ms for authentication and 7.09 ms for intent recognition on a desktop computer with an Intel Core i5-6500 processor. This could not be tested directly on the smartwatch due to a lack of library support in Tizen Studio IDE.

Gesture Overlap. In our experiment, we collected datasets of gestures and then segmented them by NFC contact point timestamp. There was always a chance that a user might make a peculiarly long gesture, such that its tail would bleed into the data segmented for the next gesture. We took precautions to prevent this by imposing a short spacing delay of 2 seconds between gestures (and a harder spacing delay of 1 second was imposed by the NFC reader hardware) and requiring the user to step away from the terminals after each gesture. We decided against plotting every gesture and rejecting those that might have trailed too long, as this could have added human error or bias to the dataset.

Antenna Size. We found that the use of a smartwatch entails a longer alignment phase than a payment card due to its smaller antenna. It is therefore likely that intent recognition is more effective on a smartwatch-based tap-and-pay system because alignment phase data is most discriminative between gestures.

Classifier Choice. We tested a number of classification techniques in our preliminary experiments, including support vector machines and the dynamic time warping approach used by Lee *et al.* [76]. We avoided computationally expensive techniques, such as neural networks, as these would be impractical to run on current smartwatch hardware. We chose to use random forest classifiers and regressors with hyperparameters tuned manually based on preliminary testing as we found them to be efficient, able to achieve comparable results, and able to estimate the importance of features. Given the number of classification tasks completed throughout this work, it was impractical to replicate each task with multiple classifier techniques. This is a limitation and it is possible that other classifiers may have performed better or provided other useful properties. We have made all of our datasets publicly available to enable further exploration.

Privacy Risk Mitigation. Countermeasures to mitigate privacy leakage from wrist motion data have been suggested by related work [83, 153, 124] and include wearing the smartwatch on the wrist that is not performing the task, disabling the inertial sensors, and deliberately causing noise in the sensor data (either via hardware, such as by adding a vibrator, or via user behaviour changes). Unfortunately, none of these is possible in our scenario because we rely on the fidelity of the inertial sensor data. We suggest that users tightly restrict access permissions to their inertial sensor data to limit the collection of it by third parties.

3.7 Related Work

Authentication. With regard to authentication, Shrestha *et al.* [131] proposed the most closely related work. They considered a system model in which the user makes mobile payments with a smart*phone* and is authenticated by tap gesture. In the authentication portion of our work, we assume a similar context but explore the use of wrist-worn sensors, producing a physiologically distinct gesture and introducing a number of additional challenges (as described in Section 3.2.2). They achieved F-measure scores of up to 0.93 for authentication, a slight improvement on our results; however, they used cross-validation to train a classifier in an authentication scenario, which violates the requirement that training (enrolment) should precede testing (authentication) [36], likely inflating their scores. Drilling into the results, we note that their classifiers consistently had greater scores for recall than for precision—ours had the opposite, suggesting that the smartwatch gesture favours security and the smartphone, usability. They found the ideal gesture size to be 1 second of sensor data and mentioned losses in accuracy for greater sizes due to the capturing of extraneous movements—we find different optimum parameters for our wrist-led gesture (as described in Section 3.5.2); furthermore, our sliding window approach makes it possible for us also to consider the case of retrospective fraud detection. They collected data at terminals set in generic positions—we set ours in positions matching real-world terminals to elicit a truer representation of real payment gestures in our data; furthermore, we include a freestyle terminal to

incorporate the common scenario of a vendor handing the terminal to the customer and to counter overfitting in our models.

Lee *et al.* [76] used inertial sensors on a smartphone to authenticate the user whenever the phone was picked up, defining the implicit *pick up gesture*, with the goal of reducing the number of explicit authentication actions required. Similar prior work by Conti *et al.* [22] authenticated the user as he made or answered a phone call. Both works show solid results with short, simple gestures using a phone. These approaches use dynamic time warping to analyse sensor data; we instead use machine learning classifiers that expose the relative importances of the features upon which they base their decisions to refine our feature set and to observe the impact of our axis-invariant features.

Johnston *et al.* [61] used inertial sensors on a smartwatch to infer gait as the user walked for identification and authentication purposes, using 10-second windows of sensor data; they claim to have achieved an EER of 0.01 but do not give details of their classification methods. Similar works [21, 101] achieved EERs of 0.03 and 0.08, respectively. Orthogonal implicit authentication systems on smartwatches have adapted heart rate biometrics to authenticate the user using electrocardiography (ECG, electrical-based) or photoplethysmography (PPG, light-based) sensors [19, 38, 116, 127, 152, 166]. These systems require a few minutes to calibrate yet show promise over time and some achieved EERs of as low as 0.02 (however, ECG sensors have been shown to be vulnerable to spoofing attack [37]). We chose to restrict our study to inertial sensors because they are more common across devices than heart rate sensors, consume less energy [96], and the limited applicability to authentication scenarios of the latter has been well explored by these related works.

Some works have used inertial sensors on a smartwatch to authenticate the user with an explicit gesture, made solely for the purpose of authentication, such as MotionAuth [160] (full arm gestures), ThumbUp [163] (hand and finger gestures), and work by Liang *et al.* [80] (a punch gesture). The use of an explicit gesture can achieve strong results, but the user must take time to perform it, remember it, and conceal the performance of it, each of which can impose an inconvenience.

Intent Recognition. With regard to intent recognition, we infer an intent to pay if we identify that a tap gesture has been performed. This is a novel contribution and to the best of our knowledge there is no closely related work. Mayrhofer *et al.* [93] showed that by holding two smartphones (or other devices) together and shaking them, their inertial sensor data can be correlated and used to pair them; Findling *et al.* [42] extended this idea to a phone and an unlocked smartwatch, such that an explicit shake gesture can be detected by both and correlated to unlock the phone (essentially, the user's intent to duplicate the authentication state of the watch onto the phone is identified and acted upon). Some works have inferred the continued presence of a user from wrist-based activity recognition for the purpose of automatically de-authenticating the user from another device if the user moves away or stops interacting with it. Mare *et al.* presented both ZEBRA [88] and CSAW [89], which inferred a sequence of interactions from the wrist motion data of the user and correlated it with workstation inputs or motion sensor data from a smartphone, respectively, to verify the continued presence of the user (however, ZEBRA was found to be vulnerable to mimicry because the attacker could control both streams [57]). Lee *et al.* [74] used the inertial sensors on a smartwatch as ancillary sensors to verify the continued interaction of the user with a smartphone. Acar *et al.* [1] showed that wrist motion data can be correlated with keystroke dynamics in a similar manner. All of these systems, like the intent recognition portion of our work, achieve their inferences in a user-agnostic manner.

3.8 Conclusion

In this chapter, we showed that a tap gesture can be used to authenticate the user and recognise intent to pay, implicitly, while the user makes a payment with a smartwatch. Our approach is software-driven and does not require any changes to terminals. Our authentication model is position-agnostic, so does not restrict the placement of the terminal, and achieves EERs of 0.07. We demonstrated that our authentication model provides resistance against an active attacker and against drift, even with 18 months between tap gestures. Our intent recognition model

is user-agnostic, so does not require the user to provide any training data during enrolment and is innately resistant to drift, and achieves EERs of 0.03 (0.01 in some single-terminal scenarios). We found the optimum gesture parameters for in-store usage and for retrospective fraud detection. We showed that our models can be optimised for security or usability and incorporated as an additional factor in an existing system to provide strict gains and by adding an unsharable factor. We identified that the tap gesture is triphasic and analysed how this and other factors contributed to our results. Finally, we explored the applicability of our approach to alternative system models with fewer input sensors, dedicated terminals, or relaxed enrolment requirements while remaining performant.

Without loss of generality, we have focused on the context of mobile payments. Our approach has wide applicability to any user authentication context in which a task or gesture is performed while wearing a smartwatch, such as access control to buildings or vehicles and interaction with smart devices or objects. We will explore some of these contexts in subsequent chapters.

4

RingAuth: User Authentication with a Smart Ring

4.1 Motivation

The miniaturisation of hardware technology and the ubiquity of smartphones has enabled the emergence of smart rings on the commercial market. Despite this form factor being new, we already see a number of finger-worn devices covering a variety of use-cases, typically administered by and monitored via an application on a paired phone. Sleep tracking rings, such as Thim, use inertial or PPG sensors to infer different stages of sleep while the user is sleeping and then plot these inferences over time to monitor sleeping patterns and circadian rhythm. HelioS uses an ultra-violet radiation sensor to monitor exposure to sunlight to infer vitamin D intake and warn the user about sunburn. Amazon Echo Loop uses a microphone and speaker to enable the user to control the Alexa virtual assistant via voice commands and to make phonecalls through the paired phone. Blinq uses inertial sensors for fitness monitoring and provides a panic button that triggers the paired phone to send a help request containing geo-location information to contacts or social media. Genki Wave is a musical instrument digital interface (MIDI) controller that has buttons for scrolling through settings, a small 9-by-5 LED grid display to indicate the currently selected setting, and uses inertial sensors to enable the user to adjust the setting by raising or lowering the finger.

Smart keys are battery-less, NFC-enabled tokens that store authentication information and are powered by the terminal via NFC as they communicate with it. For convenience, a smart key can be cast as a ring and worn on the finger as a feature-poor smart ring. Tesla Ring is a smart key that enables the user to unlock and activate a vehicle via NFC. Mastercard K-ring is a smart key that enables the user to make payments via NFC, similar to a contactless payment card. While the K-ring does not communicate directly with the user's smartphone, a payment application running on the phone can be linked to the same account to which the K-ring is linked such that payment authorisation requests can be sent from the payment provider to the phone for the user to authorise.

As smart ring technology evolves and ring-based services that require user authentication, such as tap-and-pay systems, vehicle control systems, and access control systems, grow in sophistication and demand greater confidence in the identity of the wearer, there is a need for new authentication factors in addition to the user having possession of the ring as a token. However, the size and input options available on a ring are even more limited than on a smartwatch. We showed in Chapter 3 that inertial sensors on a watch can be used to authenticate the user implicitly. We take inspiration from this work and develop a smart ring-based system using a similar underlying principle.

In this chapter, we investigate the extent to which inertial sensors can be used to authenticate the user implicitly when making a payment at a point-of-sale terminal and when knocking on a door using a smart ring. We conduct a user study—using the same terminal array that we used in Chapter 3, for consistency—to collect tap and knock gesture data. We equip each user with both a smart ring and a smartwatch worn on the same arm and have him make payments with each device. Using this data, we evaluate the effectiveness of using each device to authenticate users for each type of gesture. This enables us to explore the synergy between these devices and to evaluate each as an authentication factor to support the other.

4.2 System Design

4.2.1 Design Goals

The objective of this chapter is to investigate the use of inertial sensors on a smart ring for user authentication purposes. We consider both the use of the smart ring to authenticate the user for its own services and to support authentication on other devices and the use of other devices to support authentication for services on the smart ring.

First, we select the context of mobile payments and consider tap gestures, as measured separately by the inertial sensors on a smart ring and a smartwatch, made by the user when making both ring- and watch-based mobile payments at a point-of-sale terminal. We pose the following questions.

- To what extent can tap gestures made with a smart ring, as measured by the inertial sensors on the smart ring, be used to authenticate the user implicitly during a transaction?
- To what extent can tap gestures made with a smart ring, as measured by the inertial sensors on a *smartwatch* worn on the same arm, be used to authenticate the user implicitly (in cases where the smart ring does not have inertial sensors of its own) during a transaction?
- To what extent can implicit authentication in a smartwatch system be improved by incorporating (the inertial sensor data of) a smart ring?

Second, we select the context of access control and consider both implicit and explicit knock gestures, as measured separately by the inertial sensors on the smart ring, smartwatch, and directly mounted on the door, made by the user when knocking on a closed door. We pose the following questions.

- To what extent can knock gestures, as measured by inertial sensors, be used to authenticate the user implicitly when knocking on a door?
- To what extent can knock gestures, as measured by inertial sensors, be used to authenticate the user (explicitly) when knocking on a door with a user-chosen secret knock?

4.2.2 Design Considerations

Dominant Hand. One drawback of using wearable sensors is that users might not perform pertinent gestures with the hand on which the sensors are worn. In some of our cases, this problem is avoided as the user must move the device itself to the terminal when making a payment; in the other cases, we assume that the user is wearing the device(s) on the same arm that is performing the gesture.

Also, see **Gesture Boundaries**, **Antenna Size**, and **Privacy Risk** in Section 3.2.2.

4.2.3 System Model

For our payment model, we consider a system model in which the user is wearing both a smart ring and a smartwatch on the same arm and is using them to make an NFC-enabled payment at a point-of-sale terminal in a typical setting (*e.g.*, in a shop). To make a payment, we assume that the user performs a *tap gesture* by moving one of the devices towards the terminal until it is near enough to exchange data via NFC. The NFC contact point is when the payment provider would decide whether to approve the payment, so we assume that this marks the end of the tap gesture.

For our access control model, we consider a system model in which the user is wearing both a smart ring and a smartwatch on the same arm and is knocking on a door with that hand as a means to authenticate to an access control system. We assume that each *knock gesture* is bounded by button-presses on one of the devices.

We assume that the devices each have an accelerometer and a gyroscope and that we have access to the data generated by these sensors. We use data from only the inertial sensors. When we combine data from multiple devices, we assume that devices share (partial) feature vectors over a trusted connexion to facilitate the classification of a single, combined sample. We assume that the devices communicate with the payment provider or access control system over a trusted connexion. Aside from this, we do not consider the implementation of the wider system, only the feasibility of using the data collected for authentication purposes.

4.2.4 Threat Model

We consider an adversary that has possession of a legitimate user's smart ring (or smartwatch, or both, as appropriate), has unlocked it, is wearing it, and is attempting to use it to make a payment at a terminal or to gain entry to a locked door via an access control system. The adversary may have (maliciously) stolen the device(s) or (benignly) borrowed it. Our goal here is to authenticate the legitimate user and to reject the adversary by using only tap or knock gestures. We consider the following two types of presentation attack:

- *zero-effort attack*: for any given victim, all other users are considered to be passive, zero-effort attackers; and
- *observation attack*: an active attacker who sees (and hears) the victim perform gestures (*e.g.*, via a hidden camera) and then attempts to mimic him.

In this work, we concentrate on the extent to which gesture biometrics can be used to defend against these attacks. We do not consider threats to other components in the system, tampering of devices, poisoning of biometric templates, malware, or denial of service attacks.

4.3 Experiment Design

4.3.1 Experiment Overview

To evaluate the extent to which finger and wrist motion data can be used to authenticate users, and to compare the two, we designed and conducted a user study to collect data. Our experiment consisted of six point-of-sale terminals on an adjustable stand fixed at a height of 100 cm, an ACR122U NFC reader connected to a Raspberry Pi for timestamp collection, a second Raspberry Pi with an accelerometer and a gyroscope attached to a closed door, and a smartphone fixed in position for video recording. For our wearable devices, we used a smart ring and a smartwatch (detailed below), both commercial off-the-shelf devices and worn together on the same arm by the user. We collected motion data from the

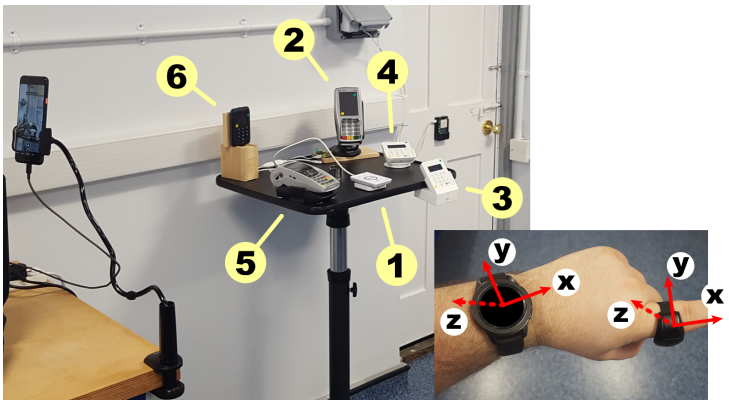


Figure 4.1: The equipment used in our experiment: six fixed terminals (labelled, see Table 3.1 for details), an NFC reader (here affixed to Terminal 1), a Raspberry Pi for timestamp collection, a Raspberry Pi attached to the door, and a fixed smartphone for video recording. Inset: our smart ring and smartwatch worn on the left arm with the sensor axes shown (each z-axis points upwards through the screen).



Figure 4.2: The attacker's restricted view of a victim interacting with Terminal 2 (left), Terminal 3 (centre), and the door (right).

wearable devices and the door-mounted Raspberry Pi (with all clocks synchronised) as the user performed gestures. Figure 4.1 shows our apparatus.

For our payment experiment, we affixed an NFC tag to the front of each wearable device and we affixed the NFC reader to the front of each point-of-sale terminal in turn. For each terminal, the user stood in front of the stand and performed tap gestures on the terminal using a wearable device, as if making mobile payments, with a short spacing delay between each tap gesture. The use of NFC tags and the NFC reader ensured consistent NFC communication between the wearable devices and the terminals. Each NFC tag stored the name of the wearable device to which it was affixed; each time an NFC contact point was made during a tap gesture, the Raspberry Pi captured its timestamp and the name of the

triggering device. This was later used to segment the inertial sensor data collected from the wearable devices to retrieve the data for each tap gesture.

For our access control experiment, the user stood in front of the closed door and performed knock gestures against it using his device-wearing hand. The user pressed the button on the smart ring before and after each knock gesture to capture bounding timestamps. These timestamps were later used to segment the inertial sensor data collected from all devices to retrieve the data for each knock gesture.

4.3.2 Point-of-sale Terminals

To emulate a real-world mobile payment setting, we captured tap gestures using *seven* terminals. See Section 3.3.2 for details.

4.3.3 Wearable Devices and Sensor Modules

For our smart ring, we used a Genki Wave. This ring was worn on the index finger and had a button that could be pressed with the thumb—we utilised this for timestamp collection in our access control experiment. We wrote a data collection script in Python using the open source library provided by the developers to interface with the ring over Bluetooth LE.

For our smartwatch, we used a Samsung Galaxy Watch running the Tizen 4.0 operating system. We built a data collection application and installed it on the watch using the Tizen Studio IDE.

From each of these wearable devices, we collected timestamped data from the following four inertial sensor data streams: the *accelerometer*, the *gyroscope*, the *linear accelerometer*, and the *GRV*. From the door-mounted Raspberry Pi, we collected data from the accelerometer and the gyroscope. We collected this data with sampling rates of 100 Hz for the smart ring (which we downsampled to 50 Hz), 50 Hz for the smartwatch, and 30 Hz for the door-mounted sensors.

The inertial sensors measured motion along three axes that were set relative to the frame of each device (as shown in Figure 4.1). Motion along the x-axis corresponded with arm extension or withdrawal; the y-axis, with side-to-side arm waving; and the z-axis, directly up- and downwards through the screen.

4.3.4 Gesture Types

To collect tap gestures for our payment experiment, we had each user perform the following types of tap gesture on each of the seven point-of-sale terminals:

- *ring tap*: the user tapped the smart ring against the terminal and moved it around, if necessary, until the NFC contact point was found to simulate a *ring*-based payment; and
- *watch tap*: the user tapped the smartwatch against the terminal and moved it around, if necessary, until the NFC contact point was found to simulate a *watch*-based payment.

To collect knock gestures for our access control experiment, we had each user perform the following types of knock gesture on the closed door:

- *3-knock*: the user knocked on the door three times;
- *5-knock*: the user knocked on the door five times; and
- *secret knock*: the user created (and then kept) a knock pattern consisting of between three and six knocks performed in a manner of his choosing (some users chose to knock to a rhythm, some included a twist or jolt of the wrist).

4.3.5 User Study

To collect our data, we conducted a user study that was reviewed and approved by our Computer Science Departmental Research Ethics Committee. We recruited 21 participants, including staff, students, and members of the public. Each participant attended three data collection sessions, performing 10 of each of the gestures detailed in Section 4.3.4 in each session. The first and second sessions occurred together, separated by a short break lasting roughly 5 minutes; the third session occurred on a different day. We collected $17 \times 10 \times 3 = 510$ gestures from each user.

In each experiment, the participant was asked to stand facing the terminals or the door. Aside from this, we did not prescribe any constraints on positioning as we wanted the user to interact comfortably as though acting in a real-world setting. The first three gestures of any type were performed in silence, to familiarise, then the reseacher engaged the participant in light conversation to simulate the

distractions of a real-world environment. This sometimes elicited additional hand and body movements if the user gesticulated naturally.

Impersonation. All 21 participants consented to having their gestures recorded. The first 3 were recorded as victims and the latter 18 watched (and heard) the video recordings and attempted to impersonate them; then, at the end of the study, the first 3 were invited back to attempt to impersonate the other 18. We recorded the following six gestures of each participant: ring tap gestures on Terminals 2 and 3, watch tap gestures on Terminals 2 and 3, 5-knock gestures, and secret knock gestures. The camera was fixed in position, as if hidden, and so the amount of observable information was controlled; Figure 4.2 shows the attacker's view of the terminals and the different information observable for each type of gesture. For each attack, the attacking participant watched a short video of the victim performing a gesture three times and then made three attempts to mimic him. The attacker wore the wearable devices on the same arm as the victim during this exercise.

User Statistics. Of our 21 participants: 15 were male, 17 wore the devices on the left arm (the decision was led by the smartwatch; everyone who wore them on the right arm was female), 15 regularly wore a watch of some kind (7 of which wore a smartwatch), 13 had paid with a smartphone before, and 5 had paid with a smartwatch (*i.e.*, 71% of those who regularly wore one). 16 participants (76%) remembered their secret knock, with an average of 4.8 days between their sessions (those who did not remember had an average of 4.2 days between sessions; across all users, there was an average of 4.6 days between sessions).

4.4 Methods

4.4.1 Data Processing

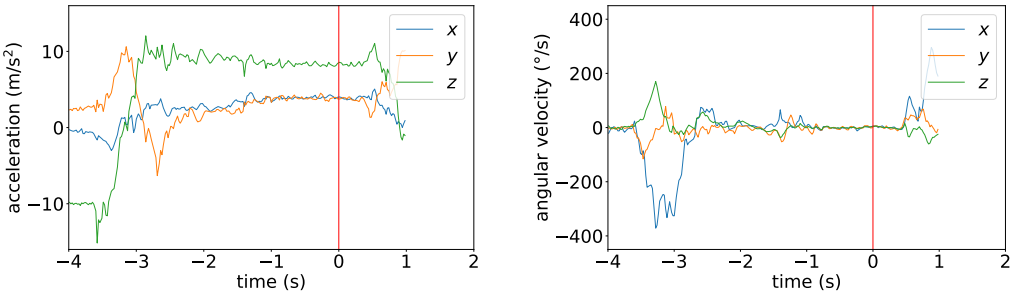
We collected time-series data from four inertial sensors on our wearable devices and from the accelerometer and gyroscope on the Raspberry Pi attached to the door. At time t, each accelerometer, gyroscope, or linear accelerometer datum is given in the form (t, x, y, z) and each GRV datum is given as a quaternion in the form (t, x, y, z, w).

We express a tap or knock gesture using inertial sensor data within a time window. In our payment experiment, we retrieve the tap gestures for each user by segmenting 4-second blocks of sensor data using the NFC contact point timestamps as the endpoint of each window. We found that a 4-second window size was sufficient to encapsulate the entirety of each tap gesture. To investigate optimum tap gesture parameters, we compare the performances of gestures bounded by different window sizes and offsets, as explained in Section 3.4.1. In our access control experiment, we retrieve the knock gestures for each user using the bounding timestamps captured when the user pressed the button on the ring before and after each gesture.

Figure 4.3 shows the sensor data for an exemplar ring tap gesture. Using data from only the smart ring, we infer from the acceleration in the y-axis and the rotation about the x-axis that the finger is furled and positioned such that the ring faces the terminal at approximately 3 seconds before the NFC contact point and then, from the acceleration in the z-axis, that the ring is pushed forward towards the terminal with little further change in orientation. Using data from only the smartwatch, we infer from the accelerometer data that the hand reached the terminal at approximately 1.5 seconds before the NFC contact point. By combining both, we infer that the withdrawal phase included a rotation of both the hand and the wrist.

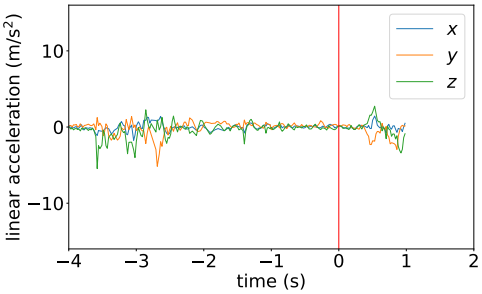
Figure 4.4 shows the sensor data for an exemplar watch tap gesture. We infer from the watch data that the hand again reached the terminal at approximately 1.5 seconds before the NFC contact point and that the ring data show peaks in acceleration and rotation that follow those in the watch data, as one would expect.

Figure 4.5 shows the sensor data for an exemplar secret knock gesture. We infer from all sources that the gesture consisted of two rapid knocks, followed by a single knock, followed by two more rapid knocks. The ring GRV data is notably smoother than its watch counterpart; this suggests that the user kept his arm relatively still and knocked by rapidly spinning his wrist into the door.

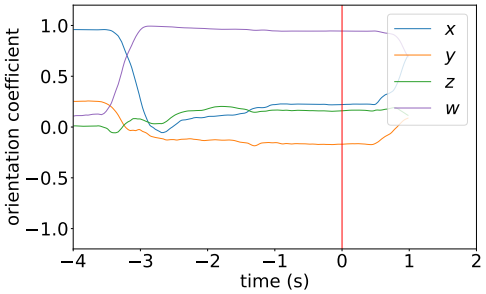


(a) ring accelerometer data

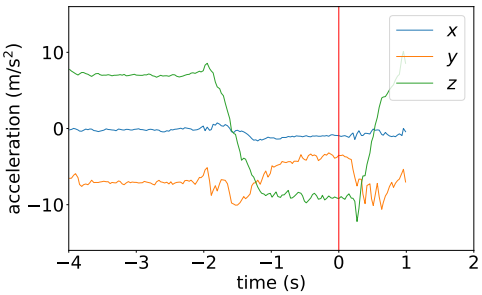
(b) ring gyroscope data



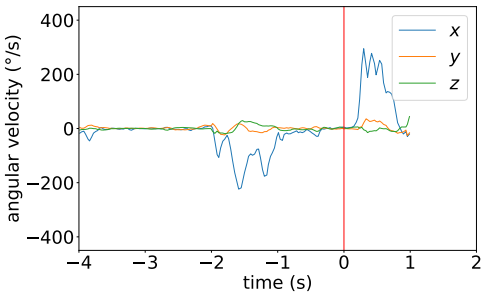
(c) ring linear accelerometer data



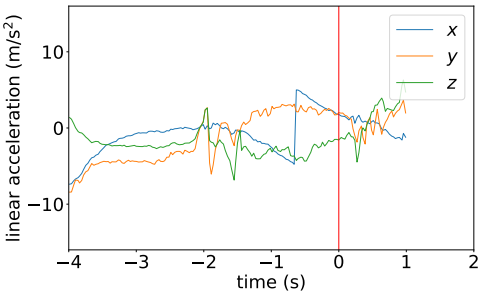
(d) ring GRV data



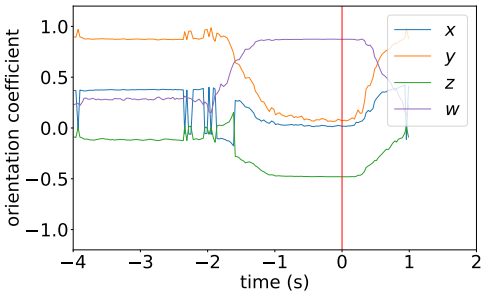
(e) watch accelerometer data



(f) watch gyroscope data

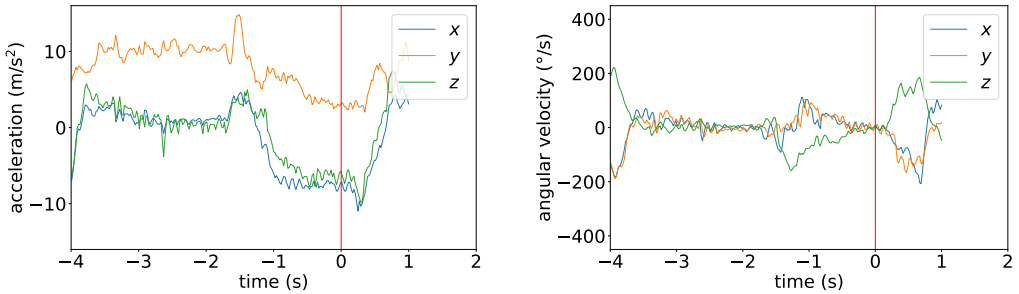


(g) watch linear accelerometer data



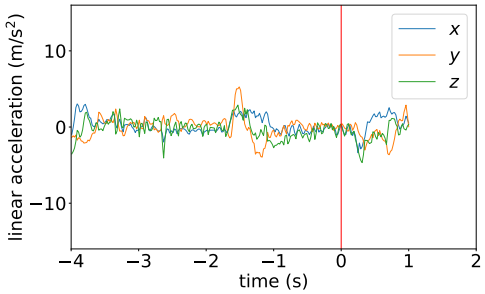
(h) watch GRV data

Figure 4.3: Plots of the same ring tap gesture (made at Terminal 1), showing data 4 seconds before the NFC contact point and 1 second after from the four inertial sensors on our smart ring (top four) and smartwatch (bottom four) used to collect our data. The NFC contact point is set at time 0 (indicated with a red line).

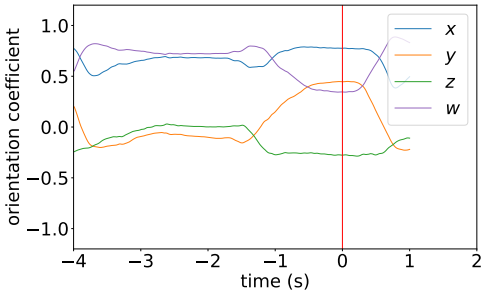


(a) ring accelerometer data

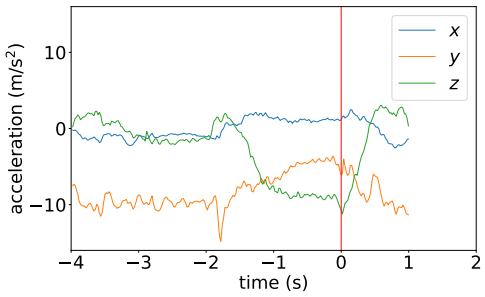
(b) ring gyroscope data



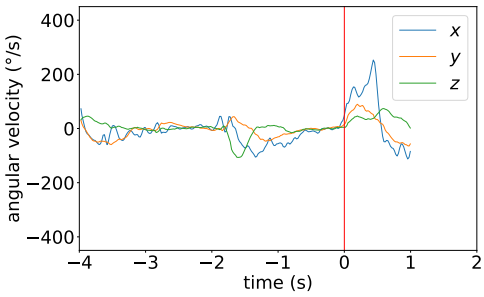
(c) ring linear accelerometer data



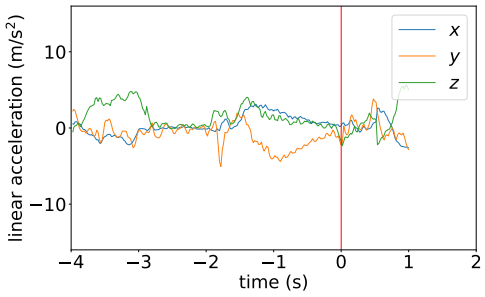
(d) ring GRV data



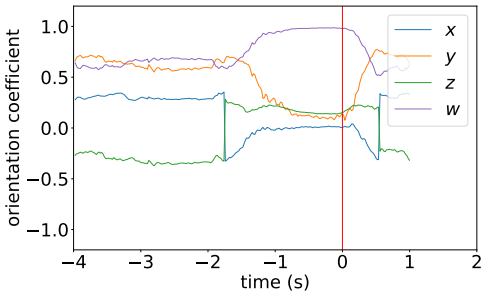
(e) watch accelerometer data



(f) watch gyroscope data

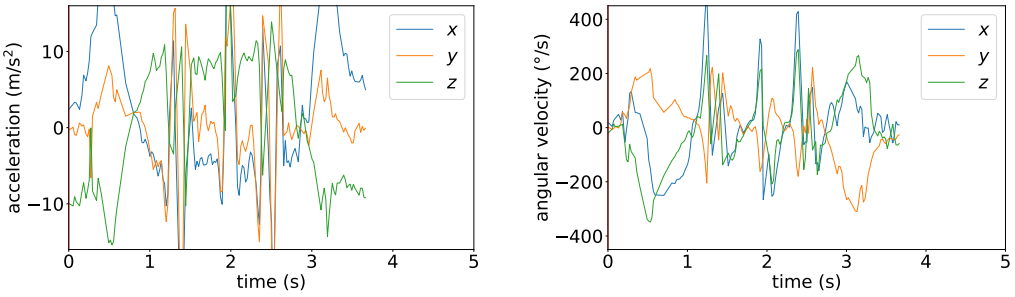


(g) watch linear accelerometer data



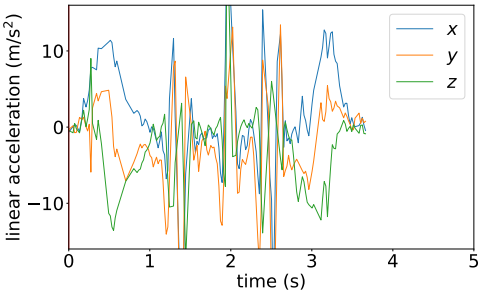
(h) watch GRV data

Figure 4.4: Plots of the same watch tap gesture (made at Terminal 1), showing data 4 seconds before the NFC contact point and 1 second after from the four inertial sensors on our smart ring (top four) and smartwatch (bottom four) used to collect our data. The NFC contact point is set at time 0 (indicated with a red line).

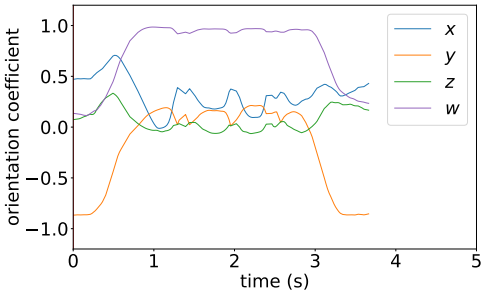


(a) ring accelerometer data

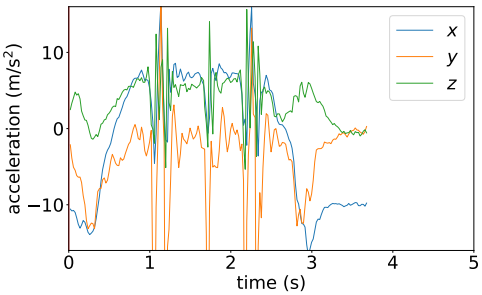
(b) ring gyroscope data



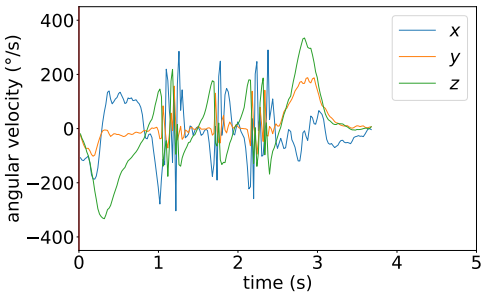
(c) ring linear accelerometer data



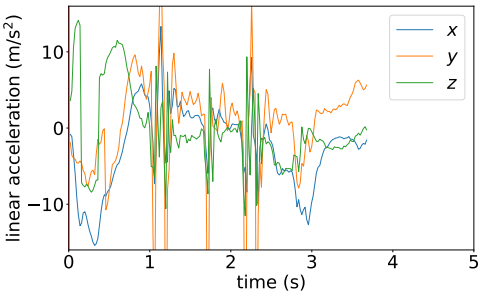
(d) ring GRV data



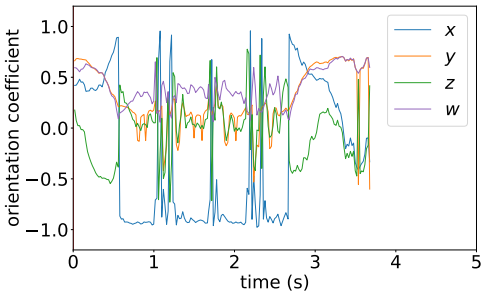
(e) watch accelerometer data



(f) watch gyroscope data



(g) watch linear accelerometer data



(h) watch GRV data

Figure 4.5: Plots of the same secret knock gesture, showing data between the bounding button-presses from the four inertial sensors on our smart ring (top four) and smartwatch (bottom four) used to collect our data.

4.4.2 Feature Extraction

For consistency across our work, we reduce each gesture to a feature vector that contains the same 220 features as before. See Section 3.4.2 for details.

In some of our models, we combine inertial sensor data from multiple devices. In these cases, we extract the full set of features for each separate device and then concatenate them together to form a linearly larger feature vector that is used for classification.

4.4.3 Classification

For consistency across our work, we use random forest classifiers in each of our payment and access control models and we find the average results over ten differently-seeded forests. See Section 3.4.3 for details.

Position-agnostic Model. For our payment model, we train a set of classifiers that are user-dependent. In each classifier, we take a given user's tap gestures as the positive class and, to evaluate the zero-effort attacker, all other users' tap gestures as the negative class. We ensure that the training data temporally precedes the testing data and that each classifier is trained in a leave-one-out manner to ensure that the model is agnostic to the position of the terminal and generalised. We train separate models for each of the two tap gestures. We train separate models using data from only the smart ring, only the smartwatch, and both devices combined.

Position-known Model. To evaluate the observation attacker, we do not exclude tap gestures based on the terminal. We pair each user as a victim with each other user as an attacker and exclude the attacker's samples from both the training and testing data to find the base-FAR of the victim in that pairing; then, we test the attacker's impersonation samples against that tuned classifier to find his attack success rate or observation-FAR.

Position-specific Model. For investigative purposes, we train a set of user-dependent classifiers in which each is trained and tested on tap gestures made only at a single terminal.

Access Control Model. For our access control model, to evaluate each of the attackers, we use a similar approach as with each respective payment model above, except that we do not need to generalise the model over multiple terminals. We train separate models for each of the three knock gestures. We train separate models using data from only the smart ring, only the smartwatch, only the door-mounted sensors, and all three devices combined.

Age, Sex, and Height Inference Models. To measure the risk to privacy posed by the misuse of tap or knock gesture data, we train a set of classifiers and regressors to infer age range, sex, or height. In each case, for each user, we train random forest classifiers or regressors using the labelled tap or knock gesture data from all other users.

4.4.4 Performance Metrics

In our models, the *true positives* is the number of times that the positive class (*i.e.*, the legitimate user) is correctly accepted; the *true negatives* is the number of times that the negative class (*i.e.*, the adversary) is correctly rejected; the *false positives* is the number of times that the negative class is wrongly accepted; and the *false negatives* is the number of times that the positive class is wrongly rejected.

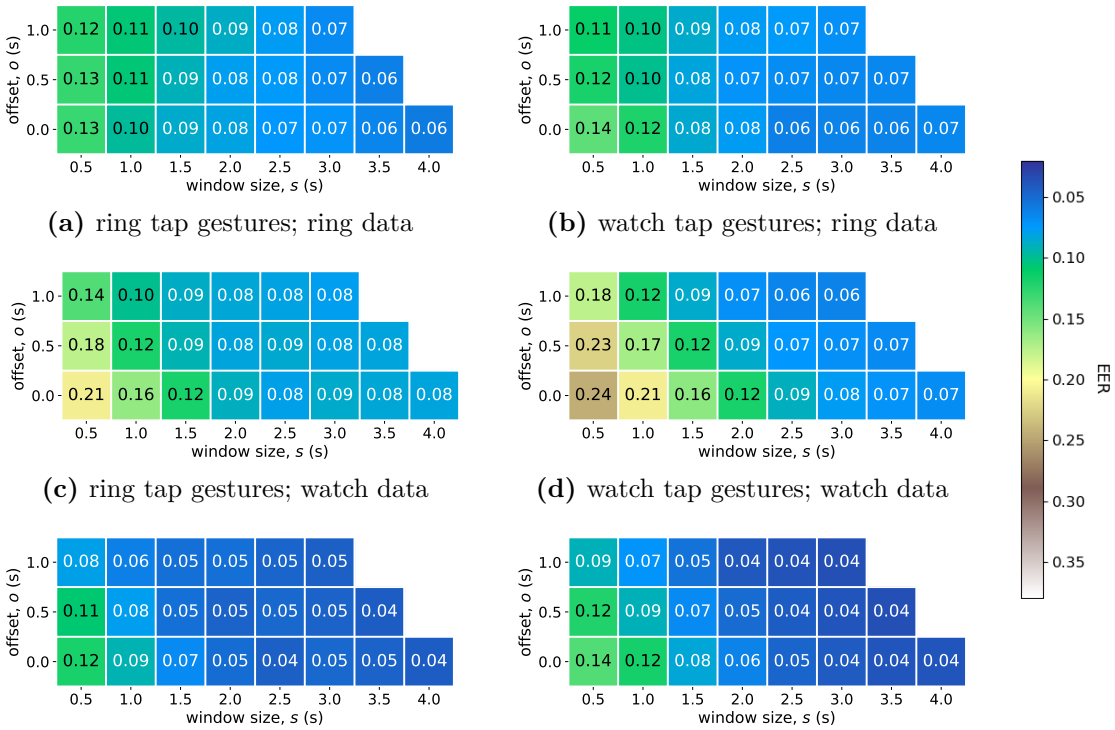
For consistency across our work, we measure performance in the same way as before. See Section 3.4.4 for details.

4.5 Results

4.5.1 Zero-effort Attack

Tap Gestures. Figure 4.6 shows the average EERs for our payment models by window size and offset. With 21 users and 6 fixed terminals, each score is the average over $21 \times 6 \times 10 = 1,260$ classifiers (see Section 4.4.3 for details).

For ring tap gestures, we saw in Section 4.4.1 that the most interesting region in the ring data occurs roughly 3 seconds before the NFC contact point and in the watch data, roughly 1.5 seconds before. We see this reflected in Figures 4.6a and 4.6c, respectively, where larger windows (or smaller windows with sufficient offset)



(a) ring tap gestures; ring data

(b) watch tap gestures; ring data

(c) ring tap gestures; watch data

(d) watch tap gestures; watch data

(e) ring tap gestures; combined data

(f) watch tap gestures; combined data

Figure 4.6: Average EERs for our payment models by window size and offset, for tap gestures made with the smart ring (left) and smartwatch (right), using data from only the ring (top), only the watch (middle), and both combined (bottom).

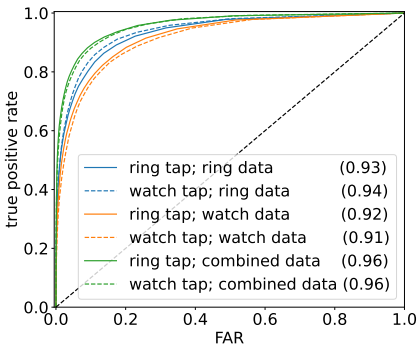


Figure 4.7: Average ROC curves for our payment models in optimum window $\{s = 2.5, o = 0\}$, using data from (i) only the smart ring, (ii) only the smartwatch, and (iii) both combined. The AUCs are given in parentheses.

that include data from the respective region achieve better results. In both cases, we find that the reaching phase is key for authentication (echoing our findings in Section 3.5.2). Figure 4.6a shows that when using data from only the smart ring our model achieves EERs as low as 0.06. Figure 4.6c shows that when using data from only the smartwatch, we get 0.08. This suggests that a smartwatch can perform

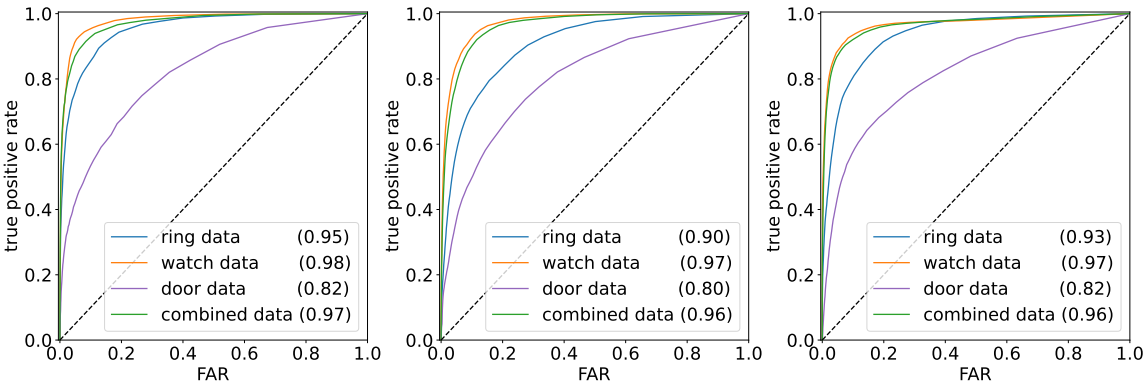
Knock Gesture	EER			
	Ring	**Watch**	**Door**	**Combined**
3-knock	0.06	**0.02**	0.17	0.03
5-knock	0.12	**0.04**	0.21	0.05
secret knock	0.09	**0.05**	0.19	**0.05**

Table 4.1: Average EERs for our access control models, using data from (i) only the smart ring, (ii) only the smartwatch, (iii) only the door-mounted sensors, and (iv) all three combined.

reasonably well as an authenticator for a smart ring payment system, in a scenario where the smart ring does not have any inertial sensors of its own (such as the Mastercard K-ring). If the data from both devices are combined, Figure 4.6e shows that we achieve EERs as low as 0.04. Favouring a smaller window size and offset to ease implementation, our optimum window parameters are again $\{s = 2.5, o = 0\}$. Figure 4.7 shows the ROC curves for these models by gesture and device in this optimum window; we achieve AUCs of up to 0.96.

When we use data from only the smartwatch, we see a pattern radiating from the bottom left corner of the grid. This shows that, when training only on data where the tapping device is near to the terminal (*i.e.*, in the alignment phase, just before the NFC contact point is found), the movements of the wrist are least discriminative between users. This is less so when using data from only the smart ring, which suggests that the finger remains active during that time, even for watch tap gestures. Figure 4.6f (as compared to Figure 4.6d) shows that including a smart ring as an additional factor can make a positive impact on the authentication of a user making watch-based payments. (Note that Figures 3.6a and 4.6d show that our approach performs consistently across our work; the former is based on watch tap gestures from 31 participants, the latter from 21.)

Knock Gestures. Across all of the participants in our user study, the average 3-knock gesture lasted 2.81 seconds, the average 5-knock, 3.32 seconds, and the average secret knock, 3.78 seconds. Table 4.1 shows the average EERs for our access control models. Each score is the average over $21 \times 10 = 210$ classifiers. Figure 4.8 shows the average ROC curves.



(a) 3-knock gestures **(b)** 5-knock gestures **(c)** secret knock gestures

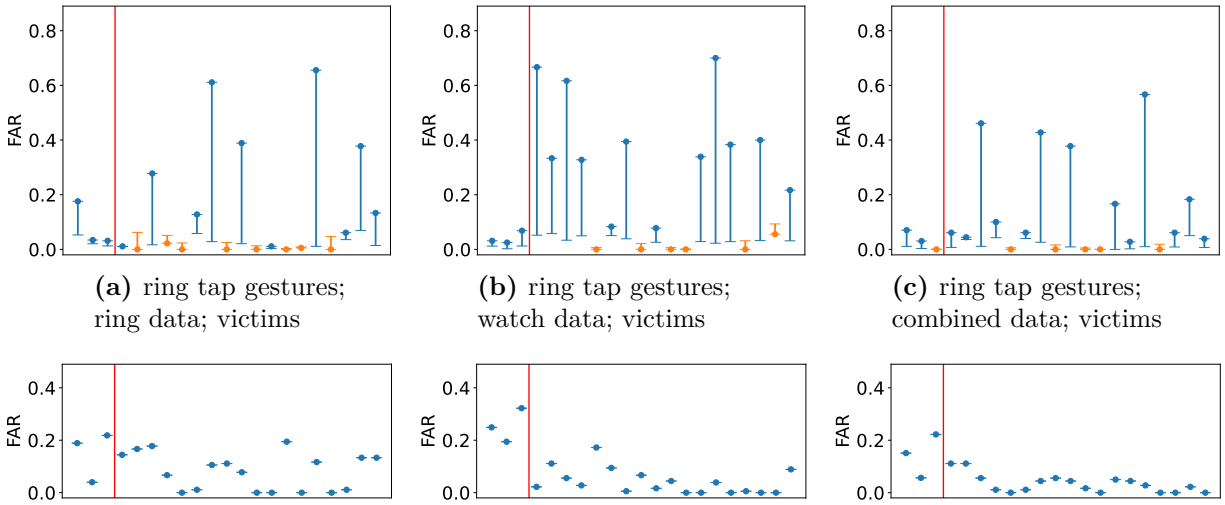
Figure 4.8: Average ROC curves for our access control models, using data from (i) only the smart ring, (ii) only the smartwatch, (iii) only the door-mounted sensors, and (iv) all three combined. The AUCs are given in parentheses.

Unexpectedly, our 3-knock model performed best, achieving an EER of 0.02 and an AUC of 0.98 using data from only the smartwatch. The weaker results of the 5-knock model might be due to some participants sometimes miscounting the number of knocks when performing a 5-knock gesture, suggesting that it is a less user-friendly gesture and leading to messier data, whereas 3-knock gestures were performed effortlessly and more consistently.

When we use data from only the smartwatch, we achieve the best error rates and class separation across all models, including those based on combined data. This suggests that wrist movements are a key discriminator in knocking, to such an extent that other factors act as pollutants. When we use data from only the door, we achieve the poorest results; while the lower sampling rate of the door-mounted sensors may have an impact, the key disadvantage of these sensors is that they lack knowledge of the user's movements surrounding each knock. Despite this sparse data, the classifiers are still able to achieve a reasonable degree of class separation, as evidenced by their average AUC of 0.81.

4.5.2 Observation Attack

Figures 4.9 and 4.10 show the results of our observation attack against our *position-known* payment and access control models, respectively. Each of the first 3 users was the victim of 1,080 ring tap impersonation attempts (18 attackers × ring tap



(a) ring tap gestures; ring data; victims

(b) ring tap gestures; watch data; victims

(c) ring tap gestures; combined data; victims

(d) ring tap gestures; ring data; attackers

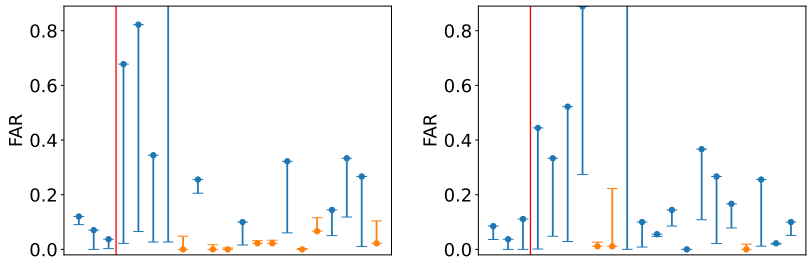
(e) ring tap gestures; watch data; attackers

(f) ring tap gestures; combined data; attackers

Figure 4.9: Results of our observation attack against our *position-known* payment models in optimum window $\{s = 2.5, o = 0\}$. The top row shows for each user as a victim the average FAR of the user-specific base model (base-FAR, flat line) and the average FAR when attacked (observation-FAR, circle); if the latter is greater, then the victim's line is coloured blue, otherwise it is orange. The bottom row shows for each user as an attacker the average FAR achieved when attempting to impersonate other users. The red line separates the first 3 users from the other 18, indicating the two groups of users who attacked each other.

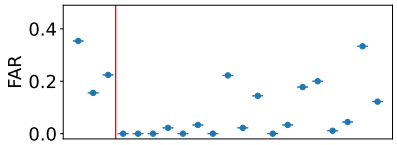
gestures on 2 terminals $\times$ 3 attempts at each gesture $\times$ 10), 540 5-knock attempts, and 540 secret knock attempts; each of the other 18 users, separated in the figures by a red line, was the victim of 180 (3 attackers), 90, and 90, respectively.

For ring tap gestures, the base model achieves average base-FARs of 0.03 when using data from only the smart ring and only the smartwatch and 0.01 when using data from both combined; when attacked, the average observation-FARs are 0.06, 0.08, and 0.05, respectively (and we find similar results for watch tap gestures). Figures 4.9a and 4.9c show that a small number of our users are lambs, where their observation-FAR is significantly greater than their base-FAR, when using data from the smart ring. Figure 4.9b seems to show that the number of lambs increases when using data from only the smartwatch; however, Figure 4.9e shows that this is due to an increase in attack success rates from attackers on the left side of the red line. This suggests that, for some attackers, the arm naturally elicits wrist movements that are more similar to the victim's when attempting to mimic

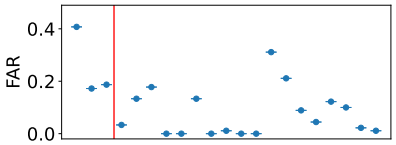


(a) 5-knock gestures; combined data; victims

(b) secret knock gestures; combined data; victims



(c) 5-knock gestures; combined data; attackers



(d) secret knock gestures; combined data; attackers

Figure 4.10: Results of our observation attack against our access control models. The top row shows for each user as a victim the average FAR of the user-specific base model (base-FAR, flat line) and the average FAR when attacked (observation-FAR, circle); if the latter is greater, then the victim's line is coloured blue, otherwise it is orange. The bottom row shows for each user as an attacker the average FAR achieved when attempting to impersonate other users. The red line separates the first 3 users from the other 18, indicating the two groups of users who attacked each other.

his finger movements, even when the finger movements are less similar. Figure 4.9c shows that the addition of the smartwatch data helps to reduce the greatest FAR deltas in Figure 4.9a, and the overall average observation-FAR, but also opens a new vector that increases the susceptibility of some users.

For knock gestures, when using data from the door-mounted sensors, smart ring, and smartwatch combined, we have average base-FARs of 0.05 for both the 5-knock and secret knock gestures and average observation-FARs of 0.08 and 0.09, respectively. Figures 4.10a and 4.10b show that we have a number of lambs, this time with greater FAR deltas. Knock gestures are notably weak against impersonation if they are loud and slow. For the secret knock, the fourth and sixth users on the right side of the red line have high base-FARs because those users chose common gesture fragments in their secret knocks. For the former (and the seventh user), the gesture was loud and slow, as evidenced by the huge observation-FARs. For the latter, curiously, the observation-FAR is much lower than the base-FAR, suggesting that the gesture was difficult to mimic despite

having commonality with other gestures. This gesture contained three fast knocks in the middle, which captured the attention of attackers only for them not to match the surrounding knocks. Figures 4.10c and 4.10d show that attentive attackers can achieve good attack success rates, but we argue that this is due to the lambs being more vulnerable—whether by poor choice of secret or poor execution (*i.e.*, loud and slow)—rather than any attacker being especially skilful.

4.5.3 Feature Informativeness

To see which features are most informative to our models, we sum the top five features of each classifier, sorted by Gini importance. (Note that, with respect to the counts, there are 6 times more classifiers for the payment model.)

For tap gestures, Table 4.2a shows that our ring tap payment models favour features derived from the GRV when using data from only the smart ring, but features derived from the accelerometer and gyroscope when using data from only the smartwatch; this echoes our findings in Section 3.5.5 and indicates that the finger moves to a position faster and remains there longer than the wrist, whose movements are smoother. The combined model, which achieves stronger results than the others (per Figure 4.6), has twice as many members in its feature vector and yet favours a similar set, re-affirming the importance of these features.

For knock gestures, Table 4.2b shows that our 3-knock access control models favour features derived from the y- and z-axes of the wearable devices; this is to be expected as these measure the movements of the hand about the wrist and towards the door, respectively. Aside from these, two notable exceptions show greater importance: `w-Acc-x-max`, which represents the maximum acceleration of the arm as it extends towards the door, and the median impact experienced by the door-mounted accelerometer, which indicates that each user struck the door with consistent force and that this is a strong discriminator between users.

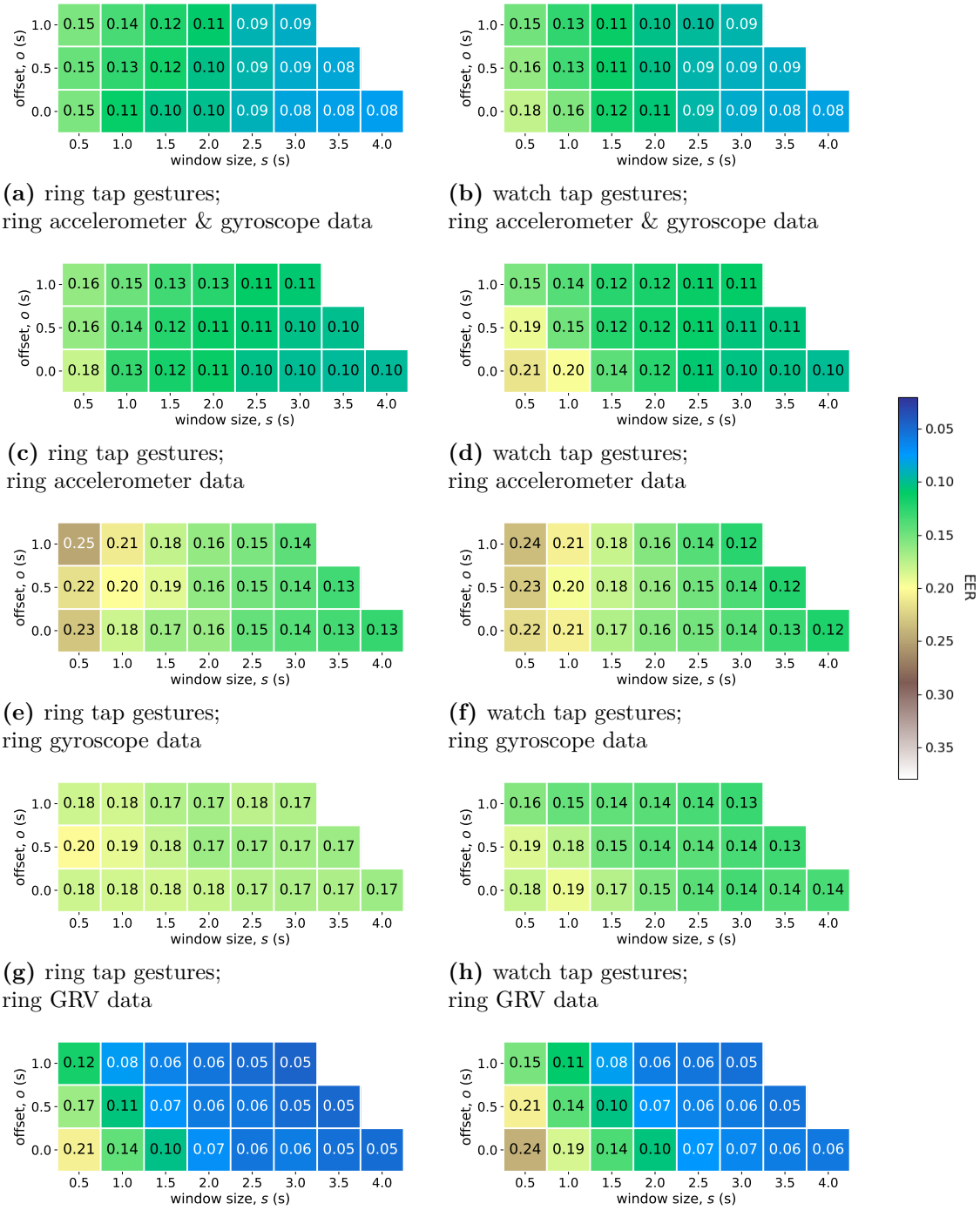
Ring		Watch		Combined	
Feature	**#**	**Feature**	**#**	**Feature**	**#**
GRV-y-med	309	Acc-x-min	297	w-Acc-x-min	185
GRV-y-mean	269	Acc-y-max	208	w-Acc-y-max	151
GRV-x-mean	239	Acc-x-max	208	r-Gyr-z-mean	145
GRV-x-med	223	Acc-z-max	189	r-Acc-x-velomax	141
GRV-x-max	222	Gyr-z-velomean	138	r-GRV-x-max	135
GRV-y-max	217	Gyr-z-mean	126	r-Acc-x-mean	128
Gyr-z-mean	192	Gyr-z-min	124	r-GRV-y-mean	122
Acc-x-mean	162	Gyr-z-disp	119	r-Acc-x-med	122
GRV-z-max	157	Acc-x-mean	117	r-GRV-x-med	116
Acc-x-velomax	155	Acc-x-velomax	116	r-GRV-y-med	116

(a) payment models; ring tap gestures

Ring		Watch		Door		Combined	
Feature	**#**	**Feature**	**#**	**Feature**	**#**	**Feature**	**#**
GRV-w-med	24	Acc-x-max	29	Acc-unf-med	100	d-Acc-y-med	22
Acc-y-disp	22	Acc-z-velomax	25	Acc-y-med	90	w-Acc-x-max	20
GRV-y-max	20	Acc-z-disp	23	Acc-z-med	72	w-Gyr-z-var	17
GRV-w-mean	19	GRV-y-var	23	Acc-x-med	51	w-Gyr-y-med	17
GRV-z-med	19	Gyr-z-velomax	22	Acc-fil-med	43	w-Gyr-z-velomax	15
Acc-z-mean	18	GRV-y-iqr	21	Acc-y-mean	39	w-Gyr-z-stdev	15
Acc-z-med	18	Acc-y-mean	21	Acc-y-max	34	w-LAc-z-disp	15
Gyr-y-velomax	18	Gyr-y-med	17	Gyr-unf-max	29	w-GRV-y-iqr	15
GRV-x-mean	18	Gyr-z-var	17	Gyr-unf-kurt	24	w-GRV-y-med	14
LAc-x-disp	17	LAc-z-disp	17	Gyr-unf-skew	24	w-Gyr-y-min	12

(b) access control models; 3-knock gestures

Table 4.2: Modal top-five features by Gini importance summed over classifiers for our payment models in optimum window $\{s = 2.5, o = 0\}$ and our access control models, using data from (i) only the smart ring, (ii) only the smartwatch, and (iii) both combined for the former and from (i) only the ring, (ii) only the watch, (iii) only the door-mounted sensors, and (iv) all three combined for the latter. Features are given in the format sensor-axis-statistic; `unf` is the energy of the unfiltered $\{x, y, z\}$ values; for models using combined data, the leading character indicates the device to which the sensor belongs (ring, watch, or door).



(a) ring tap gestures; ring accelerometer & gyroscope data

(b) watch tap gestures; ring accelerometer & gyroscope data

(c) ring tap gestures; ring accelerometer data

(d) watch tap gestures; ring accelerometer data

(e) ring tap gestures; ring gyroscope data

(f) watch tap gestures; ring gyroscope data

(g) ring tap gestures; ring GRV data

(h) watch tap gestures; ring GRV data

(i) ring tap gestures; watch accelerometer & gyroscope data

(j) watch tap gestures; watch accelerometer & gyroscope data

Figure 4.11: Average EERs for our payment models by window size, offset, and sensors, for tap gestures made with the smart ring (left) and smartwatch (right), using data from only the ring (top four rows) and only the watch (bottom).

Knock Gesture	EER											
	Ring				Watch			Door		Combined		
	A&G	Acc.	Gyr.	GRV	A&G	Acc.	Gyr.	Acc.	Gyr.	A&G	Acc.	Gyr.
3-knock	0.07	0.09	0.09	0.16	**0.01**	**0.02**	0.04	**0.17**	0.23	**0.02**	**0.03**	0.04
5-knock	**0.11**	0.14	**0.12**	0.24	**0.04**	0.05	0.08	0.22	0.26	**0.04**	0.06	0.06
secret knock	**0.07**	0.10	0.09	0.20	**0.05**	**0.05**	0.07	0.20	0.26	**0.04**	0.06	0.06

Table 4.3: Average EERs for our access control models by sensors, using data from (i) only the smart ring, (ii) only the smartwatch, (iii) only the door-mounted sensors, and (iv) all three combined. For each device, A&G is using data from only the accelerometer and gyroscope, Acc. is from only the accelerometer, Gyr. is from only the gyroscope, and GRV is from only the GRV.

4.5.4 Sensor Selection

We collected motion data from all of the inertial sensors available on our devices. To assess the feasibility of our approach on devices with fewer sensors, we train a set of sensor-specific models in which each classifier is trained and tested on data from a subset of sensors.

For tap gestures, Figure 4.11 shows the EERs for our payment models when using data from fewer sensors (*cf.* Figure 4.6, which uses data from all sensors). When using data from only the smart ring, we see poorer results whenever we remove the GRV sensor; this is to be expected given the importance of its features. We see poor results when using data from only the GRV, which suggests that it provides necessary but not sufficient information and that other features, such as `Acc-x-mean`, supply crucial supporting information for discriminating between users. When using data from only the smartwatch, especially only its accelerometer and gyroscope, we see improved results compared to Figures 4.6c and 4.6d that echo our findings in Section 3.5.6. When using data from both combined, the results remain roughly unchanged, relying on ring-based features when the data from the GRV of the ring is included and watch-based features when not.

For knock gestures, Table 4.3 shows the EERs for our access control models when using data from fewer sensors (*cf.* Table 4.1, which uses data from all sensors). The emboldened entries in Table 4.3 indicate the scores that are better than or equal to those in Table 4.1 while using data from fewer sensors.

Terminal	EER		
	Ring	**Watch**	**Combined**
1	0.07	0.06	0.04
2	0.09	0.11	0.07
3	0.09	0.08	0.07
4	0.08	0.05	0.03
5	**0.06**	**0.04**	**0.02**
6	**0.07**	**0.04**	**0.03**
F	0.10	0.11	0.09
agnostic	0.07	0.08	0.04

(a) ring tap gestures

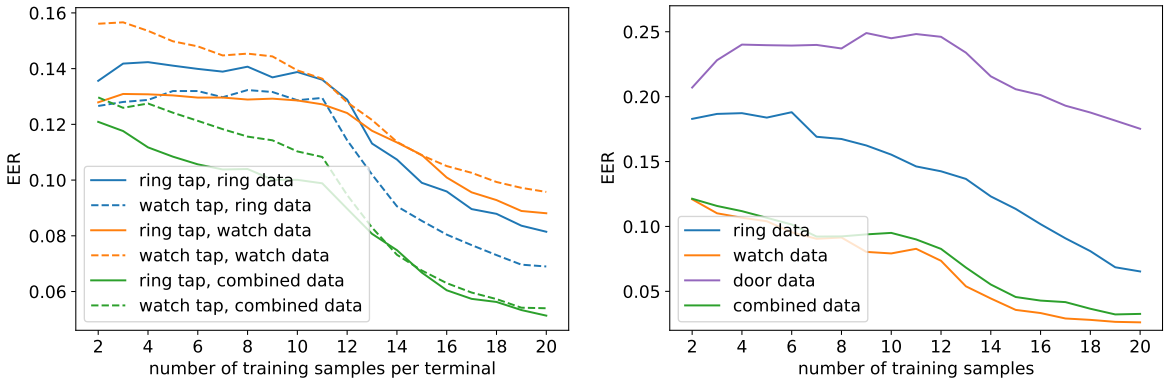
Terminal	EER		
	Ring	**Watch**	**Combined**
1	0.12	0.08	0.05
2	0.10	0.09	0.07
3	0.09	0.09	0.07
4	0.08	0.06	0.05
5	0.06	**0.06**	**0.03**
6	**0.04**	**0.04**	**0.02**
F	0.07	**0.05**	**0.04**
agnostic	0.06	0.09	0.05

(b) watch tap gestures

Table 4.4: Average EERs for our *position-specific* payment models in optimum window $\{s = 2.5, o = 0\}$, using data from (i) only the smart ring, (ii) only the smartwatch, and (iii) both combined. Our *position-agnostic* results are included for comparison.

4.5.5 Terminal Positions

We collected tap gestures performed on a range of terminals. To evaluate our approach in a single-terminal scenario, we train a set of position-specific models in which each classifier is trained and tested on data from a single terminal. Table 4.4 shows the average EERs for our *position-specific* payment models. We see a little improvement when restricting our system to a single terminal. For Terminals 5 and 6, we see slightly better results and these were the most comfortably positioned terminals for the majority of participants, who wore the devices on their left arm (indeed, if we reconstruct our models using data from only those users wearing the devices on their left arm, the relative gains are greater still). Likewise, for the freestyle terminal, we see better results for watch tap gestures, since the watch was more awkward to tap than the ring and this terminal accommodated smoother movements. However, the freestyle terminal had the opposite effect on ring tap gestures, likely because users tilted and moved the terminal a shorter distance when interacting with it when using the ring than when using the watch, eliciting a sloppier punch gesture. We again find that user comfort has a beneficial impact on our results.



(a) payment models **(b)** access control models; 3-knock gestures

Figure 4.12: Average EERs for our payment models in optimum window $\{s = 2.5, o = 0\}$ and our access control models if trained on different numbers of enrolment samples. For the payment models, each classifier is trained on six terminals.

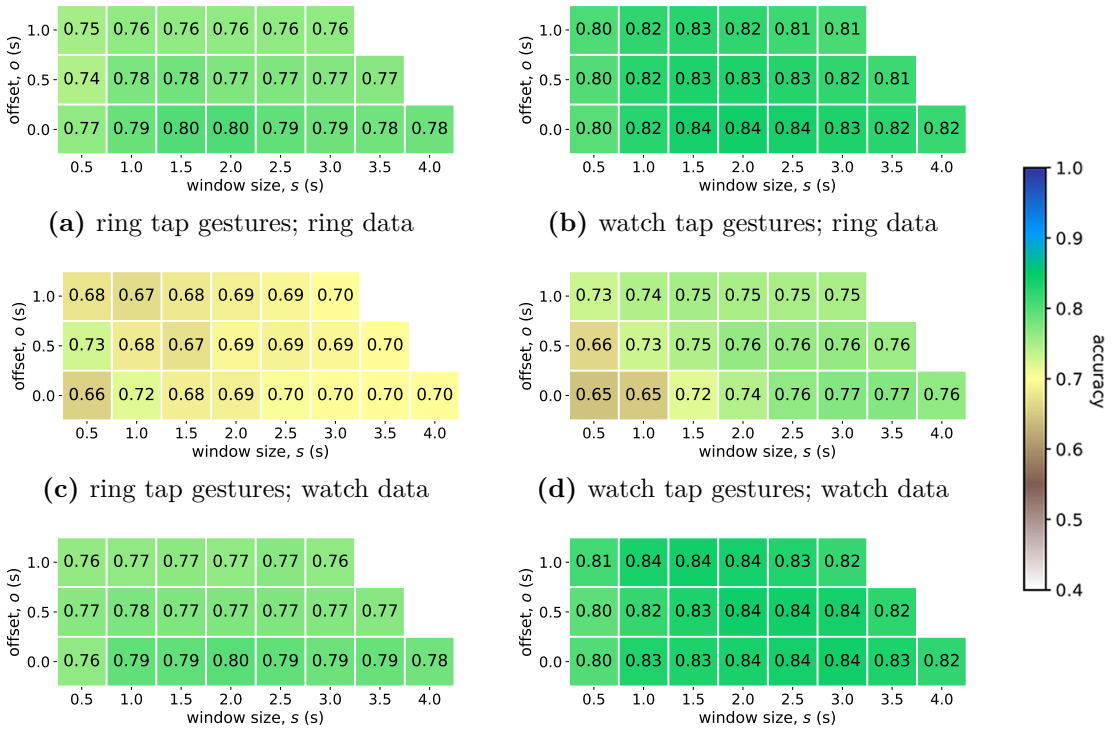
4.5.6 Enrolment Parameters

Behavioural biometric systems typically entail a burdensome enrolment phase. To evaluate the extent to which we can expedite the enrolment phase, we compare the average EERs of our authentication models when the classifiers are trained on a smaller positive class.

For tap gestures, Figure 4.12a shows that our payment models can authenticate the user with EERs as low as 0.12 when trained on just twelve of the user's tap gestures (spread evenly over six terminals), which can be performed in less than a minute.

For knock gestures, Figure 4.12b shows that, when we include data from the smartwatch, our 3-knock access control models can authenticate the user with EERs as low as 0.12 when trained on just two 3-knock gestures (the access control models for the other knock gestures show a similar pattern), which can be performed in a few seconds.

In both cases, we see that the EERs improve as more samples are included in the training set; this suggests that an update mechanism might benefit the models over time, relaxing upfront requirements and incorporating subsequent gestures as the system is used.



(a) ring tap gestures; ring data **(b)** watch tap gestures; ring data

(c) ring tap gestures; watch data **(d)** watch tap gestures; watch data

(e) ring tap gestures; combined data **(f)** watch tap gestures; combined data

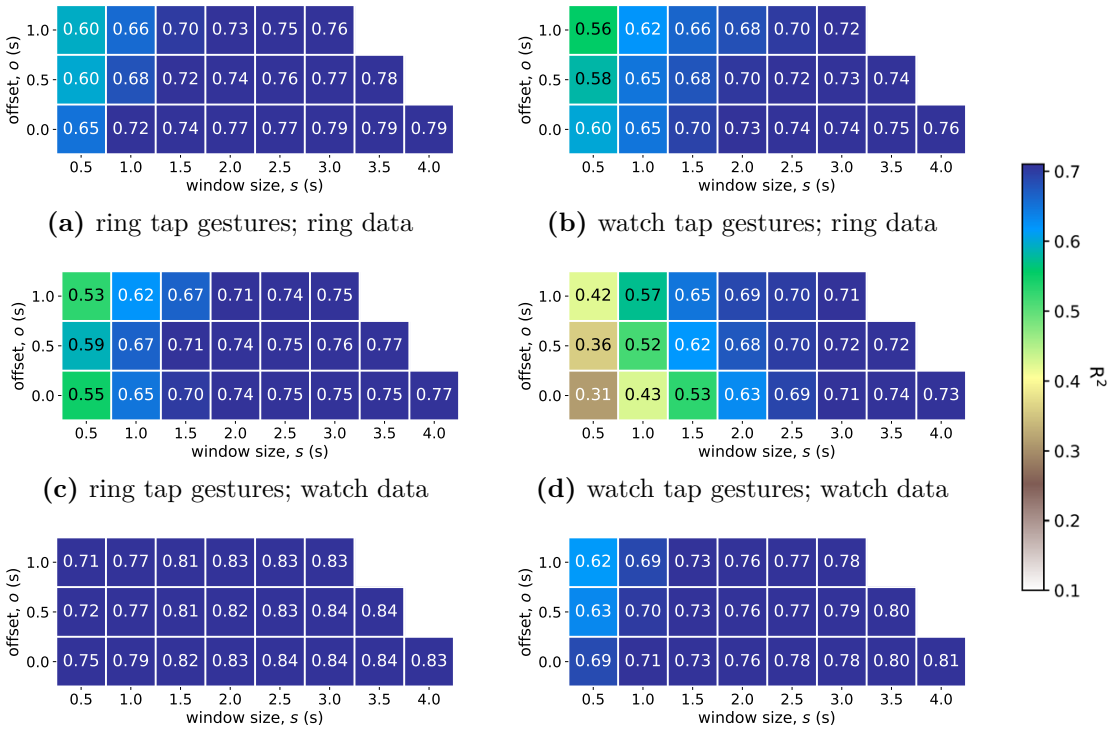
Figure 4.13: Average accuracy scores for our sex inference models by window size and offset, for tap gestures made with the smart ring (left) and smartwatch (right), using data from only the ring (top), only the watch (middle), and both combined (bottom).

4.5.7 Privacy Implications

We have shown that even a small window of inertial sensor data can be used to authenticate the user. To investigate whether any private information, such as personal characteristics, is exposed via inertial sensor data while performing a tap or knock gesture, we train a set of models to infer various traits. Figures 4.13 and 4.14 show the average accuracy and R^2 scores, respectively, for our inference models.

For age range, our models achieve average accuracy scores of between 0.56 and 0.66 in all windows for all gestures, indicating that while there is some correlation we cannot infer the age range of users.

For sex, we find that the inclusion of data from the smart ring increases the accuracy of the models in inferring the sex of the user. For tap gestures, we see that the pattern in the results in Figure 4.13d is consistent with that in Figure 3.13a. We see in Figure 4.13b that using ring data with watch tap gestures is the most



(a) ring tap gestures; ring data

(b) watch tap gestures; ring data

(c) ring tap gestures; watch data

(d) watch tap gestures; watch data

(e) ring tap gestures; combined data

(f) watch tap gestures; combined data

Figure 4.14: Average R^2 scores for our height inference models by window size and offset, for tap gestures made with the smart ring (left) and smartwatch (right), using data from only the ring (top), only the watch (middle), and both combined (bottom).

effective combination for inferring sex, regardless of window parameters, where our model achieves average scores of between 0.80 and 0.84. For knock gestures, we achieve average scores of between 0.75 and 0.80 (except when using data from only the door-mounted sensors, where our model achieves average scores of 0.60).

For height, our models perform well. For tap gestures, we see that the pattern in the results in Figure 4.14d is again consistent with that in Figure 3.14a, but that here the results are stronger because the classifiers were more successful at identifying influential features. The modal top-five features consisted of `Gyr-unf-mean` and features derived from the velocity along the x-axis of the accelerometer and gyroscope. This suggests that the slightly greater downward movement required of a taller user as he bends towards the terminal elicits a proportionally faster movement of the wrist. For knock gestures, our models achieve average R^2 scores of up to 0.87 when using data from only the smart ring, 0.93 when using data from only the smartwatch, and 0.94 when using all three devices (including the

door-mounted sensors) combined. By examining the feature importances, we see a preponderance of maxima, minima, and features derived from the energies (both filtered and unfiltered) of the gyroscope. Given that taller users have a longer reach and stood slightly further away from the door when knocking, it is likely that the models are detecting the differences in the arc of the arm as it moves and rotates towards the door.

4.6 Discussion

3-knock Impersonation. Table 4.1 shows that we achieve our best overall results from the 3-knock gesture. In our preliminary testing, this was not the case, so we chose not to include it in the impersonation exercise of our user study due to time restrictions. This is regrettable, as 3-knock gestures in the observation attack may have yielded more interesting results than 5-knock gestures.

Secret Knock Feasibility. The three drawbacks of using an explicit gesture biometric are that it (i) takes time to perform, (ii) must be remembered by the user, and (iii) must be kept secret and performed discreetly. In our user study, 76% of participants remembered their secret knock, with an average of 4.8 days between their sessions, which suggests that a secret knock is memorable. The average length of a secret knock gesture was 3.78 seconds, although we influenced this by restricting its size to be between three and six knocks. This restriction also limited the key space and resulted in each secret knock gesture consisting of permutations of 1-, 2-, or 3-knock fragments knocked at a certain cadence—despite this, we had only one collision, where two users chose the same secret knock. We found that these two users were not good at impersonating each other because, once they heard the familiar gesture, they performed their own version of it rather than listening carefully to the subtle differences. In a larger user set, we would expect to see more collisions; however, the aforementioned finding suggests that this would not necessarily weaken the system. The chief weakness of a secret knock is that its sound is difficult to conceal; however, as we show in Figure 4.10, only a few of the users in our user study were highly susceptible to having their secret

knocks impersonated; this suggests that best practice recommendations may be able to mitigate this weakness.

Smart Door Feasibility. Figure 4.8 shows that we are reasonably able to distinguish between users even when we use data only from the door-mounted sensors. This is a promising result that presents the possibility of a smart door system that can authenticate (and perhaps identify) the user using only its own embedded inertial sensors, without requiring the user to hold or wear any smart devices. We leave this project to future work.

Sampling Rates. We found that the difference between models trained and tested on smart ring data with sampling rates of 100 Hz and 50 Hz was negligible and, in each case, a similar set of features dominated in importance. This suggests that our selection of features is robust against downsampling and that using higher resolution sensors (with the same features) would not yield much improvement.

User Feedback. All of the participants in our user study found ring tap gestures to be easier and more comfortable to perform than watch tap gestures, due to the manoeuvrability of the hand compared to that of the wrist. Some terminal positions were more awkward than others, depending on the height of the user and the arm upon which the devices were worn. One female participant, who wore the devices on her right arm, commented that she considered the smartwatch used in this study to be a men's watch due to its weight and bulk and that, while she would normally wear a smaller watch on her right (dominant) wrist, she would have to wear this one on her left wrist for daily use because she would find it obstructive otherwise—*i.e.*, the model of watch used in this study likely influenced the choice of arm upon which it and then also the smart ring were worn.

4.7 Related Work

Tap Gestures. The use of inertial sensor data to authenticate the user by tap gestures in tap-and-pay systems was proposed by Shrestha *et al.* [131] for smartphone-based systems, achieving an F-measure score of 0.93 in a user study of 20 participants. We showed in Chapter 3 that a similar principle worked for

smartwatch-based systems, achieving an F-measure score of 0.88 and EER of 0.07 in a user study of 31 participants. In this chapter, we achieved an EER of 0.06 with a smart ring-based system (0.04 when combined with a smartwatch) in a user study of 21 participants. The use of a smartwatch, due to the physiology of the arm, introduces the challenge that the sensor axes frequently change reference frames because the device changes orientation during the tap gesture. We found that the use of a smart ring sits between the two in terms of complexity: the smartphone requires no major change in orientation and is the trivial case, the smart ring requires only a single change because the finger is easily manoeuvred towards the terminal, and the smartwatch orientation is changed frequently during the tap gesture.

Knock Gestures. The use of inertial sensor data to authenticate the user by explicit gestures has been explored on smartwatch-based systems. Yang *et al.* [160] considered four different full arm gestures, such as air-drawing a circle, and achieved EERs of between 0.03 and 0.23 in a user study of 26 participants, depending on user and gesture. Liang *et al.* [80] considered a punch gesture and achieved an EER of 0.04 in a user study of 20 participants. Yu *et al.* [163] considered 9 finger and hand gestures, such as making a fist, and achieved EERs of between 0.01 and 0.03 in a user study of 65 participants. We are the first to investigate the use of knock gestures for user authentication, whether implicit or explicit, either with a smartwatch, with a smart ring, or with sensors mounted on the door.

Smart Rings. Few authors have considered the use of smart rings in authentication use-cases. Sen *et al.* [125] proposed the use of a smart ring that is capable of producing a vibration to bootstrap a communication channel with another device held in the same hand that has an accelerometer to detect the vibration. Liang *et al.* [81] showed that inertial sensor data from a smart ring can be correlated with mouse movements to verify the continued presence of the user at a workstation. Roshandel *et al.* [122] showed that finger motion data can be used to authenticate the user of a smart ring while air-writing a signature or tracing it on a flat surface with the finger; they did not disclose their error rates but did achieve strong accuracy scores (notwithstanding this being an inappropriate performance metric)

in a user study of 24 participants. To the best of our knowledge, we are the first to propose the use of inertial sensors on a smart ring to authenticate a user via implicit gesture biometrics.

4.8 Conclusion

In this chapter, we showed that inertial sensor data from a smart ring can be used to authenticate the wearer. In a mobile payment context, we showed that a smart ring user can be authenticated implicitly with a single tap gesture with an EER of 0.04. We also showed that inertial sensor data from a smart ring can be used to authenticate the user when making a smartwatch payment, and *vice versa*, opening the possibility for either device to be used as an implicit second factor to support the other. This is particularly applicable in cases where the other device lacks inertial sensors of its own. In an access control context, we showed that a smart ring user can be authenticated (explicitly) with a single knock gesture with an EER of 0.06 (or 0.01 with a smartwatch). Our results further show that a smart door, which has inertial sensors embedded inside it, may be able to authenticate or identify the user knocking on it without the need for other devices. We demonstrated that our models can be trained quickly (on very few samples) and that they provide resistance against an active attacker who observed (and heard) the victim's gestures.

5

CableAuth: User Authentication over a Physical Channel

5.1 Motivation

An electric vehicle (EV) is a mode of transportation that is powered by one or more electric batteries. To recharge the battery, the driver parks the EV next to a charging station, pulls a charging cable from the station to the vehicle, connects it, and makes a payment to initiate the charging procedure. A variety of payment systems have been deployed by different station operators that include tap-and-pay systems, smart keys (more commonly known as *smart cards* in this context) issued by the operator, and QR codes printed on the station that link to a payment portal when scanned by a smartphone.

The current generation of EV batteries are slower to charge and must be recharged more often than the refuelling of vehicles with a combustion engine, so user-friendliness and convenience are of paramount importance throughout the charging process. Zero-interaction payment schemes, such as AutoCharge [104] and Plug & Charge [98], enable authentication information to be sent from the vehicle to the charger over the charging cable when it is connected. This information is linked to a payment instruction, typically administered by an application on a smartphone, to facilitate automatic billing. AutoCharge uses the MAC address of the vehicle as a unique identifier to authorise the payment (however, it was shown that the MAC address can be extracted wirelessly during an ongoing charging

session [11]). Plug & Charge stores digital certificates on the vehicle and uses TLS to secure the communication channel (however, the protocol does not mandate secure storage, leaving it vulnerable in some cases to an attacker with physical access to the vehicle [10]). These schemes treat the vehicle as a token and authenticate the vehicle rather than the user; this could allow a thief to charge a stolen vehicle or an attacker to masquerade a different vehicle as the victim's and to charge it at the victim's expense. This means that there is a need for some additional form of user authentication—and it needs to be implicit so as not to inconvenience the user. We showed in Chapter 3 that inertial sensors on a smartwatch can be used to authenticate the user implicitly. We take inspiration from this work and develop a system integrated in the handle of the charging cable using a similar underlying principle.

The implementation of our proposed system would require changes to be made to existing infrastructure. Furthermore, some charging stations allow users to connect their own portable charging cables, which may be difficult to incorporate into our system. We resolve these problems by also considering the use of a smartwatch as an interim or alternative solution for collecting gesture data as the user handles the cable.

In this chapter, we investigate the extent to which inertial sensors can be used to authenticate the user implicitly when connecting a charging cable from a charging station to a vehicle. We conduct a user study to collect unhook and plug-in gesture data. We attach an IMU to the handle of the charging cable and we equip each user with a smartwatch worn on the same arm that he uses to handle the cable. Using this data, we evaluate the effectiveness of using data from either the handle or the watch for each type of gesture to authenticate users during the charging procedure.

5.2 System Design

5.2.1 Design Goals

The objective of this chapter is to investigate the use of inertial sensors embedded in the handle of a charging cable and on a smartwatch for user authentication purposes in the context of charging an EV. We consider unhook and plug-in gestures, as measured by the inertial sensors, made by the user when interacting with a charging cable at an EV charging station. We pose the following questions.

- To what extent can gestures made by the user as he connects a charging cable to an EV, as measured by inertial sensors in the handle of the cable, be used to authenticate the user implicitly during a charging session?
- To what extent can gestures made by the user as he connects a charging cable to an EV, as measured by inertial sensors on a *smartwatch* worn on the same arm that he uses to handle the cable, be used to authenticate the user implicitly (in cases where the handle does not have inertial sensors of its own) during a charging session?
- To what extent can the explicit authentication actions required of the user be removed to improve the user experience of a charging session?

5.2.2 Design Considerations

Gesture Boundaries. When connecting a charging cable to an EV, the *handling interval* is the period of time during which the user handles the cable, where the handle of the cable is unhooked from its enclosure, moved to the vehicle, and then the connector at the end of the handle is plugged into the charging port of the vehicle. Since the space between the charging station and the parked vehicle introduces unknown variables (in terms of distance and obstacles), we control this by restricting our focus either to the unhooking action or to the plugging-in action and we retrieve a separate gesture for each. We use sliding time windows to find the optimum parameters for each gesture and we avoid gesture overlap in a handling interval by imposing hard constraints during segmentation.

Also, see **Dominant Hand** in Section 4.2.2 and **Privacy Risk** in Section 3.2.2.

5.2.3 System Model

We consider a system model in which a user has parked an EV next to a charging station in order to charge its battery. To initiate the charging procedure, we assume that the user first performs an *unhook gesture*, where he removes the handle of the charging cable from its enclosure, and then takes it to his vehicle and performs a *plug-in gesture*, where he plugs the cable into the charging port of the EV. Once the cable is securely plugged into the EV (*i.e.*, when the circuit between the station and the vehicle is closed and there is no further movement of the handle), we consider the charger to be connected and ready to begin charging pending authorisation from the payment provider. We assume that the handle has an accelerometer and a gyroscope embedded within it and that we have access to the data generated by these sensors. Due to the high current flow in a DC fast-charging session, modern charging cables are equipped with temperature sensors that are regularly polled by the charging station—we assume that the same communication channel can be used to access the inertial sensor data.

Using the motion data collected from the handle, we create two separate models to verify the identity of the user: one uses the unhook gesture and the other uses the plug-in gesture. Since the payment provider must decide whether to approve the charging session, the unhook gesture would be the preferable choice to minimise any delay to the system. We assume that we have access to the relevant timestamp (either at the removal of the handle or at the closure of the circuit, respectively) to enable the retrieval of the associated gesture from the inertial sensor data.

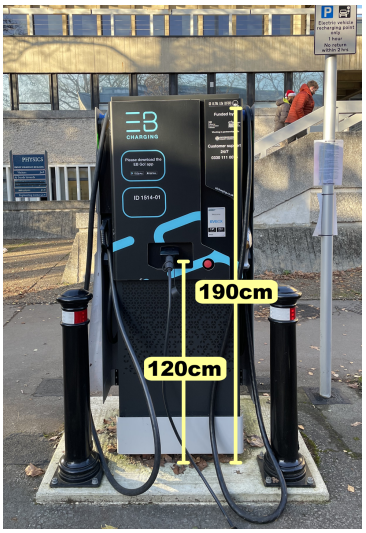
The widespread deployment of our approach would require changes to be made to existing hardware—namely, the embedding of an IMU into the handle of each hard-wired charging cable. As an interim solution, and to ensure compatibility for cables that do not get upgraded and for user-owned portable cables, we also consider a parallel model using a smartwatch worn on the same arm that is used to handle the cable. We assume that the watch has an accelerometer and a gyroscope and that we have access to the data generated by these sensors.

The principal goal of this work is to show that the unhook and plug-in gestures are biometrics capable of authenticating the user implicitly. For ease of implementation, we assume that the user is already registered with an existing authorisation system, such as AutoCharge, that can identify his vehicle to the payment provider. We assume that this system is hardened by requiring that the user authenticate himself via a smartphone application to initiate a charging session. We assume that the phone can act as the central point to which the other components can send their sensor data for processing and on which the user's template can be stored securely. With this in place, we measure the gains provided by our approach as the number of authentication actions that can be avoided by authenticating the user implicitly as he handles the cable.

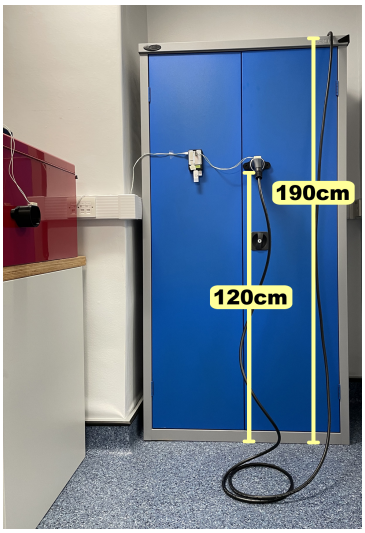
5.2.4 Threat Model

We consider two types of adversary. The first type of adversary has possession of a legitimate user's EV (and any authorisation token that may be required to initiate a charging session), has unlocked it, and is attempting to charge it at a charging station. For the smartwatch model, we also assume that the adversary has possession of the watch, has unlocked it, and is wearing it during the attack. This adversary may have (maliciously) stolen the vehicle (and associated components) or (benignly) borrowed it. The second type of adversary has possession of the credentials used by a legitimate user's EV to identify itself to the charging station and is attempting to charge a different vehicle using those credentials. This adversary may have stolen, copied, or forged a smart key, digital certificate, or unique identifier perhaps by having prior physical access to the legitimate user's vehicle, by wireless eavesdropping [11], or by relay attack [24]. In all cases, our goal is to authenticate the legitimate user and to reject the adversary by using only unhook or plug-in gestures. We treat this as a zero-effort attack.

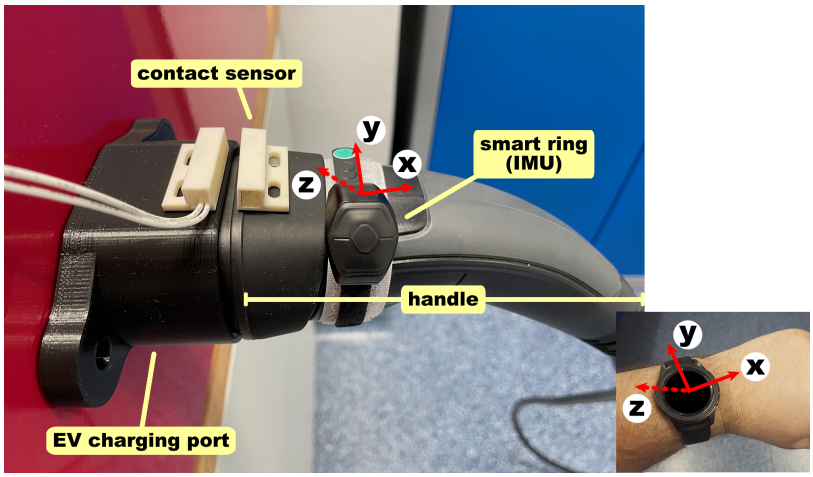
In this work, we concentrate on the extent to which gesture biometrics can be used to defend against these attacks. We do not consider threats to other components in the charging or payment systems, tampering of devices, poisoning of biometric templates, malware, or denial of service attacks.



(a) real-world charging station



(b) replica charging station



(c) charging cable handle with smart ring attached

Figure 5.1: A charging station shown for reference and the equipment used in our experiment: a replica charging station with enclosure, a replica EV with charging port, and a charging cable and handle with a smart ring attached for IMU data collection and a magnetic contact switch attached for timestamp collection. Inset: our smartwatch worn on the left wrist with the sensor axes shown.

5.3 Experiment Design

5.3.1 Experiment Overview

To evaluate the extent to which handle and wrist motion data can be used to authenticate users, and to compare the two, we designed and conducted a user study to collect data. Our experiment consisted of a replica charging station, a replica EV charging port, a charging cable, and a Samsung Galaxy Watch running

the Tizen 4.0 operating system. We attached a Genki Wave smart ring to the handle of the charging cable to collect inertial sensor data and a magnetic contact switch to enable the collection of timestamps whenever the cable was removed from the charging station or connected to the vehicle. Corresponding contact sensors were placed on the charging station enclosure and the EV charging port and each was connected to a Raspberry Pi that listened for events and recorded their timestamps. We collected handle motion data from the handle of the cable, wrist motion data from the smartwatch worn by the user, and timestamps from the contact sensors (with all clocks synchronised). Figure 5.1 shows our apparatus.

5.3.2 Cable Enclosures

To emulate a real-world EV charging setting, we used a Type-2 charging cable (including its handle) and we set up a replica charging station and vehicle. We used a 3D printer to manufacture two plastic cable enclosures matching the size and shape of the connector at the end of the handle to act as the charging station cable enclosure and the vehicle charging port. We took measurements from a DC fast-charging station (as shown in Figure 5.1a) and replicated it in our experiment by fixing the cable enclosure at a height of 120 cm and the cable feed at 190 cm. For our EV, we fixed the height of the charging port to match that of a Volkswagen ID.3 (a common EV model) and ensured that it was positioned relative to the charging station as if parked beside it.

5.3.3 Devices and Sensor Modules

To collect handle motion data, we attached a smart ring to the top of the handle (as shown in Figure 5.1c). We chose to use a ring because it had a small form factor and added negligible weight to the handle. This enabled us to emulate an integrated IMU sensor in the handle as closely as possible without adding any other factors that might influence the handling of the cable. We treat the inertial sensors on the ring as an integral part of the handle and refer to their data as the *handle data*. We wrote a data collection script in Python using the open source library provided by the developers of the ring.

To collect wrist motion data, we had the user wear a smartwatch on the same arm that he used to handle the cable. We built a data collection application and installed it on the watch using the Tizen Studio IDE.

From each of these devices, we collected timestamped data from the following four inertial sensor data streams: the *accelerometer*, the *gyroscope*, the *linear accelerometer*, and the *GRV*. We collected this data with sampling rates of 100 Hz for the handle and 50 Hz for the smartwatch.

The sensors measured motion along three axes that were set relative to the frame of each device (as shown in Figure 5.1c).

5.3.4 User Study

To collect our data, we conducted a user study that was reviewed and approved by our Computer Science Departmental Research Ethics Committee. We recruited 20 participants, including staff, students, and members of the public. Each participant attended three data collection sessions on the same day, performing 10 charges in each session. The first and second sessions occurred together, separated by a short break lasting roughly 2 minutes; the third session occurred after a longer break lasting at least 15 minutes.

The participant was instructed to stand in front of the charging station, unhook the cable from the charging station, plug it in to the vehicle, leave the room briefly and then come back, unplug the cable from the vehicle, return it to the charging station, and then leave the room again. This constituted a single charge. Aside from this, we did not prescribe any constraints on positioning as we wanted the user to stand and interact with the cable naturally as through charging in a real-world setting.

User Statistics. Of our 20 participants: 12 were male and 19 handled the cable with (and so wore the smartwatch on) the right hand.

5.4 Methods

5.4.1 Data Processing

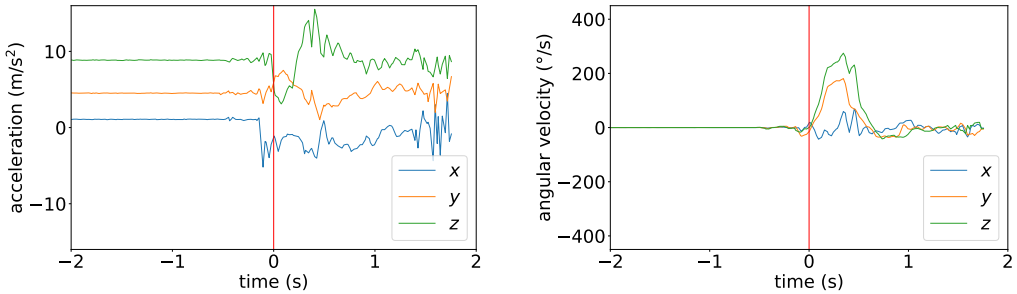
We collected time-series data from four inertial sensors on our devices. At time t, each accelerometer, gyroscope, or linear accelerometer datum is given in the form (t, x, y, z) and each GRV datum is given as a quaternion in the form (t, x, y, z, w).

We express an unhook or plug-in gesture using inertial sensor data within a time window. To retrieve the gestures for each user, we segment 4-second blocks of sensor data by using the associated contact sensor event timestamps as the midpoint of each window. We found that a 4-second window size was sufficient to encapsulate the entirety of each gesture. To investigate optimum gesture parameters, we compare the performances of gestures bounded by different window sizes and offsets, as explained in Section 3.4.1.

Figure 5.2 shows the sensor data for an exemplar unhook gesture. Using data from only the handle, we infer that the user touched the handle roughly 0.4 seconds before unhooking it from its enclosure. The peaks in acceleration just before the contact sensor event show the cable being pulled from its enclosure and the bigger peaks after it, along with the peaks in angular velocity, show the arcing movement as the handle was moved towards the vehicle—especially in the z-axis, where the height of the handle changed briefly to conform to the height of the user and then again to the height of the vehicle charging port at its destination. Using data from only the smartwatch, we infer from the trough in acceleration in the x-axis just before the contact sensor event that the user's hand moved quickly towards the handle, stopped to take hold of it, and then pulled it away.

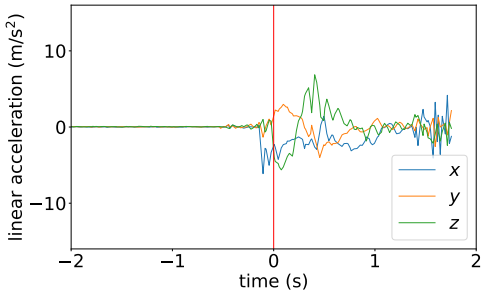
5.4.2 Feature Extraction

For consistency across our work, we reduce each gesture to a feature vector that contains the same 220 features as before. See Section 3.4.2 for details.

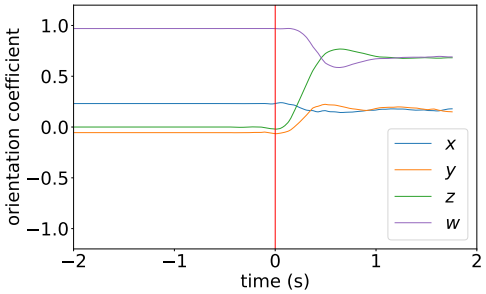


(a) handle accelerometer data

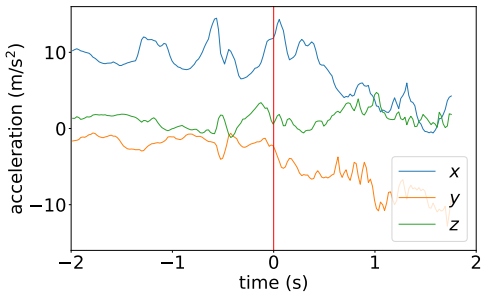
(b) handle gyroscope data



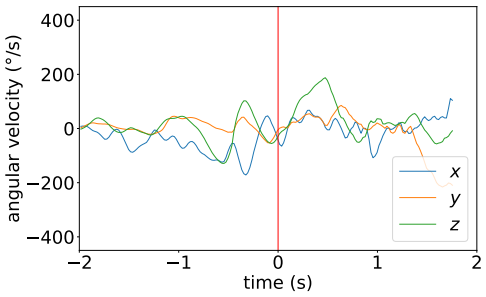
(c) handle linear accelerometer data



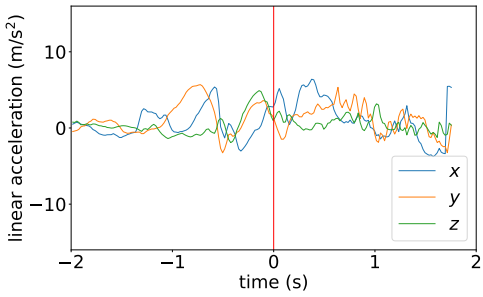
(d) handle GRV data



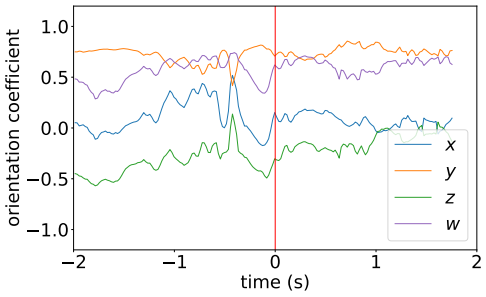
(e) watch accelerometer data



(f) watch gyroscope data



(g) watch linear accelerometer data



(h) watch GRV data

Figure 5.2: Plots of the same unhook gesture, showing data 2 seconds before the contact sensor event and 2 seconds after from the four inertial sensors on the handle (top four) and the smartwatch (bottom four) used to collect our data. The contact sensor event is set at time 0 (indicated with a red line).

5.4.3 Classification

For consistency across our work, we use random forest classifiers in each of our models and we find the average results over ten differently seeded forests. See Section 3.4.3 for details.

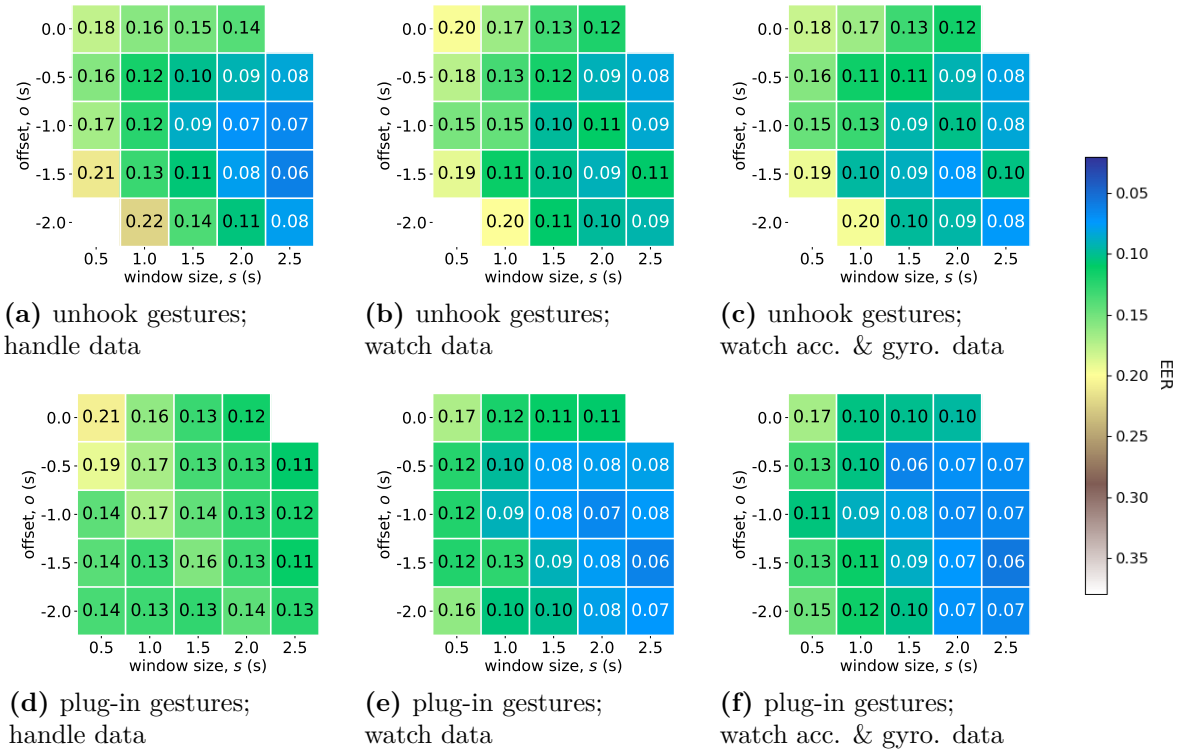
Unhook Model. For the unhook gesture, we train a set of classifiers that are user-dependent. In each classifier, we take a given user's unhook gestures as the positive class and all other users' unhook gestures as the negative class. We ensure that the training data temporally precedes the testing data by taking the unhook gestures collected in users' first and second data collection sessions as training data and those collected in the third session as testing data. We ensure that each classifier is trained in a leave-one-out manner to ensure that the model is generalised. We train separate models using data from only the handle, only the smartwatch, and (based on findings from previous chapters) only the accelerometer and gyroscope of the smartwatch.

Plug-in Model. For the plug-in gesture, we train a set of classifiers in the same way as for the unhook gesture.

Composite Model. For investigative purposes, we train a set of user-dependent classifiers in which each is trained and tested on samples that consist of both the unhook and plug-in gestures made in the same handling interval. (Note that we disregard any data in the handling interval that is not segmented to either of these two gestures, as discussed in Section 5.2.2.) Here, each sample is a composite gesture formed of two sets of features, one from its unhook gesture component and the other from its plug-in gesture component, which we concatenate together to form a feature vector that contains 440 features. To investigate the optimum parameters for the composite gesture, we treat each unhook and plug-in gesture component as independently bounded by different window sizes and offsets.

5.4.4 Performance Metrics

In our models, the *true positives* is the number of times that the positive class (*i.e.*, the legitimate user) is correctly accepted; the *true negatives* is the number



(a) unhook gestures; handle data

(b) unhook gestures; watch data

(c) unhook gestures; watch acc. & gyro. data

(d) plug-in gestures; handle data

(e) plug-in gestures; watch data

(f) plug-in gestures; watch acc. & gyro. data

Figure 5.3: Average EERs for our unhook (top) and plug-in (bottom) models by window size and offset, using data from only the handle (left), only the smartwatch (centre), and only the accelerometer and gyroscope of the smartwatch (right).

of times that the negative class (*i.e.*, the adversary) is correctly rejected; the *false positives* is the number of times that the negative class is wrongly accepted; and the *false negatives* is the number of times that the positive class is wrongly rejected.

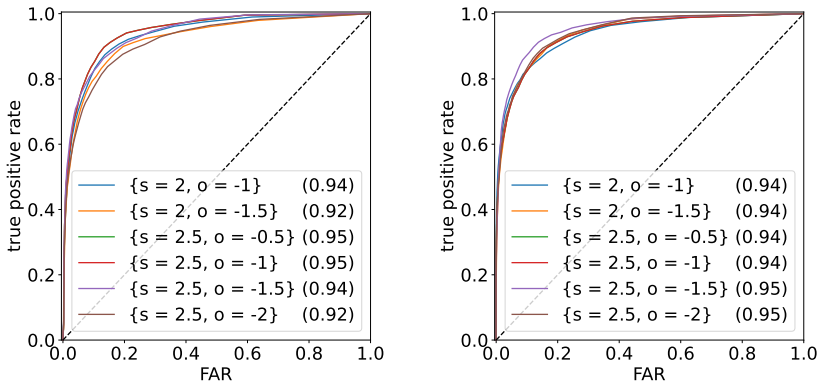
For consistency across our work, we measure performance in the same way as before. See Section 3.4.4 for details.

5.5 Results

5.5.1 Zero-effort Attack

Figure 5.3 shows the average EERs for our unhook and plug-in models by window size and offset. With 20 users, each score is the average over $20 \times 10 = 200$ classifiers (see Section 5.4.3 for details). Figure 5.4 shows the average ROC curves for the strongest performing window parameters.

For unhook gestures, we see in Figure 5.3a that the handle data yields stronger results. For plug-in gestures, we see in Figures 5.3e and 5.3f that the smartwatch



(a) unhook gestures; handle data **(b)** plug-in gestures; watch data

Figure 5.4: Average ROC curves for our unhook and plug-in models with different window sizes and offsets, using data from only the handle and only the smartwatch, respectively. The AUCs are given in parentheses.

data yields stronger results. In both cases, we find that the optimum window parameters are $\{s = 2.5, o = -1.5\}$ (*i.e.*, a window of 2.5 seconds of sensor data starting 1 second before the corresponding contact sensor event), where our models achieve an average EER of 0.06 and an average AUC of 0.95. When using data from only the accelerometer and gyroscope of the watch, we see slightly improved results that echo our findings in Section 3.5.6.

Our results show that if the IMU is embedded in the handle of the cable then the system should use the unhook gesture as the more effective at distinguishing between users, whereas if a smartwatch is used then it should use the plug-in gesture.

5.5.2 Feature Informativeness

When using data from only the handle, we find that the most important features are derived from (i) the x-value of the gyroscope data and (ii) the extrema in the GRV data. The x-axis of the handle-mounted IMU in our apparatus pointed towards the end of the handle, so the x-value of the gyroscope describes the side-to-side rotation of the handle. The GRV extrema describe sharp changes in orientation. Together, these observations suggest that the freehand wobble of the handle when held is discriminative between users and likely explains why these classifiers perform better for the unhook gesture than the plug-in gesture, as users have more freedom

that elicits natural movements when withdrawing the handle compared to having a target for which to aim that elicits deliberate movements when plugging it in.

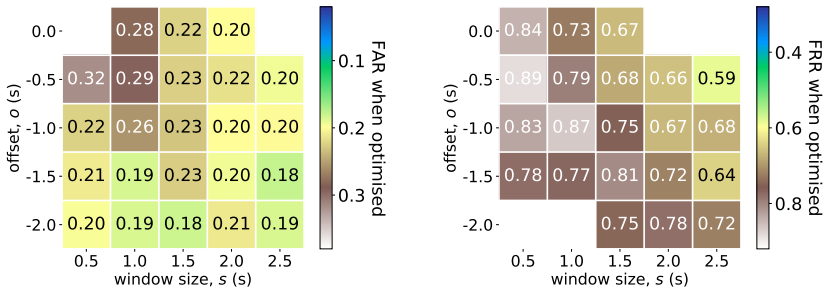
When using data from only the smartwatch, we find that the predominant features are derived from the x-value of the accelerometer data. The x-axis of the smartwatch also pointed towards the end of the handle, indicating that push and pull speeds are discriminative. When interacting with the charging station, we find that users position themselves relative to the charging station rather than the EV, meaning that pulling the handle out of the charging unit is a low-effort movement, whereas pushing the handle into the car socket elicits a change in pose as well as additional force—this likely explains why these classifiers perform better for the plug-in gesture.

5.5.3 Composite Model

We collected motion data throughout each handling interval before segmenting it into separate unhook and plug-in gestures. To see the effects of including as much pertinent data as possible per gesture, we re-combine both gesture components into a composite gesture—*i.e.*, we combine the data surrounding both of the contact sensor events that are available to the system in the same handling interval while excluding any data that accounts for traversing the space between the charging station and the parked vehicle, which will likely differ in each charging session. We choose the optimum window parameters for each type of gesture (as informed by Figure 5.3) and train a set of composite models in which each classifier is trained and tested on samples that consist of both the unhook and plug-in gesture components made in the same handling interval. Table 5.1 shows the results of our composite model for each respective pair of window parameters. Unfortunately, instead of improving on the results in Figure 5.3, we see that the composite models yield average EERs that are roughly the mean of the EERs of the single-gesture models of the two component gestures in their respective window parameters. This indicates that the data from the weaker gesture in each case clutters, rather than enriches, the classifier. We find no benefit in combining the gestures.

Device	Parameters for Unhook Gesture	EER: Parameters for Plug-in Gesture $\{s=1.5, o=-1.0\}$	$\{s=2.0, o=-1.0\}$	$\{s=2.0, o=-1.5\}$	$\{s=2.5, o=-0.5\}$	$\{s=2.5, o=-1.0\}$	$\{s=2.5, o=-1.5\}$
Handle	$\{s=2.0, o=-1.0\}$	0.13	0.11	0.12	**0.09**	0.10	0.10
	$\{s=2.0, o=-1.5\}$	0.13	0.11	0.12	**0.09**	0.10	0.11
	$\{s=2.5, o=-1.0\}$	0.12	0.11	0.11	**0.09**	**0.09**	0.10
	$\{s=2.5, o=-1.5\}$	0.12	0.11	0.12	**0.09**	**0.09**	0.10
Watch	$\{s=2.0, o=-1.0\}$	0.13	0.12	0.13	0.12	0.11	0.12
	$\{s=2.0, o=-1.5\}$	0.11	0.11	0.12	0.10	**0.09**	0.10
	$\{s=2.5, o=-1.0\}$	0.11	0.10	0.11	0.11	0.11	0.11
	$\{s=2.5, o=-1.5\}$	0.13	0.12	0.13	0.11	0.12	0.12

Table 5.1: Average EERs for our composite model with different unhook and plug-in gesture component window sizes and offsets, using data from only the handle (top) and only the smartwatch (bottom).



(a) FARs **(b)** FRRs

Figure 5.5: Average FARs when optimised to minimise false negatives and FRRs when optimised to minimise false positives for our unhook model by window size and offset, using data from only the handle.

5.5.4 Feasibility as an Additional Factor

To evaluate the strict gains that our unhook model can provide to an existing authentication system at a charging station, we adjust the decision thresholds to make any negative impact negligible and measure the FARs and FRRs under those conditions. Figure 5.5 shows the average FARs and FRRs when optimised for our unhook model by window size and offset. (Note that we choose the unhook model because it performs the best for an IMU embedded in the handle of the cable, which is our primary focus.)

Added Security. For our model to add a strict improvement to security, we tune our classifiers to minimise the occurrence of false negatives. We see in Figure 5.5a that our model can add to an existing system a layer of false acceptance detection with a FAR of 0.18 without imposing a burden on usability. This means that we can reject 82% of the adversary's charging attempts without incorrectly rejecting the legitimate user's.

Added Usability. For our model to add a strict improvement to usability, we tune our classifiers to minimise the occurrence of false positives. We see in Figure 5.5b that our model can add to an existing system a strong layer of security at the cost of a FRR of 0.59. This means that we can reduce the number of unnecessary authentication actions required of the user (*i.e.*, where he must manually approve the charging session on his smartphone) by 41% at no cost to security.

5.6 Discussion

Power Consumption. If our system uses the inertial sensors on a smartwatch, its impact on the battery of the watch is discussed in Section 3.6. If our system uses an IMU embedded in the handle of the cable, its power consumption is irrelevant because it is connected directly to a power source and it draws only a negligible amount of power compared to the charging process itself.

Response Time. If our system runs on a smartwatch, its response time is discussed in Section 3.6. If our system runs on a smartphone or as a cloud service, we expect its response time to be negligible due to greater resource allocation. Moreover, once the user has connected the charging cable to the vehicle, the charging process is able to begin whenever the payment is authorised, regardless of any response delay—even if an authentication action is required of the user via a smartphone application, there is no requirement that he must wait by the vehicle to fulfil it.

5.7 Related Work

A number of works have used inertial sensors attached to objects in a smart home context to identify occupants by interaction. Han *et al.* [51] used inertial sensors on household fixtures, such as drawers and windows, to identify which of five users was in the environment, achieving an accuracy score of 0.74 when using data from a single object and of 0.96 when fusing data from five objects. Krawiecka *et al.* [69, 70] used inertial sensors and microphones on household fixtures to identify and authenticate users. For the latter, they were able to reduce the number of unnecessary authentication actions required of the user by 97% at a small cost to security (a tolerated FAR of 0.01) in a user study of 13 participants.

For works relating to the use of inertial sensors on a smartwatch for authentication purposes, see Section 3.7.

To the best of our knowledge, we are the first to propose the use of inertial sensors, whether embedded in the handle of a charging cable or on a smartwatch, to authenticate a user implicitly during an EV charging session.

5.8 Conclusion

In this chapter, we showed that inertial sensor data from the handle of a charging cable can be used to authenticate the user handling it with an EER of 0.06. Our approach is software-driven, but requires minor changes to existing charging station hardware; to mitigate this, by providing an interim or alternative solution, we also showed that inertial sensor data from a smartwatch can be used to authenticate the user with an EER of 0.06. We showed that our model can be optimised for security or usability and incorporated as an additional factor in an existing system to provide strict gains.

This work is an application of our work in Chapter 3 and demonstrates the wide applicability of our approach.

6

VisAuth: Mutual Authentication over a Visual Channel

6.1 Motivation

In recent years, mobile payment systems have evolved rapidly, driven by user preference for enhanced security and usability features [56] and facilitated by advances in technology. One early barrier to adoption was the need for specialised hardware, where exclusionary business practices mandated the use of a dedicated point-of-sale terminal for each payment system. Merchants struggled to support them all, so brand loyalty and local trends led consumer decisions. This problem was resolved by PSD2 and the standardisation of NFC-enabled payments; now, a single terminal can accept payments made using any NFC-enabled device—whether it be a payment card, a hand-held device, or a wearable device—across various payment systems.

The size, shape, and capabilities of point-of-sale terminals have also changed. Early terminals had a slit where magnetic strip payment cards could be swiped and read; these were replaced with slots where Chip and PIN payment cards could be inserted to have their chip read. Terminals typically have a screen where the payment amount is displayed, a keypad where the PIN can be entered, and, for NFC-enabled terminals, a flat surface marked with an NFC icon where the payment device is to be tapped to communicate with the NFC module. They also need a trusted execution environment where local cryptographic materials can be handled

and stored securely and a means to connect securely to a back-end payment server. Many modern consumer devices satisfy these requirements and can be used as terminals; tap-on-phone applications enable smartphones and tablet computers to accept NFC payments, increasing the number and diversity of potential terminals and allowing the payment process to be integrated alongside other vendor services in all-in-one applications—*e.g.*, some restaurants use table management systems installed on tablets that enable waiting staff to take orders, transfer orders to the kitchen, and handle payments in the same application.

The physical presence of a terminal may give a false perception of trust to users that the terminal is legitimate, but a rogue terminal or application can easily be dressed to look genuine in order to execute a MITM attack on unsuspecting users. An effective countermeasure to MITM attacks is mutual authentication, where the payment system authenticates itself to the user before the user is asked to enter any secret information. This principle has been deployed in smartphone applications [90] and online banking interfaces [150], where a personalised greeting message is shown to the user before he is asked to enter his PIN or password. However, we are yet to see such a feature in point-of-sale terminals.

It is difficult to prevent skimming and eavesdropping over wireless channels. The use of NFC can allow payment transactions to be initiated and information stolen without the user knowing [82, 100] and attempts to limit its range to reduce the risk have been shown not to be reliable [33, 68]. Some payment systems communicate over a visual channel between the user's device and the terminal, giving the user more control over the exchange and making it difficult for an attacker to intercept without being noticed. However, these systems typically encode the information into QR codes that can be read and stolen at long range and the user's device must frequently connect to a server to request new tokens since the communication with the terminal over a visual channel is unidirectional.

In this chapter, we derive three system requirements to overcome common weaknesses found in popular, real-world mobile payment systems and propose a novel scheme that satisfies them. We show that our scheme meets these requirements and is resistant against phishing, replay, relay, and presentation attacks.

6.2 System Design

6.2.1 Design Goals

The objective of this chapter is to address common problems found in existing mobile payment systems and to propose a novel scheme that overcomes them. We consider a typical mobile payment context. We pose the following questions.

- To what extent can a visual channel be used to exchange information between the user and the terminal in a payment transaction and what security benefits does it provide?
- To what extent can mutual authentication, where the point-of-sale terminal authenticates itself to the user *before* the user is required to reveal any secret information, be achieved practically in a mobile payment system?

6.2.2 System Requirements

The purpose of a payment scheme is to authenticate a user to a verifier via a point-of-sale terminal to authorise a payment. Existing mobile payment systems present a number of common weaknesses. To ensure that our scheme is able to overcome these problems, we encapsulate them into the following three system requirements that the scheme must satisfy:

- *no specialised hardware*: the system may impose only minimal hardware requirements on the terminal that are satisfiable by typical smartphones and tablet computers, so that the system is simple for merchants to deploy;
- *no user-to-verifier connexion*: the system must not require any device of the user to communicate directly with the verifier during the authentication phase, since network connectivity cannot always be guaranteed; and
- *no expectation of trust*: the user must not be expected to trust the terminal, and therefore must not be expected to reveal any secret information to it, until the terminal has first authenticated itself (*i.e.*, proved that it is legitimate) to the user.

6.2.3 System Model

We consider a system model in which a user is making a payment at a point-of-sale terminal in a typical setting (*e.g.*, in a shop). The system consists of three components: a *user* (the prover), a point-of-sale *terminal*, and a *verifier* (the back-end payment server that authorises the payment). The terminal is a commercial off-the-shelf device that the merchant is using to take the payment; it has a camera, a screen, an input mechanism (*e.g.*, a touchscreen), and an installed application that implements our scheme. The verifier is an authentication server maintained by the payment provider and is assumed to be trusted and secure. We assume that the terminal is registered to the verifier and that these devices have shared keys and established a secure channel over which to communicate. We assume that all cryptographic materials are stored securely on their respective devices.

During the enrolment phase, we assume that the user has a user device, such as a smartphone, that he uses to enrol into the system and to administer his account. We assume that there is a secure channel between the user device and the verifier over which they exchange materials. We assume that the user is able to access a trusted terminal, authenticated using the aforementioned secure channel, to submit a number of biometric samples from which his initial user template can be constructed. We assume that the biometric being used is face geometry, but other traits could be implemented with due consideration. The user adds a payment method to his account, chooses a PIN, and sets a verification message.

During the authentication phase, we assume that there is a visual channel between the user and the terminal, and a secure channel between the terminal and the verifier. The user interacts only with the terminal. The user initiates the protocol by presenting his biometric trait to the terminal. We assume that the integrity of the biometric sample is protected with appropriate liveness detection, as is standard practise in biometric-based systems. The terminal extracts a feature vector from the biometric sample and sends it to the verifier for identification. When the verifier identifies the user, it retrieves the verification message associated with his account and returns it to the terminal, which displays it to the user.

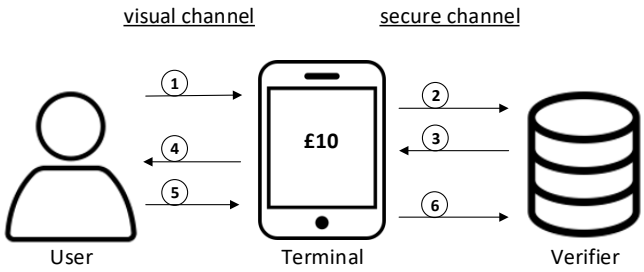


Figure 6.1: The system model of our scheme during the authentication phase. The user presents his biometric trait(s) to the terminal ①, which extracts a feature vector and sends it to the verifier ②, which attempts to identify the user. The verifier returns the verification message associated with the account of the nearest matching user to the terminal ③, which displays it to the user to authenticate the terminal to the user ④. The user then enters his PIN to the terminal ⑤, which sends it to the verifier, which verifies the match to authenticate the user and then authorises the payment ⑥.

The user verifies the message and enters his PIN to the terminal, which passes it to the verifier. The verifier checks the PIN to verify that the identification was correct and to authenticate the user. The verifier can then authorise the payment using the payment method associated with the account. We assume that the classifier used by the verifier has a low misclassification rate and that, when a user is correctly identified and authenticated, his template can be safely updated using the latest feature vector to counter the effects of drift. We assume that the verifier will reject simultaneous authentication sessions that are identified to be of the same user to prevent crossover. A visualisation of the authentication phase of our scheme is shown in Figure 6.1.

6.2.4 Threat Model

We consider an adversary that is attempting to make a payment at the expense of a legitimate user. We assume that the adversary can observe everything that is shared across the visual channel. We assume that the adversary can deploy rogue terminals and that these are dressed to look genuine. Our goal here is to authenticate the legitimate user without leaking any secret information, to facilitate legitimate mobile payment transactions, and to reject the adversary. We consider the following six types of attack:

- *phishing attack*: the adversary has deployed a rogue terminal to trick a legitimate user into revealing his PIN;
- *replay attack*: the adversary has been eavesdropping on the secure channel and has obtained encrypted messages previously sent to the verifier when a legitimate user made a payment and is attempting to re-use them to authorise a new payment;
- *relay attack (in-store)*: a legitimate user is attempting to make a payment at a rogue terminal that is passing his biometric trait (*e.g.*, a captured image of his face) to an adversary who is attempting to use it to authorise a different payment at a legitimate terminal;
- *relay attack (skimming)*: while a legitimate user is *not* involved in a transaction (*e.g.*, he might be commuting on public transport or walking on a busy street), the proximate adversary is attempting to capture his biometric trait using a concealed rogue terminal so as to pass it to a distant accomplice who is attempting to use it to authorise a payment at a legitimate terminal;
- *presentation attack (particular victim)*: the adversary has observed the PIN and biometric trait of a legitimate user in a previous transaction and is attempting to make a payment by impersonating that user; and
- *presentation attack (random victim)*: the adversary is attempting to make a payment as a random user.

In this work, we concentrate on how the proposed scheme can be used to defend against these attacks. We do not consider attacks that take place during the enrolment phase, attacks on the liveness detection system, attacks on the verifier, malware, or denial of service attacks.

6.3 System Architecture

6.3.1 Cryptographic Materials

During the enrolment phase, the user account is created and the user exchanges some materials with the verifier that are stored securely and later used in the authentication phase. A summary of these materials is shown in Table 6.1.

Material	Known to or Inherent to User	Stored on Terminal	Stored on Verifier	Purpose
a	(✓)	(✓)	×	payment amount; known at start of session
b	✓	×	(✓)	biometric feature vector; identifies the user to the verifier by nearest match
m	✓	×	✓	short alphanumeric string; authenticates the terminal to the user
PIN	✓	×	✓	short numerical password; authenticates the user to the verifier
k	×	✓	✓	secret key; secures communication between the terminal and the verifier

Table 6.1: Summary of the cryptographic materials used in our scheme.

b is the biometric feature vector. During the enrolment phase, the user submits a number of biometric samples to a terminal, which extracts feature vectors and sends them to the verifier, which constructs a biometric template for the user. This template is stored as part of the user's account. During the authentication phase, the user provides a biometric sample to the terminal. The terminal extracts the feature vector b and sends it to the verifier to identify the user. It is best practice for biometric data to be processed locally on the device that collects it and for only the feature vector to be transferred, due to the irrevocable nature of biometric data and the impact that theft may have on the security and privacy of the user (in this system and across other systems).

m is the verification message. During the enrolment phase, the user chooses a recognisable string and submits it to the verifier. This message is stored as part of the user's account. During the authentication phase, the verifier sends m to the terminal so that the terminal can authenticate to the user before the user is asked to reveal any secret information. Since the adversary can see m when it is displayed, it must be changed after each use for mutual authentication to hold. We assume that this is achieved by using seeded random string generators to generate the same messages at regular intervals (*e.g.*, every minute) on both the user device and the verifier to remove the need for any user-to-verifier connexion—however, this requires the user to carry the user device. We consider alternative implementations that prioritise usability in Section 6.5.

PIN is the PIN. During the enrolment phase, the user chooses a short (*e.g.*, 4 digits), memorable PIN and submits it to the verifier. This PIN is stored as part of the user's account. During the authentication phase, the user inputs PIN to authenticate to the verifier. More specifically, when the verifier has selected the candidate user that most closely matches b, the PIN input by the user is used to verify the match. We assume that the user will enter PIN on the terminal over a physical channel. Alternative implementations might explore the use of other channels to verify the match, such as having the user speak PIN over an audio channel, gesticulate PIN over a visual channel, or provide some input in response to a challenge on a user device. In an ideal implementation of the scheme, only *salted hashes* of PIN should be stored, transferred, and compared in order to mitigate any damage from attacks against the verifier.

k is a secret key shared between the terminal and the verifier. We assume that this is exchanged as part of the secure channel and is cryptographically secure.

6.3.2 Biometric-based Identification

We exploit the property that biometrics can be classified in a one-to-many manner (*i.e.*, for identification purposes) to enable the user to bypass initially having to reveal any concrete information to the terminal. The user presents only his biometric trait to the terminal—in our case, this is his face, which is freely observable in public. The user's account information, such as his account number, is known only to the user and the verifier and does not need to be transferred during the authentication phase.

As we have seen in previous chapters, biometric classification can result in false positives. If the identification performed by the verifier returns a false positive, this will be caught when the user is presented with a message that does not match m. If the user tries to proceed anyway, he will fail because he does not know the PIN associated with the mismatched account. For usability, one approach to resolving this problem could be for the classifier to return a shortlist of candidate users ordered by how closely they match b and for the subsequent message verification steps to repeat, iterating through the list, until the correct

m is displayed and verified. However, this approach would enable an attacker to collect verification messages of near-matching users that could be used in a phishing attack. Moreover, it would train users to tolerate false matches. In either case, the mutual authentication property would be undermined. For security, the protocol should instead terminate if the verification message is not as expected. The user can then restart it by presenting to the terminal again to give the system a fresh opportunity to identify him correctly.

6.3.3 Authentication Protocol

During the authentication phase, a visual channel between the user and the terminal and a secure channel between the terminal and the verifier are required for the system to achieve mutual authentication. Figure 6.2 shows the authentication protocol and the following steps describe it.

Steps 1 to 2: User Presents Trait to Terminal. The user approaches the terminal to initiate a payment transaction in the amount of a. The user presents his biometric trait to the terminal. The terminal samples the trait and extracts a feature vector, b.

Steps 3 to 12: Terminal Authenticates to User. The terminal sends a and b to the verifier over a secure channel. Nonces are used so that the freshness of messages can be verified by the receiver; we assume that these will take the form of timestamps. The verifier identifies the user by performing a one-to-many lookup of b amongst the user templates of all users registered on the system. When the nearest match, b', is identified, the verifier retrieves the account information of b' and gets its verification message, m', and PIN, PIN'. The verifier sends m' to the terminal, which displays it to the user. The user verifies m' by performing a string comparison against the expected message, m. This demonstrates to the user that the terminal is securely communicating with and trusted by the verifier and so authenticates the terminal to the user before the user is required to reveal any secret information.

Steps 13 to 18: User Authenticates to Verifier via Terminal. The user enters PIN on the terminal, which sends it to the verifier. The verifier verifies

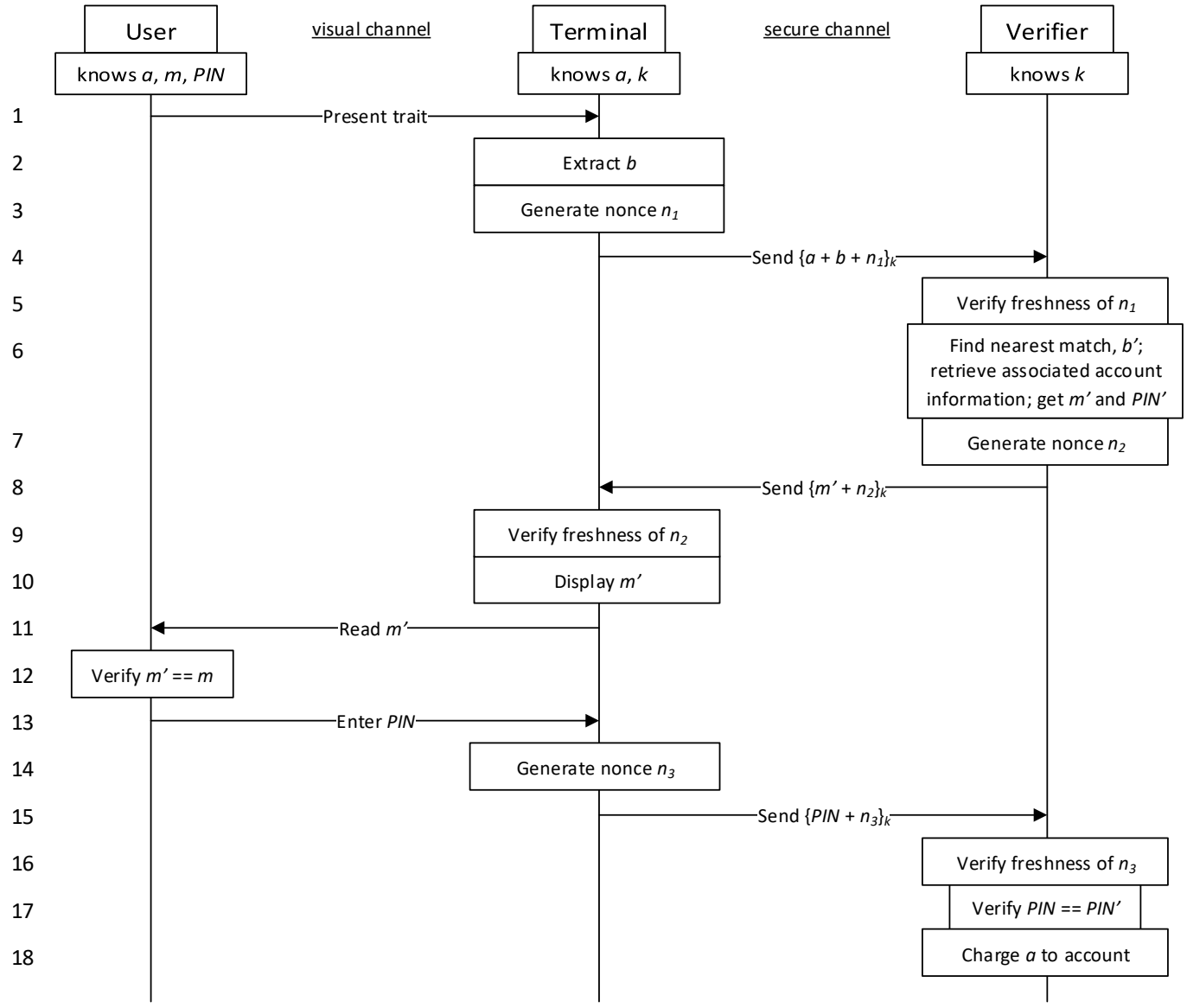


Figure 6.2: The authentication protocol of our scheme.

PIN against the expected PIN'. This authenticates the user as the identified user to the verifier and enables the verifier to process the payment of a using the payment method associated with the identified user account.

6.4 Security Analysis

Our scheme meets all of the system requirements. It requires no specialised hardware: the terminal needs only a camera, a screen, and a touchscreen or keypad for input; these requirements can be satisfied by any modern smartphone or tablet, making it easily deployable. The user is identified and authenticated to the verifier via the terminal, using a visual channel, without requiring a direct connexion between the user (or any user device) and the verifier. Finally, the authentication protocol ensures that the terminal demonstrates to the user that it is connected to and trusted by the verifier before the user is required to reveal PIN.

Phishing Attack. For the phishing attack, the adversary attempts to have a legitimate user reveal PIN by deploying a rogue terminal. At Step 10, the terminal must display m to the user before the user reveals PIN. As long as m is changed after each use, the rogue terminal will not be able to achieve this. Therefore, our scheme provides resistance against phishing attacks.

Replay Attack. For the replay attack, the adversary attempts to authorise a repeat payment by re-sending encrypted messages sent between the terminal and the verifier during a previous transaction. Nonces are included in every encrypted message sent over the secure channel to enable their freshness to be checked, so the attack will fail at Step 5. Therefore, our scheme provides resistance against replay attacks.

Relay Attack. For the in-store relay attack, the adversary attempts to authorise a different payment by passing a copy of the legitimate user's biometric trait. For the skimming relay attack, the adversary attempts to have an accomplice authorise a payment by capturing the biometric trait of an oblivious legitimate user. In each case, owing to the use of a visual channel, the legitimate terminal being used by the accomplice is able to validate the authenticity of the user. The copied biometric will

fail the liveness check and be rejected, so the attack will fail at Step 2. Therefore, our scheme provides resistance against relay attacks.

Presentation Attack. For the presentation attack on a particular victim, the adversary knows the victim's PIN and attempts to mimic his biometric trait so as to impersonate him to a legitimate terminal. Biometric identification is weaker than authentication inasmuch as the attacker only needs to achieve being matched nearer to the intended victim than to some other user. We can minimise this discrepancy by tightening the decision threshold of the classifier to ensure that the matching must meet a certain minimum accuracy, akin to authentication. The tighter this is set, the more it will increase the FRR (*i.e.*, the greater the gains in security, the greater the cost to usability). This will increase the effort required of the adversary. An implementation of the scheme can further increase the effort required of the adversary by using multiple biometric traits, since he would need to achieve being the nearest match for all traits simultaneously.

For the presentation attack on a random victim, the adversary simply presents his own biometric trait, either modified or not, so that the system matches him to a random victim in Step 6. We assume that the adversary is not registered to the system and that a match is found, even with a tightened decision threshold. The adversary does not know the PIN of the random victim, so the attack will fail at Step 17. The adversary may attempt to perform a brute force guessing attack against the PIN by presenting in the same manner repeatedly to generate the same victim each time. An ideal implementation of the scheme should use common anomaly detection and throttling techniques to defend against guessing attacks.

The first of these attacks does not work at scale, because the adversary must expend effort to obtain the victim's PIN, and is defeated by further increasing the effort (cost) required. The second does work at scale, but is defeated by the PIN. In each case, our scheme provides resistance against presentation attacks.

6.5 Discussion

Convenience. Our scheme enables the user to make a payment without needing to carry any form of cash, payment card, or user device (although he may need to consult his user device to verify m, depending on how it has been implemented). Furthermore, since the user is identified as part of the process, any relevant status checks can be made automatically against the information held on record. This means that the user does not need to carry loyalty cards, discount coupons, or proof of age or membership—as these can all be applied upon identification.

For security, we have assumed that seeded random string generators on the user device and the verifier generate a fresh m every minute. To improve usability, an implementation of the scheme might consider ways to increase the size of the interval between changes of m to free the user from the requirement that he must carry a user device during the authentication phase. This could include the use of message templates, along with shapes or colours to increase the message space, so that what needs to be memorised is more user-friendly, rather than a random string—*e.g.*, the system might allow the user to create or select a rule pertaining to the structure of an expected message that is valid for a day, then the verifier would randomly generate a fresh string every transaction that satisfies the rule so that the user only needs to verify that it fits the template (*e.g.*, 'a valid 5-letter word followed by a green triangle'). The user would then memorise the rule before a shopping session and verify the fit of each server-generated message when making payments, without needing to consult his user device. The adversary could replay such a message to perform a phishing attack, but not at scale, so the gains in usability may be worth the risks to security.

Asymmetrical Channel. The visual channel enables asymmetric communication, as the capabilities required for sending information are different to those required for receiving it. Each party can either display to or read from the channel depending on its capabilities. This means that there are constraints on what each party can do to each other and parts of the system can be restricted to unidirectional communication. Our scheme leverages this property in Steps 1 and 2,

where the user presents his biometric trait and the terminal can only read it, and in Steps 10 and 11, where the terminal displays m' and the user can only choose whether or not to verify it.

Contextual Awareness. The capabilities that can be used to read information from the visual channel can also collect incidental information from the surrounding environment. Depending on its position, the camera on the terminal can capture additional information around the user that could be used to facilitate advanced fraud detection techniques, such as verifying that the terminal is operating in the expected environment. An implementation of the scheme might leverage this property by passing an image of the scene to the verifier; expected objects, markers, or lighting effects could be placed in the environment as a form of signature, or a clock could be placed in the environment such that the time captured in the image could be extracted and cross-checked with the timestamps used as nonces to strengthen the assertion of freshness with an independent factor. A sophisticated adversary could still fabricate the entire environment, but each step would increase the effort required of the adversary and present a potential point of failure for an attack.

Privacy Risk Mitigation. The use of a visual channel—especially when collecting peripheral information—poses a risk to the privacy of the user. Any images sent to the verifier should have their utility weighed against their potential impact on privacy. Countermeasures to mitigate privacy leakage from images include reducing the resolution and blurring unnecessary details before sending. To protect the biometric data of the user, biometric traits should only be processed locally on the terminal and should be obscured from any images sent to the verifier.

6.6 Related Work

Identification-based Systems. With regard to the use of biometric identification as part of an authentication system, some payment providers have trialled the technique with the promise of improved convenience for the user. Smile-to-Pay [75], developed by Ant Financial for AliPay, uses a 3D camera to capture the user's facial

likeness, perform liveness detection, and identify the user within 2 seconds. The system then sends a verification request to the user's smartphone that requires a timely response to verify the match. Biometric Checkout Program [91], developed by Mastercard, operates in a similar manner, allowing the user to identify himself to the terminal over a visual channel using either his face or palm. Both of these systems require a user-to-verifier connexion to verify the match. To the best of our knowledge, we are the first to propose the use of biometric identification to facilitate mutual authentication and to do so without requiring a user-to-verifier connexion.

Visual Channel. With regard to the use of a visual channel, some existing mobile payment systems have explored the use of a QR code to pass information between a user device and the terminal. In Yoyo Wallet [162], the user must first authenticate to a smartphone application using a PIN and can then access a QR code that contains tokenised payment information. To make a payment, the QR code is shown to the terminal over a visual channel and can be used up to three times before it expires. When the user's smartphone next connects to the Yoyo cloud server, where the user's virtual wallet is stored, a new QR code is downloaded. The limited number of uses per QR code mitigates the damage from theft, but makes this payment system more dependent than typical tap-and-pay systems on a user-to-verifier connexion between transactions. WeChat [158] and AliPay [5], both popular payment systems in China, support the use of QR codes and barcodes to transfer information. VisAuth [137] embeds information into an image as a robust watermark to send it over a visual channel. However, the system state on the user device can become desynchronised from that on the verifier—while this can be beneficial, since it unavoidably draws attention to an attack, it can also occur if the protocol is interrupted at various steps, providing plausible deniability to an attacker and making the wider scheme impractical. All of these systems transfer confidential information over a visual channel, whereas we transfer authentication information. Our scheme makes broader use of the visual channel by observing the user's biometric trait(s) to identify him to the verifier, where his payment information is stored, rather than encoding the payment information directly into a visual token.

Smart city transport networks, such as Oxford SmartZone [107], enable bus tickets to be purchased in advance and delivered as QR codes to the user's smartphone application. Instead of buying a ticket from the driver, the user presents the QR code to a terminal on the bus to expedite boarding. In this case, a product, rather than the user, is being authenticated, so the threat model primarily considers theft. The user can only be logged in to one device at a time to prevent account-sharing on multiple devices and the screen contains additional, animated elements that are verified by the terminal to prevent token-sharing using a screenshot.

Mutual Authentication. With regard to mutual authentication, related works in the field of mobile cloud computing have focused on the mutual authentication of the user device and the verifier in a general setting without consideration for any other components that might be involved in the system, such as the point-of-sale terminal in our case, that also need to be authenticated before the user should be expected to reveal any secret information. Dey *et al.* [32] proposed a scheme that relies on the location of the user device and the current time at the verifier, and so requires a persistent user-to-verifier connexion. Other works [99, 105, 112] have proposed schemes in which mutual authentication is achieved via a trusted third party. These schemes require there to be a persistent connexion between the user device and the third party from the start of the transaction. We have sought to avoid this for the same reason that we avoid a user-to-verifier connexion.

6.7 Conclusion

In this chapter, we proposed and analysed a novel mobile payment scheme that operates over a visual channel. We showed that our scheme (i) requires no specialised hardware, imposing only minimal requirements on the terminal that can be satisfied by most commercially available smartphones and tablet computers, to ease deployment, (ii) requires no user-to-verifier connexion during the authentication phase, such that it remains usable regardless of wireless connectivity, and (iii) mutually authenticates the terminal and the verifier to the user before he is asked to reveal any secret information to authenticate himself. We explored the properties that

a visual channel provides and we showed that our scheme is extensible in various ways depending on the needs of the wider system in which it is implemented. Furthermore, our scheme provides a number of conveniences to the user, such as not having to carry any payment or loyalty cards, and provides resistance against phishing, replay, relay, and presentation attacks.

7
Summary

In this thesis, we looked at the ways in which users can be authenticated in systems with limited input capabilities. We began by surveying how users authenticate to digital services, from password-based systems to continuous MFA systems, and how the sensing capabilities of all sorts of heterogeneous devices can contribute to strengthening the process.

We showed that the tap gesture, made during a mobile payment transaction, is a gesture biometric that can be captured and used for authentication purposes. We showed that inertial sensors on a smartwatch can be used to authenticate the user when making a payment with the watch and to verify his intent. We also showed that inertial sensors on a smart ring can be used to authenticate the user when making a payment with the ring, and that the smartwatch and the smart ring can each be used to authenticate the user in payments made with the other.

We focused predominantly on the context of mobile payments, given the key importance that authentication has in supporting and enabling those systems, but we also demonstrated the wide applicability of our work by applying it to other contexts, such as access control and electric vehicle charging. We showed that we can authenticate the user by knocks on a door and when handling a charging cable, whether the inertial sensors are on the wearable devices, attached to the door, or embedded in the handle of the cable.

Finally, we looked at the current state of mobile payment systems and designed a scheme that supports a wide range of devices as terminals. We proposed a novel

scheme that requires only minimal hardware specifications that can be satisfied by most commercially available smartphones and tablets to ease the deployment burden and avoids a dependence upon a persistent user-to-verifier connexion to enable usage even where wireless connectivity is poor. We found that by using biometric identification over a visual channel, our scheme can achieve mutual authentication to protect the user from phishing attacks.

Bibliography

[1] A. Acar, H. Aksu, A. S. Uluagac, and K. Akkaya. “A Usable and Robust Continuous Authentication Framework Using Wearables", *IEEE Transactions on Mobile Computing (TMC)*, 2020.

[2] A. Adams and M. A. Sasse. “Users Are Not the Enemy", *Communications of the ACM*, Vol. 42, 1999.

[3] S. Aebischer, C. Dettoni, G. C. Jenkinson, K. K. Krol, D. Llewellyn-Jones, T. Masui, and F. Stajano. “Pico in the Wild: Replacing Passwords, One Site at a Time", *European Workshop on Usable Security (EuroUSEC)*, 2017.

[4] I. Agadakos, P. Hallgren, D. Damopoulos, A. Sabelfeld, and G. Portokalidis. “Location-enhanced Authentication Using the IOT: Because You Cannot Be in Two Places at Once", *Annual Conference on Computer Security Applications (ACSAC)*, 2016.

[5] AliPay. https://wglobal.alipay.com/products/spot (https://archive.vn/NU14f), 2017.

[6] Apple. “Use Express Travel with Apple Pay", https://support.apple.com/HT212171 (https://archive.vn/6OK5w), 2022.

[7] A. Aktypi, J. R. C. Nurse, and M. Goldsmith. “Unwinding Ariadne’s Identity Thread: Privacy Risks with Fitness Trackers and Online Social Networks", *International Workshop on Multimedia Privacy and Security (MPS)*, 2017.

[8] L. Ardüser, P. Bissig, P. Brandes, and R. Wattenhofer. “Recognizing Text Using Motion Data from a Smartwatch", *IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2016.

[9] Y. Ashibani, D. Kauling, and Q. H. Mahmoud. “A Context-aware Authentication Framework for Smart Homes", *IEEE Canadian Conference on Electrical and Computer Engineering*, 2017.

[10] K. Bao, H. Valev, M. Wagner, and H. Schmeck. “A Threat Analysis of the Vehicle-to-grid Charging Protocol ISO 15118", *Computer Science - Research and Development*, Vol. 33, 2017.

[11] R. Baker and I. Martinovic. “Losing the Car Keys: Wireless PHY-layer Insecurity in EV Charging", *USENIX Security Symposium (USENIX)*, 2019.

[12] D. V. Bailey, M. Dürmuth, and C. Paar. “Statistics on Password Re-use and Adaptive Strength for Financial Accounts", *Security and Cryptography for Networks*, Vol. 8642, 2014.

[13] D. Basin, R. Sasse, and J. Toro-Pozo. “The EMV Standard: Break, Fix, Verify", *IEEE Symposium on Security and Privacy (S&P)*, 2021.

[14] B. Biggio, L. Didaci, G. Fumera, and F. Roli. “Poisoning Attacks to Compromise Face Templates", *International Conference on Biometrics*, 2013.

[15] M. Bond, O. Choudary, S. J. Murdoch, S. Skorobogatov, and R. Anderson. “Chip and Skim: Cloning EMV Cards with the Pre-play Attack", *IEEE Symposium on Security and Privacy (S&P)*, 2014.

[16] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. “Classification and Regression Trees", 1984.

[17] A. Buriro, B. Crispo, M. Eskandri, S. Gupta, A. Mahboob, and R. van Acker. “SnapAuth: A Gesture-based Unobtrusive Smartwatch User Authentication Scheme", *Emerging Technologies for Authorization and Authentication*, 2018.

[18] R. Caruana and A. Niculescu-Mizil. “An Empirical Comparison of Supervised Learning Algorithms", *International Conference on Machine Learning*, 2006.

[19] S. Y. Chun, J.-H. Kang, H. Kim, C. Lee, I. Oakley, and S.-P. Kim. “ECG Based User Authentication for Wearable Devices Using Short Time Fourier Transform", *International Conference on Telecommunications and Signal Processing (TSP)*, 2016.

[20] F. Ciuffo and G. M. Weiss. “Smartwatch-based Transcription Biometrics", *IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference*, 2017.

[21] G. Cola, M. Avvenuti, F. Musso, and A. Vecchio. “Gait-based Authentication Using a Wrist-worn Device", *International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2016.

[22] M. Conti, I. Zachia-Zlatea, and B. Crispo. “Mind How You Answer Me!", *ACM Asia Conference on Computer and Communications Security (AsiaCCS)*, 2011.

[23] M. Conti, G. Lovisotto, I. Martinovic, and G. Tsudik. “FADEWICH: Fast Deauthentication over the Wireless Channel", *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2017.

[24] M. Conti, D. Donadel, R. Poovendran, and F. Turrin. “EVExchange: A Relay Attack on Electric Vehicle Charging System", *European Symposium on Research in Computer Security (ESORICS)*, 2022.

[25] C. Cornelius and D. Kotz. “Recognizing Whether Sensors Are on the Same Body", *Journal of Pervasive and Mobile Computing*, 2012.

[26] C. Cornelius, J. Sorber, R. Peterson, J. Skinner, R. Halter, and D. Kotz. “Who Wears Me? Bioimpedance as a Passive Biometric", *USENIX Workshop on Health Security and Privacy (HealthSec)*, 2012.

[27] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. F. Wang. “The Tangled Web of Password Reuse", *Network and Distributed System Security Symposium (NDSS)*, 2014.

[28] D. Dasgupta, A. K. Nag, and A. Roy. “Adaptive Multi-factor Authentication System", US Patent 9912657, 2018.

[29] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann. “Touch Me Once and I Know It’s You!: Implicit Authentication based on Touch Screen Patterns", *ACM Conference on Human Factors in Computing Systems (CHI)*, 2012.

[30] Y. Desmedt. “Threshold Cryptography", *Transactions on Emerging Telecommunications Technology*, Vol. 5, 1994.

[31] Y. Desmedt, M. Burmester, R. Safavi-Naini, and H. Wang. “Threshold Things That Think (T4): Security Requirements to Cope with Theft of Handheld/Handless Internet Devices", *Symposium on Requirements Engineering for Information Security*, 2001.

[32] S. Dey, Q. Ye, and S. Sampalli. “AMLT: A Mutual Authentication Scheme for Mobile Cloud Computing", *IEEE International Conference on Internet of Things (iThings), IEEE Green Computing and Communications (GreenCom), IEEE Cyber, Physical and Social Computing (CPSCom), and IEEE Smart Data (SmartData)*, 2018.

[33] T. P. Diakos, J. A. Briffa, T. W. C. Brown, and S. Wesemeyer. “Eavesdropping Near Field Contactless Payments: A Quantitative Analysis", *The Journal of Engineering*, 2013.

[34] M. Dürmuth, F. Angelstorf, C. Castelluccia, D. Perito, and A. Chaabane. “OMEN: Faster Password Guessing Using an Ordered Markov Enumerator", *Engineering Secure Software and Systems*, 2015.

[35] S. Eberz, K. B. Rasmussen, V. Lenders, and I. Martinovic. “Preventing Lunchtime Attacks: Fighting Insider Threats with Eye Movement Biometrics", *Network and Distributed System Security Symposium (NDSS)*, 2015.

[36] S. Eberz, K. B. Rasmussen, V. Lenders, and I. Martinovic. “Evaluating Behavioral Biometrics for Continuous Authentication: Challenges and Metrics", *ACM Asia Conference on Computer and Communications Security (AsiaCCS)*, 2017.

[37] S. Eberz, N. Paoletti, M. Roeschlin, A. Patane, M. Kwiatkowska, and I. Martinovic. “Broken Hearted: How to Attack ECG Biometrics", *Network and Distributed System Security Symposium (NDSS)*, 2017.

[38] D. Ekiz, Y. S. Can, Y. C. Dardagan, and C. Ersoy. “Can a Smartband Be Used for Continuous Implicit Authentication in Real Life", *IEEE Access*, Vol. 8, 2020.

[39] M. Emms, B. Arief, L. Freitas, J. Hannon, and A. van Moorsel. “Harvesting High Value Foreign Currency Transactions from EMV Contactless Credit Cards without the PIN", *ACM Conference on Computer and Communications Security (CCS)*, 2014.

[40] M. Espinoza, C. Champod, and P. Margot. “Vulnerabilities of Fingerprint Reader to Fake Fingerprints Attacks", *Forensic Science International*, Vol. 204, 2011.

[41] FIDO Alliance. “FIDO Alliance Implementation Draft 20 February 2018", https://fidoalliance.org/specs/fido-uaf-v1.2-id-20180220/fido-security-ref-v1.2-id-20180220.html (https://archive.vn/ebGHT), 2018.

[42] R. D. Findling, M. Muaaz, D. Hintze, and R. Mayrhofer. “ShakeUnlock: Securely Transfer Authentication States Between Mobile Devices", *IEEE Transactions on Mobile Computing*, Vol. 16, 2017.

[43] D. Florencio, C. Herley, and P. C. van Oorschot. “An Administrator’s Guide to Internet Password Research", *USENIX Conference on Large Installation System Administration (LISA)*, 2014.

[44] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. “Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication", *IEEE Transactions on Information Forensics and Security*, Vol. 8, 2013.

[45] D. M. Freeman, S. Jain, M. Dürmuth, B. Biggio, and G. Giacinto. “Who Are You? A Statistical Approach to Measuring User Authenticity", *Network and Distributed System Security Symposium (NDSS)*, 2016.

[46] L. Fridman, S. Weber, R. Greenstadt, and M. Kam. “Active Authentication on Mobile Devices via Stylometry, Application Usage, Web Browsing, and GPS Location", *IEEE Systems Journal*, 2017.

[47] M. Ghasemisharif, A. Rames, S. Checkoway, C. Kanich, and J. Polakis. “O Single Sign-off, Where Art Thou? An Empirical Analysis of Single Sign-on Account Hijacking and Session Management on the Web", *USENIX Security Symposium (USENIX)*, 2018.

[48] I. Griswold-Steiner, R. Matovu, and A. Serwadda. “Handwriting Watcher: A Mechanism for Smartwatch-driven Handwriting Authentication", *IEEE International Joint Conference on Biometrics (IJCB)*, 2017.

[49] I. Griswold-Steiner, R. Matovu, and A. Serwadda. “Wearables-driven Freeform Handwriting Authentication", *IEEE Transactions on Biometrics, Behavior, and Identity Science*, Vol. 1, pp. 152-164, 2019.

[50] A. Gupta, M. Miettinen, N. Asokan, and M. Nagy. “Intuitive Security Policy Configuration in Mobile Devices Using Context Profiling", *ASE/IEEE International Conference on Privacy, Security, Risk and Trust and International Conference on Social Computing*, 2012.

[51] J. Han, S. Pan, M. K. Sinha, H. Y. Noh, P. Zhang, and P. Tague. “Smart Home Occupant Identification via Sensor Fusion across On-object Devices", *ACM Transactions on Sensor Networks (TOSN)*, Vol. 14, 2018.

[52] E. Hayashi, S. Das, S. Amini, J. Hong, and I. Oakley. “CASA: Context-aware Scalable Authentication", *Symposium on Usable Privacy and Security (SOUPS)*, 2013.

[53] A. Hern. “Hacker Fakes German Minister’s Fingerprints Using Photos of Her Hands", 2014.

[54] D. Hintze, M. Füller, S. Scholz, R. D. Findling, M. Muaaz, P. Kapfer, E. Koch, and R. Mayrhofer. “CORMORANT: Ubiquitous Risk-aware Multi-modal Biometric Authentication across Mobile Devices", *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 3, 2019.

[55] C. G. Hocking, S. M. Furnell, N. L. Clarke, and P. L. Reynolds. “Authentication Aura: A Distributed Approach to User Authentication", *Journal of Information Assurance and Security*, Vol. 6, 2011.

[56] J. H. Huh, S. Verma, S. S. V. Rayala, R. B. Bobba, K. Beznosov, and H. Kim. “I Don’t Use Apple Pay Because It’s Less Secure...: Perception of Security and Usability in Mobile Tap-and-pay", *Workshop on Usable Security (USEC)*, 2017.

[57] O. Huhta, P. Shrestha, S. Udar, M. Juuti, N. Saxena, and N. Asokan. “Pitfalls in Designing Zero-effort Deauthentication: Opportunistic Human Observation Attacks", *Network and Distributed System Security Symposium (NDSS)*, 2016.

[58] A. Jain, R. Bolle, and S. Pankanti. “Biometrics: Personal Identification in Networked Society", *Science & Business Media*, 2006.

[59] M. Jakobsson, E. Shi, P. Golle, and R. Chow. “Implicit Authentication for Mobile Devices", *USENIX Conference on Hot Topics in Security (HotSec)*, 2009.

[60] M. Jakobsson. “Two-factor Inauthentication - The Rise in SMS Phishing Attacks", *Computer Fraud & Security*, Vol. 2018, 2018.

[61] A. H. Johnston and G. M. Weiss. “Smartwatch-based Biometric Gait Recognition", *IEEE International Conference on Biometrics Theory, Applications, and Systems (BTAS)*, 2015.

[62] T. Kaczmarek, E. Ozturk, and G. Tsudik. “Assentication: User De-authentication and Lunchtime Attack Mitigation with Seated Posture Biometric", *Applied Cryptography and Network Security*, 2018.

[63] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun. “Sound-proof: Usable Two-factor Authentication Based on Ambient Sound", *USENIX Security Symposium (USENIX)*, 2015.

[64] H. G. Kayacik, M. Just, L. Baillie, D. Aspinall, and N. Micallef. “Data Driven Authentication: On the Effectiveness of User Behaviour Modelling with Mobile Device Sensors", *Workshop on Mobile Security Technologies (MoST)*, 2014.

[65] H. Khan and U. Hengartner. “Towards Application-centric Implicit Authentication on Smartphones", *Workshop on Mobile Computing Systems and Applications*, 2014.

[66] J. K. Kim, K. B. Lee, and S. G. Hong. “Random Forest Based Biometric Identification Using Smart Shoes", *International Conference on Sensing Technology (ICST)*, 2017.

[67] H. Kong, L. Lu, J. Yu, Y. Zhu, F. Tang, Y.-C. Chen, L. Kong, and F. Lyu. “Push the Limit of WiFi-based User Authentication towards Undefined Gestures", *IEEE Conference on Computer Communications (INFOCOM)*, 2022.

[68] H. Kortvedt and S. Mjolsnes. “Eavesdropping Near Field Communication", *The Norwegian Information Security Conference (NISK)*, 2009.

[69] K. Krawiecka, S. Birnbach, S. Eberz, and I. Martinovic. “Biometric Identification System based on Object Interactions in Internet of Things Environments", *IEEE Security and Privacy Workshops (SPW)*, 2022.

[70] K. Krawiecka, S. Birnbach, S. Eberz, and I. Martinovic. “BeeHIVE: Behavioral Biometric System based on Object Interactions in Smart Environments", arXiv preprint arXiv:2202.03845, 2022.

[71] K. Krol, S. Aebischer, D. Llewellyn-Jones, C. Dettoni, and F. Stajano. “Seamless Authentication with Pico", *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2017.

[72] P. Laperdrix, N. Bielova, B. Baudry, and G. Avoine. “Browser Fingerprinting: A Survey", *ACM Transactions on the Web*, Vol. 14, 2020.

[73] G. Laput and C. Harrison. “Sensing Fine-grained Hand Activity with Smartwatches", *Conference on Human Factors in Computing Systems (CHI)*, 2019.

[74] W. H. Lee and R. B. Lee. “Implicit Sensor-based Authentication of Smartphone Users with Smartwatch", *ACM Hardware and Architectural Support for Security and Privacy (HASP)*, 2016.

[75] A. Lee. “Alipay Rolls Out World’s First ‘Smile to Pay’ Facial Recognition System at KFC Outlet in Hangzhou", https://www.scmp.com/tech/start-ups/article/2109321/alipay-rolls-out-worlds-first-smile-pay-facial-recognition-system-kfc (https://archive.vn/r0yTR), 2017.

[76] W. H. Lee, X. Liu, Y. Shen, H. Jin, and R. B. Lee. “Secure Pick Up: Implicit Authentication When You Start Using the Smartphone", *ACM Symposium on Access Control Models and Technologies (SACMAT)*, 2017.

[77] A. Levy, B. Nassi, Y. Elovici, and E. Shmueli. “Handwritten Signature Verification Using Hand-worn Devices", *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, Vol. 2, 2018.

[78] F. Li, N. Clarke, M. Papadaki, and P. Dowland. “Active Authentication for Mobile Devices Utilising Behaviour Profiling", *International Journal of Information Security*, Vol. 13, 2014.

[79] S. Li, A. Ashok, Y. Zhang, C. Xu, J. Lindqvist, and M. Gruteser. “Whose Move Is It Anyway? Authenticating Smart Wearable Devices Using Unique Head Movement Patterns", *IEEE International Conference on Pervasive Computing and Communication (PerCom)*, 2016.

[80] G. C. Liang, X. Y. Xu, and J. D. Yu. “User Authentication on Wearable Devices Based on Punch Gesture Biometrics", *International Conference on Information Science and Technology*, Vol. 11, 2017.

[81] X. Liang and D. Kotz. “AuthoRing: Wearable User-presence Authentication", *ACM Workshop on Wearable Systems and Applications (WearSys)*, 2017.

[82] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis. “On the Security Issues of NFC-enabled Mobile Phones", *International Journal of Internet Technology and Secured Transactions*, 2010.

[83] X. Liu, Z. Zhou, W. Diao, Z. Li, and K. Zhang. “When Good Becomes Evil: Keystroke Inference with Smartwatch", *ACM Conference on Computer and Communications Security (CCS)*, 2015.

[84] G. Lovisotto, S. Eberz, and I. Martinovic. “Biometric Backdoors: A Poisoning Attack against Unsupervised Template Updating", *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2020.

[85] S. G. Lyastani, M. Schilling, S. Fahl, M. Backes, and S. Bugiel. “Better Managed Than Memorized? Studying the Impact of Managers on Password Strength and Reuse", *USENIX Security Symposium (USENIX)*, 2018.

[86] C. Mainka, V. Mladenov, and J. Schwenk. “Do Not Trust Me: Using Malicious IdPs for Analyzing and Attacking Single Sign-on", *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016.

[87] C. Mainka, V. Mladenov, J. Schwenk, and T. Wich. “SoK: Single Sign-on Security - An Evaluation of OpenID Connect", *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2017.

[88] S. Mare, A. M. Markham, C. Cornelius, R. Peterson, and D. Kotz. “ZEBRA: Zero-effort Bilateral Recurring Authentication", *IEEE Symposium on Security and Privacy (S&P)*, 2014.

[89] S. Mare, R. Rawassizadeh, R. Peterson, and D. Kotz. “Continuous Smartphone Authentication Using Wristbands", *Workshop on Usable Security and Privacy (USEC)*, 2019.

[90] C. Marforio, R. J. Masti, C. Soriente, K. Kostiainen, and S. Čapkun. “Evaluation of Personalized Security Indicators as an Anti-phishing Mechanism for Smartphone Applications", *CHI Conference on Human Factors in Computing Systems (CHI)*, 2016.

[91] Mastercard. “With a Smile or a Wave, Paying in Store Just Got Personal", https://www.mastercard.com/news/press/2022/may/with-a-smile-or-a-wave-paying-in-store-just-got-personal (https://archive.vn/Rm23Z), 2022.

[92] R. A. Maxion and K. S. Killourhy. “Keystroke Biometrics with Number-pad Input", *IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, 2010.

[93] R. Mayrhofer and H. Gellersen. “Shake Well Before Use: Intuitive and Secure Pairing of Mobile Devices", *IEEE Transactions on Mobile Computing*, , Vol. 8, 2009.

[94] W. Melicher, B. Ur, S. M. Segreti, S. Komanduri, L. Bauer, N. Christin, and L. F. Cranor. “Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks", *USENIX Security Symposium (USENIX)*, 2016.

[95] M. Miettine, S. Heuser, W. Kronz, A. R. Sadeghi, and N. Asokan. “ConXsense: Automated Context Classification for Context-aware Access Control", *ACM Symposium on Information, Computer and Communications Security (AsiaCCS)*, 2014.

[96] M. Migliardi, M. Guerar, S. Marzio, and C. Ferrari. “Continuous Authentication on a Smartwatch", *International Conference on Ubiquitous Computing & Ambient Intelligence*, 2022.

[97] M. Muaaz and R. Mayrhofer. “Smartphone-Based Gait Recognition: From Authentication to Imitation", *IEEE Transactions on Mobile Computing*, Vol. 16, 2017.

[98] M. Mültin. “The Basics of Plug & Charge", https://www.switch-ev.com/blog/basics-of-plug-and-charge (https://archive.vn/ngOul), 2020.

[99] E. Munivel and A. Kannammal. “New Authentication Scheme to Secure against the Phishing Attack in the Mobile Cloud Computing", *Security and Communication Networks*, 2019.

[100] S. J. Murdoch, S. Drimer, R. Anderson, and M. Bond. “Chip and PIN Is Broken", *IEEE Symposium on Security and Privacy (S&P)*, 2010.

[101] P. Musale, D. Baek, N. Werellagama, S. S. Woo, and B. J. Choi. “You Walk, We Authenticate: Lightweight Seamless Authentication based on Gait in Wearable IoT Systems", *IEEE Access*, Vol. 7, 2019.

[102] T. Mustafa, R. Matovu, A. Serwadda, and N. Muirhead. “Unsure How to Authenticate on Your VR Headset? Come on, Use Your Head!", *ACM International Workshop on Security and Privacy Analytics (IWSPA)*, 2018.

[103] M. F. M. Noor, S. Rogersy, and J. Williamson. “Pre-interaction Identification by Dynamic Grip Classification", *International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2017.

[104] Open Fast Charging Alliance. “Autocharge - Automatic Charging Start and Authorization of Electric Vehicles", https://github.com/openfastchargingalliance/openfastchargingalliance/blob/master/autocharge-final.pdf (https://archive.vn/LWqKr), 2017.

[105] O. O. Olakanmi and S. O. Oke. “MASHED: Security and Privacy-aware Mutual Authentication Scheme for Heterogeneous and Distributed Mobile Cloud Computing Services", *Information Security Journal: A Global Perspective*, Vol. 27, 2018.

[106] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas. “How Many Trees in a Random Forest?", *IAPR International Conference on Machine Learning and Data Mining in Pattern Recognition*, 2012.

[107] Oxford Bus Company. “Mobile Ticketing Technology Launched for Oxford SmartZone", https://www.oxfordbus.co.uk/press-release-oxford-smartzone-app (https://archive.vn/IS3Vk), 2022.

[108] C. Palow. “After Watching This Talk, You’ll Never Look at Passwords the Same Again", 2013.

[109] A. Parate, M.-C. Chiu, C. Chadowitz, D. Ganesan, and E. Kalogerakis. “Risq: Recognizing Smoking Gestures with Inertial Sensors on a Wristband", *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2014.

[110] S. Pearman, J. Thomas, P. E. Naeini, H. Habib, L. Bauer, N. Christin , L. F. Cranor, S. Egelman, and A. Forget. “Let’s Go in for a Closer Look: Observing Passwords in Their Natural Habitat", *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017.

[111] K. Pfeuffer, M. J. Geiger, S. Prange, L. Mecke, D. Buschek, and F. Alt. “Behavioural Biometrics in VR: Identifying People from Body Motion and Relations in Virtual Reality", *CHI Conference on Human Factors in Computing Systems (CHI)*, 2019.

[112] A. T. Purnomo, Y. S. Gondokaryono, and C.-S. Kim. “Mutual Authentication in Securing Mobile Payment System Using Encrypted QR Code based on Public Key Infrastructure", *IEEE International Conference on System Engineering and Technology (ICSET)*, 2016.

[113] N. Quermann, M. Harbach, and M. Dürmuth. “The State of User Authentication in the Wild", *Symposium on Usable Privacy and Security (SOUPS)*, 2018.

[114] A.-I. Radu, T. Chothia, C. J. P. Newton, I. Boureanu, and L. Chen. “Practical EMV Relay Protection", *IEEE Symposium on Security and Privacy (S&P)*, 2022.

[115] R. Raghavendra and C. Busch. “A Low Cost Wrist Vein Sensor for Biometric Authentication", *IEEE International Conference on Imaging Systems and Techniques (IST)*, 2016.

[116] D. A. Ramli, M. Y. Hooi, and K. J. Chee. “Development of Heartbeat Detection Kit for Biometric Authentication System", *Procedia Computer Science*, Vol. 96, 2016.

[117] J. Ranjan and K. Whitehouse. “Towards Recognizing Person-object Interactions Using a Single Wrist Wearable Device", *ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2016.

[118] K. B. Rasmussen, M. Roeschlin, I. Martinovic, and G. Tsudik. “Authentication Using Pulse-response Biometrics", *Network and Distributed System Security Symposium (NDSS)*, 2014.

[119] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman. “Activity Recognition from Accelerometer Data", *Association for the Advancement of Artificial Intelligence (AAAI)*, Vol. 3, 2005.

[120] Q. Riaz, A. Vögele, B. Krüger, and A. Weber. “One Small Step for a Man: Estimation of Gender, Age and Height from Recordings of One Step by a Single Inertial Sensor", *IEEE Sensors Journal*, Vol. 15, 2015.

[121] O. Riva, C. Qin, K. Strauss, and D. Lymberopoulos. “Progressive Authentication: Deciding When to Authenticate on Mobile Phones", *USENIX Security Symposium (USENIX)*, 2012.

[122] M. Roshandel, A. Munjal, P. Moghadam, S. Tajik, and H. Ketabdar. “Multi-sensor Finger Ring for Authentication Based on 3D Signatures", *Human Computer Interaction. Advanced Interaction Modalities and Techniques (HCI)*, 2014.

[123] S. Sen, K. K. Rachuri, A. Mukherji, and A. Misra. “Did You Take a Break Today?: Detecting Playing Foosball Using Your Smartwatch", *IEEE International Workshop on Sensing Systems and Applications Using Wrist Worn Smart Devices*, 2016.

[124] S. Sen, K. Grover, V. Subbaraju, and A. Misra. “Inferring Smartphone Keypress via Smartwatch Inertial Sensing", *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017.

[125] S. Sen and D. Kotz. “VibeRing: Using Vibrations from a Smart Ring as an Out-of-band Channel for Sharing Secret Keys", *Journal of Pervasive and Mobile Computing*, Vol. 78, 2020.

[126] A. Serwadda, V. V. Phoha, S. Poudel, and L. Hirshfield. “fNIRS: A New Modality for Brain Activity-based Biometric Authentication", *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2015.

[127] J. Shang and J. Wu. “A Usable Authentication System Using Wrist-worn Photoplethysmography Sensors on Smartwatches", *IEEE Conference on Communications and Network Security (CNS)*, 2019.

[128] Y. Shen, H. Wen, C. Luo, W. Xu, T. Zhang, W. Hu, and D. Rus. “GaitLock: Protect Virtual and Augmented Reality Headsets Using Gait", *IEEE Transactions on Dependable and Secure Computing*, Vol. 16, 2019.

[129] C. Shi, J. Liu, H. Liu, and Y. Chen. “Smart User Authentication through Actuation of Daily Activities Leveraging WiFi-enabled IoT", *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2017.

[130] D. M. Shila and K. Srivastava. “CASTRA: Seamless and Unobtrusive Authentication of Users to Diverse Mobile Services", *IEEE Internet of Things Journal*, 2018.

[131] B. Shrestha, M. Mohamed, S. Tamrakar, and N. Saxena. “Theft-resilient Mobile Wallets: Transparently Authenticating NFC Users with Tapping Gesture Biometrics", *Annual Conference on Computer Security Applications (ACSAC)*, 2016.

[132] I. Sluganovic, M. Roeschlin, K. B. Rasmussen, and I. Martinovic. “Using Reflexive Eye Movements for Fast Challenge-response Authentication", *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016.

[133] J. Somorovsky, A. Mayer, J. Schwenk, M. Kampmann, and M. Jensen. “On Breaking SAML: Be Whoever You Want to Be", *USENIX Security Symposium (USENIX)*, 2012.

[134] O. Stannard and F. Stajano. “Am I in Good Company? A Privacy-protecting Protocol for Cooperating Ubiquitous Computing Devices", *Security Protocols XX*, 2012.

[135] E. Stobert and R. Biddle. “The Password Life Cycle: User Behaviour in Managing Passwords", *Symposium on Usable Privacy and Security (SOUPS)*, 2014.

[136] B. Stock and M. Johns. “Protecting Users Against XSS-based Password Manager Abuse", *ACM Symposium on Information, Computer and Communications Security (CCS)*, 2014.

[137] J. Sturgess and I. Martinovic. “VisAuth: Authentication over a Visual Channel Using an Embedded Image", *International Conference on Cryptology and Network Security (CANS)*, 2017.

[138] J. Sturgess, S. Eberz, I. Sluganovic, and I. Martinovic. “Inferring User Height and Improving Impersonation Attacks in Mobile Payments Using a Smartwatch", *IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2022.

[139] J. Sturgess, S. Eberz, I. Sluganovic, and I. Martinovic. “WatchAuth: User Authentication and Intent Recognition in Mobile Payments Using a Smartwatch", *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2022.

[140] J. Sturgess, S. Eberz, I. Sluganovic, and I. Martinovic. “Wearable Authentication in Mobile Payments Using a Smartwatch", *Inspired Research*, Issue 21, 2022.

[141] J. Sturgess, S. Eberz, I. Sluganovic, and I. Martinovic. “WatchAuth Dataset (Original)", *University of Oxford*, UUID: 8c7cbd89-7953-40ff-b28f-3aa3f8c59a01, 2022.

[142] J. Sturgess, S. Birnbach, S. Eberz, and I. Martinovic. “RingAuth: Wearable Authentication Using a Smart Ring", *arXiv preprint*, arXiv:2301.03594 [cs.CR], 2023.

[143] J. Sturgess, S. Köhler, S. Birnbach, and I. Martinovic. “CableAuth: A Biometric Second Factor Authentication Scheme for Electric Vehicle Charging", *ISOC Symposium on Vehicle Security and Privacy (VehicleSec)*, 2023.

[144] J. Sturgess, S. Köhler, S. Birnbach, and I. Martinovic. “Using Inertial Sensors to Authenticate Users", *Inspired Research*, Issue 23, 2023.

[145] J. Sturgess. “WatchAuth Dataset (Extended)", *University of Oxford*, DOI: 10.5287/ora-xqkqzwkex, 2023.

[146] J. Sturgess. “RingAuth Dataset", *University of Oxford*, DOI: 10.5287/ora-no9q44mzq, 2023.

[147] J. Sturgess and S. Köhler. “CableAuth Dataset", *University of Oxford*, DOI: 10.5287/ora-xvpaxmjmj, 2023.

[148] S. Sun and K. Beznosov. “The Devil Is in the (Implementation) Details: an Empirical Analysis of OAuth SSO Systems", *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2012.

[149] S. Tahir, A. Raheel, M. Ehatisham-Ul-Haq, and A. Arsalan. “Recognizing Human-object Interaction (HOI) Using Wrist-mounted Inertial Sensors", *IEEE Sensors Journal*, Vol. 21, 2021.

[150] Tangerine. “DoubleSafe: Your Picture, Your Phrase", https://www.tangerine.ca/en/security (https://archive.vn/rjbOp), 2017.

[151] B. Ur, S. M. Segreti, L. Bauer, N. Christin, L. F. Cranor, S. Komanduri, D. Kurilova, M. L. Mazurek, W. Melicher, and R. Shay. “Measuring Real-world Accuracies and Biases in Modeling Password Guessability", *USENIX Security Symposium (USENIX)*, 2015.

[152] S. Vhaduri and C. Poellabauer. “Multi-modal Biometric-based Implicit Authentication of Wearable Device Users", *IEEE Transactions on Information Forensics and Security*, Vol. 14, 2019.

[153] C. Wang, X. Guo, Y. Wang, Y. Chen, and B. Liu. “Friend or Foe?: Your Wearable Devices Reveal Your Personal PIN", *ACM Asia Conference on Computer and Communications Security (AsiaCCS)*, 2016.

[154] D. Wang and P. Wang. “The Emperor’s New Password Creation Policies", *European Symposium on Research in Computer Security (ESORICS)*, 2015.

[155] H. Wang, T. T.-T. Lai, and R. R. Choudhury. “MoLe: Motion Leaks through Smartwatch Sensors", *International Conference on Mobile Computing and Networking (MobiCom)*, 2015.

[156] M. Wang, W.-T. Zhu, S. Yan, and Q. Wang. “SoundAuth: Secure Zero-effort Two-factor Authentication Based on Audio Signals", *IEEE Conference on Communications and Network Security (CNS)*, 2018.

[157] R. Wang, S. Chen, and X. Wang. “Signing Me onto Your Accounts through Facebook and Google: a Traffic-guided Security Study of Commercially Deployed Single Sign-on Web Services", *IEEE Symposium on Security and Privacy (S&P)*, 2012.

[158] WeChat. “WeChat Pay", https://pay.weixin.qq.com/index.php/public/wechatpay (https://archive.vn/nxbi3), 2017.

[159] UnifyID. “UnifyID Implicit Authentication Platform", https://unify.id/product (https://archive.vn/8hUBi), 2018.

[160] J. Yang, Y. Li, and M. Xie. “MotionAuth: Motion-based Authentication for Wrist Worn Smart Devices", *IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2015.

[161] F. Yao, S. Y. Yerima, B. J. Kang, and S. Sezer. “Continuous Implicit Authentication for Mobile Devices based on Adaptive Neuro-fuzzy Inference System", *International Conference on Cyber Security and Protection of Digital Services*, 2017.

[162] Yoyo. https://www.yoyowallet.com/support.html (https://archive.vn/FYksa), 2017.

[163] X. Yu, Z. Zhou, M. Xu, X. You, and X. Li. "ThumbUp: Identification and Authentication by Smartwatch Using Simple Hand Gestures", *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2020.

[164] Y. Zeng, A. Pande, J. Zhu, and P. Mohapatra. "WearIA: Wearable Device Implicit Authentication based on Activity Information", *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2017.

[165] L. Zhang-Kennedy, S. Chiasson, and P. C. van Oorschot. "Revisiting Password Rules: Facilitating Human Management of Passwords", *APWG Symposium on Electronic Crime Research*, 2016.

[166] T. Zhao, Y. Wang, J. Liu, and Y. Chen. "Your Heart Won't Lie: PPG-based Continuous Authentication on Wrist-worn Wearable Devices", *International Conference on Mobile Computing and Networking (MobiCom)*, 2018.

[167] H. Zhu, W. Jin, M. Xiao, S. Murali, and M. Li. "BlinKey: A Two-factor User Authentication Method for Virtual Reality Devices", *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 4, 2020.