WatchAuth: User Authentication and Intent Recognition in Mobile Payments using a Smartwatch

Jack Sturgess, Simon Eberz, Ivo Sluganovic, and Ivan Martinovic

Department of Computer Science, University of Oxford, Oxford, UK
{firstname.lastname}@cs.ox.ac.uk

2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P) | 978-1-6654-1614-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/EUROSP53844.2022.00031

***Abstract*—In this paper, we show that the tap gesture, performed when a user 'taps' a smartwatch onto an NFC-enabled terminal to make a payment, is a biometric capable of implicitly authenticating the user and simultaneously recognising intent-to-pay. The proposed system can be deployed purely in software on the watch without requiring updates to payment terminals. It is agnostic to terminal type and position and the intent recognition portion does not require any training data from the user. To validate the system, we conduct a user study (n=16) to collect wrist motion data from users as they interact with payment terminals and to collect long-term data from a subset of them (n=9) as they perform daily activities. Based on this data, we identify optimum gesture parameters and develop authentication and intent recognition models, for which we achieve EERs of 0.08 and 0.04, respectively.**

1. Introduction

The popularity of cashless and contactless payment systems continues to grow. In the UK, card payments surpassed cash payments for the first time in 2018 [33]. Going forward, NFC-enabled mobile payments are projected to account for over 27% of market share by the end of 2021 [42], driven by user preference for usability and enhanced security features [19].

Mobile payment systems (also known as tap-and-pay), such as Google Pay, enable the user to provision one or more payment cards to a virtual wallet that is accessible via a smartphone, which can then be used to make payments over NFC (using tokenisation to protect card details). Typically, these systems require two factors to authenticate the user and utilise the fingerprint reader or face recognition camera of the device where possible.

In recent years, mobile payment systems have had their functionality extended onto smartwatches. When paired with a smartphone, a watch can access the same virtual wallet and then make payments over NFC, even when the phone is not present. Current smartwatches do not offer fingerprint readers or cameras, but use short passcodes for authentication. For convenience, a smartwatch can be configured to remain unlocked (*i.e.*, in an authenticated state) after a single passcode entry until it is restarted or removed from the wrist. Apple Pay Express Travel mode [2], rolled out in 2020, enables payments to be made by an Apple phone or watch at busy transport barriers without it needing to be unlocked at all, but limiting the scope of misuse by working only for certain payments. In each case, explicit authentication actions are obviated to improve usability at a potential cost to security. Moreover, without explicit user interaction with the terminal, an uncertainty arises for the payment provider as to whether the user intended to make the payment at all.

In 2020, the European Union introduced the Updated Payment Services Directive (PSD2) [43], overhauling payment regulations for the banking sector and establishing a precedent that other nations are likely to follow. One of its core principles, entitled Strong Customer Authentication, mandates the use of multi-factor authentication 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). For mobile payments, possession of the smart device (and its tokens) is one factor, so at least one more is needed.

Given the trend towards convenience, implicit factors that do not require any user effort are becoming more desirable. Fuelled by the availability of multifarious embedded sensors, recent work has demonstrated the use of a variety of factors, such as GPS location, usage habits, environmental conditions, proximity to other devices, and behavioural biometrics, in combination, to authenticate the user continuously and unobtrusively on Web browsers [13], smartphones [15, 17, 18, 20, 22, 23, 32, 40, 46], and wearable devices [28, 29, 45]. In one particular system, Shrestha *et al.* [41] showed that users making tap-and-pay payments with a smartphone can be authenticated by their tap gestures using various sensors in the smartphone.

In this work, we focus on implicit authentication and intent recognition in mobile payments using a smartwatch. We are motivated by the lack of biometric authentication options that are available to smartwatch users, coupled with advances in usability that may cast doubt on user intent during the payment process. We conduct a user study to collect wrist motion data from users as they interact with point-of-sale terminals and show that the tap gesture is sufficiently distinct between users that it can be used to authenticate users implicitly. We also collect a large dataset of relevant non-tap gestures from users as they perform other activities and show that the tap gesture is sufficiently recognisable between gestures that it can be used to infer whether a payment is intentional, providing a new technique to allay uncertainty in convenience-optimised payment schemes and to strengthen system security by rejecting unintentional payments.

DOI 10.1109/EuroSP53844.2022.00031

Contributions.

- Using only wrist motion data, we show that a tap gesture performed by the user while making a payment with a smartwatch can implicitly authenticate the user and recognise intent-to-pay. Our system runs on the watch and does not require any changes to terminals.
- We show that our approach can be applied to real-time data for in-store usage and to historic data for retrospective fraud detection, offering a defence against malicious, shared, and accidental payments.
- Our authentication model is terminal-agnostic, so does not need any specific terminal type or position.
- Our intent recognition model is user-agnostic, so does not need training data from the user during the enrolment phase and is innately resilient to drift.
- We implement our system to allow for real-world evaluation and show that our models can be tuned for use as a second factor in an existing system, providing a strict improvement to security (by adding a layer of false acceptance detection and introducing an unsharable factor) with negligible cost to usability.
- We make the code and data required to reproduce our results available at https://github.com/jacksturgess.

Paper Structure. The rest of this paper is organised in the following way. Section 2 presents a summary of gesture biometrics. Section 3 outlines the challenges of using a smartwatch compared with a smartphone and details our system and threat models. Section 4 describes our experimental apparatus. Section 5 explains the methods that we employ to collect and process data, train classifiers, and measure performance. Section 6 presents and analyses our results. Section 7 discusses peripheral topics. Section 8 compares our approach to related work. Section 9 considers limitations. Section 10 concludes the paper.

2. Background

In the context of authentication, a biometric is a measured characteristic of an individual that should be unique, persistent, and hard to impersonate. A biometric can be categorised as either physiological or behavioural. Physiological biometrics measure a physical feature of the user, such as a fingerprint or retina scan. Behavioural biometrics measure patterns of movements exhibited by the user, such as gait or keystroke dynamics. Biometric authentication systems first require an enrolment phase, in which features from the user's biometric measurements are extracted and encapsulated in a template; when the user attempts to authenticate, a fresh measurement is taken and features are extracted anew and matched against the template. Behavioural biometrics tend to need a longer enrolment phase, which had rendered them largely impractical before ubiquitous sensors became available.

Physiological biometrics are typically measured in discrete, explicit actions performed solely for the purpose of authentication, such as touching or looking at a sensor. Behavioural biometrics are measured over time and without any effort on the part of the user—such factors are referred to as *implicit*, because they can be measured while the user engages in other tasks. Implicit factors facilitate *continuous* authentication, where a system authenticates the user often and unobtrusively to maximise its confidence in his identity at all times.

In terms of PSD2, all biometrics are inherence-based. Biometrics cannot be forgotten or guessed (like knowledge-based factors) or lost or stolen (like possession-based factors), but they can *drift* (naturally change) over time. Lee *et al.* [24] demonstrated a mitigation technique for drift using an update mechanism to replace the old user template with a new one by averaging it with the latest signal. User-agnostic systems are resistant to drift as they are not trained on the individual user, so changes in his behaviour are irrelevant. A template could also be *poisoned* (maliciously changed) by exploiting the update mechanism such that the template is gradually morphed to wrongly accept an impersonator's signals [4, 27], although this too can be mitigated, here by limiting the frequency of updates or gating the mechanism behind another authentication factor. Biometrics cannot be shared between users, ensuring a one-to-one identity mapping, but mimicry attacks are typically feasible given sufficient resources and trait collisions can occur in large user sets.

A gesture is a series of movements made by the user; it could be explicit, such as a nod to indicate an affirmative response or a touchscreen swiping pattern to activate some functionality (*e.g.*, the *pinch gesture* to resize or zoom), or it could be performed innately as part of another activity and thus be implicit, such as the movement of the wrist while typing. A gesture as a biometric is typically measured using inertial sensors, such as accelerometers and gyroscopes. Early work in gesture-based authentication focused on explicit gestures, showing that users could be distinguished from one another based on their performing an explicit action, such as various arm-swinging gestures, with handheld devices [26, 30, 36] or using wrist-worn sensors [25, 45, 47]. More recent work has considered implicit gestures and their feasibility for use in continuous authentication. Frank *et al.* [12] use sensors in a smartphone to authenticate the user based on touchscreen interaction over time. Han *et al.* [16] use sensors mounted on smart home devices to identify occupants based on object interaction. Nassi *et al.* [34] use wrist-worn sensors to verify users as they hand-write signatures and Griswold-Steiner *et al.* [14] extend the idea to general handwriting.

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.

3. Objectives and Assumptions

3.1. Design Considerations

We have chosen to focus on the use of a smartwatch, rather than a smartphone, due to the following two challenges that make it a more interesting problem.

Firstly, the starting point of a tap gesture is more difficult to determine on a watch. 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 on all devices to initiate a payment (*e.g.*, double-clicking a side button); 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 study 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.

Secondly, a watch undergoes a much greater change in orientation to perform a tap gesture. A phone can be tapped against 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 relative to the orientation of the device (see Section 4.3); 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 in which to stabilise this behaviour would be to transform the sensor data into the external co-ordinate system (*e.g.*, as used by Ardüser *et al.* [3]). Unfortunately, this would require some ground truth from both reference frames to calibrate the transformation, which is not practical in our case—either we would need (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.

3.2. System Model

We consider a system model in which a user is wearing a smartwatch on his wrist and is using it to make NFC-enabled payments at point-of-sale terminals in a typical setting—namely, in a shop or at the entry barriers to a transport 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 gyroscope and that we have access to the data generated by these sensors. We use data from the inertial sensors only, regardless of the availability of any medical, environmental, or location sensors that the watch may have; this enables us to compare our results fairly with those of related works (in Section 8).

While the current generation of smartwatches are dependent upon a paired smartphone for administration purposes and the installation of apps, 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 or proximity data, that one might provide.

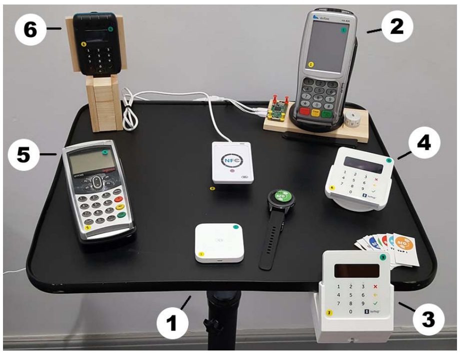


Figure 1: The equipment used in our experiment: six fixed terminals (labelled, see Table 1 for details), an NFC reader, a Raspberry Pi for timestamp collection, and a smartwatch.

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 1: Details of the terminals used in our experiment; the indices match those labelled in Figure 1 and 'F' is the freestyle terminal. *Height* indicates the height from the floor to the lowest point of the terminal; *Tilt* is the angle of inclination at the lowest point of the terminal from the flat platform; and *Distance* is the distance from the front of the platform to the point of the terminal that is closest to the user. Terminals 2 and 6 match terminals on self-service checkouts at supermarket chains, roughly an arm's length from the user.

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 obscure, 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 combination of these models forms our system, which we call WatchAuth.

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 against our threat model in three use-cases. Firstly, 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. Secondly, 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. Thirdly, for use as an *additional factor*, we optimise our models to minimise the occurrence of false negatives, so that implementing our models alongside an existing authentication system provides only a strict improvement to security with negligible cost to usability.

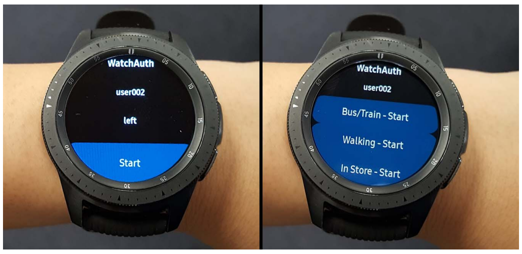


Figure 2: The two modes of our data collection app, for in-lab (left) and out-of-lab (right) usage.

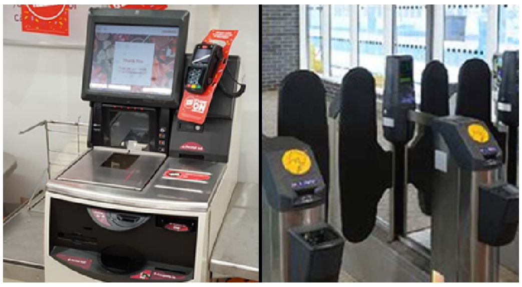


Figure 3: Two examples of point-of-sale terminals commonly found in the UK: the terminal on a self-service checkout at a supermarket (left) and the terminals on a train station barrier (right, in yellow). The positions of these terminals are replicated by our Terminals 2 and 3, respectively.

3.3. Threat Model

Given that our system model branches into two, our threat model considers a separate attacker against each.

For the authentication model, we consider an adversary that has possession of a legitimate user's smartwatch, has unlocked it, and is attempting to use it to make a payment at an unstaffed 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 other users by using only tap gestures.

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 skimming attack, or it could be accidental. We 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 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 or biometric templates, malware, or denial of service attacks.

4. Experimental Design

4.1. Experiment Overview

To evaluate the extent to which wrist motion data can be used to achieve our goals, we designed and conducted a user study to collect data. Our experiment consists 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, and a Samsung Galaxy Watch running the Tizen 4.0 operating system (as shown in Figure 1). The experiment also includes a screen, connected wirelessly to the Raspberry Pi, that instructs the user when to perform each tap gesture.

We built a data collection app with the Tizen Studio IDE and installed it on the smartwatch to continuously collect timestamped wrist motion data as the user wears it (as shown in Figure 2). To collect data for a single tap gesture, the NFC reader is first affixed to the front of the terminal and the user performs the tap gesture on it as if making an NFC-enabled payment. Each NFC contact point timestamp is captured by the Raspberry Pi and a short spacing delay is initiated before the user is instructed to perform the next tap gesture.

A subset of participants also collected data outside of the lab by wearing the smartwatch during various non-payment activities. The user selects the setting and the app collects and labels the motion data until stopped.

4.2. Point-of-Sale Terminals

To emulate real-world mobile payment scenarios, we capture tap gestures using *seven* terminals: six in fixed positions (as shown in Figure 1 and detailed in Table 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 (as shown in Figure 3). For the freestyle terminal, the user picks up the NFC reader directly with his other hand and performs a tap gesture against it, returning it after each interaction, just as if a vendor 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 remain deactivated throughout the experiment as their functionality is not required. For consistent data collection, we affix the NFC reader to the front of each terminal when using it. As such, the terminals should be regarded only as fixtures that enforce heights and angles, as well as a tool for immersing the user in a payment scenario.

During the user study, each participant is instructed to stand in front of the terminal platform while performing his tap gestures. Aside from this, we do not prescribe any constraints on positioning as we want the user to stand and interact with the terminals comfortably and naturally as though making payments in a real-world setting.

4.3. Sensor Module

The Tizen platform provides four inertial sensors directly or derived from the MEMS accelerometer and gyroscope in the smartwatch. The *accelerometer* measures change in velocity and the *gyroscope* measures angular velocity. The *linear accelerometer* is derived from the accelerometer with the effects of gravity excluded. The *gyroscope rotation vector* (GRV) is a fusion of accelerometer and gyroscope readings to compute the orientation of the device. Our app collects timestamped data from all four at a sampling rate of 50 Hz.

The inertial sensors measure wrist motion along three axes that are relative to the frame of the watch (as shown in Figure 4). Motion along the x-axis corresponds 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.

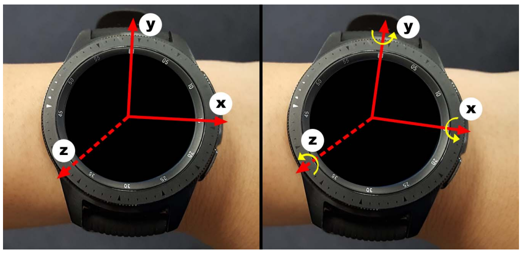


Figure 4: The sensor axes relative to the frame of the smartwatch; the positive x-axis points towards the hand when worn on the left wrist; angular velocity is measured along the yellow arrows.

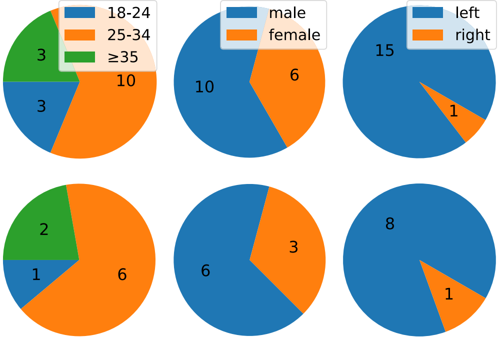


Figure 5: The distribution of age (left), sex (centre), and on which wrist the smartwatch was worn (right) of users in our main lab experiment (top, n=16) and the subset of users that collected additional data (bottom, n=9).

5. Methods

5.1. User Study

To collect data, we conducted a user study that was reviewed and approved by the relevant research ethics committee at our university. We recruited 16 participants, including students and members of the public; a breakdown of participant demographics is shown in Figure 5.

Each participant attended 3 data collection sessions. In each session, the participant performed 10 tap gestures on each of the seven terminals. The first and second sessions occurred back-to-back, separated by a short break lasting roughly 5 minutes; the third session occurred on a different day. In total, we collected 210 tap gestures from each user.

A subset of 9 participants also collected and labelled data outside of the lab while walking, commuting on a bus or train, or in a shop—daily activities identified in our threat model as likely settings for an unintentional payment. In total, we collected 1,088 minutes of out-of-lab activity data across all users (601 minutes walking, 317 minutes on a bus or train, 148 minutes in-store, and 22 minutes of combined activities[1]).

Of the participants, 19% indicated that they regularly wore a non-smart watch and 38% wore a smartwatch; only 6% had ever made a payment using a smartwatch compared with 81% who had paid using a smart*phone*. 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.

5.2. Data Processing

We collect timestamped data from four inertial sensors. Each accelerometer, gyroscope, or linear accelerometer sample is given in the form (t, x, y, z) and represents the change in velocity or angular velocity of the smartwatch along each axis at time t. Each GRV sample is given as a quarternion in the form (t, x, y, z, w) and approximates the orientation at time t.

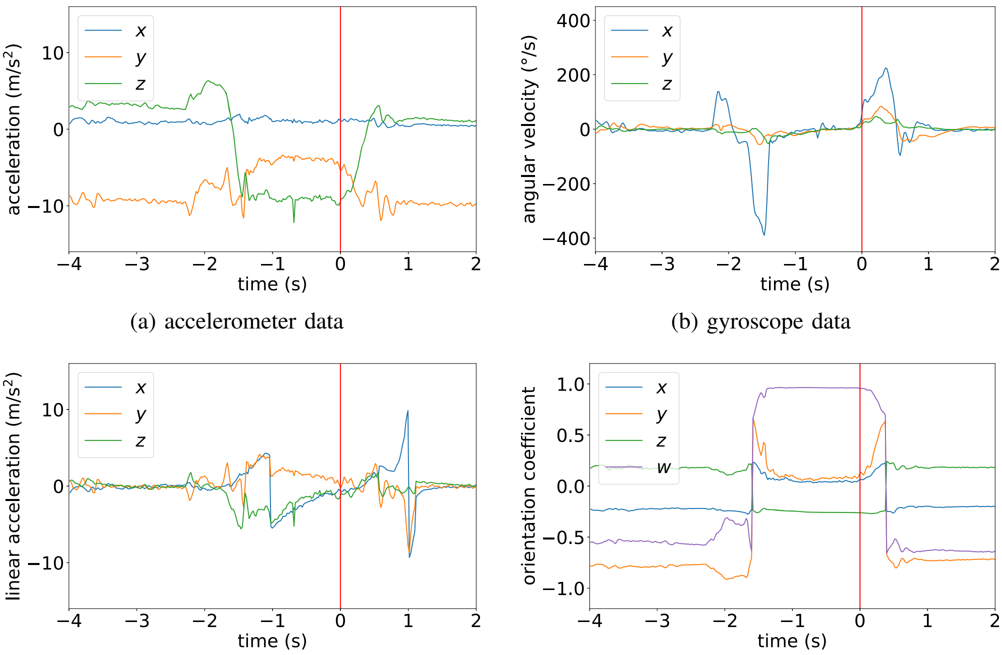
We express a tap gesture using series of inertial sensor data samples within a time window. To retrieve the tap gestures for each user, we segment 4-second blocks of sensor data by using the NFC contact point timestamps as the endpoint of each window. We found that a 4-second maximum window size was sufficient to encapsulate the entirety of each gesture.

Sensor data for an exemplar tap gesture is shown in Figure 6. Here, 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 connection. An additional 2 seconds of data after the NFC contact point is included for context, showing the user's arm withdrawing.

To investigate optimum tap gesture parameters, we compare (in Section 6) the performances of gestures bounded by various 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 $T_E = T_0 - o$ and $T_S = T_E - s$.

We segment the out-of-lab sensor data into comparable 4-second blocks. In total, we obtained 32,441 4-second, non-tap gestures by segmenting the data every 2 seconds to ensure a 50% overlap so as not to eliminate any interesting regions.

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



(a) accelerometer data

(b) gyroscope data

(c) linear accelerometer data

(d) GRV data

Figure 6: Visualisations of the same tap gesture, showing data 4 seconds before the NFC contact point and 2 seconds after from the four sensors used to collect our data. The NFC contact point is set at time 0 (indicated with a red line).

5.3. Feature Extraction

For each windowed gesture, we apply a low pass filter to the data to reduce noise and then process the following five dimensions for each accelerometer, gyroscope, or linear accelerometer sample: the filtered x-, y-, and z-values, the magnitude of those filtered values, and the magnitude of the unfiltered (original) values, where the magnitude of $\{x, y, z\}$ is $\sqrt{x^2 + y^2 + z^2}$. As GRV samples are expressed as quaternions, for those we process only the four filtered dimensions (since the Euclidean norm of a quaternion is always 1). In total, we process each gesture in 19 dimensions.

We extract the following statistical features in each dimension: *minimum*, *maximum*, *mean*, *median*, *standard deviation*, *variance*, *inter-quartile range*, *kurtosis*, *skewness*, and *peak count*. For each gesture, 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. Ultimately, we extract a feature vector containing 220 members for each gesture.

We began with a larger set of features that had been used successfully by other authors in similar scenarios [1, 8, 28, 39] and we pruned it down by using normalised Gini importances to reject the least informative features. The Gini importance of a feature is a measure of its effect to decrease the impurity of the model [5] and thus its positive impact on classification. 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 calculate velocity and displacement due to the variability in orientation of the watch in transit between the user and terminal.

5.4. Classification

We employ three supervised learning approaches, training separate models for authentication and intent recognition.

For our authentication model, we train a set of classifiers that are user-dependent and terminal-agnostic. In each, 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 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 template is created) and those collected in the third session as testing data (analogous to an authentication phase). For each user, we train multiple classifiers, each one excluding a different fixed terminal; we train the classifier on users' tap gestures performed on the other terminals and then test it on those performed on the excluded terminal, to ensure that the model is terminal-agnostic and generalised. With 16 users and 6 fixed terminals, this gives $16 \times 6 = 96$ separate classifiers.

Furthermore, for investigative purposes, we also train a *terminal-specific* authentication model, in which each classifier is trained and tested on tap gestures performed on a single terminal. This model enables us to compare the effectiveness of dedicated, terminal-specific classifiers for systems that have standardised terminals, such as public transport systems.

For our intent recognition model, we train a set of tap gesture recognition classifiers that are not user-dependent. We take all users' tap gestures as the positive class and all users' non-tap gestures as the negative class—that is, we treat tap gestures performed on a terminal as intentional and other gestures as unintentional. As the two classes are highly imbalanced in this model (with the negative class being many times larger than the positive), we apply a stratified 10-fold cross-validation approach to preserve class proportionality in training and testing folds and to avoid bias towards the more populous class. For each user, we train classifiers using only the data of other users to ensure that they are user-agnostic.

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 [6, 31]. To balance relevance with learning time, we include 100 trees in each forest [37]. To reduce the impact of random generation on our results, and to avoid the deceptive practice of selecting results only from the most performant forest, we train and test each classifier ten times with different forest randomisation seeds and average the outcomes.

5.5. Performance Metrics

In each model, the *true positives* is the number of times that the positive class (*i.e.*, the legitimate user or intentional gesture) is correctly accepted; the *true negatives* is the number of times that the negative class (*i.e.*, the adversary or unintentional gesture) 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.

To quantify the performance of our models and to compare our results with those of our closest related work [41] using the same metrics, we calculate precision, recall, and F-measure. Precision indicates *security*, by measuring how well the model rejects the negative class, and recall indicates *usability*, inasmuch as it measures how well the model avoids misclassifying the positive class and causing inconvenience to the user; F-measure is the harmonic mean of the precision and recall (equally weighted), offering a rough fusion of the two, which is ideal for our purposes as we want to consider a balance of both security and usability.

To quantify the performance of our models when used as an additional factor in an existing system, we want to measure the security benefit that we provide without incurring any cost to usability in the form of false negatives. To evaluate our models in this regard, we find for each model the optimum decision threshold that yields minimal false negatives and we measure the false acceptance rate (FAR) there. The FAR inversely indicates *security*, by measuring the likelihood that the negative class will be wrongly accepted. The false rejection rate (FRR) inversely indicates *usability*, by measuring the likelihood that the positive class will be wrongly rejected.

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 the 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 minimise the occurrence of false negatives, we optimise our models by selecting θ such that the FRR is less than 0.01% and the corresponding FAR is as low as possible (an example is shown in Figure 11).

The FAR and FRR are antagonistic insofar as setting θ to favour one will disfavour the other. The crossover point is called the equal error rate (EER) and is a measure of system performance when consideration is balanced evenly between security and usability. To measure the impact of optimisation on a model (*i.e.*, how much of the potential security gains are sacrificed to minimise the impact on usability), we compare the FAR when optimised with the EER (and therefore the FAR) when not.

6. Results

6.1. 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 connection, owing to the short-ranged nature of NFC technology. Once a connection 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 0.5 to 1.5 seconds before the NFC contact point, depending on how quickly the NFC connection is established, and ends 0 to 1 seconds after, depending on how quickly the user reacts to the notification.

6.2. Optimum Window Parameters

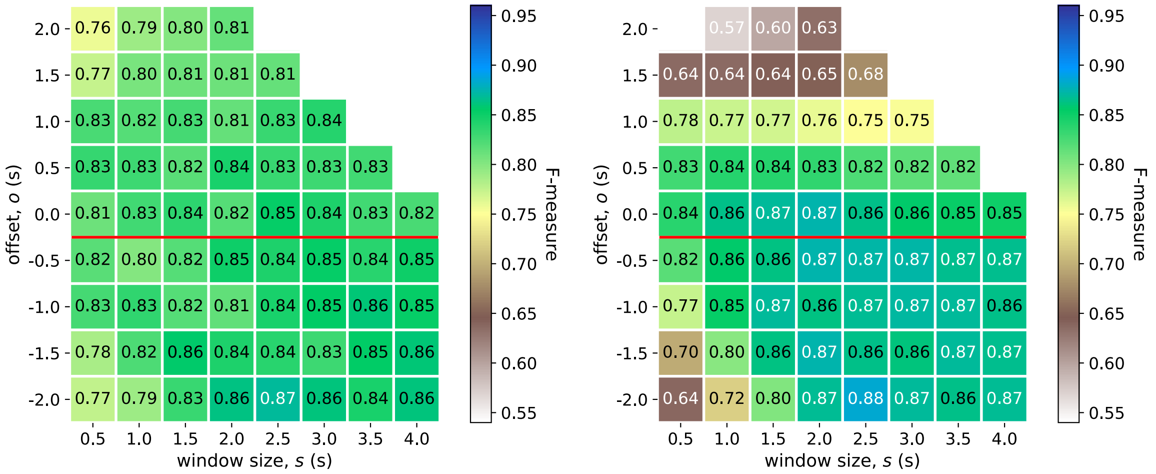
Figure 7 shows the F-measure scores for our authentication and intent recognition models by window size and offset. Each score in Figure 7a is the average of scores from $16 \times 6 \times 10 = 960$ classifiers and in Figure 7b, from $16 \times 10 = 160$ (see Section 5.4 for details).

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 7a that our model achieves an average F-measure score of 0.83 with very little deviation across window sizes with offsets $o \leq 1$; our best score is 0.85 at $\{s = 2.5, o = 0\}$. We find that even 0.5 seconds of wrist motion data is sufficient to authenticate the user with a tap gesture.

For intent recognition, we find in Figure 7b that offset is the determining factor and that smaller offsets yield stronger results; the model achieves its best in-store results when $o = 0$, with an average F-measure score of 0.86.

Considering Figures 7a and 7b together, we find that the optimum window parameters for in-store usage, favouring authentication, 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 authenticate the user with an average F-measure score of 0.85 (precision 0.92, recall



(a) authentication model

(b) intent recognition model

Figure 7: Average F-measure scores 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.

0.84, EER 0.10) and recognise intent-to-pay with an average F-measure score of 0.86 (precision 0.93, recall 0.82, EER 0.04).

Retrospective Fraud Detection. For historic analysis, we consider all results in the heatmaps. We see a slight improvement in F-measure scores for windows containing sensor data collected after the NFC contact point, especially where the tap gesture spans it (*i.e.*, where $o < 0$ and $s > |o|$); this suggests that a user's withdrawal is at least as distinctive as his movement towards the terminal. The optimum window parameters here are $\{s = 2.5, o = -2\}$.

For authentication, our results suggest that the withdrawal phase is more distinctive between users than the alignment phase, as windows that contain more data from that phase tend to yield stronger results, although the differences are too small to be conclusive. We can see that the EERs in Figure 9a corroborate this; they also suggest that larger window sizes produce a better balance of security and usability. The results begin to decline at the top- and bottom-left corners of Figure 7a; these windows have the highest likelihood of containing data that is irrelevant to the tap gesture, data that is collected from random movements before or after the tap gesture, respectively, and is therefore harder to classify.

For intent recognition, we see that the alignment phase of the gesture is the most distinctive between gesture types. 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 in the inverse, as we see that the fewer alignment phase samples a window has, the weaker its results: at the top of Figure 7b, the larger the positive offset, the fewer alignment phase samples it is likely to contain, and at the bottom-left, the larger the negative offset and the smaller the window, the fewer alignment phase samples 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 connection are peculiar to the tap gesture and so act to distinguish it from other gestures. Strong results are given by those windows that span all three phases, in particular $\{2 \leq s \leq 3, o = -0.5\}$.

6.3. Feature Informativeness

To see which features are most informative to our models, we sum the top five features, sorted by Gini importance, of each classifier. Table 2 shows the modal top-five features summed over classifiers with optimum window parameters $\{s = 2.5, o = 0\}$ and across all windows. (Note that, *w.r.t.* the counts, there are six times more classifiers for authentication.)

For authentication, we see in Table 2a that features derived from the y-axis 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 extremes in acceleration along the x-axis, representing the rapidity of the extension and withdrawal of the arm, is also shown to be important.

For intent recognition, we see in Table 2b that the number of peaks in the magnitude of linear accelerometer samples is of particular importance, far exceeding any other 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 saw 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-value of the gyroscope data, which measures the tilt of the wrist from side to side (see Figure 4), 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-value of the accelerometer data 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.

6.4. 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 gauge the feasibility of our approach on devices with fewer sensors, we trained a set of sensor-specific models in which each classifier is trained and tested on data from a subset of sensors. Figure 8 shows the F-measure scores for models using data from (i) the accelerometer and gyroscope and (ii) only the accelerometer (*cf.* Figure 7, which uses all four).

For authentication, we see that there is a monotonic improvement in results the more sensors are included, with few exceptions.

For intent recognition, we see comparable results across all windows in Figures 7b and 8d, but improved results up to 0.89 in Figure 8b; this suggests that the inclusion of the linear accelerometer and GRV is unnecessary and pollutes the classifiers. Table 2c shows the modal top-five features for the intent recognition model with these sensors omitted; compared with Table 2b, 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 largely unchanged.

6.5. Terminal Positions

Table 3 shows the F-measure scores for our terminal-specific authentication and intent recognition models, in which each classifier is trained and tested on tap gestures performed on a single terminal, for the four optimum windows identified above for in-store usage. We also include the terminal-agnostic results from our core authentication model, trained on tap gestures performed on all terminals other than the one under test, for comparison.

For authentication, we see in Table 3a that, per terminal, a similar trend is presented across the different window sizes, suggesting that window size is a more important factor than terminal position. We see that the results for Terminal 3 and the freestyle terminal are consistently better than those for the terminal-agnostic model. Terminal 3 protrudes towards the user and elicited a change in pose from users in the study as they interacted with it, resulting in a smoother, more comfortable tap gesture; the freestyle terminal accommodated this as well. The strength of results for these two terminals suggest that the reaching phase is more significant in the terminal-specific classifiers, perhaps because it is less constrained than the alignment phase and so offers the opportunity for user-distinctive traits to present. This appears to be less pronounced in the terminal-agnostic classifiers, where the training sets are broader.

For intent recognition, we find in Table 3b that the distance between the user and the terminal appears to correlate well with our results. As shown in Table 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. Here, the protrusiveness of Terminal 3 works against us, as the smoother gesture that results from the change in pose is less distinctive; whereas, for Terminals 1, 4, and 5, no such change in pose was prompted, so users interacted with these 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 gesture in that case. We find that user comfort is beneficial in authentication, but detrimental in intent recognition.

$s = 2.5, o = 0$		All Windows	
Feature	**Count**	**Feature**	**Count**
`Acc-x-min`	218	`Acc-x-min`	9261
`Gyr-y-velomean`	207	`Acc-x-max`	9225
`Gyr-y-mean`	178	`Gyr-y-mean`	8129
`Gyr-y-disp`	169	`Acc-x-velomean`	7513
`Gyr-y-max`	146	`Gyr-y-velomean`	6459
`Gyr-y-med`	143	`GRV-x-min`	6304
`Gyr-y-velomax`	125	`Gyr-y-med`	6234
`Acc-x-max`	121	`GRV-x-mean`	6085
`Gyr-z-max`	120	`Gyr-y-velomax`	6018
`GRV-x-min`	116	`Gyr-y-disp`	5627

(a) authentication model

$s = 2.5, o = 0$		All Windows	
Feature	**Count**	**Feature**	**Count**
`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

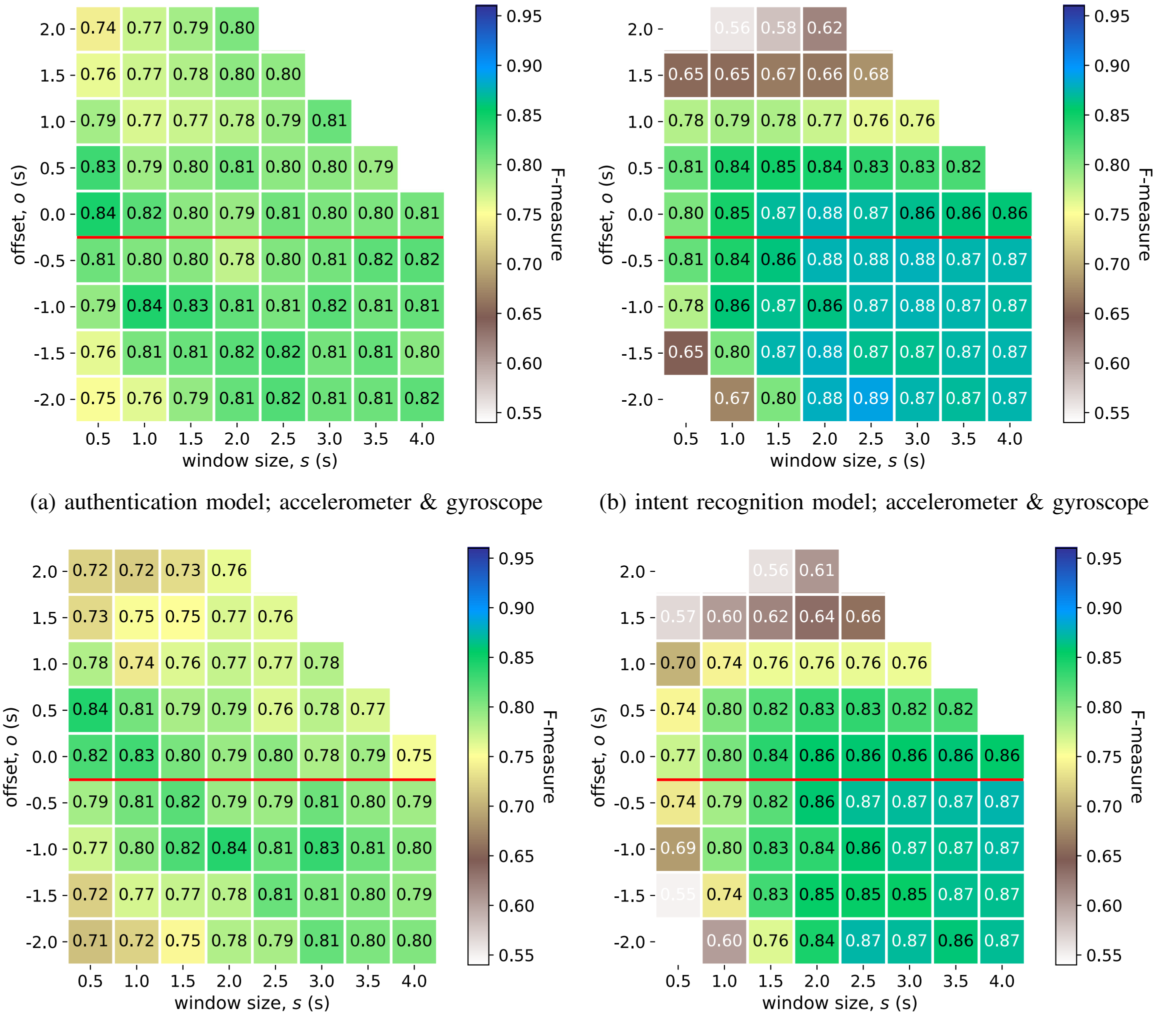
(b) intent recognition model

$s = 2.5, o = 0$		All Windows	
Feature	**Count**	**Feature**	**Count**
`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

TABLE 2: 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 magnitude of the unfiltered $\{x, y, z\}$ values and `disptotal` is the Euclidean displacement.

The strongest results in Table 3b are given by Terminal 1 and the weakest by Terminals 2, 3, and 6. The former is flat on the surface and so demands the greatest wrist rotation from the user, whereas the latter three are inclined at angles of 45° or greater (see Table 1) and so require the



(a) authentication model; accelerometer & gyroscope

(b) intent recognition model; accelerometer & gyroscope

(c) authentication model; accelerometer

(d) intent recognition model; accelerometer

Figure 8: Average F-measure scores for our authentication and intent recognition models by window size, offset, and sensors. Tap gestures that end at or before the NFC contact point, which are therefore compatible with in-store usage, are above the red line.

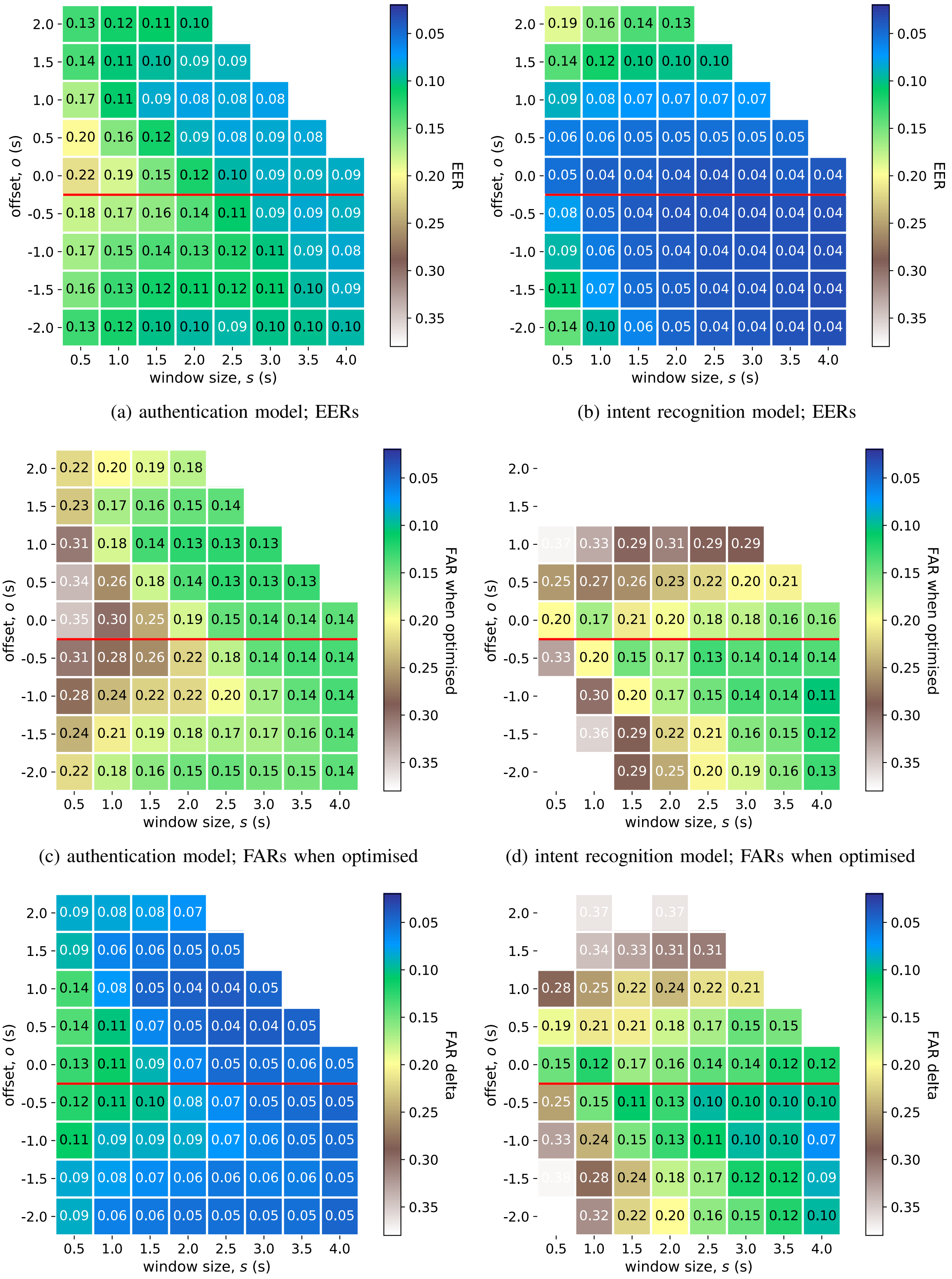
	F-measure			
Terminal	$s = 1.5$	$s = 2$	$s = 2.5$	$s = 3$
1	0.79	0.80	0.81	0.80
2	0.77	0.77	0.77	0.80
3	0.86	0.85	0.86	0.86
4	0.79	0.80	0.81	0.81
5	0.84	0.88	0.90	0.88
6	0.81	0.84	0.86	0.85
F	0.86	0.86	0.86	0.87
agnostic	0.84	0.82	0.85	0.84

(a) terminal-specific authentication model

	F-measure			
Terminal	$s = 1.5$	$s = 2$	$s = 2.5$	$s = 3$
1	0.88	0.87	0.86	0.86
2	0.67	0.72	0.68	0.68
3	0.65	0.65	0.68	0.67
4	0.81	0.80	0.80	0.79
5	0.81	0.81	0.80	0.79
6	0.69	0.65	0.73	0.67
F	0.66	0.73	0.81	0.90
agnostic	0.87	0.87	0.86	0.86

(b) intent recognition model

TABLE 3: Average F-measure scores for our terminal-specific authentication and intent recognition models with optimum window parameters ($o = 0$ in each case) for in-store usage by terminal. Terminal-agnostic results are included for comparison.



(a) authentication model; EERs

(b) intent recognition model; EERs

(c) authentication model; FARs when optimised

(d) intent recognition model; FARs when optimised

(e) authentication model; FAR deltas

(f) intent recognition model; FAR deltas

Figure 9: Average EERs, FARs when optimised for minimal false negatives, and the differences between these values (indicating the cost of optimisation) 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.

Activity Type	Number of Samples	Proportion of Samples (%)	FAR (%)
Walking	17890	55.15	2.99
Bus or Train	9463	29.17	4.66
In-store	4417	13.62	6.08
all	32441	100	3.77

TABLE 4: 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.

least. This echoes our finding in Section 6.3 and suggests that wrist rotation is a key discriminator between tap gestures and other gestures.

The results for the freestyle terminal improve significantly with larger windows. Here, the freedom to manipulate both the smartwatch and the terminal leads not only to a smoother gesture, but also to a shorter alignment phase, both of which likely contribute to weaker results; however, for $s = 3$, the results are better than those for the agnostic model, suggesting that the preparatory movements made in the reaching phase by users when interacting with the freestyle terminal, which included lifting the arm across the chest, are highly distinctive.

The terminal-agnostic approach is clearly superior to the terminal-specific approach for intent recognition, for all but the most awkwardly-positioned terminals. This shows that a classifier trained on tap gestures from a broader range of terminals becomes more effective at distinguishing a tap gesture from other gestures.

6.6. 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. 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 10 shows that our model can authenticate the user with an average EER of 0.16 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 might benefit the model over time, relaxing upfront requirements and incorporating subsequent tap gestures as the system is used. (Note that our intent recognition model is user-agnostic, so does not require training data from the user.)

6.7. 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 4 shows the number, proportion, and FAR of gesture samples of each type. We see that in-store gestures account for the greatest proportion of false positives; it is likely that actions such as reaching for a product on a shelf and rotating the wrist to read the label on a product exhibit some similar movement pattern fragments as those found in a tap gesture.

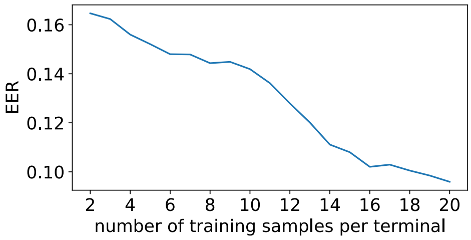


Figure 10: Average EERs for our authentication model if trained on different numbers of enrolment samples in optimum window $\{s = 2.5, o = 0\}$. Each classifier is trained on six terminals.

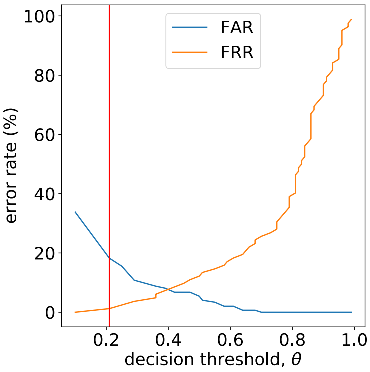


Figure 11: An example plot showing an EER of 7% at $\theta = 0.42$. At $\theta = 0.21$ (indicated with a red line), the model is optimised to minimise the FRR and has an FAR of 18%.

6.8. Cost of Optimisation

For our models to add a strict improvement to the security of an existing system, we need to ensure that we do not impose a burden on its usability, so we optimise the classifiers to minimise the occurrence of false negatives that would impose inconvenience (see Figure 11). Figures 9a and 9b show the EERs of our models (*cf.* Figure 7, which shows the F-measure scores). Figures 9c and 9d show the FARs of each classifier once its decision threshold has been adjusted to minimise the FRR. Figures 9e and 9f show the differences between the respective FARs before and after optimisation and therefore the cost.

We see that the cost of optimisation for authentication is relatively low; whereas, for intent recognition, the EERs start lower but the costs are much higher, suggesting steep FAR curves in those windows with fewer alignment phase samples. We find that the optimum window parameters for the optimised models for in-store usage are $\{3 \leq s \leq 4, o = 0\}$, which conveys a cross-model average FAR of 0.15, and for historic analysis, $\{s = 4, -2 \leq o \leq -1\}$, with an average FAR of 0.13.

Our approach adds two components to the security of an existing system. Firstly, we add a layer of false acceptance detection, the effectiveness of which is shown by these low FARs. Secondly, we introduce to the system an unsharable factor that ensures that only the legitimate user can make payments with the smartwatch. (Note that unsharability is an oft overlooked yet advantageous property of biometrics and one that cannot otherwise be achieved by knowledge- or possession-based factors.) The results above have shown that we are able to provide these security gains without imposing a burden on usability.

7. Discussion

Power Consumption. Smartwatches are designed to facilitate always-on sensing (*e.g.*, in health and fitness monitoring). To measure the impact of our data collection app in practical terms, we wore two Samsung Galaxy Watches in an identical state, but with one running our app. Without any effort put into performance optimisation, our app 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 i5-6500 processor. Using a benchmarking tool [35], we found that a Samsung Exynos W920 (a modern smartwatch processor) performs 26 times slower, so we would expect a response time of roughly 185 ms on a smartwatch for in-store usage. This could not be tested directly due to a lack of library support in Tizen Studio IDE.

8. Related Work

Authentication. With regard to authentication, Shrestha *et al.* [41] present the most closely related work. They consider 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 3.1). They achieve F-measure scores of up to 0.93 for authentication, a slight improvement on our results; however, they use cross-validation to train a classifier in an authentication use-case, which violates the requirement that training (enrolment) should precede testing (user verification) [9], potentially inflating their scores. Drilling into the results, we note that their classifiers consistently had higher scores for recall than for precision—ours had the opposite, suggesting that the smartwatch gesture favours security and the smartphone, usability. They find the ideal gesture size to be 1 second of sensor data and mention 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 6.2); furthermore, our sliding window approach makes it possible for us also to consider the case of retrospective fraud detection. They collect data with 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. We tabulate these differences in Table 5.

Key Aspects	This Work	[41]
device used for tapping	watch	phone
authentication	✓	✓
intent recognition	✓	×
real-world inspired terminal set-up	✓	×
inclusion of a non-fixed terminal	✓	×
in-store usage/real-time use-case	✓	✓
retrospective fraud detection use-case	✓	×
additional factor use-case	✓	×

TABLE 5: Comparison of the key aspects of this work with those of the most closely related work, Shrestha *et al.* [41].

Lee *et al.* [24] use inertial sensors on a smartphone to authenticate the user whenever the phone is picked up, defining the implicit *pick up gesture*, with the goal of reducing the number of explicit log-in actions required. Similar prior work by Conti *et al.* [7] authenticates the user as he makes or answers 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.* [21] use inertial sensors on a smartwatch to infer gait as the user walks for identification and authentication purposes, using 10-second windows of sensor data. Acar *et al.* [1] use inertial sensors on a smartwatch, in combination with keystroke dynamics measured at a workstation, to continuously authenticate the user against insider attacks when typing at a keyboard, achieving strong results with 20 seconds of sensor data. Lee *et al.* [23] consider the use of inertial sensors on a smartwatch (or other wearable device) as ancillary sensors to an authentication system based on a smartphone, although not in isolation. 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 [11, 38, 44]. These systems require a few minutes to calibrate yet show promise over time, although ECG sensors have been shown to be vulnerable to spoofing attacks [10].

Some works use inertial sensors on a smartwatch to authenticate the user with an explicit gesture, made solely for the purpose of authentication, such as MotionAuth [45] (full arm gestures), ThumbUp [47] (hand and finger gestures), and work by Liang *et al.* [25] (a punch gesture). The use of an explicit gesture can achieve strong results, but the user must take time to perform it and must remember it, each of which can impose an inconvenience.

Intent Recognition. With regard to intent recognition, we infer an intent-to-pay if we identify a tap gesture. This is a novel contribution and to the best of our knowledge there is no closely related work. Loosely, we know of two works that infer a security feature from wrist-based activity recognition: Mare *et al.* present both ZEBRA [28] and CSAW [29], which infer activities from wrist-worn sensor data and correlate them, respectively, with a stream of workstation inputs to ascertain the user's continued presence at that workstation (and to de-authenticate him automatically) or with motion sensor data from a smartphone to continuously authenticate the user to that phone. These systems, like the intent recognition portion of our work, achieve their inferences in a user-agnostic manner.

9. Limitations and Future Work

The main limitation of this work is the size of the dataset (unfortunately, our experimental work was stopped abruptly by national lockdowns in 2020). Having samples from 16 users enables us to demonstrate the feasibility of our approach; however, to validate our findings, more users are required and this should be a focus of future work. The collection of tap gestures in an artificial lab setting, notwithstanding our efforts to immerse the user, is also a limitation; future work should gather tap gestures from payments made in the wild to ensure that the system is robust against noise caused by real-world obstacles and distractions that affect the user.

10. Conclusion

In this paper, 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 terminal-agnostic, so does not require the use of any specific terminal type or position, and achieves F-measure scores of up to 0.87 and EERs as low as 0.08. Our intent recognition model is user-agnostic, so does not require the user to provide any training data during enrolment and is resistant to drift, and achieves F-measure scores of up to 0.89 and EERs as low as 0.04. We identified the optimum gesture parameters for in-store usage and for retrospective fraud detection. We showed that our models can be optimised for usability and incorporated as an additional factor in an existing system to provide a strict improvement to security (in terms of FAR and by adding an unsharable factor) at negligible cost to usability (in terms of FRR). We explored the factors that contributed to our results and 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 building access control, vehicle access control, or interaction with smart devices or objects.

Acknowledgement

This work was supported by Mastercard; the Engineering and Physical Sciences Research Council [grant number EP/P00881X/1]; and the PETRAS National Centre of Excellence for IoT Systems Cybersecurity [grant number EP/S035362/1]. The authors would like to thank these organisations for their support and the anonymous reviewers for their feedback.

References

[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] Apple. “Use Express Travel with Apple Pay”, https://support.apple.com/HT209495, accessed August 2020.

[3] 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.

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

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

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

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

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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*, 2018.

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

[18] C. G. Hocking, S. M. Furnell, N. L. Clarke, and P. L. Reynolds. “Co-operative User Identity Verification using an Authentication Aura”, *Computers & Security*, Vol. 39B, 2013.

[19] 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.

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

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

[22] 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.

[23] 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.

[24] 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.

[25] 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 (IST)*, Vol. 11, 2017.

[26] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan. "uWave: Accelerometer-based Personalized Gesture Recognition and Its Applications", *Pervasive and Mobile Computing*, Vol. 5, 2009.

[27] 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.

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

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

[30] K. Matsuo, F. Okumura, M. Hashimoto, S. Sakazawa, and Y. Hatori. "Arm Swing Identification Method with Template Update for Long Term Stability", *Advances in Biometrics*, 2007.

[31] R. A. Maxion and K. S. Killourhy. "Keystroke Biometrics with Number-pad Input", *Dependable Systems and Networks*, 2010.

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

[33] Mobile Transaction. "Contactless Payments Continue to Grow in the UK", https://www.mobiletransaction.org/contactless-payments-uk, accessed August 2020.

[34] B. Nassi, A. Levy, Y. Elovici, and E. Shmueli. "Handwritten Signature Verification Using Hand-worn Devices", *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, Vol. 2, 2016.

[35] Notebook Check. "Intel Core i5-6500 vs Samsung Exynos W920", https://www.notebookcheck.net/6500-vs-Exynos-W920_7839_13823.247596.0.html, accessed February 2022.

[36] F. Okumura, A. Kubota, Y. Hatori, K. Matsuo, M. Hashimoto, A. Koike. "A Study on Biometric Authentication based on Arm Sweep Action with Acceleration Sensor", *International Symposium on Intelligent Signal Processing and Communications (ISPACS)*, 2006.

[37] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas. "How Many Trees in a Random Forest?", *Machine Learning and Data Mining in Pattern Recognition (MLDM)*, 2012.

[38] 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.

[39] 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.

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

[41] 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.

[42] Technavio. "NFC Mobile Payments Set to Grow Its Market Dominance", https://blog.technavio.com/blog/mobile-payment-trends-nfc-payments-leads-growth, accessed August 2020.

[43] Thales. "PSD2 Regulation - Get Ready with Thales", https://www.thalesgroup.com/en/markets/digital-identity-and-security/banking-payment/digital-banking/psd2, accessed August 2020.

[44] 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.

[45] 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.

[46] 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.

[47] 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.