Advancing Touch-based Continuous Authentication by Automatically Extracting User Behaviours

by

Peter Aaby

School of Computing, Engineering & The Built Environment

A thesis submitted in partial fulfilment of the requirements of

Edinburgh Napier University for the award of
Doctor of Philosophy

August 2023

Abstract

Smartphones provide convenient access to online banking, social media, photos, and entertainment, all of which have become integral to our daily lives. However, this mobile nature also opens up new avenues for unauthorised access to user data. To combat this, device manufacturers often provide a screen lock mechanism. Yet, traditional lock screen authentication can be inconvenient and only offers protection at the point of user verification. Therefore, this thesis demonstrates the potential of touch-based behavioural biometrics for continuous authentication, which could significantly enhance smartphone security. The behaviours studied here are exclusively modelled from smartphone touchscreen inputs obtained from publicly available data, and the results show promising effectiveness in addressing security concerns.

The initial objective was to assess, through feature selection, the behaviours universally exhibited by users and those unique to individuals to improve the performance of the 60 different continuous authentication models being tested. Of the 30 features used over five feature selection methods, results showed that features related to pressure appeared in 81% of models, demonstrating their importance for most users while not negatively affecting performance when non-important features were removed. The following research sought to model users independent of their directional navigation and instead rely more on these essential features.

In this field, several models are typically produced for each user depending on gesture direction when testing authentication methods. However, since features describe behaviour, this thesis demonstrates that the proposed single omni-directional model can be employed while prioritising lesser complex hyperparameters. Results show that using an omni-directional model to evaluate 35 users can achieve an AUC score of 89% and 17.9% EER when authenticating using five gestures while outperforming more complex bi-directional techniques. Furthermore, the omni-directional model performs better when using the oldest feature set than more recent efforts in engineering new features.

When considering the necessary prioritisation of features unique to individuals, it becomes immediately apparent that engineering and evaluating these manually is impossible at scale. To address this, this thesis proposes a move towards automatic feature extraction through the innovative TouchEncoding method. This method transforms touch behaviour into image encodings that enable computer vision to authenticate users. The results of this approach were superior to all the related work, with AUC scores of 96.7% and EER of 8.5% across 74 users while authenticating on a single gesture. The performance further improved to 99.1% AUC and 3.6% EER when authenticating using five gestures. This underscores the effectiveness and superiority of the TouchEncoding method, paving the way for future work in this area.

Denne afhandling er dedikeret til min søster, Tine, hvis hjertevarme og 6. juli-menu har været afgørende for succesen og en konstant påmindelse om, at alt nok skal gå. Tusind tak for dit nærvær, selvom vi har været langt fra hinanden.

Acknowledgements

First and foremost, I am eternally grateful to my parents and family, who not only raised me but also shaped me through their teachings of kindness, curiosity, and courage. Their unwavering support in all my pursuits, academic or otherwise, has been a constant source of strength and motivation.

I am also profoundly thankful to Dr. Zhiyuan for his guidance and patience when listening to my promising and flawed ideas and his mentorship in shaping me into an independent researcher. I am also indebted to Dr. Valerio for the great conversations and overstimulating espressos in his office. His insights into AI have proven to be instrumental, surpassing my expectations and significantly contributing to the success of my research. Lastly, I cannot express enough gratitude to Professor Bill OBE, who has provided unwavering support throughout my PhD journey. Bill's mentorship has been invaluable from the initial stages with an industry partner to facilitating collaboration and securing completion with a new team.

Besides the supervisory team, I would like to thank everyone at the university who has shared their knowledge. In particular, I would like to thank Dr Craig Thomson, who became a great friend during our undergraduate degrees and started our PhD's together. Craig has been a rock throughout and, more importantly, made Scotland feel like home in times when the family was far away.

Lastly, I would like to thank all my friends! My flatmates shared countless hygge nights, the "sneaky peas" friend group, and everyone who has been constant in laughter, workouts, quizzes, and support. To all my friends who were ever-ready to offer a distraction, travel, or extend an invitation to their home country when a break was needed, your presence has been a source of immeasurable comfort and joy. Your friendships, though too numerous to list here, have made it all so much more pleasant and I can't wait to hang out again.

Tusind tak,
Peter.

Supervisory team and examiners

Director of Studies

Dr Zhiyuan (Thomas) Tan

Second supervisor

Dr Valerio Mario Giuffrida

Third supervisor

Professor William (Bill) J Buchanan OBE

Independent Panel Chair

Professor Amir Hussain

Internal examiner

Dr Jawad Ahmad

External examiner

Professor Steven Furnell

Author's declaration

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

Signed:

Peter Aaby

August, 2023

Table of Contents

List of Tables

List of Figures

List of Acronyms

2FA Two-Factor Authentication. 4, 8–10, 15

AB AdaBoost. 42

ANN Artificial Neural Network. 45, 61

AUC Area Under the Curve. 28, 33, 34, 80, 82, 83, 85, 90, 99, 101–104, 106, 107, 109, 123–125, 127, 131

BBA Biometric-Based Authentication. 1, 4, 11, 14, 16, 17, 68

BPNN Back Propagating Neural Network. 46, 48, 50, 51, 60, 92

CA Continuous Authentication. 1, 3, 11, 16, 17, 19, 21, 22, 26, 28, 29, 37, 38, 40, 43, 46, 47, 53, 58, 65, 68–73, 80, 87–91, 110–112, 114, 132–135, 138

CART Classification And Regression Trees. 44

CM Confusion Matrix. 32

CNN Convolutional Neural Networks. 25, 136

CV Cross-Validation. 41, 47, 48, 54, 58, 59, 79, 97, 98

DBN Deep Belief Networks. 42, 43, 60, 61

DFS Discriminative Factorized Subspaces. 54

DL Deep-Learning. 20, 24, 25, 27, 61, 62, 66, 67, 111, 112, 114, 115, 121, 132

DNN Deep Neural Network. 61

DT Decision Tree. 42, 46, 47, 50–52, 54, 56, 64, 71, 73, 92

EA Explicit Authenticating. 3, 18, 45

EER Equal Error Rate. 32–35, 38, 40–44, 47–49, 51–58, 60–64, 71–74, 83, 87, 90, 91, 112, 123, 127, 128, 137

ET Extra Tree. 42, 91, 92, 97, 98, 101, 104, 106, 108, 109, 134

FAR False Acceptance Rate. 32, 34, 35, 41, 42, 50, 51, 57, 59, 60, 74

FN False Negatives. 30–33

FP False Positives. 30–33

FRR False Rejection Rate. 32, 34, 35, 41, 42, 50, 51, 57, 59, 60, 74

FTA Failure to Acquire. 14, 15

FTC Failure to Capture. 14

FTE Failure to Enroll. 14, 15, 53, 56

GBC Gradient Boosting Classifier. 42, 97, 99, 101, 134

GBM Gradient Boosting Model. 47

GCHQ Government Communications Headquarters. 5

GMM Gaussian Mixture Models. 48, 52, 53, 72, 73

GMM-UBM Gaussian Mixture Models with Universal Background Model. 52, 53

HG Horizontal Gesture. 41, 42, 58, 60, 72, 78, 85, 89, 92

HOTP HMAC-based One-Time Password. 9

HTER Half Total Error Rate. 34, 35, 55, 59

iForest Isolation Forest. 92

KBA Knowledge-Based Authentication. 1, 4, 5, 7, 9, 11, 15, 16, 68

KNN K-Nearest Neighbours. 41, 44, 46–49, 52, 54–56, 70, 71, 73, 80, 81, 83, 84, 92, 93, 97, 98, 100, 101, 109, 134

LP Linear Perceptron. 44

LR Logistic Regression. 47, 54, 56, 60, 71, 73, 92

MFA Multi-Factor Authentication. 4, 8, 10

MI Mutual Information. 41, 58, 71–73, 81, 133

ML Machine-Learning. 20, 23, 24, 72, 112, 119, 131, 132, 136, 139

MLP Multi-Layer Perceptron. 54, 59–61, 92

MVP Median Vector Proximity. 49

NB Naive Bayes. 43, 44, 46, 47, 50, 51, 54, 56, 64, 92

NIST National Institute of Standards and Technology. 6, 8, 12, 15

NN Neural Network. 59–62, 66, 111, 115, 116, 132, 136, 137

OBA Object-Based Authentication. 1, 4, 9, 11, 15, 16

OCSVM One-Class Support Vector Machine. 73, 92

OTP One-Time Passwords. 9, 10, 16

OvR One versus Rest. 28, 29, 70, 71, 73, 77, 92, 98, 123

PSO-RBFN Particle Swarm Optimisation Radial Basis Function Network. 46, 50, 51, 59

RBFN Radial Basis Function Network. 46, 50, 51

RF Random Forest. 42–44, 46–49, 52, 55, 59–62, 64, 91, 92, 97–99, 134

RGB Red, Green, and Blue. 114, 119–121, 131

ROC Receive Operation Characteristics. ix, 34, 125, 127

SBS Sequential Backwards Selection. 82–84, 133

SFBS Sequential Floating Backwards Selection. 82–84, 133

SFFS Sequential Floating Forward Selection. 52, 53, 72, 82–84, 133

SFS Sequential Forward Selection. 47, 70, 82–84, 86, 133

SVM Support Vector Machine. 45, 52, 59, 62, 64, 93, 99, 114

SVM-RBF Support Vector Machine with Radial Basis Function kernel. 30, 38, 41, 46–48, 52–58, 70–73, 80, 81, 83, 84, 87, 91, 92, 97–101, 109, 133, 134

TA TouchAnalytics. 42, 46, 53–55, 61, 100, 133, 134, 138

TN True Negatives. 30–33

TOTP Time-based One-Time Password. 9, 16

TP True Positives. 30–33

VG Vertical Gesture. 42, 58, 60, 65, 71, 72, 78, 85, 89, 92

WVW Which Verifiers Work. 36, 53–55, 60, 61, 102, 134, 138

XGB eXtreme Gradient Boosting. 42, 59

Chapter 1

Introduction and Background

This first chapter aims to provide the necessary background information to grasp the problem statement better and outline the challenges of authenticating smartphone users. It also defines the problem and thesis statements and identifies the thesis contributions. The first section provides a scope of the problem domain in the context of user or device authentication, followed by an overview and background of the three most common user authentication factors, namely Knowledge-Based Authentication (KBA) in Section 1.2.1, Object-Based Authentication (OBA) in Section 1.2.2, and Biometric-Based Authentication (BBA) in Section 1.2.3. The benefits, use cases, challenges, and potential attack vectors are explained for each authentication factor. Lastly, the behavioural biometric and the Continuous Authentication (CA) concept is introduced in Section 1.2.5 before summarising the open challenges with CA in Section 1.3.

1.1 Problem Statement and Motivation

In London, the capital of the United Kingdom, a smartphone is stolen every six minutes; however, only about two per cent of these thefts result in the recovery of the device [18]. This leaves users vulnerable to identity theft and unauthorised access to their data, including potentially sensitive Personally Identifiable Information (PPI). Since smartphones provide access to numerous apps, such as web browsing, banking, social media, and private photos, the implications of such attacks are not limited to the loss of a physical device. Criminals can observe users before the theft through shoulder-surfing [19], although this may not be necessary since many smartphone users apply no protection to their device [20]. After stealing a smartphone, criminals can explore the stolen devices to gain access to other user accounts stored on the device. These credentials can then be sold or weaponised in credential-stuffing attacks [21], [22], where the attacker may gain access to other accounts that use the same username and passwords. Moreover, the criminal can extort victims if sensitive data is on the device, such as private photos [23]. Mitigating shoulder-surfing or credential-stuffing attacks typically involves additional authentication, such as a secondary authenticator, to reduce the risk of unauthorised access through stolen credentials. However, attackers can still bypass this defence, as seen in the recent Uber breach [24], where a user was tricked into forwarding a secondary authenticator due to too many pop-ups.

Consequently, securing smartphones is necessary to protect against device and identity theft. As shown in Figure 1.1, this problem domain can be separated into two concepts: machine-to-machine authentication protects the device's digital communication channel, and user authentication protects which human can access the devices. For example, advanced mathematical protection such as encryption can be implemented in the machine-to-machine domain, such as with Signal protocol [25]. However, it is more challenging to establish the same mathematical protection between a device and the end user since humans cannot compute maths as fast. Figure 1.1 shows the separation between the user and machine-to-machine authentication domain. For example, User A may rely on the protocol protecting the communication channel to User B's device. Still, there is no guarantee that another user cannot access the device without adequate user authentication. Thus, various lock screen authentication techniques are used to prevent unauthorised users' access to a smartphone. E.g., passwords, fingerprints, pin patterns, or, more recently, a proposed double pin pattern to enhance the security of pin patterns [26]. However, unlocking and accessing the device with two rather than a single pattern would take longer and harm the user experience. In a study by [27], users already spend up

[redacted]

Figure 1.1: User versus machine authentication [6]

to 9% of their time using Explicit Authenticating (EA) and unlocking their smartphones. Annoying and long authentication processes exacerbate the frustration of smartphone users since devices are frequently used for short intervals of about 50 seconds due to their portability and use in the wild [28]. Consequently, authentication mechanisms on a smartphone can quickly become a hassle for users, who might opt for usability over security, which may explain why about 200 million devices remain unprotected [20]. To address the drawbacks of traditional authentication factors such as pin patterns and passwords, [20] suggests a shift towards Implicit Authentication (IA). This involves using a user's behaviour as a biometric to eliminate shoulder-surfing issues and reduce users' need to create strong passwords. The benefits of using behavioural biometrics are two-fold. It makes it more difficult and time-consuming for a potential attacker to observe while removing any demands for users to consciously choose or enter their authentication information while protecting their accounts over time. In the literature, IA is also called Passive Authentication or CA. The remainder of this thesis will use the term CA since it implies the benefit of protecting users over time. More specifically, accurately capturing and modelling behaviour over time is the key to advancing and enabling CA. In 2016, Google attempted to promote behavioural biometrics for smartphone authentication through a *Trust* Application Program Interface (API), announced during the annual I/O conference [29]. This API was meant to be powered by several behavioural modalities. However, the API is yet to be released, suggesting that some challenges must be addressed before the technology can be widely adopted. While this section defines the problem and motivation, the following section provides more context and background information concerning the three common authentication factors, how behavioural differs from physiological biometrics, and the challenges with the former factors.

1.2 Background of Traditional Authentication

User authentication through a lock screen typically secures smartphone information, ensuring the person accessing a device is genuine. Measuring the authenticity of a user on a device can be achieved using one or more of three traditional authentication factors visualised in Figure 1.2, where each factor covers several authentication methods in an overall category. The three factors are commonly known as "Something you *know*", "Something you *have*", and "Something you *are*" [6]. These factors will be referred to as KBA, OBA, and BBA authentication, respectively. These three authentication factors are explained in the following subsections, briefly describing their strengths and weaknesses.

Typically, each of the traditional factors carries a weakness that another factor overcomes, and when combined, results in Multi-Factor Authentication (MFA) or Two-Factor Authentication (2FA). When combining factors, the strength of authentication is assumed to be increased since attackers must gather multiple factors, including secret knowledge, items in possession, and potential biometrics linked to a user account. An example of a standard that enforces 2FA is the Payment Card Industry Data Security Standard (PCI-DSS) [30], which defines multiple factors as strong authentication by requiring at least two different factors to access systems that store payment card data. However, as shown in the following sections, each factor and MFA/2FA combinations can still be exploited. At the same time, the user experience worsens for each factor required during the authentication process.

1.2.1 Knowledge-Based Authentication

The first authentication factor is "something you *know*". It includes methods such as passwords, Personal Identification Numbers (PIN), PIN patterns, and challenge questions designed to be secret knowledge only known by the genuine user. These

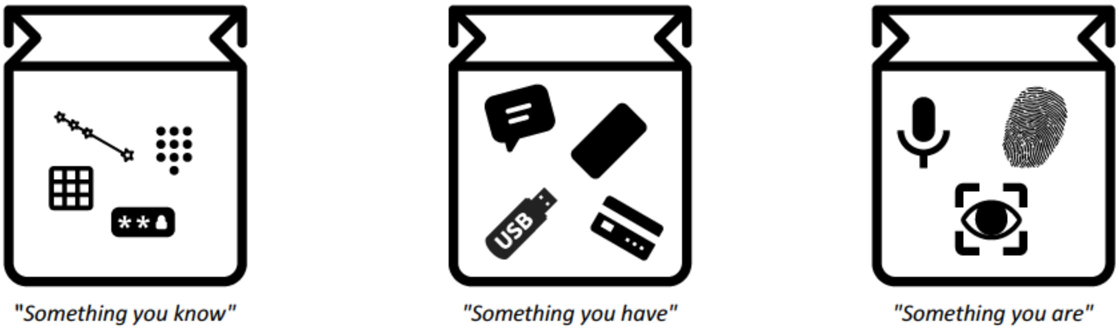


Figure 1.2: Overview of Traditional Authentication Factors

methods can be used when interacting with websites, purchasing goods with payment cards, and logging into secure online banking accounts. The benefit of KBA revolves around the balance of simplicity and complexity required when implementing the authenticator and the user selecting their answers. Naively, a system could store the username and secret in clear text, while the user would be free to choose a simple, memorable password such as 1234. However, developers should carefully consider implementing KBA using best practices and frameworks to avoid clear text storage and users selecting easily guessable passwords. A password policy is essential, but it should also consider that human memory can be volatile. Complex passwords, answers to challenging security questions, and unique numbers may be complicated to remember and could lead to losing access to one's account [31]. Similarly, clear text storage is insecure because anyone accessing the system can look up and steal the user credentials information. As a result of the challenges with KBA, Government Communications Headquarters (GCHQ) [32], the information security arm of the United Kingdom has compiled a selection of guidelines to follow. They include preventing password overload to reduce user stress, e.g., steering away from password renewals, which is seen as a vulnerability that causes users to develop, reuse and adapt passwords to fit system requirements rather than promoting solid and secure passwords [33]. The guidelines from GCHQ [32] also highlight how companies should store and manage passwords securely. A popular and recommended method is to use one-way functions to protect passwords by applying an algorithm to transform clear-text passwords into unique cryptographic strings that change entirely when a single letter in a password is updated. Further guidelines from National Institute of Standards and Technology (NIST) [34] also recommend a minimum password length of eight characters, and although short, the standard now specifies defensive measures. These include prohibiting passwords from previously known breaches, dictionary words, repetitive characters, and context-specific words. Furthermore, guidelines are defined to protect against offline attacks, such as cracking, including salting techniques and more robust one-way functions on database passwords [34].

1.2.1.1 Challenges with KBA

One of the challenges in verifying knowledge is that it can be widely shared, which makes it challenging to authenticate whether the user is genuine or a malicious person has obtained the secret knowledge. The issue is exacerbated by the prevalence of significant data breaches and large password dumps appearing frequently, and criminals can often reverse the one-way functions if implemented or chosen poorly [35]. Users often select passwords such as "123456" or "password", which lack uniqueness and are effortlessly deciphered using modern High-Performance Computing (HPC) techniques despite applying one-way functions as protective measures [36], [37]. One way to crack poorly protected passwords using HPC equipment is to apply Graphical Processing Units (GPUs) that can break any mixture of eight characters, digits, or symbols as a password within an hour through brute force techniques [37], [38]. Another way is to guess the password using dictionaries and precomputed lookup tables [39] or through more advanced guessing methods, including Markov Chains [40]. Markov chains are used to build a probability algorithm which filters and attempts the most probable passwords based on natural language. On the same topic, [41] suggests that simple 16-character passwords are more effective than complex eight-character passwords that require a mix of upper and lower-case letters, symbols, and digits or may not include specified words. This contradicts outdated NIST guidelines that considered both types of passwords equally strong. Despite the contradiction, NIST recommends several defensive measures. These include prohibiting passwords from previously known breaches, dictionary words, repetitive characters, and context-specific words. Furthermore, technical guidelines are defined to protect against offline attacks such as cracking, including salting techniques and more robust one-way functions on database passwords [34]. In this context, a one-way function seeks to cryptographically scramble passwords from plaintext into a unique value that can only be produced with the original input.

Before focusing on a more practical smartphone example and attack, credential-stuffing attacks [42] should also be considered, which involves reusing credentials across different sites. To overcome the reuse, password managers offer a way for users to create unique credentials across many websites, which is encouraged since they can store and automatically fill in their credentials and avoid password reuse. However, as seen in [43], various issues follow. These include the policy of which auto-fill is supported and allowed in unencrypted communication channels, together with potential malware able to log keystrokes or otherwise steal credentials. Moreover, password managers can become insecure when master passwords are weak, enabling attackers to compromise multiple credentials simultaneously. A complex master or lengthy password may also not be practical on smartphones with limited user input interfaces. However, pin patterns could enhance the adoption of KBA security by smartphone users who prefer usability over security [44]. Pin patterns provide a quick swipe interface on touchscreens where a user selects a secret and correct order of numbers in a single gesture. Unfortunately, shoulder surfing [45] can enable a malicious user to observe the genuine user remotely while keying in the unique pin pattern. The early work discussed in [7] also demonstrates how a photo captured of an Android device and the residual oil left by a finger sliding over the screen exposes the PIN pattern. An example from their paper is shown in Figure 1.3. Figure 1.3a presents a screen where the genuine user keyed in their pattern. A clear pattern is observed in the oily residue, along with directional changes that indicate the order of the pattern. Figure 1.3b presents another user who keyed in their pattern, wiped the screen and pocketed the phone before the photo was taken. In Figure 1.3a, an apparent smudge is observed, and results show that patterns can be recovered from this information.

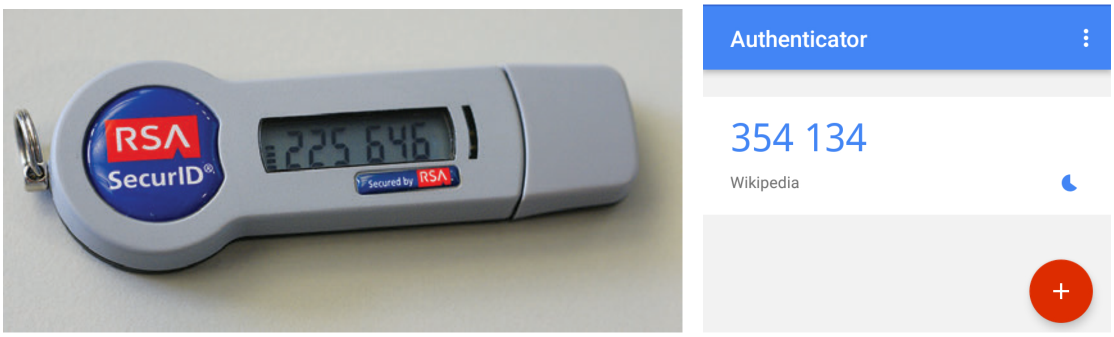
[redacted]

Figure 1.3: Attacks against pin pattern-based authentication on smartphones [7]

1.2.2 Object-Based Authentication

The second factor refers to "something you *have*" and involves binding an object to credentials, ensuring the genuine user possesses a specific physical token when authenticating. For example, a token can be a device with an associated telephone number or any other dedicated hardware or software token [6]. This factor can be used independently with properties similar to traditional physical house locks. In this case, a key is the object required to open the lock but suffers from being easily compromised by anyone obtaining the key. Because of this, combining "something you *know*" with "something you have" is often used to protect against the exploitation of individual factors, known as MFA or 2FA. NIST describes 2FA as Authenticator Assurance Level 2 (AAL2), which is required for higher security and risk levels [34]. By applying AAL2, the combined factors seek to define a high confidence level in the authenticity of users since they must prove themselves using at least two distinctively and cryptographically linked factors. In practice, SMS authentication has been a common and cost-effective way to bind knowledge with an object, e.g., the smartphone. It requires users to register a telephone number to their credentials and verify initial possession of the registered number. Any future logins can then require SMS verification to increase confidence and mitigate the KBA factor is not compromised. SMS authentication has understandable benefits, such as no additional cost in user hardware or software requirements. However, issues can arise where phone coverage is low, and the reception of SMS messages is unavailable.

Instead, other ways to prove possession of an object involve using offline One-Time Passwords (OTP) generators, either through hardware or software tokens. Figure 1.4a is an example of a widely used RSA SecureID [46] hardware token, generating passwords through the Time-based One-Time Password (TOTP) algorithm [47]. Figure 1.4b pictures the Google Authenticator project [48], a software token that implements TOTP and also supports the older HMAC-based One-Time Password (HOTP) algorithm [49]. The algorithmic difference can be spotted in Figure 1.4b, where the generated token "354 134", associated with "Wikipedia", is timing out as indicated by a circle and specified by the TOTP algorithm. On the contrary, HOTP token is not bound by time but by a counter. HOTP and TOTP tokens are linked to accounts using cryptographic pairing and keys, ensuring no other user may generate the same token and protecting it using time-boundaries or counter-cycles uniquely, TOTP and HOTP, respectively. Users can apply these tokens to establish trust beyond knowledge using the OTP algorithm, proving they can generate a token verified against their account. Thus, the possession of a specific object becomes the secure factor. However, since OBA is typically used as 2FA, the issues with credential and password overload [32] are exacerbated by demand-



(a) RSA SecureID

(b) Google Authenticator

Figure 1.4: Example of hardware (a) and software tokens (b)

ing users to spend even more time to authenticate. Regardless, the objective is to require the second factor such that attackers must simultaneously compromise two authenticators with different properties. The following section covers the challenges and attacks against the design and implementation of this aspect.

1.2.2.1 Challenges with OBA

Using 2FA can help ensure user authenticity, but selecting appropriate methods to combine factors is crucial [50]. For example, SMS verification can be problematic due to phone coverage issues and vulnerabilities using interception and malware attacks. A study by Dmitrienko et al. [51] highlights the weaknesses of SMS authentication, particularly the risk of dual infection attacks that target both the login device and associated phone receiving SMS tokens. Such attacks require targeted malware deployment, but exploiting telephone provider network protocols to intercept SMS messages is also possible [52]. Criminals can, therefore, remotely attack and exploit SMS authentication mechanisms without deploying malware or physically infecting targets. Due to these potential threats, NIST recommends against using SMS messages or Public Switched Telephone Networks (PSTN) for secure out-of-band communication [34], [50].

Hardware or software tokens can add additional protection for credentials. However, adoption amongst Dropbox users is reported at about one per cent, indicating low willingness from customers to apply the extra security factor to their accounts [53], [54]. The additional step required when accessing accounts may cause low adoption as users spend longer authenticating. SMS and OTP tokens are obtrusive and take time away by asking users to either wait for SMS codes to arrive or to prepare their tokens for use during login. Additionally, the protection provided by these authenticators may be outdated, as cybercriminals are targeting victims to install malware that intercepts and bypasses 2FA authentication, such as demonstrated in [55]. Similarly, in a recent security incident, the ride-hailing company Uber [24] also fell victim to a social engineering attack. The attacker fatigued a genuine user with requests for MFA authentication and sent WhatsApp messages pretending to be from the IT department. Unfortunately, both authentication factors failed to deny the attacker, and they were able to penetrate the internal networks and gain unauthorised access to databases. While smartphones can be secured through lock screens with passwords or pin patterns, the time and annoying components cause users to choose usability over security [27], [53]. Thus, smartphone developers such as Apple, amongst others, have adopted BBA through fingerprint scanners [56] or facial recognition [57]. These methods allow users to unlock a smartphone without entering knowledge or presenting an object but instead verifying "something they *are*". The following section provides an overview of the BBA domain and how biometrics are separated into physiological and behavioural biometrics.

1.2.3 Biometric-Based Authentication

The third authentication factor involves "something you *are*" or something that can be inferred from a person. Commonly, this is known as biometrics and is used to accurately identify users based on physiological traits or behavioural patterns [8], as seen in Figure 1.5. Compared to KBA and OBA, delegating or sharing biometrics with others is more complicated. The biometric domain is divided into two groups: physiological biometrics can be instantly captured from static body measurements, and behavioural biometrics involves observing user patterns over time. Categorising biometrics is essential because each category has unique strengths and weaknesses. Biometric authentication, although stable and fast, is susceptible to replay attacks where an attacker can present a copied biometric, observed and replicated from traces left in the public domain, e.g., fingerprint on glass, to bypass the authenticator [58]. More details on replay and this particular presentation attack are in the following Section 1.2.3.1. On the contrary, behavioural biometrics can be complex to model accurately and take time to capture. It also makes it challenging for a malicious attacker to observe and replay instantly [6], [59]. The overall weaknesses of behavioural biometrics define the gaps and reasons for this researcher, while the strength supports a password-less future powered by CA on smartphones. Something that Google hoped to roll out in 2016 but has yet to be released seven years

[redacted]

Figure 1.5: Difference between physiological and behavioural biometrics [8]

after [29]. In pursuing more precise and secure biometric identification methods, workshops such as “Applying Measurement Science in the Identity Ecosystem” [60], organised by NIST, focus on enhancing biometric performance. The evaluation of biometrics took centre stage in these discussions. However, the discussion avoids distinguishing between physiological and behavioural biometrics. Similarly, the NIST digital identity guidelines refrain from specifying best practices for each category and mainly focus on the physiological biometrics [34], [61]. The identity workshop [60] also highlights a need to consider usability and user experience to ensure users adopt security, and more critically, the biometric signals must resist replay attacks and accurately verify that the presented biometric signal belongs to a live human (liveness) [60]. These aspects are discussed in the following subsection.

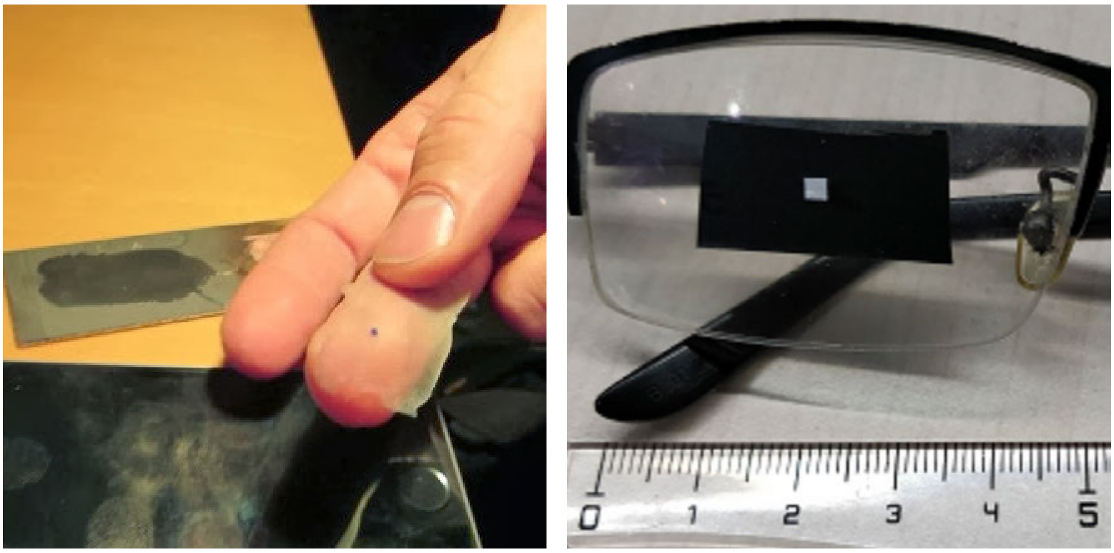
1.2.3.1 Challenges with BBA

Physiological biometrics provide users with accurate and quick smartphone authentication due to their stability. Nevertheless, unlike behavioural biometrics, they

[redacted]

Figure 1.6: Afghan girl identified by iris patterns [9]

pose two distinct issues. The main issue with using physiological biometric signals for authentication is that they are inherently stable and cannot be easily revoked or changed, leading to privacy and security compromises. For example, Figure 1.6 presents two photographs taken for National Geographic by Steve McCurry, together with iris codes [62] used to identify an Afghan girl [9]. Figure 1.6a is taken in 1984 when the girl was 12 years old, and Figure 1.6b is photographed 18 years later. Using the iris recognition codes, it is almost statistically impossible to reject that the person on Figures 1.6a and 1.6b are different humans. Thus, iris codes have the potential to identify an individual despite their privacy preferences unless they cover their eyes. The second issue is identifying whether a physiological biometric signal is from a live human, defined explicitly as “liveness detection”. Liveness detection is a complex task for fingerprint and facial recognition technologies. The reason lies in the static nature of these traits, which makes them vulnerable to copying and used to replay an otherwise genuine user’s fingerprint through this type of presentation attack. A replay or presentation attack is designed to compromise the effectiveness of liveness detection by replicating the original signal and tricking the system into believing it is provided by a genuine human, as illustrated in Figure 1.7. The figures present clear evidence of detection challenges experienced by TouchID [56] and FaceID [57], which are physiological biometric authentication techniques used for Apple smartphones. In Figure 1.7a, a high-resolution photograph of the residue from a fingerprint has been used to generate a dummy finger using household items such as pink latex mild or wood glue [58]. The researcher claims the technique can bypass several modern fingerprint scanners. Slightly different, Figure 1.7b demonstrates how a pair of glasses with tape on can evade the liveness detection of Apple's FaceID as presented at BlackHat in 2019 [63]. In this case, the researcher places the glasses on a sleeping or unconscious victim, and the glasses can trick the authenticator into thinking the person is alive and attentive. Additionally, two other issues can affect the usefulness of BBA. Firstly, Failure to Acquire (FTA), interchangeably called Failure to Capture (FTC), refers to issues with the quality of the captured biometric signal. Secondly, Failure to Enroll (FTE) relates to problems with a biometric system's ability to model users' biometric template accurately [64]. The former relates to instability in the provided biometric signal compared to the model known by the system. This can happen due to changes in finger humidity that can severely affect the ability of fingerprint scanners to record an accurate sig-



(a) TouchID exploit [58]

(b) FaceID exploit [63]

Figure 1.7: Fooling liveness detection on physiological biometrics

nal. For example, when trying to unlock a smartphone after taking a long shower or in the case of facial recognition, obscuring part of the camera lens may prevent the system from working. The prevalence of high FTA rates could also indicate issues with liveness detection or attackers trying to spoof, such as in the example shown in Figure 1.7. On the contrary, FTE happens when a user is incompatible with the biometric system. For example, some fingerprints may lack sufficiently unique characteristics to allow enrolment. This can happen due to tissue damage or other kinds of wear and tear to the fingers. Beyond these metrics, further performance metrics describe how accurately a user is detected versus impostors. However, further discussion on the different methods to measure this aspect is better suited in the literature review alongside comparing the other works, as seen in Section 2.2.

1.2.4 Summary of Traditional Authentication Challenges

With KBA, users must choose a secret they only know. However, the effectiveness of this secret in providing protection depends on its uniqueness. Unfortunately, users often choose simple and easily memorable passwords, such as “123456”, without fully comprehending the potential consequences. This preference for weaker passwords arises from the difficulty of remembering and inputting more complex ones, leading to convenience but compromising security. Consequently, attackers can exploit these weak passwords through guessing, cracking, or eavesdropping methods, thus gaining unauthorised access to user accounts. The latter threat is particularly significant for smartphones, where mobility and frequent use in various environments make users vulnerable to shoulder surfing attacks. To address these challenges, employing multiple and diverse authentication factors, such as 2FA, to enhance confidence in user authenticity is possible. However, this can add stress to the end user. In a 2FA setting, combining KBA and OBA requires users to provide knowledge as a secret and then confirm possession of an object by keying in a generated token. For example, using the Google Authenticator app, as depicted in Figure 1.4b, generates secure out-of-band 6-digit pin codes with a short expiry, aligning with NIST guidelines for secure identity management [34].

The Google Authenticator project, as depicted in Figure 1.4b, provides an open-source software solution for generating TOTP tokens. Although cryptographically protected and time-restrained, implementing cryptographic algorithms requires careful attention to security. Bardou et al. [65] demonstrate the potential risks of side-channel attacks on hardware tokens. Moreover, the Google Authenticator database can be copied from one phone to another, enabling attackers to generate valid codes on different devices [51]. Additionally, TOTP tokens rely on time-based passwords, necessitating synchronisation and attention between the user and the authentication process when keying in the token. Nonetheless, OTP technology has gained widespread adoption as a secure second factor. Prominent companies such as Dropbox recommend OTP to enhance security, especially after experiencing breaches caused by users choosing weaker passwords [66], [67].

To overcome the issues of KBA and OBA on smartphones, manufacturers have increasingly incorporated BBA into devices since it can overcome some of the usability concerns relating to remembering complex passwords and the challenges associated with keying such passwords in on a smartphone with a limited input interface. However, the stable nature of physiological biometrics authentication is also a risk for systems with poor liveness detection. For example, the fingerprint reader on Apple iPhones can be tricked into accepting a replica of a dummy finger, as seen in Figure 1.7a. Additionally, and shared between the traditional authentication factors, each factor cannot ensure user authenticity over time due to being a one-off process. Once a user has been authenticated, each process completes, and sessions either timeout or the user manually logs off.

Instead, behavioural biometrics and CA seek to overcome these challenges by providing confidence in the user over time through monitoring and analysing changes in their behaviour. However, several challenges arise in monitoring and learning individual users' behaviour, which provides the overarching motivation for this research.

1.2.5 Behavioural Biometrics and CA

Since traditional authentication factors are annoying and susceptible to attacks, research into CA using behavioural biometrics has received increasing attention from the research community, with large companies such as Google invested in furthering the development [29]. However, the technology has yet to mature due to several open challenges, opportunities for research, and common pitfalls among the related work [10], [68], [69]. Behavioural biometrics are combined with CA to passively authenticate users by collecting biometrics from input and evaluating user authenticity over time, mitigating the weaknesses of BBA. Although to better describe the challenges of CA, a high-level overview of the main objective of using touch-based behaviour as the biometric input signal is provided in Figure 1.8. In this example, a CA system secures a phone by 1. capturing touch data from user input, 2. comparing against a biometric user template, and 3. evaluating whether the genuine user is granted access to the phone 3.1 or locking the device 3.2. Conceptually, the continuous component is visualised as the arrow that loops around points 1, 2, and 3. The constant loop then repeats the steps for each interaction to secure the device over time. However, the performance relies heavily on the inter-operation between components and the design choice of each underlying point in the continuous loop. In this example, point one captures touch data from the smartphone screen. However, other researchers utilise other sensors, such as the many options in Figure 1.9 [10]. The figure categorises several modalities and defines groups associated with different actions. For example, raw touch input can be captured when a user types on the

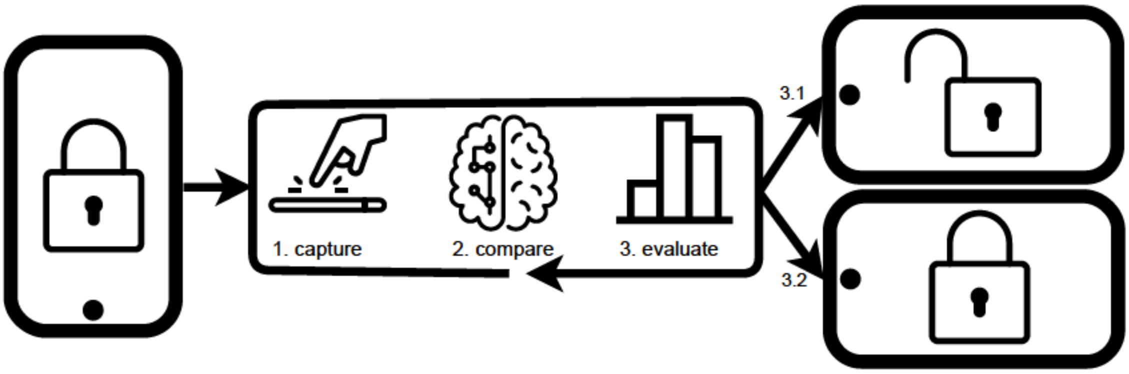


Figure 1.8: Touch-based CA concept

[redacted]

Figure 1.9: Smartphone sensors capable of collecting behaviour [10]

keyboard, enters navigational swiping patterns, or a combination of both. The fusion of behaviour from different sensors is also possible with careful consideration towards increased modelling complexity, which is the responsibility of point two. In this second step of the continuous loop, a comparison against the owners' template behaviour must be stable enough to not require any fallback to EA factors. The last step defines the evaluation criteria based on a policy decision that depends on the quality of the input signal, the performance of the behavioural modelling approach, and selecting a trade-off between erroneously misclassifying the genuine user or an impostor. Section 2.2 further define the performance metrics, evaluation criteria, and how these differ amongst the related work.

1.3 Challenges and Opportunities for Research

Despite Google's efforts [29], behavioural biometrics' commercial deployment and adoption still appear limited. While no public records explain why Google decided not to release their implementation, issues such as authentication performance or the ability to generalise on many users may play a significant role. However, considering the attention from researchers and Google's motivation, further research is necessary to advance the field and meet the demand for accurate behavioural biometric solutions, as defined in previous studies [10], [68], [69].

Close to the announcement of Google's trust API [29], Patel et al. [10] surveyed the recent progress and remaining challenges of continuous smartphone user authentication. The work defined eight directions of future research. The first challenge is domain adaptation and transfer learning, as authentication models should support adaption to changing behaviour, and the knowledge of overlapping behavioural patterns should be transferable between users. However, privacy concerns also arise in cases where third parties handle behavioural data, highlighting the need for secure processing of sensitive information. Similarly, protecting biometric templates representing user behaviour is essential for preventing unauthorised access and potential misuse. Another suggested direction involves improving feature engineering and selection to ensure CA systems accurately capture user behaviour. Nevertheless, the absence of a standardised evaluation framework makes it difficult to compare and assess different approaches. Additionally, the limited availability of comprehensive and high-quality datasets for training and evaluation further exacerbates the issue of comparing studies. Considering usability and acceptability from the end-users perspective is crucial to developing user-friendly and widely adopted systems. Lastly, the author reiterates issues with physiological biometrics being susceptible to replay attacks.

More recently, Zaidi et al. [68] also surveyed the challenges and opportunities specifically for touch-based CA. Their observations repeat some of the challenges identified by Patel et al. [10] and expand on other issues. To begin, they find a demand for more accurate detection of genuine users, and fast detection of impostors is crucial, necessitating a clear definition of "fast". For impostor detection, research should not rely solely on other classes' random attacks to develop a robust threat model [69], [70], e.g., draw samples from the negative class at random and use these for testing attacks. Furthermore, good inter-class variance but high intra-class variance requires domain adaptation and a stable classifier capable of handling the potential of concept drift. This aligns with Patel et al. [10] first point.

Additionally, Zaidi et al. [68] highlight several other interesting challenges where three are related to this thesis: (i) Without addressing and learning the diverse nature of touch interactions, models may become unstable and cause poor classification performance with specific interactions. (ii) Feature engineering and selecting quality behaviour are critical to ensure the system's effectiveness. To this end, datasets must support extraction, analysis, and proper evaluation of distinct features. The dataset should also be publicly available to enable transparent performance comparisons across research. Moreover, investigating computational time and resource consumption is necessary to ensure the feasibility and practicality of the research implementation, e.g., how many parameters to test when training and what the impact is on performance. (iii) Lastly, Zaidi et al. also observe a lack of studies utilising Deep-Learning (DL) and suggest "more research is needed to explore the potential of newer DL techniques in this context". Their survey also discusses issues relating to computational time and resources required and that DL often demands more data than Machine-Learning (ML). However, recent advancements in DL may have changed these demands, and it could be attractive to investigate further the application of DL.

To advance the field and guide further research, the following literature review in Chapter 2 will centre on current solutions, specifically emphasising essential topics to shape the thesis. Firstly, the study will closely examine the criteria researchers employ to evaluate their work, facilitating fairer comparisons based on similar performance metrics and any additional methods that can impact performance over time, such as authenticating users within a specific window of observations rather than at any given time. Secondly, the related work will be thoroughly assessed concerning the data sets utilised and their availability to the public. Additionally, the review will quantify the number of extracted behavioural features and their potential analysis, inclusion, or exclusion in performance evaluation. Lastly, considering the diverse nature of smartphone interactions, each article under examination should describe the specific scenarios considered and how these scenarios influence the chosen modelling approach. Thus, the literature review aims to illuminate existing solutions by delving into these aspects and establishing a foundation of knowledge to improve and contribute to the research domain.

1.4 Thesis statement

This thesis states that smartphone touch inputs can exclusively and accurately authenticate benign users from malicious ones by monitoring and extracting behavioural biometrics over time. Because traditional authentication factors can be exploited, users are vulnerable to impersonation attacks during and after logging into their devices. A behavioural biometric solution can passively generate user profiles and form a more robust, less susceptible platform to bind users closer to their devices while reducing the burden and attention users require while passively authenticating over time.

This thesis focuses on improving authentication performance by recognising distinctive behavioural traits specific to each user. By analysing data collected from mobile devices based on user touch behaviour, the aim is to enhance authentication performance by creating a more personalised user model. Similarly, a model should be able to verify a user independent of their directional use as long as their features are descriptive enough. Lastly, relying solely on manual feature engineering and selection restricts the ability to generalise to a broader population. Therefore, this thesis answers these issues and ultimately showcases how touchscreen behaviour can be transformed into images, enabling automatic and personalised feature extraction. This breakthrough could encourage greater technology adoption by the research community since there is no longer a need for manual feature engineering and selection, which tackles a significant challenge with the traditional approaches.

1.5 Research Objectives and Contributions

The thesis statement presented three aspects of challenges and improvements to drive better performance of touch-based CA. (i) behaviour is personal, and this context should be considered in the modelling phase when including features; (ii) the directional navigation of users' behaviour should be captured by features and not overly complex individual models; and (iii) features should be automatically extracted to accommodate personal behaviour better. Consequently, three distinct contributions build on the continued knowledge gained from the literature, and each contribution supports the claim in the thesis statement. These contributions provide compelling evidence for the advancement of touch-based CA, especially with the touch-based computer-aided design that automatically extracts personal behaviour from image encodings, as presented in Chapter 5.

The following subsections describe each contribution and the overall research objective for each. Further research questions and motivation for the individual contribution are located within their respective chapters.

1.5.1 Research Objective 1

The main objective of this contribution is to demonstrate how behaviour may be personal and, therefore, should be carefully considered during behavioural modelling. To demonstrate this hypothesis, an initial experiment is implemented to compare the baseline performance of the original features and expert knowledge presented by [1] against the proposed approach. Rather than using expert knowledge to select features for all users, several feature selection methods are implemented and evaluated for each user against the baseline performance. The results are expected to improve in cases where users expose personal behaviour through specific features. Consequently, an evaluation is given around popular features selected by the different feature selection algorithms and how the performance changes considering the selected features.

First Contribution

This contribution studies the users with available intrasession, such as within a single device usage session, and intersession gestures between sessions of usage from the public dataset released by [1]. Considering CA and behavioural biometrics seek to identify users based on unique patterns, the classic approach of using the same behavioural features for any user is unwise. Instead, this contribution states that behaviour should be considered on a personal user level in the context of a chosen classifier and device usage. Analysis and selection of unique features can be achieved at different levels, such as with filtering techniques or wrapper methods that interact with the classifier. Results demonstrate the importance of evaluating features since some users' behavioural features are more common, whereas others are unique. Unsurprisingly, the performance also appears to improve when choosing personal features for the individual user. The contribution was submitted, published, and remotely presented at the Cyber Science and Technology Congress 2020 [71]. Further details on this contribution can be found inChapter 3.

1.5.2 Research Objective 2

The performance of ML models depends on the quality of data and extracted features. Popular approaches in this field create several models depending on the user and the navigational direction of the interaction. However, this contribution investigates whether a proposed single and exclusive omnidirectional model can accurately authenticate users independent of directional movements while using a new hyperparameter selection approach. Specifically, the objective is to capture directional behaviour through features rather than segregate models. To that end, an experiment is designed to implement the traditional directional modelling approaches alongside the proposed omnidirectional model to establish a comparison between the two. Subsequent objectives involve quantifying which classifier performs better, trains faster, and produces better results when aggregating gestures or without. A final and essential objective evaluates the proposed hyperparameter tuning method that prioritises less complex parameters, which may generate simpler models.

Second Contribution

Results from the first contribution highlighted the importance of personally selecting features in the context of users, classifiers, and directional modelling approaches. With this knowledge, this contribution proposes an omnidirectional model that competes with and often outperforms traditional multi-model approaches through a new hyperparameter tuning approach favouring less complex hyperparameters. The proposed approach is rigorously investigated using five different and broader feature sets while empirically exploring the technique on a richer data set published by Serwadda et al. [2]. Results demonstrate that the proposed approach is faster and simpler to model and that despite new research suggesting better features, the original feature set offered by Frank et al. [1] in 2012 remains superior to others.

1.5.3 Research Objective 3

Since DL has been overlooked [68], this contribution takes the first steps towards applying DL to automatically extract behavioural features from touch-data, rather than relying on manual feature engineering and selection. Instead, the objective is to utilise computer vision to overcome the issues of manual feature engineering and selection. Thus, the main objective is designing the properties required for the screen canvas dimension and image size to enable the encoding of touch behaviour suitable for computer vision. The canvas should be large enough to represent the raw device screen. At the same time, the image size should be fast to process for DL models to avoid heavy resource demands to allow efficient smartphone deployment. Further, the canvas must accommodate the majority of gestures in the dataset for meaningful analysis. After designing the canvas, raw touch behaviour must be transformed and scaled to fit into colour channels ranging from 0-1 or 1-255, depending on the image format. With three available colour channels, considering which behaviour should be encoded and which formula to use when deriving the encoding is required. The final objective is to analyse different plotting styles using the encodings in the context of selected neural networks and the performance impact when tuning several hyperparameters. The contribution is compared against related work using the same dataset for single or multi-gesture performance.

Third Contribution

As shown in the first two contributions, features are critical to performance, with researchers attempting to engineer new and better feature sets that work on a personal level. The third contribution successfully addresses the challenges associated with ML by introducing a novel approach to transform and encode touch behaviour into images. Consequently, the proposed approach enables automatic feature extraction by applying DL techniques such as Convolutional Neural Networks (CNN) on the behavioural images, eliminating the need to engineer and select personal features. The methodology has been executed and rigorously assessed using a publicly accessible dataset [2], and the results demonstrate that computer vision and DL surpass all prior research. Despite the reluctance of some researchers to utilise DL, this approach has proven to be exceptionally effective.

1.6 Publications

During this PhD, the following publications have been submitted, peer-reviewed, and published in related conferences and journals. Each publication supports the contributions and is presented as individual chapters for this thesis.

- P. Aaby, M. V. Giuffrida, W. J. Buchanan, and Z. Tan, ‘Towards Continuous User Authentication Using Personalised Touch-Based Behaviour’, in 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress, Calgary, AB, Canada, Aug. 2020, pp. 41–48. doi: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00023

- P. Aaby, M. V. Giuffrida, W. J. Buchanan, and Z. Tan, ‘An omnidirectional approach to touch-based continuous authentication’, Computers & Security, vol. 128, p. 103146, May 2023, doi: 10.1016/j.cose.2023.103146

- P. Aaby, W. J. Buchanan, Z. Tan, and M. V. Giuffrida, ‘TouchEnc: a Novel Behavioural Encoding Technique to Enable Computer Vision for Continuous Smartphone User Authentication’, in 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2023. doi: 10.1109/TrustCom60117.2023.00115

1.7 Structure of the Thesis

The thesis is structured into six chapters. This first chapter introduces the topic area and offers an overview of traditional authentication factors. The background section aims to give readers a good grasp of conventional authentication factors, including "something you *know*", "*have*", and "*are*". The chapter also briefly introduces behavioural authentication as the potential solution to the challenges, emphasising the need to improve model performance to identify smartphone users better over time. Finally, the chapter sets out the problem and thesis statements, defines the contributions and sets out the research objectives. The following outlines the remaining structure of the thesis.

Chapter 2 contains a literature review focusing on accurately identifying smartphone users exclusively using touch-screen data. The chapter begins with an overview of the performance metrics used to evaluate CA systems and how they differ before surveying the related work. The literature review critically analyses the current state-of-the-art in touch-based behavioural biometric authentication and compares the related work from 2012 to date. The goal is to comprehend current approaches and their challenges and recognise any shortcomings. From there, informed decisions can be made, leading to the empirical analysis required to introduce new solutions and fill the identified gaps.

Chapter 3 hypothesises that behaviour is unique to each user; thus, behavioural features should be selected personally for each user model rather than in general. This work builds on the foundations of public data and the seminal work introduced by Frank et al. [1]. The chapter begins with a brief review of the most related literature before proposing the suggested approach containing the following points. User selection regarding the number of samples available for each, data cleaning and filtering of poorly defined gestures, and class balancing to allow fair model selection and parameter tuning. The last two sections analyse the personal features selected for each user before discussing the experimental results.

Chapter 4 builds on the contribution and findings from Chapter 3 but proposes a different and less complex modelling approach. The chapter begins with a comparison of modelling approaches and popular classifiers used in the related work. An experimental setup describes the proposed parameter selection method focusing on less complex models. Furthermore, five feature sets are summarised and compared as part of the implementation. The results section discussed the selected modelling parameters and the impact of combining gestures. Finally, modelling results are ranked and analysed for statistical significance before discussing limitations in the last section.

Chapter 5 overcomes the challenges of Chapters 3 and 4 and the issues around manually engineering features. The chapter proposes image-based encodings enabling computer vision to extract personal user behaviour automatically. A brief review of papers utilising the same data set but relying on manual feature engineering is given. The proposed approach is then described in terms of the data and user selection, cleaning, preprocessing, and how the behaviour is encoded into images. The following section covers the implementation of DL models with training recipes used for modelling. The evaluation describes the performance of two different combinations of encodings, the performance of each DL architecture, whether one plotting style is better, how merging gestures can improve performance even further, and visual verification that the computer vision attention is not random but focusing on the users' gestures before concluding in the final section.

Chapter 6 summarises the work and discusses and draws together the overall findings of this thesis. A section describes the limits of the work before finalising, with a section presenting preliminary results showing a promising direction for future work using the encodings from Chapter 5, but which has yet to be completed.

Chapter 2

Literature Review

2.1 Introduction

This chapter begins with a thorough discussion of the metrics utilised to evaluate and measure the effectiveness of behavioural biometrics in Section 2.2. Section 2.3 provides an overview of the available datasets and their characteristics. Sections 2.4 and 2.5 elaborate on the relevant research, classified into feature-based and image-based methodologies. Lastly, Section 2.6 concludes the literature review.

2.2 Performance Metrics

A brief outline of the pros and cons of standard performance and evaluation metrics is required to effectively compare the touch-based CA across the related work. While numerous metrics can be used to evaluate a model's performance, some offer better insights and are more effective in determining its broader effectiveness. For example, certain metrics require a specific decision threshold, whereas the Area Under the Curve (AUC) is independent of such and measures performance as a function of several thresholds [72]. To classify the owner of a smartphone, touch data is typically collected from a group of subjects using the same device. As such, multiple classes exist in the form of a numeric identifier for each subject. However, the related work typically applies One versus Rest (OvR) [73]. The objective of OvR is to transform a multi-class problem into a two-class classification, which reduces the complexity of classifying and authenticating genuine users. The genuine user is considered the positive class; collectively, all other users are the negative class. Naturally, this can cause class imbalance since all the other users are grouped, often producing a negative majority class. Fortunately, this can be overcome using sampling techniques as seen in [1], [2], amongst others. However, various metrics have been used in the literature and will be discussed in the following subsections using examples and metrics on the same dummy data for comparison to help better describe, plot, and inspect the difference between methods.

The performance metric examples can be applied to any datasets described in Table 2.1. Still, visually observing the positive class and changes in metrics would be more challenging due to a larger OvR class imbalance in these data sets. Thus, dummy data with less imbalance is useful when visualising and describing the class distribution in these examples while still reflecting the impact of changing the decision threshold and how it affects a given metric. The following subsections introduce the performance metrics used through the literature with examples that reflect the OvR scenario [73]. To help better describe, plot, and inspect the different performance metrics found in the literature, an example count of observations is defined for a genuine dummy user that should be identified amongst other users. Throughout this literature review, the number of observations differs, and this section can be used to reiterate why or how specific metrics are helpful or not.

2.2.1 Outline of Dummy Data

Rather than defining a specific experiment or proposing a hypothesis, a small set of dummy data can be generated to simplify describing and discussing the performance metrics in the context of CA. The dummy data must contain two classes because the objective is to identify a genuine user from others. Class one (1) identifies the genuine users as the positive class, whereas class zero (0) is the negative class identifying others. Aggregating other uses in one class naturally causes imbalance; thus, the example has a 20% skew in favour of the positive class to arbitrarily demonstrate this. In reality, the skew is more prominent, but 20% is enough to demonstrate the impact while preventing the smaller positive class from visually disappearing on distribution plots. Sci-Kit learn [74] provides functions to produce dummy data points given a total sample size while accommodating the defined skew. The dummy data has a total of 2500 data points and 28 features similar to [1], where 80% is used to train a Support Vector Machine with Radial Basis Function kernel (SVM-RBF) classifier with default Sci-Kit learn parameters [74]. The classifier can be substituted with any other classifier without changing the interpretation of the difference between the performance metrics. Thus, the example and comparison of metrics use the remaining 860 data points used in the following subsections when describing each metric using this dummy data.

2.2.2 Decision Thresholds

During user authentication, interactions are classified based on predicted probability scores from a classification algorithm. The scores range from zero, which indicates that the classifiers have low confidence in predicting the class, to one, which suggests a higher likelihood that the classifier recognised the class. Sometimes, classifiers can be imperfect, and it can be challenging to model the data accurately. As a result, there may be an overlap in predicted probability scores between the two classes, and a decision must be made about which misclassification to accept. Figure 2.1 illustrate the probability distributions from the two classes after predicting the test data created in Section 2.2.1.

In this example, Class 0 denotes other users, whereas Class 1 signifies the genuine user. The classification of each observation in these two distributions is predicated upon setting a decision threshold using the predicted probability scores. For instance, any probability score that surpasses 0.5 is deemed to be Class 1, the genuine user, whereas scores below this are classified as 0, another user. Based on the scores and decision threshold, each observation can be evaluated against the true labels to determine True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). These are annotated in Figure 2.1 where FN results in the rejection of the genuine user and FP falsely allows access to another user. TP and

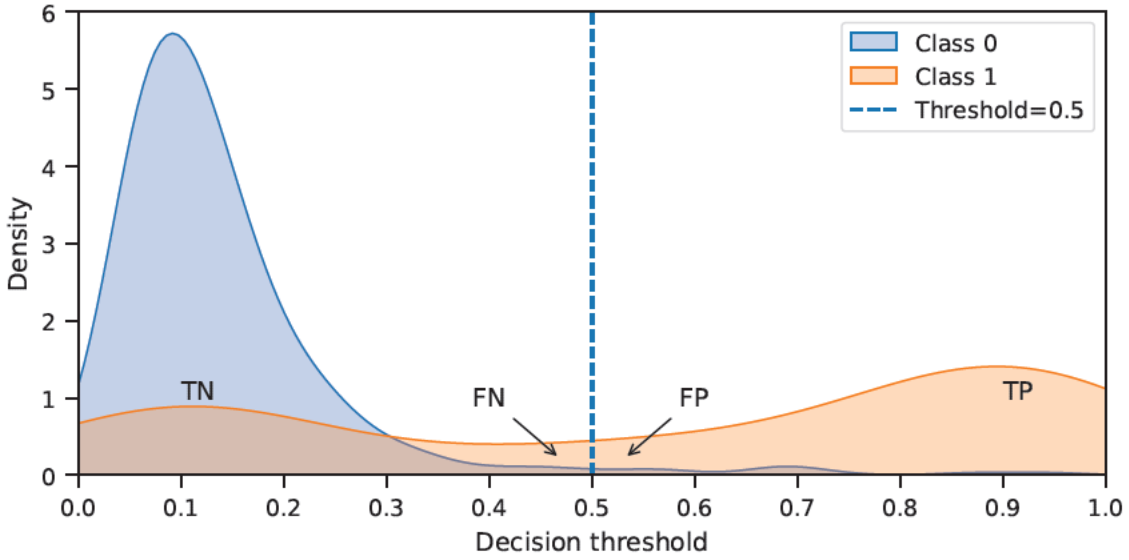


Figure 2.1: Predicted probability distributions for a dummy two-class classification problem and the impact when setting the decision threshold to 0.5

TN refers to correct classification of the respective classes.

Naturally, the decision threshold can be changed to decrease the rate of one type of error, but this often leads to an increase in error for the other class. For example, a high-security environment might dictate a policy prioritising and preventing erroneous access for unauthorised users, even if it means genuine users are mistakenly denied access more frequently. Based on the decision threshold, summary metrics such as accuracy can describe the performance at a certain threshold. The following section will describe the miss classification when measuring accuracy using a default decision threshold.

2.2.3 Accuracy

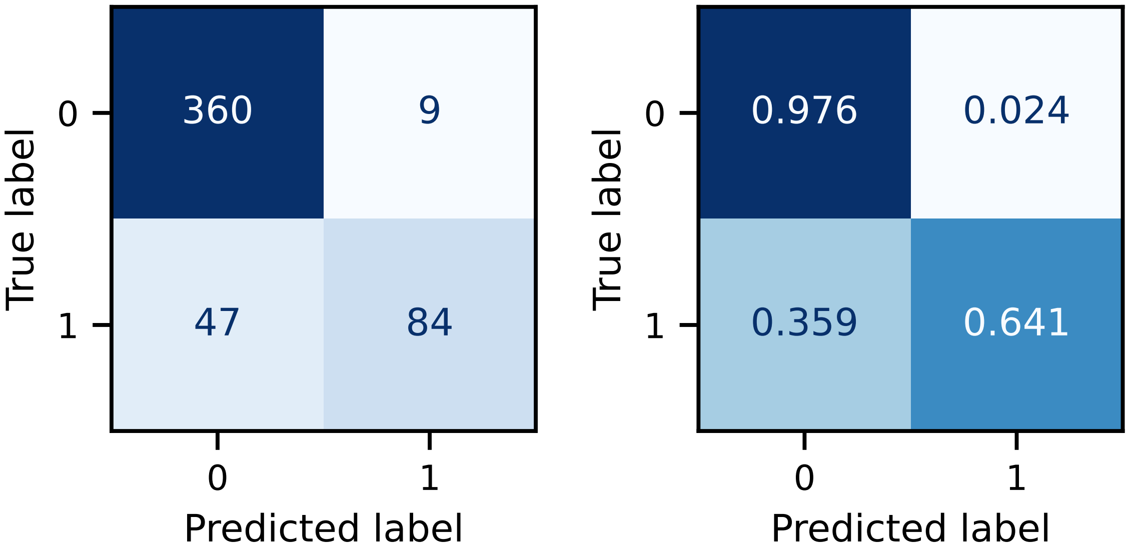
Choosing and configuring the decision threshold using the predicted probabilities is essential to produce a final classification [75]. Usually, the default decision threshold is set to 0.5, as demonstrated in Figure 2.1. Under this condition, the TP, TN, FP, and FN can be determined by comparing the predicted class to the actual label. Assuming the decision threshold is appropriate, the *accuracy* metric can be calculated using the formula in Equation (2.1).

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number Of Predictions}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.1}$$

In Figure 2.2, the impact of setting a decision threshold to 0.5 and evaluating the classifier is shown using a Confusion Matrix (CM) each to visualise the relationship between the TP, TN, FP, and FN when considering real and normalised numbers, Figures 2.2a and 2.2b, respectively. The numbers from Figure 2.2a can then be used with the formula in Equation (2.1) to produce an accuracy of 0.888. Considering Figure 2.1, the visual impact of miss classification appears where the two curves overlap. In the case of a 0.5 decision threshold, the number of miss classified samples for the negative class is FP=9 (2.4%), whereas the genuine user is mistaken FN=47 times (35.9%). Since the dataset has more data for the negative class, the classifier may have prioritised these samples, or the decision threshold is poorly configured. While the performance is great on the negative class, balancing the error rates may be more interesting since the objective is to predict the genuine user better while also accurately rejecting the negative users. Thus, a low Equal Error Rate (EER) is desired and often used to measure biometric authentication. The next section will describe how shifting the decision threshold to balance error rates can impact the accuracy score.

2.2.4 Equal Error Rate

The EER is a popular measure in biometrics since it balances the error rates for both classes. More specifically, the EER happens where the False Acceptance Rate (FAR) and False Rejection Rate (FRR) intersect. As such, it defines a balance between incorrectly accepting or rejecting users during authentication at a particular decision threshold. Figure 2.3 presents the false accept and reject error rates at different thresholds for the dummy data created in Section 2.2.1. In the previous section, the default threshold of 0.5 produced an accuracy of 0.888 but with skewed misclassification between the two classes. Analysing the error curves in Figure 2.3, a balance between the error rates of the two classes requires the decision threshold to



(a) Real numbers

(b) Normalised

Figure 2.2: Confusion Matrices when setting the decision threshold to 0.5

be set to 0.16 instead, resulting in EER=0.267 (2.67%). When the decision threshold is adjusted from 0.5 to 0.16, the values for TP, TN, FP, and FN also change, as depicted in Figure 2.4. This directly impacts the accuracy score, which is affected in the following way: the FP increased from 9 to 98, while the FN decreased from 47 to 35. As a result, the number of errors increased from 56 to 133. Consequently, this affects the accuracy score, which drops from 0.888 to 0.734, emphasising the importance of reporting accuracy scores at different decision thresholds or preferably reporting threshold-independent metrics such as AUC scores.

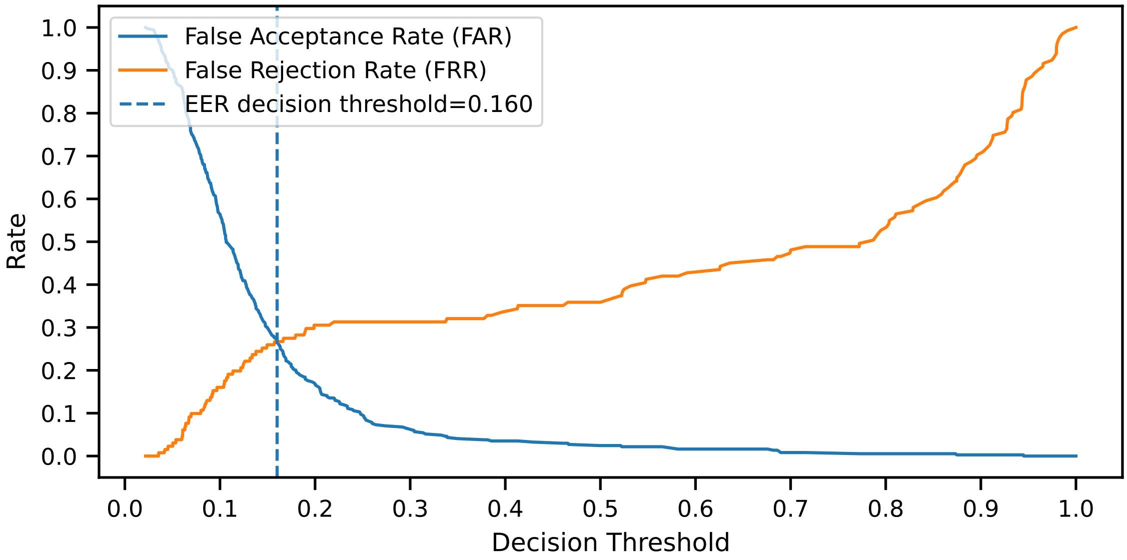
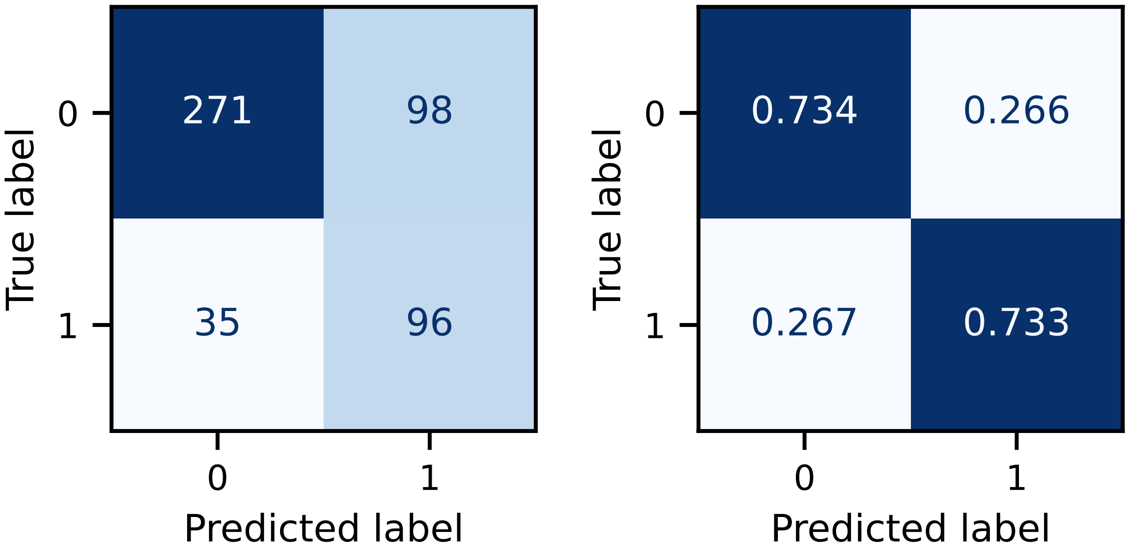


Figure 2.3: Plotting and identification of the Equal Error Rate



(a) Real numbers

(b) Normalised

Figure 2.4: Confusion Matrices when setting the threshold according to the EER

2.2.5 Area Under the Curve

Unlike accuracy and EER, the AUC score is unique in that it considers a model's performance across various thresholds instead of just one. It is derived from the ROC, which shows the relationship between true and false positives. If a classifier detects both true and false positives correctly, it will receive a perfect AUC score of 1.0, with the ROC curve reaching 1.0 on the y-axis and staying at 0.0 on the x-axis. As the true positive rate decreases or the false positive rate increases, the AUC score decreases. A model with a perfect AUC score can accurately accept or reject users regardless of the decision threshold, but if the ROC is below the 50/50 chance line, the model is no better than tossing a fair coin with half the chance of being right or wrong.

2.2.6 Half Total Error Rate

The EER and Half Total Error Rate (HTER) are commonly used metrics to evaluate biometric systems and are sometimes mistakenly used interchangeably. However, the EER represents the point at which the FAR equals the FRR, implying an equal balance between accepting impostors and rejecting genuine users. In contrast, the HTER is the average of the FAR and FRR, dividing the total error equally between

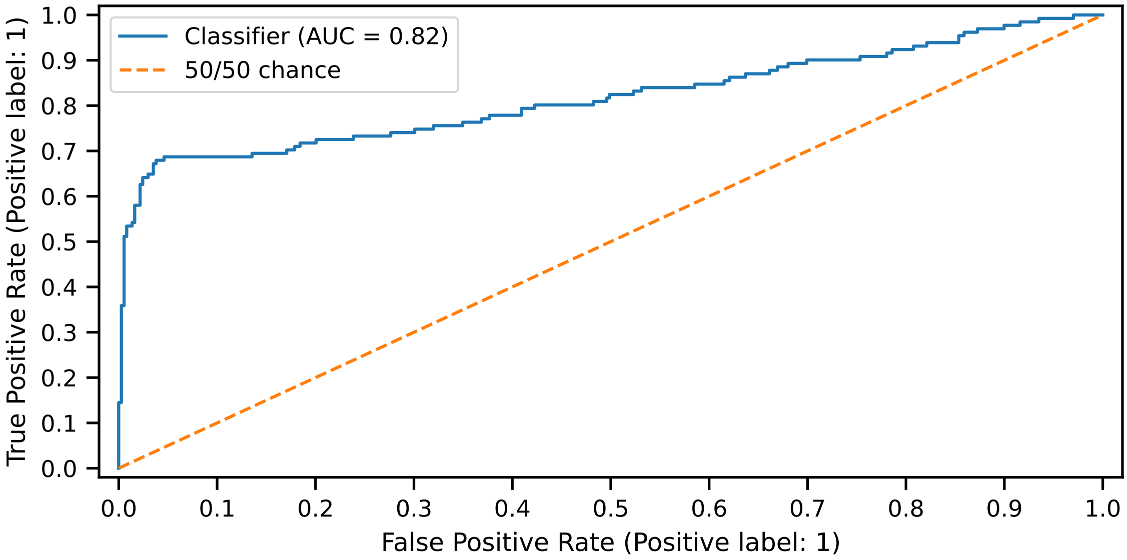


Figure 2.5: Receive Operation Characteristics curve

the two rates as shown in Equation (2.2) [76].

$$\text{HTER} = \frac{\text{FAR} + \text{FRR}}{2} \tag{2.2}$$

While both metrics provide valuable insights into the performance of a biometric system, they describe different aspects. The EER highlights the balance between security and convenience by discovering the decision threshold where both types of errors are equal. On the contrary, the HTER offers a balanced view of the overall system performance, giving equal weight to the two types of errors. Therefore, the choice between EER and HTER depends on the specific requirements and priorities of the evaluated biometric application. While the HTER can offer valuable insights into system performance, it might be less commonly used due to its equal weighting of FAR and FRR, which can obscure the specific trade-offs between security and convenience in a system.

2.3 Data-sets and Availability

As part of developing reproducible results, the availability of high-quality datasets is required in line with the findings by Patel et al. and Zaidi et al. [10], [68]. However, as described in [77], many papers collect and experiment on private datasets with incredible EER but unverifiable performance. By refraining from publishing the

Table 2.1: Summary of touch-based data sets and availability. Public (Y=Yes, R=Request, D=Declined). Operating System, OS (A=Android, I=iOS). Number of Users, NU. Orientation, O (P=Portrait, L=Landscape). Pressure, P. Area, A. Devices, Dev. Unknown setting, -

Release	Name	Public	OS	NU	Sessions	O	Usage	P	A	Dev
'12 [1]	TA	✓	A	41	1 week	P	Text/Image	✓	✓	5
'13 [2]	WVW	R	A	190	2 days	P/L	Text/Image	✓	✓	1
'15 [78]	BioIdent	✓	A	71	4 weeks	P/L	Text/Image	✓	✓	8
'15 [79]	Syed	D	A	31	2-3 weeks	P/L	Navigation	-	-	4
'16 [80]	UMDAA-II	✗	A	48	2 months	-	Background	✓	✗	1
'19 [81]	Brainrun	✓	A/I	2344	+1105	-	Playing	✗	✗	2418
'19 [82]	MobTouchDB	R	A	217	3 weeks	P/L	Draw numbers	✗	✓	94
'22 [69]	CEP	✓	I	470	31 days	P	Social/Image	✓	✓	9

data, other researchers cannot confirm the experiments' data, method, or outcome. Furthermore, the size and quality of data sets must be adequate in terms of the objective of the specific studies [10]. Additionally, size issues appear when studies propose modelling n number of users, but not all users use the same device. In such cases, comparing results fairly amongst all users or across other related work may be challenging unless their data is gathered under similar conditions. Similarly, the lack of adequate users or samples for specific users may prohibit meaningful analysis. Thus, Table 2.1 summarises and compares the available public datasets. The table lists the associated datasets and studies sorted by the year of release, from 2012 to date. Names in the related literature commonly refer to each study, and the public nature of datasets has been confirmed by attempting to access the resources freely or by direct contact with the corresponding author. For example, the Which Verifiers Work (WVW) dataset is available on request, whereas the author of the UMDAA-II did not respond to multiple access requests, and Syed denied access to the raw datasets.

As shown in Table 2.1, eight datasets have been publicly released as of date or previously. In the attempt to access these, it appears some are no longer available or only on request to some. Thus, research relies on only a few options, with further limitations depending on the experimental design or research objective. For example, research on iOS is limited to one dataset if an experiment requires pressure and area recordings. Similarly, the Android datasets are limited to three options when considering the same pressure, area constraints, and availability. The following section reviews the feature-based approaches, and a column in Table 2.2 specifically defines which dataset the method is tested on together with several other exciting attributes when comparing the related work and advancements in the field.

2.4 Feature-based Approaches

The related literature for CA is vast, especially considering all the available sensors that capture behaviour. However, this thesis exclusively focuses on touch-based biometrics and further separates the area into feature and image-based approaches. Feature-based refers to the work that relies on engineering or extracting features from raw touch data to model user behaviour. In contrast, as the subsequent section covers, only some researchers approach the topic by transforming touch input into images.

Table 2.2 chronologically list the related work focusing on touch-based CA using features. Each paper is analysed according to which dataset is used for the analysis and whether it is publicly available. Since not all papers use a public dataset with a known user count, the number of users in the work is also listed. Despite not being explicitly defined in most papers, the device orientation used for experimental results regarding portrait and landscape is given. Further to the device orientation, most researchers also model different behaviours exclusively depending on directional navigation, such as up, down, left, right, or grouped as horizontal and vertical.

Depending on the work, several classifiers may have been investigated, but only the best-performing classifier is listed in the table for brevity. Another interesting perspective is the number of features used to train the classifiers, whether multiple gestures are required, and, if so, how many. Lastly, several metrics are used throughout the literature, as covered in Section 2.2. In the table, the different metrics are described as acronyms and reported as median or mean, depending on the author's choice. The following subsections will describe each paper, the objective, and group observations to contrast patterns and discrepancies in the literature.

2.4.1 Pitfall in Touch-based CA

While Table 2.2 begins with the seminal work published by Frank et al. [1], it makes more sense to summarise the work by Georgiev et al. [69] first because it covers several pitfalls that appear throughout the literature. Knowing these pitfalls allows a better understanding of why some related work excels while others don't, despite using nearly the same experimental design or data. [69] explores six potential pitfalls in touch-based CA and suggests best practices to improve reporting and compatibility when comparing articles. To better support their claim, empirical evidence is provided using a newly collected dataset from 470 subjects using one of nine Apple iOS devices. The collected data contains up to 31 sessions for 68 users, while the average user provided 13 sessions each. Each experiment is implemented using an SVM-RBF classifier with default Sci-Kit learn [74] parameters, and the same 28 features proposed by [1]. However, as seen in Table 2.1, the availability of datasets collected using iOS is limited; thus, it may be difficult to directly compare against others due to differences in hardware and software.

Regardless, the pitfalls encompass a range of factors, and the following state the Pitfall (P) together with the observed results and recommended best practise. (P1) Small sample sizes and population of users in the datasets. Results show that EER improves when increasing the user sample count from $n = 40$ to $n = 400$, from 9.14 to 8.41%, a difference of 0.73% EER. Further, no evidence is found of users needing to acclimatisation to sessions to improve performance, nor was there a clear improvement in using excessive samples per user. However, there is no investigation of the lower bounds required for reasonable performance. For comparison amongst works, the author advises collecting and reporting results for a minimum of 40 users. (P2) Mixing phone models. When training a genuine user's behavioural model, other users are aggregated and considered a single negative class. However, the different phones used by the negative population can lead to an overestimated performance of up to 8.9% EER. Thus, the author suggests that devices should not be mixed in

Table 2.2: Overview of the related work. Notation: Public Dataset, PD. Number of Users in the study, NU. Device Orientation, DO (P=Portrait, L=Landscape, T=Tablet). Gesture 'Direction' of modelled scenario. Best classifier, CLF. Number of Features, NF. Number of Gestures combined, NG. Metric: measured in E=EER, A=AUC, C=Accuracy, F=F1, H=HTER, and $\widetilde{x}$=median, $\overline{x}$=mean. Metric reported for portrait orientation where possible

Release	Data	PD	NU	DO	Direction	CLF	NF	NG	Metric(%)
'12 [1]	[1]	✓	41	P	Hs,Vs	SVM	28	1	$\widetilde{E}13.0$
'13 [2]	[2]	✓	106	P(/L)	Hs,Vs	LR	28	10	$\overline{E}15.5$
'13 [11]	[11]	✗	20			PSO-RBFN	21	10min	$\overline{H}02.9$
'14 [83]	[1]	✓	14	P	Hs,Vs	HMM		1	$\widetilde{E}09.3$
'14 [84]	[84]	✗	10		Sliders	NB	4		$\overline{C}88.9$
'14 [85]	[85]	✗	23		U,D,L,R,ZI,ZO	1NN-DTW		8	$\overline{C}90.0$
'15 [86]	[78]	✓	71	P	Hs,Vs	SVM+CPANN	15	2	$C78.8^{\dagger}$
'15 [87]	[1]	✓	41			XGBoost	64		$\overline{C}83.6$
'15 [88]	[88]	✗	20			PSO-RBFN	14		$\overline{E}02.0$
'15 [79]	[79]	✗	31	P(/L)	Hs,Vs	RF	18	5	$\overline{E}07.1$
'15 [78]	[78]	✓	71	P	Hs+Vs	SVM	15	1	$C64.0^{\dagger}$
'15 [89]	[89]	✗	60			SVM	3	20	$H0.04$
'15 [12]	[12]	✗	80			PSO-RBFN	22		$\overline{H}04.3$
'15 [90]	[2]‡	✓	137	P(/L)	Hs,Vs	SVM-RBF	27	11	$\overline{E}05.9$
'16 [80]	[80]	✗	48			RF	24	16	$\overline{E}22.1$
'16 [91]	[79]	✗	31	T	Hs	RF	18	1	$\overline{E}11.5$
'16 [92]	[92]	✗	84			RF	13		$\overline{E}08.3$
'16 [93]	[93]	✗	71	P	U+D+L+R	RF	22-27	1	$\overline{E}35.0$
'16 [15]	[1]	✓	41			RF	162	10	$\widetilde{E}02.6$
'16 [94]	[94]	✗	30	P		DT	32		$\overline{E}20.9$
'17 [95]	[2]	✓		P(/L)	U,D,L,R	GMM-UBM	5	10	$\overline{E}15.0$
'18 [96]	[2]‡	✓	104	P(/L)	U,D,L,R	GMM+SVM	5	10	$\overline{E}07.0$
'18 [97]	[97]	✗	20		Hs,Vs	KNN	8	1	$\overline{E}05.3$
'18 [98]	[98]	✗	24			PSO-RBFN	21	10min	$\overline{H}04.0$
'19 [4]	[4]	✗	45		Hs,Vs,Os,Co	OCSVM	16	9	$\overline{C}95.9$
'19 [99]	[2]‡	✓	106	P(/L)	U,D,L,R	Temporal RF	112	4	$\overline{E}07.9$
'20 [100]	[2]‡	✓			U,D,L,R	GMM+SVM	33	10	$\overline{E}15.0$
'20 [101]	[101]	✗	10	T		ANN	35		$\overline{C}92.0$
'20 [102]	[2]‡	✓		P	U,D,L,R	DFS	33	10	$\overline{E}24.2$
'20 [103]	[2]‡	✓		P	Hs	DNN	28	1	$\overline{E}20.0$
'21 [104]	[2]‡	✓				RF	47	5	$\overline{H}06.0$
'22 [105]	[2]‡	✓		P(/L)	Hs,Vs	DES-P	28	10	$\overline{H}16.8$
'22 [69]	[69]	✓	40		U,D,L,R	SVM-RBF	28	1	$\overline{E}10.1$
'22 [106]	[1]‡	✓	14	P		SVM+RF+NN	125	1	$\overline{E}21.0^{\dagger}$

†, Metric extrapolated from chart in publication

‡, Multiple datasets but reported for [2]

training sets to avoid classifiers becoming biased towards other users and devices rather than their behaviour.

(P3) Non-contiguous training data selection. Without care, randomly sampling train and test sets from datasets can lead to data from the future becoming training data. Experiments show that random sampling can lead to an overestimation of up to 3.8% EER. Similarly, training and testing on intra-session yielded the best results due to a lesser risk of changing behaviour over time, although this goes against the design goal of CA. Refraining from testing on data collected before the training data is recommended. (P4) Analysing adversarial data in training sets. The negative class containing other users could be considered as random attacking users. However, this consideration leads to better and lower EER than excluding attackers during training. Results show 0.3-6.9% EER improvements depending on the number of adversaries in the training set. Therefore, like excluding test observations from training sets to prevent information leaks, the behaviour of adversarial users must not be part of the negative class during training when specifically analysing random adversarial attacks.

(P5) Different gesture aggregation methods and window size. Experimental results reveal that score-level fusion improves performance from 8.2 to 5.9% EER when using two gestures while further decaying and improving with more gestures. However, various aggregation windows and methods hinder meaningful comparison of works when measuring authentication performance. E.g., comparing score-level fusion with others who report feature-level fusion [2] or voting-base performance [3], [79]. Thus, it is always recommended that the performance of models be reported using individual gestures. When looking into aggregation, it's crucial to consider the method used compared to the same windows and method as the related research when comparing. (P6) A lack of public datasets and non-existent code availability. Similar to the observations in Table 2.1, they highlight that researchers may often collect data but rarely make data available for free. Similarly, they found no existing source code to implement and replicate the experiments in any of the 30 related works surveyed.

2.4.2 TouchAnalytics

Frank et al. [1] published one of the seminal works, TouchAnalytics, in 2012. Their paper can be seen as a catalyst for the research field by investigating the feasibility of authenticating smartphone users exclusively by their touch behaviour. More importantly, they opened up for further research due to their data collection. As part of the study, they collected data from 41 users and made the dataset freely available for download along with source code to extract their features. An example of raw data collected from eight unique users can be seen in Figure 2.6a, whereby visual differences in the preference of screen area can be observed. For modelling behaviour, they proposed 28 features designed to capture various user patterns during interactions where users read or play an image comparison game. The image comparison game can be seen in Figure 2.6b, requesting users to swipe horizontally between the screens but with a blank screen injected between the two images to provoke more gestures and to make the game more challenging. As part of their evaluation, they measure the Mutual Information (MI) between the features and user labels to better understand and explore the discriminate power of each feature. According to their expert knowledge, some features are removed because they do not benefit the classifier in deciding on the user label. Using the extracted features, a K-Nearest Neighbours (KNN) and SVM-RBF classifier is trained while tuning hyperparameters. The performance is then evaluated for single and multiple gestures. Results show that a single gesture achieves 13% median EER, where aggregating gestures can reduce the error rate to around 4%.

Roy et al. [83] propose using Hidden Markov Models (HMM) instead of traditional classification algorithms for training and testing. They suggest setting the decision threshold at FAR=0 and FRR=0 to test for extreme security and usability, respectively. As discussed in Section 2.2.2 and visualised in Figure 2.1, selecting the decision threshold can dramatically impact the performance. A policy to accept less security for more usability may be more valuable for users getting mistakenly locked due to misclassification. Regardless, Roy et al. trained two models using five-fold Cross-Validation (CV) to group directional gestures into Horizontal Gesture (HG)

[redacted]

Figure 2.6: Original figures from TA [1]

and Vertical Gesture (VG). Using a single gesture, they achieved intersession EER of 7.42% and 8.17% for HG and VG, respectively. If less security is preferred, setting FRR to zero will result in higher FAR. The median FAR was found to be HG=7.13% and VG=6.78%. However, a high-security setting requires low FAR. Setting FAR to 0 resulted in a median FRR of HG=19.19% and VG=18.28%. Therefore, their method may be better suited for low-security settings since FRR increases roughly twice as much as FAR in the low-security setting.

Using the TouchAnalytics (TA) dataset [1], Budulan et al. [87] model user behaviour, focusing on engineering 64 features, an increase of 36 features compared to TA, but lacking detailed specifications on why and how these are engineered and extracted. Seven classifiers are trained on the features using an unknown hyper-parameter grid, including eXtreme Gradient Boosting (XGB), AdaBoost (AB) over Decision Tree (DT), AB over Extra Tree (ET), ET, Random Forest (RF), Gradient Boosting Classifier (GBC), and Bagging over DT. The ET classifier identifies ten of the best features, with four related to pressure, such as the maximum, mean, median, and initial pressure. Interestingly, the "document ID" is also included as part of the best ten feature, although the authors of the dataset [1] argue that it should have been considered a label. Despite this discrepancy, the XGB classifier achieves the best performance, with 83.6% accuracy.

As a continued effort to improve performance using the TA dataset, a Deep

Belief Networks (DBN) and RF approach is proposed by Lee et al. [15] in their efforts to develop touch-based CA. They utilised the dataset collected by [1], which contains 41 users but differs in their approach to modelling behaviour. Specifically, they computed 37 stroke-based features and merged a group of ten gestures to generate seven session-based features. Together, this is reported to form a 162-dimension feature vector for training. However, the details of how the features are fused are unclear. For classification, 80% of the data is used for training an RF with 200 trees and DBN with two layers and dropout to classify the users. Results show that RF consistently outperforms the DBN, 2.58 to 9.93% EER, respectively. While providing excellent performance, the paper does not detail how much data each user provided in natural numbers, whether results are computed for intra or intersession, whether gestures are aggregated for authentication, or how the results compare to the related work. Furthermore, the limited depth or configuration of the DBN could affect its ability to match the performances of the RF classifier.

2.4.3 Analysis Conducted on Tablets

In the study called “LatentGestures”, conducted by Saravanan et al. [84], data was collected from 20 participants using a custom application that recorded touch interactions with user interface elements such as radio buttons, checkboxes and sliders implemented in a custom Android application, which is deployed on smartphones and tablets. The researchers suggest that training a multi-class classifier with ten users is sufficient, as this is a natural limit for a single household. However, the performance was evaluated on all 20 subjects, and it is unclear which data was used for training, validation, and testing. Moreover, their Naive Bayes (NB) classifier achieved 1.0 accuracy for five users, which may indicate overfitting and a lack of generalisation of new users. It is also challenging to understand which users use a phone or tablet and whether both are used for training. Lastly, the features used in their approach are poorly described, making it difficult to comprehend their proposed method fully. Despite the shortcomings, collecting data from tablet users had yet to be investigated, and this work took early steps towards this objective.

On the contrary, Syed et al. [79] rigorously investigate how different user postures affect the performance of touch-based CA and whether profiles can transfer between smartphones and tablets. In their work, they collect data from 31 subjects over eight sessions with an average length of nine minutes. The user models were trained for four data sessions, which proved effective during a pilot study. To evaluate the device posture, each user was tasked to use two different tablets and a phone to record their interactions: (T10) 10” tablet, (T7) 7” tablet, and (S3) 4.7” smartphone, respectively. Further, each device is used in three different postures: (P1) device on a table, (P2) device held in portrait orientation, and (P3) device held in landscape orientation. They found that larger device sizes yield better results and attribute this to more freedom for users to express their behaviour on the larger screen area. Since this research focuses on smartphones, we report this particular result in Table 2.2 but note the tablet performance achieves a mean EER of 5.16% when held in portrait and 3.8% in the landscape mode.

Following the work by Syed, Palaskar et al. [91] use a subset of 31 users using a tablet within the data gathered by [79]. They focus exclusively on horizontal gestures to answer whether classifiers degrade over time and if retraining classifiers can solve any degradation. Each user training data is divided into n blocks with equal periods. The first block is used for training, and testing is done on subsequent blocks to test degradation. Several classifiers are applied, including RF, Classification And Regression Trees (CART), NB, SVM-Linear, KNNKNN (k=10) and Linear Perceptron (LP). RF consistently performs better than other classifiers with an average EER of 11.48% without majority voting and using a ratio of 4-to-1; train-to-test sample size, respectively. For optimal results when testing 74 samples using their method, it is advisable to train with 300 gestures. Further, the error rate increases by 32% when the distance between training and testing is 600 gestures, and the author recommends retraining the classifier using the latest available data.

Rather than using the feature set collected by Syed [79], Sarhan et al. [101] collected data from ten subjects using a Samsung Galaxy Note 10.1 tablet—each of the ten subjects provided around ten touch sessions over two months of collection. From the raw data, two categories of features are extracted: gesture-level and session-level features. Gesture-level refers to the traditional modelling of each gesture, whereas session-level aggregates a sequence of gestures within a time window. For gesture-level training, 27 features are used along with 50 per cent of samples for training, 25 for validation, and the remaining 25 per cent for testing. As part of the modelling, the 27 features are compared against a reduced set of 13 of the best Principal Components. For session-level training, 35 features are used to train a model using 65 per cent, 15 for validation, and 20 per cent for testing. Thus, the training, validation, and testing sample size differs between the gesture and session-level experiments due to the sessions aggregating gestures, which causes the sample size to shrink. Regardless, two classifiers are applied for each experiment. First, an Artificial Neural Network (ANN) is built using an input layer, a single hidden layer, and an output classification layer using the sigmoid function. Second, Support Vector Machine (SVM) models using various kernel functions. Results show that principal component analysis is generally ineffective independent of the experiment. Further, the session-level classification results are 90% accuracy versus 70% gesture-level accuracy. However, the results are difficult to verify due to being private, and the number of users providing data is limited. Lastly, the ANN is configured as a relatively simple network compared to the benefit deeper networks could provide.

2.4.4 Enhancing EA with CA

A study by [85] proposes a Touch-based Identity Protection Service (TIPS) to protect users post EA. TIPS creates individual models based on application context and navigational direction, such as up, down, left, right, zoom in, and zoom out. This requires six new models for each installed application. The study collected data from 23 users who used eight different phone models and evaluated it on 123 participants. However, the author encountered pitfalls one and two, defined in [69], due to assessing a small dataset and training on multiple devices. This may lead to overestimated results. Regardless, Dynamic Time Warping (DTW) is used to compute the Euclidean distance between gestures over time, while the one-nearest neighbour is applied for classification. TIPS achieve roughly 90% accuracy when considering a sequence of eight gestures but significantly degrades with shorter authentication length. For example, the false positive rate is only $\sim 58\%$ when considering single-gesture authentication.

In 2015, Nader et al. [88] proposed combining a touch-based CA system with implicit authentication, such as a pattern lock. Their study gathered five minutes of data from 20 users and trained user models using DT, NB, JRIP, Back Propagating Neural Network (BPNN), and an Radial Basis Function Network (RBFN). Fourteen features are used to train each classifier, of which six are proposed by the author and the remaining eight overlap with [12]. They achieve excellent performance using Particle Swarm Optimisation Radial Basis Function Network (PSO-RBFN); however, several details are missing, such as what data is used for training and testing, whether the author authenticates users on single or multiple interactions, and whether they model users per directional navigation or as a whole. The limited amount of data may also suggest a lack of evaluation of unseen data, which can result in overfitting the five minutes of user interactions.

2.4.5 Approaches using Random Forests

According to the surveyed literature in Table 2.2, a RF is the most common classifier to perform well. Several articles successfully deployed the classifier using different data, user count, device types, features, and evaluation criteria [15], [79], [80], [91]–[93]. This section describes the majority of these papers, except [79], [91] who are represented in Section 2.4.3, which relate to tablet use, and [15] who is described in Section 2.4.2 relating to papers utilising the TA dataset.

Antal et al. [78] collected four datasets. Datasets one and two are for user classification, where the former has horizontal and vertical, and the latter only contains vertical gestures. Dataset three is used for gender classification, whereas dataset four contains four labels that refer to different user experience levels. They optimise the hyperparameters for KNN, RF, and SVM-RBF classifiers to perform user classification with unknown parameter grids while training on 100 samples for each user model. Results show that the RF classifiers outperform the others when combining gestures, but SVM appears superior, with 64% accuracy on a single gesture. Performance converges when combining 16 or more gestures. However, RF outperforms SVM when combining more than one gesture.

In 2016, Mahbub et al. [80] released the UMDAA-II datasets separated into two pieces, face-detection and touch behaviour. The face-detection dataset is free to download, whereas the touch data is restricted and unavailable due to missing responses by the author when requesting access [106]. While face detection is interesting, this thesis exclusively considers their work's touch-based CA component. Their paper collected data from 48 users using an Android-powered Nexus 5 smartphone over two months. Touch data is preprocessed to filter clicks from gestures using a minimum criteria of four touchpoints to qualify as a gesture. Several classifiers and hyperparameters are trained using ten-fold CV, including KNN, SVM-RBF, NB, Logistic Regression (LR), DT, RF, and Gradient Boosting Model (GBM). The search space is not documented but details the best parameter for the ensemble classifiers: "max tree depths = 10" and "the number of estimators = 200". The best classifier is RF, with 22.1% EER when merging 16 gestures using the mean score-level fusion method.

Alariki et al. [92] collected data due to "no public data being available" and enrolled 84 subjects for data collection. The users can perform gestures in any direction, but once they are comfortable with the device, they are asked to complete six gestures in any direction. Thus, it remains unknown which directions are gathered and whether users must draw six gestures for *each* direction. Nevertheless, if the user provides data for up, down, left, and right, then the dataset is estimated to have a size of 84 (users) X 4 (directions) X 6 (samples) = 2016 samples, which is relatively few compared to other publicly available datasets [1], [2]. They train and test an RF classifier without specifying if or how the training and testing data are separated. The results are evaluated using Sequential Forward Selection (SFS), but the best result is reported using all features, with an accuracy of 91.67% and EER of 8.33%.

Shen et al. [93] propose using different feature sets for each navigational direction, up, down, left, right, and a holistic approach. They collected data from 71 users across three different smartphones during three sessions. The three sessions are recorded with a minimum of one day between sessions one and two and a minimum of seven days between two and three. Several classifiers are trained using ten-fold CV, including KNN where k in the range 2-20 with $k = 11$ as the best parameter, SVM-RBF, linear, sigmoid, and polynomial kernels, BPNN with input nodes, $(2m + 1)$ first-hidden-layer nodes, three second-hidden-layer nodes, and one output node—learning rate of 0.001. RF is trained with 1000 trees. The author notes that two navigations are more common than others: sliding upwards and leftward, scrolling down on pages and moving to the next image, respectively. Further, a limited number of gestures are available in the landscape since users prefer portrait mode; thus, the paper focuses on gestures recorded in portrait only. Results show the model for left operations is superior to a holistic approach, with an EER of 25% compared to 35%.

Temporal Regression Forests is proposed by [99], focusing on the relationship between interactions. In their work, they evaluate the suggested method on two public datasets [1], [2] while comparing their Temporal RF against similar classifiers as [2] and [96], SVM-RBF and Gaussian Mixture Models (GMM)+SVM-RBF, respectively. To model temporal behaviour, their method fuses the features of consecutive gestures into a flat feature vector, in contrast to [2], which averages the feature vector of ten gestures. Thus, the final feature vector considers $n_{features} * k_{gestures}$ where $k = 4$ and $n = 28$, thus producing a 112-dimension feature vector for each interaction. Consequently, a minimum of four gestures are required for a single training sample or authentication. Forty samples for the target user are used for training, causing 2.8 times more features than samples for each target user when comparing the ratio of features to samples. The other class size is not specified. Due to the fusion of gestures, many features may positively impact the classification results at the cost of increased model complexity.

Unlike the other authors advocating for RF, Alghamdi et al. [97] provide a contrarian view. Their work collected data from 20 subjects using a smartphone that records touch input using the QWERTY and 12-digit keyboard and general scroll and drag gestures over three sessions. Six features are extracted from the QWERTY, five from the 12-digit inputs, and eight for drag and scroll features. As such, the models contain a mixture of touch-biometrics and key-stroke analytics to measure performance. They propose using Median Vector Proximity (MVP) for single-gesture authentication but suggest KNN ($k = 1$) for multi-gesture authentication. They also evaluate RF (five trees) but do not find it to be "ideal for user authentication" in contrast to several other works [15], [79], [80], [91]–[93]. The models are trained on three sessions of data and tested on another two sessions. However, it remains unclear how much data each session contains and how they differ, either by the time between them or by different tasks. Regardless, when authenticating with a single gesture, they achieve a remarkable 5.25% EER using an MVP classifier.

2.4.6 PSO-RBFN Approaches

In 2013, Meng et al. [11], [12], [98] began working on three contributions that ultimately let them develop TouchWB [98]. In the initial work, Meng et al. [11] captured data from 20 users over 120 ten-minute sessions as part of the first work. In difference to the related work, the data capturing application is embedded into the Android system by modifying the source code and deploying a customised version of the Cyanogen operating system where the pointer application records touch input. An example of the application can be seen in Figure 2.7. The benefit of this data collection approach is the ability to capture system-wide gestures rather than those contained in a specific application.

Opposed to the studies by [1], [2], several touch inputs appear to be aggregated into 21 features over ten minutes. The assumption is based on the proposed features named "number of touch movements per session", "number of single-touch events per session", and "number of multi-touch events per session", which may imply that statistics are gathered as a group of interactions. Regardless, they

[redacted]

Figure 2.7: Data collector embedded into the Cyanogen OS as used in [11], [12]

model user behaviour by applying several classifiers, including DT, NB, K-star, RBFN, BPNN.The best performance is achieved using RBFN with FAR=7.08% and FRR=8.34%. However, optimising the model using Particle Swarm Optimisation, the PSO-RBFN outperforms the former with FAR=2.5% and FRR=3.34% using the proposed signatures.

In their second work, Meng et al. [12] further surveyed the literature and extended prior work [11] by collecting data from 80 users rather than the original 20. For each user, they collect 25 sessions within three days, and each session includes 100 gestures. The main difference between this work and their prior study is the different data collected from users and a different feature set consisting of 22 variables. However, they provide no comparison or discussion on why they needed new data or changed the features. Similar to their original work, it is challenging to understand whether raw gestures are used for training or grouped into ten minutes of device usage. Regardless, they achieve an average FAR=3.82 and FRR=4.79 with the PSO-RBFN classifier. Compared to their previous study [11], the error rates slightly increased, which may be due to modelling more users.

The third article by Meng et al. [98] extends the preliminary work released in [11]. However, this work collects data from 48 users using the same Android operating system and data collection as in [11], [12]. Compared to their previous work [12], a similar number of features and training methods are taken. However, in this work, the users were divided into two groups of 24, where Group A had all their gestures recorded, but Group B only recorded during web browsing. Each user is required to complete 20 sessions of ten minutes each over three days. Twelve sessions are used for training and the remainder for testing. Each training set is then modelled using the various classifiers, including DT, NB, K-star, RBFN, BPNN, and PSO-RBFN, and parameters optimised using ten-fold cross-validation. They found that the error rate for Group A is higher than Group B and argued that performance is better when exclusively modelling web browsing due to fewer deviations in behaviour compared to freely navigating and using the device freely. However, developing a single model for each application would be costly and scale poorly when installing more apps. Regardless, the performance of Group A. is FAR=3.67% and FRR=4.13%, compared to Group B. with FAR=2.22% and FRR=2.54%.

While not the same author or applying the same classification algorithm, Kroeze et al. [94] also collected data from 30 subjects using Cyanogen, a customised version of Android. Fourteen raw features were gathered from the operating systems' pointer location over 20-minute sessions. In contrast, 32 features were derived from the gestures. Information gain was used to evaluate feature importance, ranking pressure among the most important. They train DT and K-Star classifiers for each user on the raw data and the computed gestures. Each classifier is trained on 70% of the users' data and tested on the remaining 30%. They note that testing sets should not contain the same unauthorised users used in the negative class to train the classifiers, in line with pitfall 4 [69]. Results show that the classifiers trained on the raw data outperform those using the feature set, and the best classifier is K-star for the raw data, whereas DT is better on the feature set, 14.8% EER versus

20.9%, respectively. The author suggests and attributes this to two points: (i) the features fail to describe the raw data efficiently, and (ii) there is more raw data available than features that aggregate the touchpoints from each raw stroke into a single vector. Further, they found a negative impact on performance when modelled using a global maximum of training gestures for each user, according to the user who provided the fewest gestures.

2.4.7 Statistical Modelling Approach

Many studies concentrate on classifiers that distinguish between classes based on distances, like KNN and SVM, or tree ensembles like DT and RF. Nevertheless, some researchers also explore using statistical modelling such as GMM, using the models individually or improving distance-based classification results by combining these with statistical models at the score level.

Pozo et al. [95] investigate statistical modelling using Gaussian Mixture Models with Universal Background Model (GMM-UBM) as a classifier on the public data collected by [2]. Besides using a different classifier, they also individually compute directional models for up, down, left, and right. Sequential Floating Forward Selection (SFFS) is applied to individually discover the best five features for horizontal and vertical gestures from 100 features designed for handwritten signature verification and adopted from [107], published by one of the co-authors. For each directional model, the training data is taken from the first session of data and testing is performed in the second session. Since the dataset contains data collected in portrait and landscape, a further separation is made between the two device orientations. They observe that landscape generally outperforms portrait models slightly. Still, the authors caution against the observation due to a lack of other public datasets with landscape data to further support the claim. Regardless, ten strokes are aggregated using an average over the scores to produce evaluation metrics. The best performance ranges between 15 to 22% EER depending on the classifier's configuration and the gestures' direction.

Fierrez et al. [96] is a coauthor of [95] and extends the work by including a SVM-

RBF classifier and combining the two using score-level fusion. Furthermore, they include an additional feature set [2], [107] and attempted to evaluate the approach on four datasets [1], [14], [78], [80]. In line with their previous research, they employed SFFS to identify the top five unique features for horizontal and vertical models across all users. The datasets were analysed based on the number of gestures per direction per user, revealing a preference for portrait usage over landscape orientation. Both SVM-RBF and GMM-UBM classifiers are trained individually. Combining both classifiers achieves the best performance at the score-level fusion between SVM-RBF and GMM-UBM. Results show an average EER of 6.98% for portrait and 7.6% for landscape when using the WVW dataset from [2]. The other datasets did not have enough landscape data, but portrait EER was 4.5% for the TA [1] dataset; however, it had fewer users and training samples. The remaining two datasets did not have intersession or enough data for meaningful evaluation [78], [80].

While not strictly focusing on statistical classifiers, Santopietro et al. [100] sought to quantify the quality of gestures used for training touch-based CA using a similar classifier as proposed by Fierrez et al. [96]. The paper states that models may improve when excluding gestures of poor quality. The method in this study is tested by incorporating features from two distinct sources [2], [96]. The GMM+SVM-RBF fusion technique similar to [96] is employed as a classifier to train models for each user to understand how models trained on low, medium, and high-quality gestures perform. This approach's effectiveness is evaluated on three public datasets [1], [2], [78]. The study concentrates on portrait usage since the WVW [2] dataset is the only experiment to have been collected in landscape mode. Furthermore, analysis reveals insufficient data to model "down" usage for TA [1] users. Low, medium, and high-quality gestures perform differently, averaging 23.48, 18.24, and 15.04% EER. Thus, the author argues that error rates can improve when removing low-quality gestures. On the contrary, it may also indicate that the classifiers have issues learning from the features while also causing issues with further authentication of users who may not always provide high-quality samples. It would be interesting to see how the impact of their approach concerning the balance between FTE and users who provide low-quality gestures more often.

2.4.8 Stacking Classifiers

While most work focuses on individual classifiers or the fusion of these, others also consider training multiple models and stacking those as a group of classifiers. In [102], the author proposes a feature-level fusion classifier called Discriminative Factorized Subspaces (DFS) and argues that their method performs well when the training data is limited. They test the approach on four datasets [1], [2], [78], [80], and compare against the following classifiers: Partially Shared Latent Factor (PSLF), Generalised Multi-view Analysis (GMA), Score Level Fusion for Mobile Authentication (SCF) such as the model presented in [96], Discriminative Correlation Analysis, Sparse Multi-modal Biometric Recognition (SMBR), and Multi-View Metric Learning (MVML). They combine the 28 swipe-based features from WVW [2] and five signature-based features [107] to form a 33-dimension feature vector. Models are computed using five repetitions of ten-fold CV. Due to limited data, the smallest dataset from TA [1] cannot model the up direction. Nonetheless, the author claims to perform better, but the results do not support this since TA [1] obtained a median EER$\sim$4% while this author reports a mean EER=7.44%. Similarly, they report 26.16% for intersession portrait mode on WVW [2], but the original work reported mean EER=15.5%. According to the author, their performance is strong even with limited training data. However, the evidence provided fails to validate this assertion. There may be several factors causing these differences, including distinct ways of preprocessing data or including/excluding certain users from the dataset. However, the paper lacks specific information regarding the implementation of these aspects.

Opposed to DFS [102], Zaidi et al. [105] propose a Dynamic Selection (DS) that implements a pool of classifiers and selects the best-performing classifier for specific touch gestures. The pool comprises six traditional classifiers: KNN, SVM-RBF, DT, NB, LR, and Multi-Layer Perceptron (MLP). The DS method is divided into Dynamic Classifier Selections (DCS) and Dynamic Ensemble Selections (DES).

DCS is designed to select a single classifier from a pool, whereas DES selects a subset of classifiers and combines them for classification. They normalise the data in the range of 0-1 and train using the full training set for validation to measure the competence of classifiers. Four public datasets are used for training, including [1], [2], [78], [80]. To meet the eligibility requirements for their analysis, each user must provide a minimum of 60 samples. As seen in [96], some of the public datasets, such as TA [1], contain limited data, and it may not be possible to model enough users given the eligibility requirement. The author defines $EER = \frac{FAR+FRR}{2}$ in equation three, but this appears to be the HTER as per [76] and as described in Section 2.2.6. Regardless, the score is computed using score-level fusion over ten gestures. The individual RF classifier achieves mean EER of 18.15% while taking 0.19 seconds to computer; in contrast, the best DS method is the DES-P classifier, which achieves mean EER=16.58 but takes 2.40 seconds. Thus, the performance may improve using DS but at the expense of computational resources.

2.4.9 Adversarial Concerns

Most related work focuses on detecting the genuine user, but another exciting aspect is accurately rejecting other users and measuring the impact of adversarial attacks. However, accurately measuring adversarial attacks must be done accordingly to avoid over-estimating performance, as defined in the fourth pitfall category described in [69]. One of the first works aiming to illuminate the adversarial aspects appears in [13]. They developed a LEGO Mindstorms robot, shown in Figure 2.8, which is custom-built to replay and forge touch gestures using electrically charged Play-Doh. They use the WVW dataset [2] and select all users with a minimum of 80 gestures in session one. Then, two classifiers are trained using KNN and SVM-RBF to see how well users can be authenticated before attacking them with the robot. Any users who already perform poorly are excluded from the robot attack since it would take zero effort to bypass these models. Furthermore, they focus on portrait mode since no new information can be gained from the landscape.

Without the robot and under the zero-effort attack, the performance is between

[redacted]

Figure 2.8: LEGO robot that replays and forges touch gestures proposed by [13], [14]

3.5-13% EER depending on how many users are excluded according to their FTE policy. However, results show that EER significantly increase under the robot attack, while 20-40% of users show a reduced error rate. Interestingly, the author argues that poor performance under the robotic attack is an issue and that 20-40% of users are unique. They caution that the dataset, features, and classifiers all depend on each other, and the attacks may be less effective on different variations, especially concerning other features. The higher error rates positively indicate an ongoing attack, which should prompt the locking mechanism to request further authentication information from the user or lock the device. Later, [14] extended their work [13] by adding population-based and user-specific attacks where 1, 5, and 10 gestures are stolen from the most resilient users. They also expanded with more classification algorithms and demonstrated that their robot could be implemented using essential software that school kids could program. Similar to their previous observation, results show that 20-30% of users are immune to population-based attacks, although these can be successfully compromised by observing five gestures. In terms of the applied classification algorithm, they find no significant improvements when using one classifier over another.

Lu et al. [89] collected data from 60 subjects over one month. The authors utilise 14 features and deploy five classifiers: DT, NB, KNN, LR, and SVM-RBF.

They found SVM-RBF performs well and tunes the γ and C parameters. The search space is defined as $C = \{2^{11}, 2^{13}, 2^{15}, 2^{17}\}$ and $\gamma = \{2^{-1}, 2^{1}, 2^{3}, 2^{5}\}$. Data grouping and filtering are performed using Cumulative Distribution Functions on a subset of 20 users to demonstrate the discriminate power of the selected “angle”, “distance”, and “pressure intensity” features. While their performance is impressive and close to 0% EER, it is unclear how they implement the training and testing process given their data grouping and filtering.

In their paper, they describe grouping as a method to evaluate the direction of a gesture into the four quadrants in a circle. However, the interaction between these groups and the training or inference is not documented beyond their results in Table II of their paper. In this table, two columns are labelled “Number of testing movements”, where one may refer to the number of samples grouped when training and testing, respectively. Regardless, the given number for those columns is shown in ranges, but the resulting performance is a single number. Thus, it is difficult to interpret whether the lower or higher “Number of testing movements” is used to get their final result. Nevertheless, they achieve FAR=0.03% FRR=0.05% using 1-20 sliding movements. As the final part of their experiment, five users are chosen to mimic target users by observing and attempting to emulate their behaviour. Each of the five impostors provides 1000 samples, and results show that the FAR increase slowly and reach 25% when considering 40% of training data. When considering more training data, the FAR reduces to 10

In 2016, Gong et al. [108] proposed an Adaptive Touch-based Continuous Authentication (ATCA) designed to be forgery-resistant and protect against adversarial attacks. They collected data from 25 users and created five models for each user according to varying screen settings that distort the raw user inputs. The input is distorted on the x axis for horizontal and y for vertical every 30 seconds. Each screen setting is denoted by setting s_x where $x \in \{a, b, c, d, e\}$. The screen distortion and transition between, e.g., s_a and s_b, is not noticed by the users, which indicates that users are subconsciously adapting to the changes. As such, the distortion challenges attackers with an additional layer to attack beyond simply observing touch inputs. However, the mean sample size for each user and distorted setting s_x is around 64, which is relatively low compared to other works [2], [4], [91]. Nevertheless, several SVM-RBF models are trained for each user using the features proposed by [2], and five-fold CV is applied to optimise parameters. The models are trained for horizontal and vertical directions.

To train the ATCA models, each user is modelled five times according to the different distortions. For each user, the individual distortion is used as the user's positive class, where the other distorted gestures are grouped with the other users in the negative class. As part of feature analysis, the author applies MI [109] to evaluate the importance of each feature. Analysis shows that features derived from "pressure" are the most critical variables, and the "acceleration standard deviation" is dropped due to being the worst-performing feature. Further, an additional five "area" features are dropped due to correlation analysis with the "pressure" features. The performance is measured exclusively on the validation set from CV and when applying random and targeted attacks. Random attacks occur without the attacking obtaining touch gestures from the target user but may have obtained gestures from others. Targeted attacks are more sophisticated and require obtaining gestures from the target user, similar to the user-specific attacks explored in [14]. Results show an average EER of 4% for HG and 8% for VG during random attacks, while the targeted attacks cause higher error rates, 32 and 33%, respectively. While the results are impressive, the study's sample size is relatively small, a potential issue according to pitfall 1 [69]. It's also unclear whether the authentication is evaluated using individual gestures and whether the training includes adversarial users, which goes against pitfalls 4 and 5 in [69].

Agrawal et al. [104] also focus on adversarial attacks against touch-based CA systems. They investigate two attack types: the traditional zero-effort attack, where the negative class aggregates other users as impostors. The second attack is population-based, where behaviour is sampled from the entire data population and synthesised using a generator and discriminator agent. All users are considered at any given time; thus, the results may overestimate performance according to [69] due to adversaries being part of the training set. For their population-based attack, they combine the touch data from two public datasets [2], [110]; it is unclear whether that is the case for the zero-effort attack. To maintain a balance between legitimate users and impostors, ADASYN [111] is employed for adaptive synthetic sampling. The balancing involves selecting four samples from each impostor as input to the negative class while the target users' data make up the positive class.

SVM, RF, MLP, and XGB classifiers are trained using five-fold CV for zero-effort, same population, and different population-based attacks. The paper does not detail the "same" and "different" variations of the population-based attacks, but it appears to be intra and inter-users between [13] and [110] datasets. Authentication is evaluated using score-level fusion over five gestures and reported as FAR, FRR, and HTER. Performance under zero-effort attacks is HTER=5% for RF and XGB, independent of using their generative approach or not. Under the population-based attacks, a RF classifier tested on inter-dataset performance with the generative model presents more resilience than the others, resulting in HTER=5% versus 14% without generating samples. Interestingly, the inter-dataset performance does not degrade much, with HTER=13% for the standard method and 6% for the generative approach. However, the models are still trained with adversarial users present, causing the performance to be unreliable or over-estimated, according to [69].

2.4.10 Neural Network and Deep Learning Performance

While Neural Network (NN)'s are widely explored for facial recognition [112], the prevalence of touch-based authentication taking this approach is limited. For most research listed in Table 2.2, the best-performing classifier belongs to the machine learning domain. However, a few researchers also successfully implement variations of NN in their analysis, as seen in Section 2.4.6. The PSO-RBFN appears to work well, but the depth of the network is unknown, and other researchers cannot verify the results since the data and source code are private. Instead of the PSO-RBFN, other researchers design and implement NN such as shown in Figure 2.9. However, NN architectures require careful design regarding the provided features used as the

[redacted]

Figure 2.9: Neural Network concept as described by [15]

input layer, the number of hidden layers and neurons, the activation function between layers, regularisation methods, attention modules, and the training recipe, including the chosen optimiser, loss functions, and data augmentation. Liu et al. [113] demonstrate some of these choices and their impact in the work towards updating and optimising the ResNet [114] model, which is helpful for facial recognition [112].

Serwadda et al. sought to answer which classifier works best in the seminal work releasing the WVW [2] dataset. As part of the evaluation, they trained an MLP classifier but provided no details of the number of hidden layers or neurons in each layer. The performance of their MLP (NN) is HG=16.0% VG=20.7% EER, but the LR is superior with HG=13.8% and VG=17.2%. Three years later, [93] implements a BPNN with an input layer and two hidden layers. The paper provides no further information about the network architecture besides fixed learning rate=0.001. Results are measured using FRR with a fixed FAR set to 0.1% and combining 11 gestures for authentication. The BPNN achieves FRR=46.87% compared to 37.75% for a RF classifier.

In the same year, Lee et al. [15] applied DBN with a single input layer and two hidden layers. Additionally, they experiment with 0.05, 0.2, and 0.15 dropout rates in the input layer, hidden layer one, and hidden layer two, respectively. The hidden layers use a tanh activation function. However, their DBN model is inferior to a RF classifier and takes longer to train. More specifically, the RF achieves EER=2.58% compared to 9.93% for the DBN. Fierrez et al. [96] also attempted to model users with a MLP classifier. They obtained the best performances with two hidden layers and 25 units each. The results reached 36% EER and concluded that more complex architectures are required. Ooi et al. [99] configures a four-layer MLP and argues it resembles a DL model. However, four layers are still shallow compared to popular networks such as ResNet [114] with 50 hidden layers. Further, they experimented with Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), but the preliminary results were not in favour of the NN architectures.

Like other studies, Sarhan et al. [101] implement an ANN with a single hidden layer and mention various configurable parameters but no detail or reason for the settings. A lack of detail in NN implementation details appears to be standard across the literature, with poor performance and missing discussion on why. However, Keyhaie et al. [103] are among the few authors who argue in favour of NN in their proposed Match-On-Card (MOC). The MOC method quantises data to reduce the NN model size and allow storage on a SIM card. This study utilises the identical 28 features proposed by Fieres et al. [96], emphasising horizontal gestures as they possess better distinguishing qualities. The approach is evaluated on the TA and WVW [1], [2] public datasets with a focus on the portrait mode as the dominating device orientation. The data is split into 80% training and 20% testing, and 20% of the training data is used for validation during training. Deep Neural Network (DNN) is used for classification and employs a single hidden layer with 14 neurons, followed by a ReLu activation function and a one-node output layer. They optimise the cross-entropy loss using the Adam optimiser. Similar to others, the depth of the DNN is shallow, likely due to the size restrictions of the SIM card. The EER performance for a single gesture is 20%, while it stabilises when aggregating 15 gestures, causing the performance to improve to 2.6%.

Georgiev et al. [69] design a feed-forward network with three hidden layers of sizes 30, 30, and 15, adding batch normalisation and a dropout layer with a rate of 0.3 between the layers. The optimiser is Adam, and the activation function is ReLU. However, the performance of the NN is dismissed because an SVM classifier generally performs better. Despite this observation, Georgiev et al. [106] adopt NN in a later work using a different configuration. In [106], they built a feed-forward neural network with three hidden layers of 150, 150, and 75 neurons with a ReLU activation function. The optimiser and loss function are the same as in prior work. Thus, the number of hidden layers remains the same compared to [69], but the number of neurons is more significant, but the author offers no insights as to why. They also create an ensemble classifier consisting of SVM, RF, and NN. Interestingly, the performance of the NN is 15.41%, and the ensemble is 14.73% EER. Thus, the NN performs almost as well as the ensemble.

Despite not analysing or implementing NN in their work, Stefania et al. [87] suggests NNs as a future direction, including DL methods such as recurrent learning or auto-encoders for obtaining new features. Similarly, Kroeze et al. [94] also suggest NN classification techniques in their future work. Pokhriyal et al. [102] also describe how other areas successfully apply DL but decide not to pursue the method since it often requires millions of samples. Lastly, Zaidi et al. surveyed the literature and concluded that more work should investigate deeper architectures.

2.5 Image-based CA

While several authors focus on tabular data to create feature-based models, only three articles take an image-based approach. Zhao et al. authored the first work, Graphic Touch Gesture Feature (GTGF) [16]. They later extended the work by proposing Statistical Touch Dynamics Images (STDI) [17], which relies on their (GTGF) contribution. Lastly, Ahmad et al. [115] proposed “Trace Maps” as the last image-based approach. In this section, a summary is given for each of the methods. A similar lack of image-based approaches is identified by Georgiev et al. [106].

Zhao et al. [16] collected six data sessions from 30 subjects with three days

[redacted]

Figure 2.10: GTGF images from two different users [16], [17]

between. Each subject provided 20 navigational touch traces in the categorical direction of Up (U), Down (D), Left (L), Right (R), Zoom-In (ZI), and Zoom-Out (ZO) while allowing the users to decide how they hold and touch the device freely. To model the user behaviour, each trace is converted into a Graphic Touch Gesture Feature (GTGF) image with a dimension of 100x150 pixels and used for training behavioural models. An example of the GTGF images can be seen in Figure 2.10, where Figure 2.10a presents five touch gestures from a random user in their dataset, whereas Figure 2.10b is five gestures from a different user. Differences can be spotted between Figures 2.10a and 2.10b, but intra-image differences are more complicated for the specific user.

The images are drawn on a $100x150$ pixel canvas, aggregating information from 50 gestures. Each gesture is represented using three pixels in width, thus fitting within the 150-pixel image width. Raw x-axis values from gestures are represented on the canvas y-axis from 50 and upwards, whereas the raw y-axis is drawn on the canvas y-axis from 50 and below. Colour intensity represents movement speed. To authenticate, the distance between target and query sets is measured using Manhattan (L1), Euclidean (L2), and Normalised Cross Correlation (NS). They found that L1 distance was superior, with a mean EER score of 11.28% for verification when considering all directional navigation, including up, down, left, right, right, zoom in, and zoom out. The author notes that up and down gestures perform better than the others.

The second work by Zhao et al. [17] extends and improves on the GTGF images by proposing Statistical Touch Dynamics Images (STDI). The new method applies a Statistical Feature Model (SFM) and reduces the original feature vector through a proposed and modified Principal Component Analysis (PCA). They varied and grid-searched three parameters as part of the GTGF extraction, including L_p, U_x, and U_y. L_p defines an upper bound for the tactile pressure, where U_x and U_y refer to relative upper bounds in the x and y axes. Similarly, they search parameters p and k for the STDI. k defines the number of eigenvectors to retain using PCA, whereas p limits the "variability of synthesising new instances". The average EER for the six directional models is 9.7% when $k = 0.9$ and $p = 1.0$. Thus, they improve the performance of GTGF from 11.28 to 9.7%. However, they do not publish their data, so it doesn't remain easy to replicate and verify the results.

Interactive Trace Maps (ITM) is proposed by Ahmad et al. [115]. to extract textural and shape features for training an SVM classifier. They collect data from 25 users interacting with different applications, including the Android launcher, SMS messaging, dialling numbers and navigating the phone book, interacting with Facebook through browsing and liking photos, and finally, typing in URLs in the web browser and scrolling on websites. Several models are trained for each application. To construct the ITM, texture features are extracted using an adaptive Edge Orientation Histogram (EOH) and applied over $m \times n$ sub-images depending on the device screen size.

The shape features are extracted using $w \times h$ grid applied to the entire images, and the rectangular shape of each cell in the grid is calculated. The texture and shape features are then combined to form a single feature vector that can be used to train classifiers. As such, it appears the author disregards any potential behaviour observed from the tactile pressure exuded onto the screen. The method is evaluated using NB, RF, DT, and an SVM, where the SVM slightly outperforms RF as the superior classifier. They further compared the performance between different application contexts and found scrolling, gestures, navigating the phone book, and browsing web pages to outperform app launching, typing, and clicking. Results show that ITM can obtain 88.5% accuracy when excluding poor-performing gestures, but whether gestures are evaluated alone or aggregated is unclear.

2.6 Conclusion

In conclusion, the adoption and deployment of touch-based CA face several challenges, primarily due to their poor performance. This chapter has provided an extensive overview of evaluation metrics and a rigorous review of the related work, as demonstrated in Table 2.2. The review focuses on feature-based approaches, which currently dominate the field. However, it's worth noting that two researchers have explored an image-based approach, albeit with limited performance. Several discrepancies have become apparent upon reviewing the articles and analysing Table 2.2. Notably, there is an approximately equal ratio between research that evaluates private and public data. This discrepancy raises concerns as it goes against a common pitfall that Georgiev et al. [69] emphasise the importance of research verification. Similarly, most papers conduct experiments on a small population without providing adequate cautionary statements about the potential implications of the validity of their results.

Personalised Modelling: The findings from the research reviewed in Table 2.2 also highlight the various approaches to modelling scenarios in the context of directional navigation. Some authors opt for grouping directions, such as up and down, into VG models, while others focus on modelling individual directions. While either approach may seem wise, it is crucial to recognise that the ultimate performance of these models relies on the quality of data and feature sets used. Moreover, the effectiveness of the models is greatly influenced by how well features capture user behaviour through the collected data and engineered features. Thus, a key observation from the literature is the diverse usage of different feature sets in the models without thoroughly discussing the rationale behind their engineering, extraction, or selection. Whether these features perform optimally for all users or a certain combination of features synergises better with different classification methods remains unanswered. As a result, the motivation behind Chapter 3 is to address this gap and delve into the extent to which features are personalised. The ultimate goal is to investigate whether individual feature selection can enhance personalised models.

Omni-directional Modelling: In addition to the observations related to personalised modelling, the literature review also reveals a limited focus on the complexity and cost of modelling concerning different features and classifiers. For instance, some papers train an explicit classifier for each directional action (e.g., up, down, left, right, zoom in, and zoom out) and each application [85]. This approach raises concerns about training time with hyperparameter search, feature selection, and the relatively unstable performance of different models concerning directional navigation. To address these challenges, a high-quality model should be able to generalise well to an individual user's behaviour regardless of the gesture direction. Implementing a single direction-independent model empowered by features that can accurately model users irrespective of the direction of use becomes essential. Furthermore, tuning models using different parameter selection methods may improve situations where a model overfits according to specific behaviours. As such, Chapter 4 aims to model each user with a single omnidirectional classifier and reduce parameter complexity to minimise training time while achieving accurate and reliable results.

Deep-Learning Opportunity: Although some articles have utilised neural networks, as seen in Section 2.4.10, they have predominantly employed shallow architectures, resulting in subpar performance. This limitation could be addressed by leveraging the power of DL, where deeper architectures can automatically learn hierarchical representations from the data, potentially enhancing model performance significantly. By not fully exploring deeper architectures, the current research may miss out on capturing touch-based data's intricate patterns and complexities using more up-to-date techniques, as further identified by [105] and highlighted in 1.3. The related work also overlooks optimising hyperparameters for NN, which can also impact the results. As such, an exciting opportunity lies in the potential application of DL for touch-based authentication.

Moreover, the opportunity to integrate computer vision techniques with DL in touch-based authentication is largely untapped. While some articles have utilised neural networks, they have mostly been limited to shallow depths, and feature sets have often been used as the input layer without considering all the raw behavioural inputs accordingly. The potential of computer vision in touch-based authentication remains largely unexplored. Deep learning architectures utilising computer vision have the potential to overcome the limitations observed in shallow neural networks and offer more robust and accurate touch-based authentication models. Considering these factors, Chapter 5 seeks to further research and investigate the benefits of embracing DL and computer vision techniques. By exploring novel architectures and methodologies, new approaches could unlock the full potential of touch-based authentication and pave the way for more secure, reliable, and user-friendly authentication systems through computer vision. This opportunity represents a promising frontier for future research and can lead to significant advancements in touch-based authentication.

Chapter 3

Personalised Behavioural Modelling

3.1 Introduction

Smartphones typically provide a range of KBA and physiological BBA authentication methods to secure access through lock screens. The former includes PIN codes, passwords, and drawing patterns, whereas the latter integrates specific hardware, such as biometric fingerprint scanners or facial recognition. However, such solutions are typically used for one-off authentication, where the users authenticate once before starting a new session, with secrets being keyed in or by providing irrevocable fingerprint/facial images for more user-friendly authentication. KBA methods are inherently vulnerable since secrets can be shared, lost, or stolen, whereas physiological BBA are susceptible to presentation and replay attacks [58], [63], [116]. In these contexts, users must also actively engage with the authenticator, where up to nine per cent of the time is spent unlocking devices, taking away valuable time and requiring conscious attention by the user [27].

Instead, CA aims to ease the burden on users by binding their behaviour closer to a digital profile by passively collecting sensory input and measuring signals against known behaviour. CA then compares if an incoming stream of signals is within an acceptable confidence level of an owner's behaviour. As discussed in Figure 2.1, the acceptable level can be adjusted based on a decision threshold and tuned for better usability or higher security depending on a desired security policy [83]. Thus, CA attempts to address the shortcoming of traditional authentication by removing the demand for active user input while following the user's dynamic behavioural pattern, making it more difficult to capture and replay. The popularity of smartphones and their inherent mobility also present an increased risk of theft and consequent property loss compared to computers. Smartphones may also carry more Personal Private Information (PPI) data and allow financial transactions where users have adopted mobile payment methods.

Consequently, by utilising high-quality models of observed behaviour, CA could enable a paradigm shift from traditional one-off authenticators toward continuous seamless and unobtrusive user authentication over time. However, the challenge of uniquely creating a high-quality model remains since users behave differently; therefore, the same solution can only be applied across some users. Different smartphone sensors support behavioural detection, such as accelerometer and gyroscopic data that may be combined to detect hand movement, orientation, and grasp [117], [118]. However, this chapter focuses on touch-based CA using information that can be gathered exclusively from smartphone touchscreens. Touch data includes (x, y) coordinates of finger touch-down movement and when the finger is lifted together with auxiliary information, including timestamps, device orientation, pressure, the area covered by a finger, and application IDs. Through collecting raw touch data, researchers have focused on advancing CA by engineering features, selecting appropriate classifiers, and tuning hyperparameters while training models using varying sample sizes. We extend the body of work by exploring and empirically evaluating features for individual users. We also highlight that, within CA, a behaviour is expressed through features. Thus, including or excluding specific features should improve or decrease model performance depending on how well a feature aligns with a user's unique behaviour.

3.1.1 Motivation and Challenges

CA is still in its infancy with several challenges [10]. We highlight two significant challenges motivating this chapter: (i) human behaviour is unpredictable and subject to change over time as users adapt to various environments; (ii) different users may expose individual behaviour through distinct feature sets. Therefore, feature selection should be done at a user level.

3.1.2 Research Questions and Contributions

This chapter contributes answers to the following questions:

1. How does the user-level feature selection impact behavioural models' performance in context-aware applications?
2. Which features are more commonly important to most users, and which have limited influence on performance?

This chapter thoroughly analyses user-level feature selection for CA applications. A OvR approach is introduced to create a training set for each user of interest, allowing for the analysis of feature importance in the context of unique and individual user behaviour. OvR needs to be thoroughly explored in related work. Different types of behaviour are expressed through 30 features, and since humans may behave differently, selecting the most discriminative features is essential. Selecting minimal but highly discriminative features could reduce noise in behavioural models and improve performance. This work empirically tests features using KNN and SVM-RBF classifiers while applying several feature-selection algorithms for each classifier. We evaluate our method using a subset of the “TouchAlytics” dataset [1]. The experimental results show that our approach improves the state-of-the-art by identifying SFS as the optimal feature selection technique in combination with an SVM-RBF classifier for the selected users. The rest of the chapter is structured as follows: Section 3.2 reviews the related work. Section 3.3 describes the proposed method. Section 3.4 presents the feature selection techniques and analysis, with

Section 3.5 extending through results and discussion before concluding the chapter in Section 3.6.

3.2 Related Work

TouchAlytics. The dataset presented in [1] includes touch-based behavioural data as a viable sensory input for CA. They acquired data by developing an Android application that offers a user Wikipedia articles to read or a "find five differences in a picture" game. Reading articles was designed to collect VG, while the game caused users to slide horizontally between pictures. While using the app, touch data was recorded and allowed the extraction of 32 features. The Pearson correlation and MI were used to rank such features. Using expert knowledge obtained by evaluating the two ranking methods, three features were removed, including the "average velocity", "length of trajectory", and "orientation of end-to-end line". KNN and SVM-RBF classifiers were applied, producing results to support touch data as a viable sensory input for CA with a median EER between 2-9% when combining 10-13 gestures.

Which Verifiers Work? Similar to [1], Serwadda et al. [2] collected data from 190 subjects focusing on which classifier works while separating behaviour into four templates such that horizontal and vertical behaviour is modelled individually for portrait and landscape modes. They trained models using 80 samples from a target user while drawing 80 randomly chosen gestures from imposters, i.e., the OvR approach [73]. Each model uses the same 28-dimensional feature set. During testing, ten gestures were averaged using a sliding window to allow for more stable authentication. Individual classifiers achieved a mean EER between 10.5% and 42% using LR and DT, best and worst, respectively. Interestingly, horizontal models generally outperformed the vertical ones in portrait mode, while there was little change in horizontal or vertical scores in the landscape mode. Furthermore, SVM-RBF seemed to be the most stable classifier when considering both mean EER and its variance across all models, while KNN scored second worst.

Horizontal Versus Vertical Behaviour. Fierres et al. [96] supported the evidence found by [2], whereby HG are more discriminative. Their system of classification works by training a model using T randomly chosen samples from the legitimate user and $T/10$ samples from an imposter population. Two classifiers were trained using two different feature sets. The first model applied SVM-RBF with a 28-dimensional feature set proposed by [2]. The second model implemented a GMM using another signature feature set comprising the five best features selected through SFFS from a 61-dimensional feature set [107]. They tested behavioural models by averaging ten gestures using a sliding window. They found that HG were faster than VG with EER of around 10%, independent of device orientation. Additionally, gestures performed in the portrait mode were more stable than in the landscape mode.

Dot-To-Dot CA. In 2016, a hybrid authenticator called TMGuard was introduced by Meng et al. [119], which combines Android's dot-to-dot unlock patterns with touch-based CA. Since users must draw patterns to unlock, this method must be more transparent to qualify as CA fully. Nevertheless, the research surveyed 75 participants and demonstrated that individuals might expose stable but unique behaviour when interacting with the dot-to-dot unlock pattern. TMGuard evaluates gestures separately by grouping them up, down, left, and right. Contrary to earlier work, this work defines unique behaviour only using two features: the Speed of Touch Movement (STM) and the Angle of Touch Movement (ATM). Behaviour is then evaluated using a statistic-based profile-matching approach over several gestures, which distances the work from those applying ML methods. Regardless, the work finds similarities by concluding that users may expose consistent behaviour when performing the same gestures, although this varies across users.

Users and Their Devices. Zyed et al. [3] investigated the effect of user posture and the difference in screen size across smartphones and tablets and provided insights into inter-session variation. They extracted 18 features from their raw data and discarded four features using MI similar to [1]. Their result shows that the

EER exponentially improves when increasing the training sample size from 10, 20-30 per cent with a flat performance at 40 per cent and gradually decays using further training data. After training, user authentication combines five gestures, providing a mean EER between 3.8-8.8 per cent, min and max rates, respectively. Models from tablets perform better than smartphones with smaller screen sizes, and transferring user profiles between devices appears to degrade authentication performance.

One-Class Classifier Approach. In [4], the authors present an evaluation of 45 participants using WeChat over two weeks. This work differs by approaching CA using One-Class Support Vector Machine (OCSVM) classification and categorising behaviour into four significant groups: vertical, horizontal, oblique, and clicks. Up to 16 features were extracted from each category and selected using Fisher scores [120]. Models were also trained with varying sample sizes and hyperparameters, with the best performance found by combining nine gestures and using 80 samples for training. Results are presented using F-scores with oblique gestures outperforming others while clicks are inferior.

Summary. CA has dramatically improved due to the engineering of behavioural features tested against several classification approaches. KNN and SVM-RBF are commonly used and provide a good foundation for comparability against the related literature, using EER as a performance metric. At the same time, other classifiers may also prove suitable, such as GMM, LR, DT, and Neural Networks [2]. In this chapter, we limit our investigation to KNN and SVM-RBF classifiers as the focus remains on identifying the distinctive features in the context of individual users. While feature selection was mentioned in the related works, the application should be more rigorously explored, especially in modelling individual users. In work applying feature selection, statistical ranking techniques such as MI have been used to estimate significant features before manual removal using expert knowledge; thus, the correlation between features and applied classifiers still needs to be discovered. Furthermore, applying feature selection combined with OvR distances this work by uncovering features that may be important to most and only to some, potentially improving the EER score.

For most related works, EER is reported to describe model performance, which defines the decision threshold where FAR and FRR are equal. Our results are presented using the average EER score for all feature selection techniques for comparability. However, several related works [1], [2], [4], [96] consider multiple gestures for authentication, which prohibits exact performance comparison between the work of others. Our work will be pragmatic by reporting EER and authenticating users using singular gestures. Consequently, all EER scores may be improved by considering multiple gestures, but it is currently beyond the scope of identifying the most significant features for individual users.

3.3 Proposed Methods

In this section, we present the methods used to select users of interest, clean the chosen data and ensure class balancing for model fairness, and the methods used for model selection and hyper-parameter tuning.

3.3.1 Data Set and Users of Interest

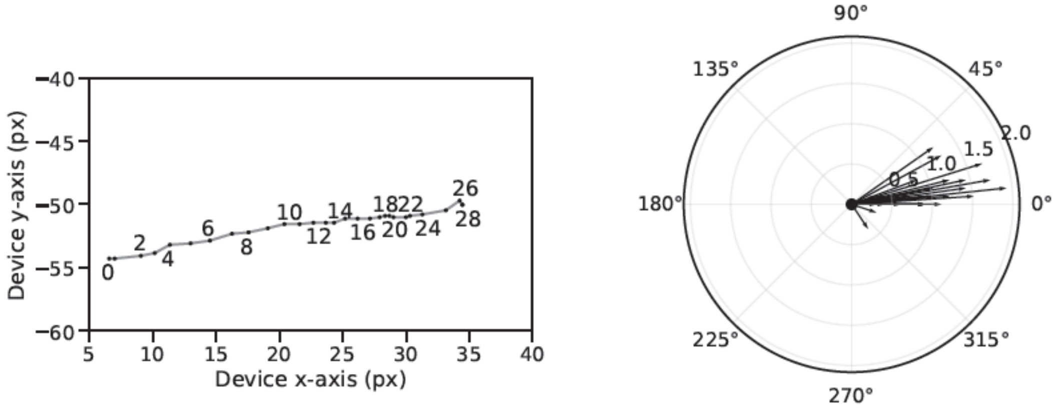
The data used for this research is extracted from a public dataset collected by Frank et al. [1], containing touch inputs from 41 subjects interacting with seven documents over two weeks. However, not all users participated in the entire experiment. Thus, we only selected users who had provided data for the whole experiment duration (2 weeks) because of the interest towards assessing model stability over time. todoC3.8Consequently, the dataset is reduced to 14 users and separated into intrasession (week one) and intersession (week two). Amongst the 14 users, two more were removed, namely, user IDs 5 and 35, as they exhibited inconsistent behaviour. All users performed two general tasks: reading Wikipedia articles and playing an image comparison game. The activities are referred to as document IDs. Documents 1, 2, 3, and 6 are Wikipedia articles, whereas 4, 5, and 7 are Gaming (comparing pictures).

3.3.2 Data Cleaning and Filtering

Filtering Clicks and Long or Idle Gestures. Figure 3.1 presents the raw data collected from a single gesture performed by a user playing the picture comparison game described in [1]. Any device interaction produces several touch points (x, y) for a click or sliding gesture. Figure 3.1a visualises the 29 touch points in the example gesture as they are drawn on the device screen. Every second touch point is annotated, and a higher density can be observed at the end of this gesture. Like others [1], [2], this work focuses on sliding gestures, distinguished by having more than five touch points. Thus, any interaction with more than five touchpoints is typically deemed a sliding gesture, whereas any interaction with lass is discarded as clicks.

Figure 3.2 visualises how many interactions exceed n points in the gesture using the log scale on the y-axis to allow variations in minority interactions to be visible. For example, 10% of the interactions have more than 100 points, and only 0.1% have roughly 550 or more points. Given only a few gestures have more than 550 points, an upper limit is applied to filter excessive outliers. We also remove gestures with inter-gesture time exceeding 1000ms. These filters remove brief and lengthy gestures such as clicks, sticky fingers, and gestures far from each other.

Missing Values. Features with no value may arise in gestures with few points, such as when calculating “20% pairwise velocity” or “median acceleration over the



(a) Raw gesture with 29 points

(b) Directional pairwise vectors

Figure 3.1: Example gesture from 1 vertical touch-screen interaction

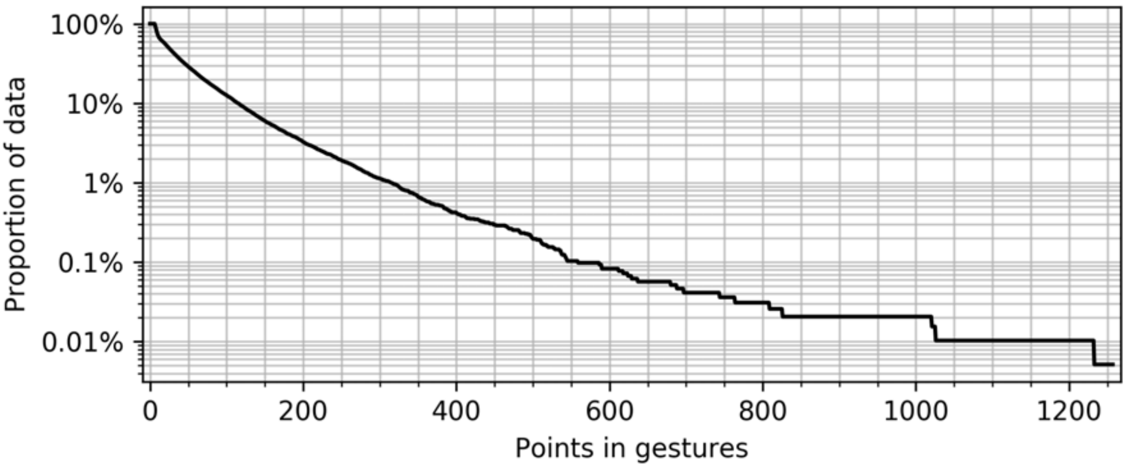


Figure 3.2: Distribution of points within gestures across all users

first five points" over gestures with less than 10 points. Missing values also occur for the last gesture performed by users, as intra-gesture times are unavailable due to being the last interaction. Gestures with incomplete values are discarded as they provide no value and rarely occur.

Finger Orientation. In contrast to the original feature set proposed by [1], we remove the "change of finger orientation" feature, as the variable remains the same across all samples and therefore provides no distinctive behavioural information. However, all other features are kept for the feature selection technique to analyse, which is contrary to the original work by Frank et al., who removed "average velocity", "length of trajectory", and "orientation of end-to-end line". Section 3.5 further highlights why these features should be included since they may present important biometric properties for some users.

Stroke Direction. Similar to [1], each gesture is categorised as up, down, left, or right by evaluating directional data. An example is shown in Figure 3.1b, highlighting the spread over pairwise vectors from a horizontal gesture. Each pairwise vector reveals minutiae behavioural detail within a gesture. As such, extracting the right features based on raw data and selecting the most discriminate features to identify an individual user is essential. Overall, we evaluate 30 features, as shown in Table 3.1, of which a subset is selected for each user individually. Further details described in Section 3.4.

3.3.3 Class Balancing and Model Fairness

Since the classification task remains to tell the device owner apart from a non-owner, the multi-class challenge can be transformed into a two-class classification consisting of n subsets with binary class labels. Binarising multi-class with this approach is also known as OvR, signifying a single user as the positive class while grouping the remaining users into another negative *rest* class. However, transforming a multi-class problem into OvR causes class imbalance, as the negative samples are more than the positive ones, which may cause classification bias. We overcome class imbalance by applying OvR and down-sampling the majority class, as shown in Figure 3.3. However, each user contributes a different number of samples, and balancing should be fair amongst models to ensure the approach is stable and comparable between users. As such, the user contributing the lowest maximum gestures will define an upper limit of allowed gestures in the models per class. Thus, for each training set, the positive class is limited to include only the 30 first gestures from a target class and roughly three samples from each remaining user in the negative class. The remaining gestures are discarded, allowing model fairness and comparability between users despite some contributing more gestures than others. Furthermore, the feature selection technique is quicker to evaluate when applying smaller sample

Table 3.1: Features included in the feature selection step

#	Description	#	Description
1	Inter-gesture time	16	80 perc. Pairwise acceleration
2	Gesture duration	17	Median velocity at last 3pts
3	Start X	18	Largest dev. end-to-end line
4	Start Y	19	20 perc. dev. end-to-end line
5	Stop X	20	50 perc. dev. end-to-end line
6	Stop Y	21	80 perc. dev. end-to-end line
7	Direct end-to-end distance	22	Average direction
8	Mean resultant length	23	Length of gesture
9	Up/down/left/right flag	24	Ratio F7:F23
10	Direction of end-to-end line	25	Average velocity
11	20 perc. Pairwise velocity	26	Median acceleration first 5 pts.
12	50 perc. Pairwise velocity	27	Median pressure
13	80 perc. Pairwise velocity	28	Median area covered
14	20 perc. Pairwise acceleration	29	Median finger orientation
15	50 perc. Pairwise acceleration	30	Phone orientation

sizes. At the same time, related work indicates adequate performance with small sample sizes [3], [4].

Each user interacted with the document IDs in a different order. Therefore, we construct each training set by including document ID in the order of target users and their interaction with the Android data collection app. We supply the first two of three IDs when training Wikipedia models, while less data is available for game models. Only one document ID out of two is applied to Game models. For example, user #2 interacted with document IDs in the order of [1, 3, 2, 4, 5, 6, 7], including ID 1 and 3 in the behavioural Wikipedia model and using ID 2 as an intersession test and seven as intersession testing data using the sampling strategy illustrated in Figure 3.3.

3.3.4 Model Selection and Parameter Tuning

As part of the original experiment developed by Frank et al. [1], each user is tasked to read Wikipedia articles and play picture comparison games. Each task was designed to provoke specific interactions, such as VG with Wikipedia and HG with games. As such, it is possible to model each interaction separately, and we define reading as *Wiki* and comparing pictures as a *Game*. Since we are interested in model stability, we must test the selected features and train models on data over time. For the first

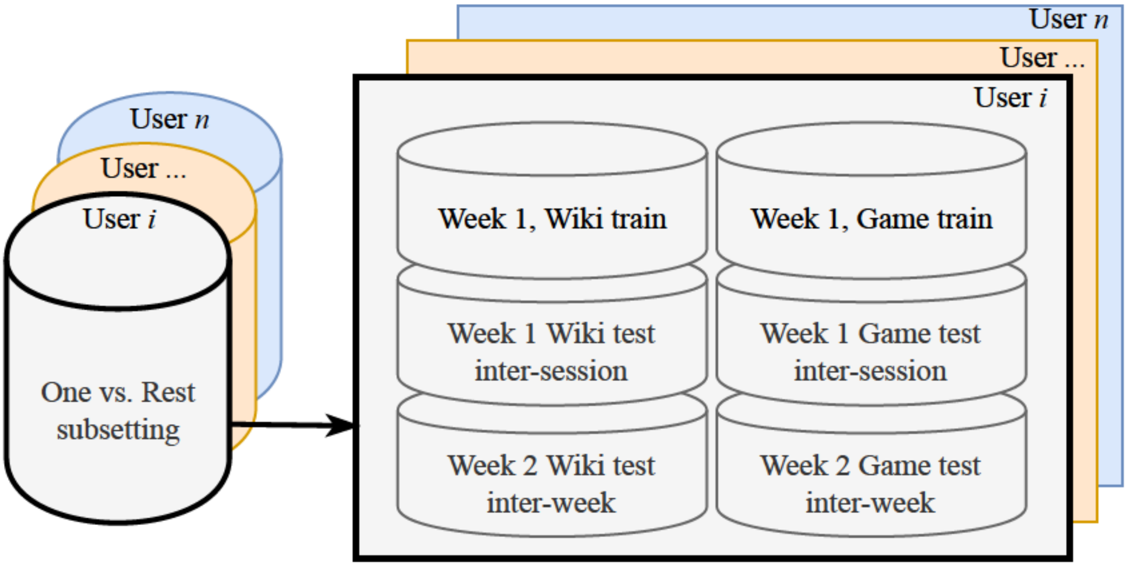

User *n*
User ...
User *i*
One vs. Rest subsetting
User *n*
User ...
User *i*
Week 1, Wiki train
Week 1, Game train
Week 1 Wiki test inter-session
Week 1 Game test inter-session
Week 2 Wiki test inter-week
Week 2 Game test inter-week

Figure 3.3: OvR sub-sampling approach for each of the 12 users selected

week, the last document ID from week one is held out from training and used to calculate *intersession* performance. For the second week, the trained model from week one is tested using document IDs from week two, constituting *intersession* results. Thus, results can be evaluated over time by comparing intra versus intersession scores.

Figure 3.4 illustrates the training pipeline to select features and tune hyperparameters. The proposed approach applies to Wiki and Game models, wherein models are trained on the individual user's first session(s) of data as previously demonstrated in Figure 3.3. Since we learn from relatively small training data sets, repeating 5-fold CV ten times minimises bias between each feature and parameter test. Training data is standardised within each test to have one standard deviation with zero means and scaled to a min-max range in [0, 1], as in [2], [96]. Pre-processing adjusts the data for feature selection and hyperparameter tuning. Features are then included or excluded based on performance rank together with hyperparameter tuning of the classifier using the selected features in each test.

In this work, we focus on illuminating which feature, or combination of, are

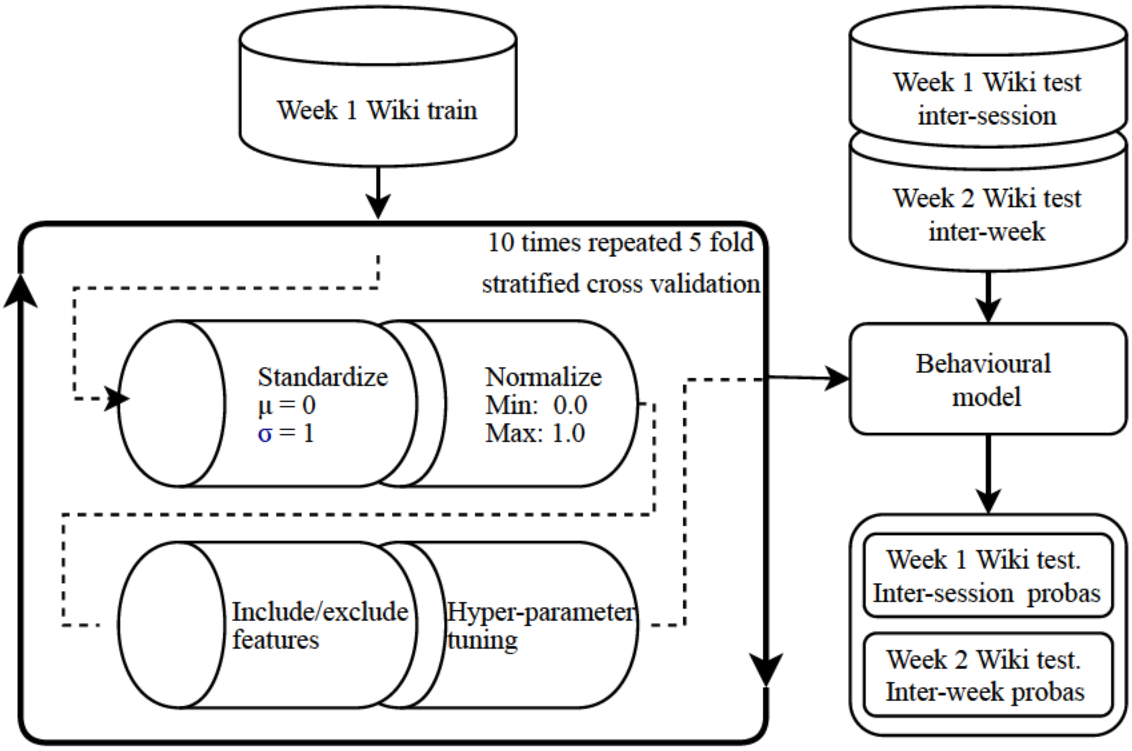


Figure 3.4: The modelling pipeline for each user

essential when modelling users using the same two classifiers, KNN and SVM-RBF proposed in the related literature [1]–[3], [96]. We evaluate the KNN setting $k = \{3, 5, 7, 9\}$ with neighbour weight estimated by the inverse of their Euclidean distance or uniformly distributed. For SVM-RBF, all combinations of γ and C values of {0.0001, 0.001, 0.01, 1, 10, 100, 1000} are searched. The parameter grids are defined following the original literature to make comparisons fairer and to limit the resources required when running on mobile devices. For example, a high k setting would demand comparing new gestures against more reference points, which may not be ideal for resource-constrained environments like smartphones. For all cases, models are selected and optimised to maximise the AUC since this score is threshold independent while also allowing identification of the best error trade-off between both classes [72].

3.4 Feature Selection and Analysis

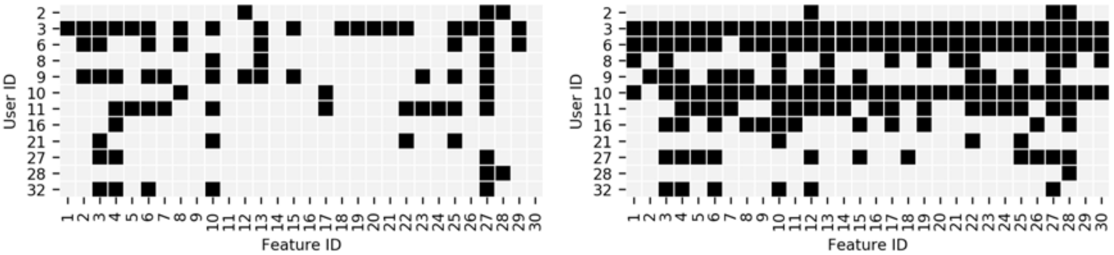
This section presents several feature selection methods and our results for each while analysing the different outcomes among the approaches. Feature selection is a type of dimensionality reduction that aims to determine the smallest feature set required to predict a target class. It not only allows faster computation but also reduces model complexity. When modelling user behaviour, it may be necessary to consider the importance of the feature concerning the target user dynamically. In the case of CA, the positive class usually consists of the data produced by the owner of a device. In contrast, other users are collectively considered to be the negative class. In these experiments, the selected features returned by all selection techniques for Wikipedia and Game interaction are always identical. We report only one feature set for brevity.

3.4.1 Expert Knowledge

Features such as the “change of finger orientation” may logically provide valuable information during feature engineering. However, all of the included users kept their finger orientation the same. Thus, the feature does not add any further information and is removed. Similarly, “phone orientation” rarely changes but could identify a few users orienting their phones differently. However, removing this feature is not recommended as such behaviour may be highly discriminate for specific users. Therefore, this feature is included and empirically tested as part of the implemented feature-selection techniques. In this work, all features except the “change of finger orientation” remain included for empirical testing by the selection algorithms. For all users, features for Wikipedia and Game models are always the same. Thus, we present feature maps that are valid for both models.

3.4.2 Univariate Feature Selection

Filtering techniques, known as univariate selection, rank each feature by applying a scoring function. In the related work, MI is used as the scoring function [1], which returns a statistical measure of information gained between an individual feature and the class label. MI [109] is fast to compute since it does not apply a classification algorithm, but at the same time is also unable to describe how features interact with a classifier. Therefore, features are tested using the modelling approach in Figure 3.4 by iterating and including k highest-ranked features for hyperparameter tuning in the range of k between 1 and 30. Our results are shown in Figure 3.5, which presents the selected features by applying MI for both KNN and SVM-RBF classification, where included features are marked with a black square. Overall, in Figure 3.5a, it can be observed that KNN has selected fewer features than SVM-RBF, as shown in Figure 3.5b.



(a) KNN MI

(b) SVM-RBF MI

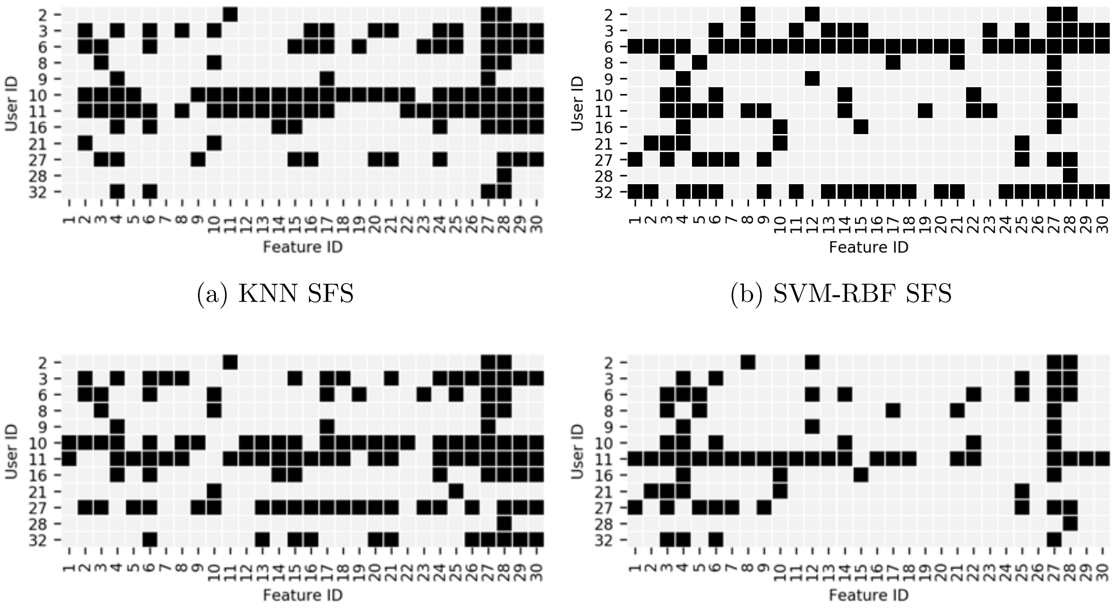
Figure 3.5: Selected features using Mutual Information

3.4.3 Sequential Feature Selection

To overcome the drawbacks of univariate selection, such as the inability to measure feature interaction, applying *Sequential Feature Selection* provides insight into such interaction between features and classification algorithms while testing different subsets. Two modes are available, including or excluding features, forward and backwards, respectively. For each method, a binary float option controls whether the selector can add features from the exclusion list back during the incremental inclusion steps as long as the decision function improves or maintains performance. This section presents these four sequential selection techniques, including SFS, SFFS, Sequential Backwards Selection (SBS) and Sequential Floating Backwards Selection (SFBS) [121].

Forward Selection Using SFS, the feature selection technique begins with an empty feature set and iteratively tests the performance of each feature for inclusion in the forward selection step. If performance persists or increases, the feature remains; Otherwise, the feature is marked as insignificant and excluded from the final user model. As such, this approach attempts to find the least features possible. Figure 3.6a and Figure 3.6b present the selected features in search of the optimal AUC score for each user. Like the SFS approach, Figure 3.6c and Figure 3.6d present the impact of allowing the forward selector to float backwards. As such, the number of selected features increases only if the previously excluded feature positively interacts with selected features. Selected features remain intact for eight out of 12 users. In contrast, the remaining four users are significantly affected, as seen with user #32, reducing the selected features from 22 to four when comparing SFS with SFFS, respectively.

Backwards Selection Contrary to the forward selection, SBS begins with a complete feature set while iteratively testing and excluding insignificant features. This approach aims to reduce a feature set by identifying noise. Figure 3.7 presents the selected features using backward selection techniques, which, compared to a forward



(a) KNN SFS

(b) SVM-RBF SFS

(c) KNN SFFS

(d) SVM-RBF SFFS

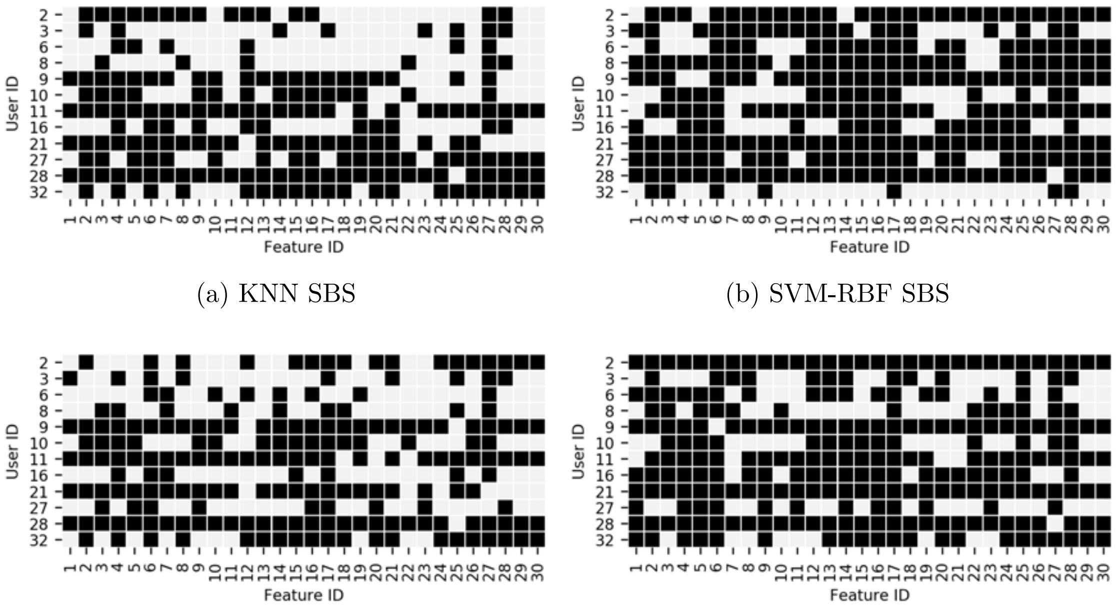
Figure 3.6: Selected features using Forward Selection methods

selection, such as seen in Figure 3.6, a significant increase in the included features can be observed. Like SFS, SBS allows for floating operations, enabling previously excluded features to be included in each step; thus, the exclusion list is considered part of the floating stage until the decision function decays.

SFFS and SFBS are computationally more expensive since the methods reintroduce previously excluded features. However, the techniques also provide better coverage regarding feature interaction and generally produce smaller feature sets. Therefore, the touch behaviour of different users can be described with distinct groups of features, which confirms research question (ii) *Different users may expose individual behaviour through personal feature sets.*

3.5 Experimental Results and Discussion

In this section, we present the average results of all user models concerning the selected features using the selection techniques presented in Section 3.4. To allow for comparison with related work, Figure 3.8a and Figure 3.8b present the average EER scores across all individual users, while Figure 3.9a and Figure 3.9b present the average AUC scores. The results are separated into intrasession and intersession to



(a) KNN SBS

(b) SVM-RBF SBS

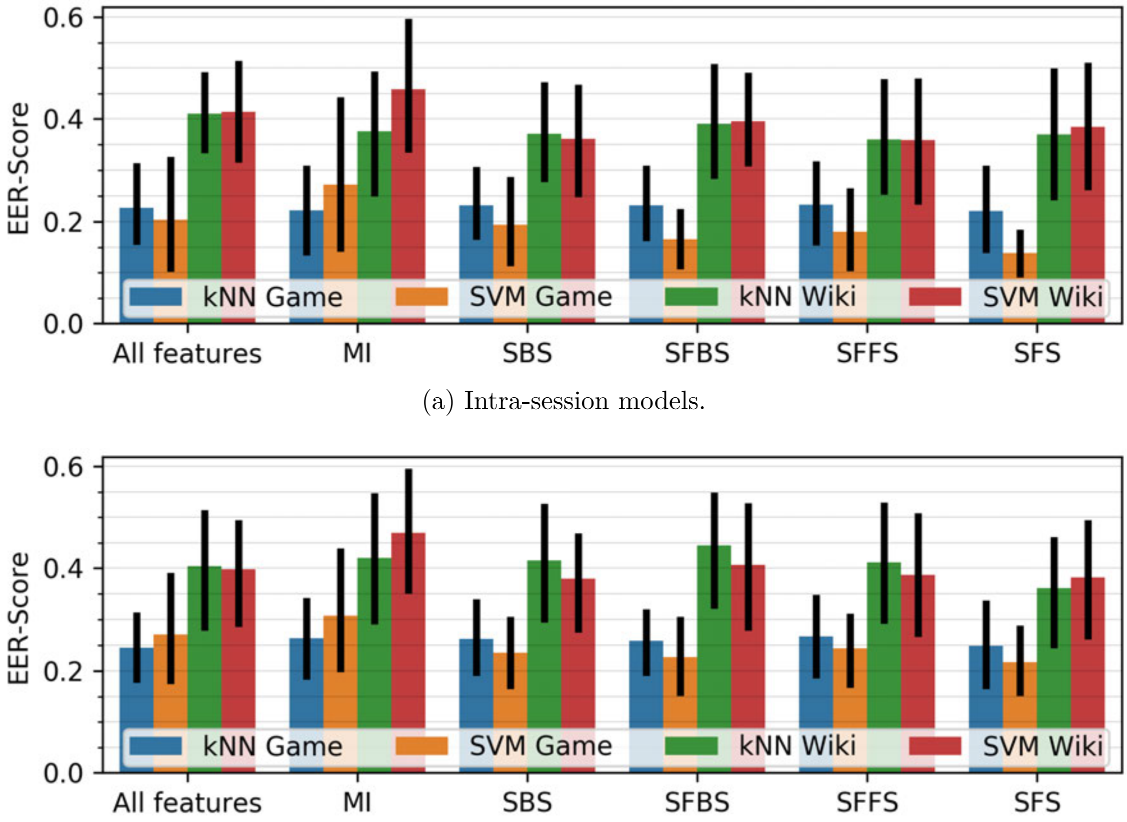
(c) KNN SFBS

(d) SVM-RBF SFBS

Figure 3.7: Selected features using Backwards Selection methods

show model stability over time. For example, intrasession refers to instant authentication after training. In contrast, intersession illuminates long-term performance between sessions of device usage, e.g., training in week one and testing on data from week two. All figures include error bars signifying the 95% confidence interval. We find that SFS maintains or outperforms all other selection methods when applied in an SVM-RBF classifier.

Personalised Behaviour As shown in Section 3.4, specific features are selected for different users when applying our modelling approach as previously illustrated in Figure 3.4. Our method highlights that users express behaviour through various features, and reducing model complexity without affecting model performance is possible. We observe in Figure 3.9a and Figure 3.9b that SFS generally outperforms all other feature-selection techniques by maintaining or improving model performance, even over time. Thus, some features can be removed as they likely introduce noise, as users may not conform with specified behaviour calculated by some features. Furthermore, floating options (SFFS and SFBS) only improve model performance after consuming more computational resources. As such, it is not advisable to use floating options when applying sequential feature selection on the



(a) Intra-session models.

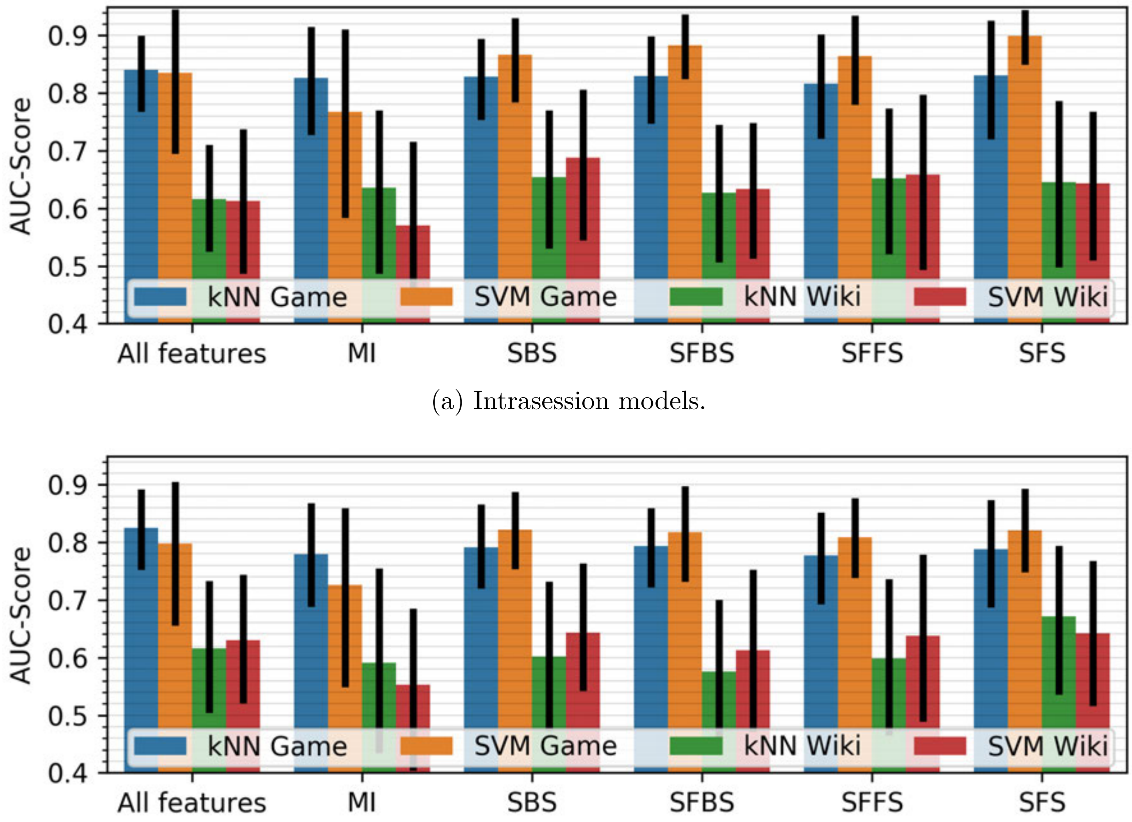
(b) Inter-session models.

Figure 3.8: Mean EER scores per feature selection technique with 95% confidence

selected users.

Cross-Model Performance We support the general hypothesis [2], [96] that HG are more discriminate. However, we extend the work by observing that all feature-selection techniques independently identify the same feature sets when measuring HG and VG. As such, we compared model performance by testing predictive Game behaviour against a trained Wiki model and confirmed that Game models could predict Wiki behaviour and vice versa. Thus, training two models may be unnecessary as they could be interchangeable.

Stability Over Time The selection of features for each user may affect model stability over time. Figure 3.9a and Figure 3.9b compare the AUC score over time, expecting reduced performance because human behaviour tends to change over time, and the proposed method is limited to one-off training. Despite the expectation, most applied feature selection techniques sustain performance over time with a



(a) Intrasession models.

(b) Intersession models.

Figure 3.9: Mean AUC scores per feature selection technique with a 95% confidence

limited reduction.

Shared Feature Importance Figure 3.10 presents an overview of selected features across all 60 models, 12 for each selection technique. The lowest occurrence of a single feature is 15 times across all models, whereas the most common feature was included 50 times. Interestingly, features 10, 23, and 25 were removed by Frank et al. in their work [1]; however, the empirical evaluation shows these features may be significant to specific users. Feature 25, "average velocity", appears to be a robust generic feature across all the selected users. Besides being robust, certain unique features, such as those selected infrequently, might help identify specific people. Therefore, models should be trained on a mixture of robust and unique features while selected using an empirical technique. i.e. SFS.

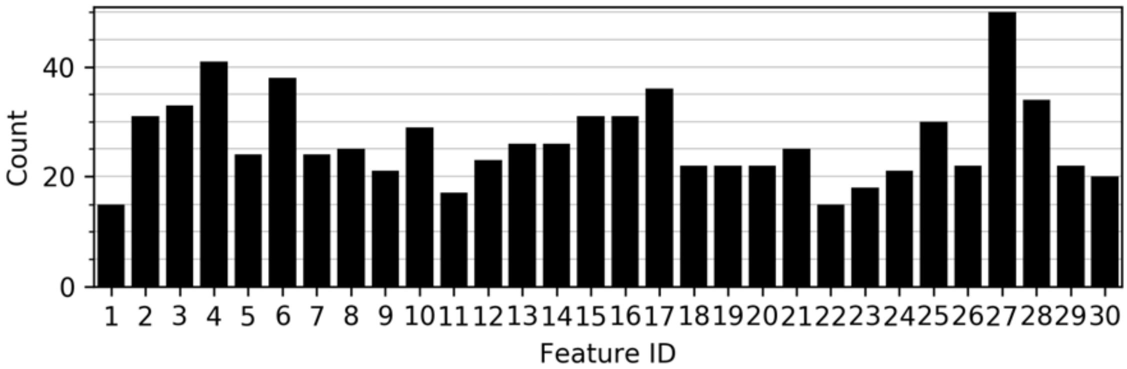


Figure 3.10: Frequency of feature across a total of 60 models trained

3.6 Conclusion

This research empirically evaluated standard features computed and used in touch-based CA. Applying the proposed method confirms that features should be personally considered for each user, while feature selection techniques reduce complexity and often improve performance. On average, the best feature selection technique is Sequential Forward Selection in combination with an SVM-RBF classifier, especially over time. The final approach results in a horizontal (Game) average EER score of 15% and 22% for intrasession and intersession, respectively, while vertical (Wiki) EER reached 37% for both intrasession and intersession. The EER scores are higher than related work since each gesture is evaluated independently. As seen in related work, combining gestures suggests that the error rates are conservative results.

The most common features amongst the selection techniques are "median pressure" and "median area covered", appearing in 81 and 73 per cent out of 60 models tested, respectively. On the other hand, "inter-gesture time" was rarely included but not necessarily insignificant. In the future, having a more extensive selection of features and excluding screen-size-dependent ones would be interesting.

Chapter 4

Omni-directional Modelling

4.1 Introduction

Apple caused a paradigm shift by releasing its first smartphone with a touch screen in 2007. Since then, smartphones have become ubiquitous, with an 81 per cent penetration rate in the US [122]. With the adoption of smartphones, a single device can now provide access to the entire life of its owner, e.g., entertainment profiles such as Netflix, social media accounts with instant messaging, and online banking, amongst others. However, user authentication on touch devices is challenging due to the limited input interfaces.

With facial recognition and smartphone fingerprint reading, biometric lock screen authentication can confirm legitimate users conveniently but cannot continuously maintain user authenticity through user sessions. These physiological biometrics also require sensors, which are vulnerable to presentation and replay attacks [123], [124]. Finally, active authentication methods are time-consuming and may interrupt or delay productivity [20], [27]. The early work by [1] sought to establish the viability of touchscreen data as input for behavioural biometrics and explicitly argued for user authentication through touch gestures. Contrary to other types of CA, the touch-based method only requires a touchscreen and may thus be applied across any device with a touch interface. For instance, humans in smart factories could use a touchscreen to operate a conveyor belt or pickers in a warehouse to use a smartphone for packing orders. However, several challenges remain, such as identifying high-quality behavioural traits and defining a standard to compare touch-based CA methods [10], [68].

4.1.1 Motivation and Challenges

Commonly, researchers model touch behaviour in the context of HG or VG, with the horizontal model typically outperforming the vertical [1], [2], [71]. Figure 4.1 presents the typical modelling approach for a user touching their device four times. Each gesture is evaluated for direction and then processed by the respective bi-directional model. The model then predicts a probability score between zero and one for gestures 1 through 4. Higher probability results define a favourable decision to unlock the device as there is a high probability that the user is genuine. As shown in the rightmost Figure 4.1, a moving average can be applied as a window over the gestures and probabilities to smooth and improve authentication accuracy. However, a model should be agnostic to the direction such that a single omni-directional model can authenticate the owner regardless of touch direction or the number of gestures.

The previous study in Chapter 3 and published in [71] took early steps towards investigating distinct traits concerning individual user behaviour. However, this work focuses on limited features and does not consider mixing the directional gestures into a single omni-directional model. Consequently, this chapter is motivated by the challenges and discrepancies between modelling approaches, the need to compare behavioural feature sets, and a desire to create user profiles based on high-quality behavioural features. Lastly, we define a new method to select the model parameters to reduce complexity at the cost of minor performance drops. The fol-

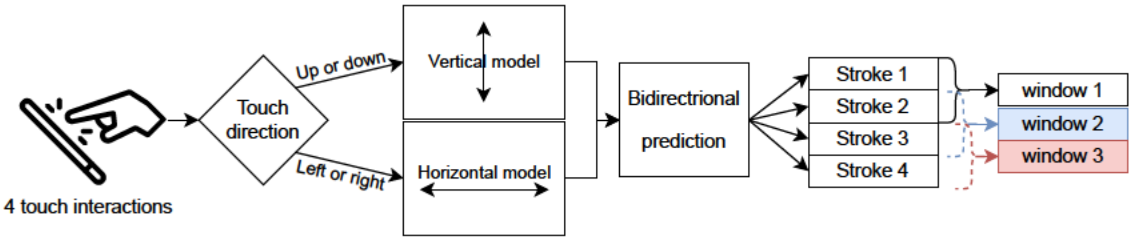


Figure 4.1: A high-level overview of the traditional bi-directional approach using score-level fusion through sliding windows.

lowing section will detail each contribution related to these challenges.

4.1.2 Research Questions and Contributions

Following the challenges outlined, this chapter contributes answers to the following questions:

1. What is the performance difference between the proposed approach versus a typical bi-directional where parameters are highly optimised?

2. What is the impact of combining n gestures when using the typical and proposed approach?

3. Which feature set should be used considering the different directional modelling approaches?

We focus on balancing the model complexity and performance to answer these questions while choosing the optimal feature set for a single omni-directional model and two independent Hs and Vs models. Each user is modelled using five feature sets to evaluate the best overall behavioural traits. We report the AUC since it is threshold-independent while assessing the best trade-off between classes as a function of all thresholds [72]. We also report the EER since it is the most popular metric across the literature, noting that such a rate only represents a specific decision threshold. Finally, our results are reported for single-gesture and combining gestures, measured by the AUC and EER scores when combining a sequence of gestures in ranges 1 through 20. The rest of this chapter is structured as follows: Section 4.2 covers the related work. Section 4.3 describes the experimental design and the applied methods to implement and complete the experiment. Section 4.4 presents the results before concluding in Section 4.6.

4.2 Related work

Touch-based CA relies on distinct features to authenticate an owner from other users. Several approaches from the literature have exhibited promising results using different feature sets. However, some overlap where others use existing feature sets from the literature [1]–[5]. This work focuses on five different feature sets chosen because they offer the best variation amongst the related literature. The study presented in [5] explains how to differentiate between child and adult smartphone users using touch-based features. While their feature set is not used in authentication, the features they extract could also define distinct behavioural features used for authentication; thus, applying these in continuous authentication would be attractive. The authors conclude that gestures perform better than clicks in differentiating children from adults and that an ET classifier outperforms others. The ET classifier is not commonly used for CA and could also be interesting in the context of CA. The following sub-section will compare the classifiers often used for CA.

4.2.1 Classifier, Parameters, and Metrics

Rather than identifying children from adults, Table 4.1 presents work using various classifiers to implement touch-based CA and specifies the parameters used across the literature. Although each classifier behaves differently, they can also differ internally depending on parameter settings. The CA literature often evaluates several classifiers with varying parameters [2]. However, it is challenging to characterise the best overall classifier amongst the related literature without specifying comparable parameter search space or using the same metrics. For example, [1] achieves a 13% EER on a single gesture using a SVM-RBF. However, [78] finds RF superior, with a single gesture accuracy of 65%. In [93], the RF classifier offers the best EER score of 25% instead. Finally, [4] optimises for a balanced F-score with a single gesture performance between 0.7 and 0.8 - depending on the type of gesture. Thus, the related works are challenging to compare beyond their different approach due to varying metrics.

4.2.2 Modelling Approach and the Impact of Training Size

When modelling touch-based CA, gestures can be categorised and processed depending on the direction of the trajectory or independent of the direction. For example,

Table 4.1: Classifiers and parameters across related work. Bayesian Network (BN). Undefined parameters (N/A)

Paper	Classifier (parameter)
[1]	SVM-RBF (C=N/A, γ=N/A), KNN (k=1,3,5,9)
[2]	SVM-RBF (C=N/A, γ=N/A), KNN (k=9), RF (n=1000), LR, NB, MLP, DT
[5]	SVM-RBF (C=1.4, γ=0.15), KNN (k=7), RF (n=200), ET (n=200)
[78]	SVM-RBF (C=2,8, γ=8), KNN (k=3), RF (n=10, 100)
[93]	SVM-RBF (C=0.03, γ=0.006), KNN (k=2-20. Best=11), RF (n=1000), BPNN (2 + 1 layers and learning rate=0.001)
[4]	OCSVM (nu=0.1), iForest (contamination=0.1)

grouping left and right gestures in a HG model is used in [1], [2], [4], [78], [93]. In contrast, [93] also models each direction individually while evaluating against mixed directions. These papers approach the classification as a binary challenge using a OvR scheme [73]. The device owner then forms the positive class, and the negative class groups the remaining users. OvR causes a class imbalance that can be mitigated through sampling techniques [1], [2].

However, it becomes increasingly challenging to compare works since the directional approach differs, and OvR sampling may further affect the characteristics of training data. Thus, we highlight the varying amount of training data used and the potential effect on performance. Eighty samples are used for HG/VG models in [2], 100 for HG/VG in [78], and roughly 160 per direction-specific models in [93] for each class, respectively. The concept of model stability through varying training data size is partially studied in [3], [78], with [4] showing minor improvement using more than 80 training observations.

4.2.3 Removing clicks and combining gestures

Defining gestures from clicks is essential as a precursor to modelling since clicks appear to cause poor performance [4]. There are different ways to identify gestures, e.g., counting the points within a trajectory and removing gestures with less than four [2] or five [1] touchpoints. Others assess the directional angle and exclude gestures that change direction, such as sliding up and then down without releasing the finger [3], [4], [93], or a minimum length can be required [78]. Since a user's touch gestures may have slight variations, authenticating based on a single gesture is challenging because it requires the classifier to identify each touch operation perfectly. In [1], 1-20 gestures are combined using different techniques based on KNN neighbours' distance and the SVM's hyperplane, with 11 and 13 gestures working well.

In contrast, [2] takes a sequence of ten feature vectors and applies a moving average when predicting the user. Rather than averaging the feature vector, [78] uses a moving average over the predicted probabilities of a sequence of gestures and concludes that ten gestures are optimal. The latter approach can be seen in Figure 4.1 with a moving average window of two gestures. However, [93] found 11 gestures a reasonable trade-off. [3] groups gestures by five and authenticated based on a majority vote. Lastly, [5] combines 9-11 gestures. Consequently, improving performance by combining around ten gestures is common, but the method and outcome vary across the literature. As such, we seek to answer research question two by varying the number of combined gestures in the context of the different approaches and feature sets. The following section presents the experimental design and describes the proposed omni-directional method, training size, data set, and the implemented behavioural feature sets.

4.3 Experimental setup

The central hypothesis of this chapter argues that behaviour can be generalised by an omni-directional model - matching or outperforming the traditional approach where the horizontal and vertical gestures are modelled independently. If true, the time to model a user can be reduced by roughly half. Further, selecting and evaluating essential behavioural features may be more straightforward as only one model needs to be inspected. The traditional approach is configured as a baseline and omni-directional as the contender to evaluate our method. We define an omni-directional model to process any gesture independent of the direction of the gesture. In contrast, bi-directional models separate gestures depending on the underlying direction. Furthermore, five different feature sets are used to illuminate which behaviour works in the context of the proposed method.

4.3.1 Data and feature sets

This chapter uses the raw data collected by [2] as it contains more users, observations, and extended periods of data compared to others [1], [70], [78], [80], [117], [125], [126]. The data was collected over two sessions to enable intersession authentication. The raw data includes a user ID, swipe ID, timestamp, (x, y) coordinate pairs, pressure, and the area covered by the finger for each recorded touch point. Portrait and landscape data are separated into different sets. We exclusively focus on the portrait mode as the most preferred smartphone orientation [96]. In Figure 4.2, a single gesture is visualised and shows a user moving their finger from Point

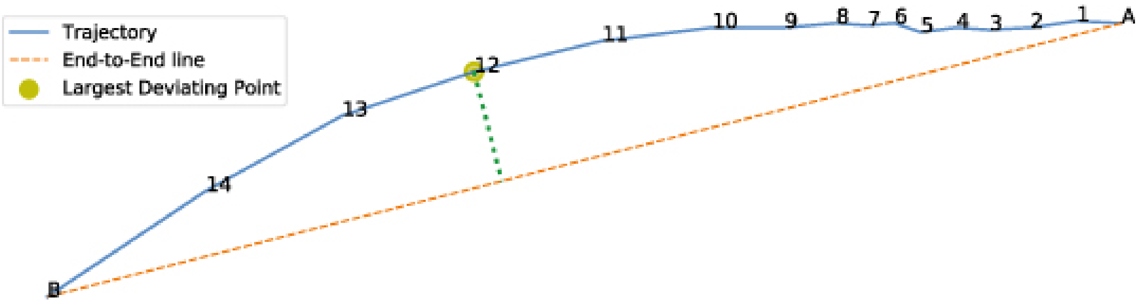


Figure 4.2: Visualisation of the touchpoints from a single gesture and three features.

4.3.2 Compatible users and raw data

Since this work is not looking to vary the training data, we follow the recommendation by [4]. Thus, eligible users must provide more than 80 training observations with each horizontal and vertical direction to establish the two models for the bi-directional approach. An even number of observations is selected for each direction among up, down, left, and right gestures. In contrast, the omni-directional model uses all directions. When authenticating, users must also provide enough test data to combine a sequence of n gestures. [1] combined up to 20 gestures with ten gestures producing good results; thus, we combine gestures in the range of 1 through 20 for comparability.

4.3.3 Features

Figure 4.2 presents a single gesture drawn between a finger down and up, Point A and Point B, respectively. A blue line is illustrated as the trajectory, which defines the length between the 16 points collected as part of the Android operating system when capturing touch interactions. An orange dotted line is also shown to highlight the End-to-End (E2E) length feature, which defines the flight distance. The Largest Deviating Point (LDP) is another feature at point 12, measured by the dotted green line. Several feature sets have been proposed throughout the literature to describe behaviour, as seen in Table 4.2. Several papers overlap without directly comparing or discussing which features are included or excluded in each set.

4.3.4 Feature extraction and data cleaning

We selected five papers [1]–[5] from the literature because they provide a broad spectrum of different behavioural traits. We remove clicks and interactions with less than or equal to five points or if the trajectory length is shorter than three pixels to focus on gestures. Besides filtering clicks, some features cause undefined values, such as the E2E line slope for perfect horizontal gestures. Similarly, the inter-gesture time is unavailable for each user's first gesture. After removing gestures and data cleaning, the data set consists of 78,423 gestures that qualify for all five feature sets.

4.3.5 Selecting users of interest

Each model must have 80 observations to generate a stable behavioural model for the target user [4]. Thus, valid users are chosen based on the requirement in Figure 4.3. For each left, right, up, and down direction, 50 training and 30 testing gestures are required. Thus, the training size is 100 for each of the two bi-directional models

Table 4.2: Overlap of features between Frank et al. [1], Serwadda et al. [2], Syed et al. [3], Yang et al. [4], and Cheng et al. [5]. Acceleration, ACC. Velocity, VEL, End-to-End, E2E

F#	Feature name	[1]	[2]	[3]	[4]	[5]	F#	Feature name	[1]	[2]	[3]	[4]	[5]
1	Inter-gesture time	✓		✓		✓	39	Std. pressure		✓			✓
2	Gesture duration	✓	✓	✓		✓	40	25% pressure		✓			
3	Start X	✓	✓	✓	✓	✓	41	50% pressure		✓			
4	Start Y	✓	✓	✓	✓	✓	42	75% pressure		✓			
5	Stop X	✓	✓	✓	✓	✓	43	Mean area		✓			✓
6	Stop Y	✓	✓	✓	✓	✓	44	Std. area		✓			✓
7	Length E2E	✓	✓	✓		✓	45	25% area		✓			
8	Mean Resultant Length	✓				✓	46	50% area		✓			
9	Numeric direction	✓					47	75% area		✓			
10	Direction E2E	✓	✓	✓		✓	48	Start pressure			✓	✓	✓
11	20% VEL	✓		✓			49	Stop pressure			✓	✓	
12	50% VEL	✓	✓	✓			50	Cat. direction			✓		
13	80% VEL	✓		✓			51	X @ max velocity				✓	
14	20% ACC	✓					52	X @ min VEL				✓	
15	50% ACC	✓	✓				53	Y @ max VEL				✓	
16	80% ACC	✓					54	Y @ min VEL				✓	
17	Mid VEL last 3 pts	✓				✓	55	Max VEL				✓	✓
18	Largest dev. E2E	✓					56	Min VEL				✓	
19	20% dev. E2E	✓					57	E2E Slope				✓	
20	50% dev. E2E	✓					58	E2E Intercept				✓	
21	80% dev. E2E	✓					59	X @ LDP				✓	
22	μ Direction						60	Y @ LDP				✓	
23	Length of trajectory	✓	✓	✓		✓	61	LDP pressure				✓	
24	Ratio F7:F23	✓		✓			62	μX VEL pre. LDP				✓	
25	μVEL	✓	✓	✓		✓	63	μY VEL pre. LDP				✓	
26	Mid ACC last 5 pts	✓					64	μX VEL post. LDP				✓	
27	Mid pressure	✓		✓		✓	65	μY VEL post. LDP				✓	
28	Mid area	✓				✓	66	Start pressure					✓
29	Mid finger orientation	✓					67	Time to max VEL					✓
30	Phone orientation	✓					68	X disp. down-down					✓
31	Std. VEL		✓			✓	69	Y disp. down-down					✓
32	25% VEL		✓				70	X disp. down-up					✓
33	75% VEL		✓				71	Y disp. down-up					✓
34	Mean ACC		✓				72	Mid VEL first 3 pts					✓
35	Std. ACC		✓				73	Mid VEL					✓
36	25% ACC		✓				74	Mid ACC first 3 pts					✓
37	75% ACC		✓				75	Mid ACC last 3 pts					✓
38	Mean pressure		✓			✓	76	Mid ACC					✓

and 200 for the omni-directional models. The testing size is set to 30 samples per direction to ensure enough data when combining gestures. Consequently, the data set is reduced from 138 to 35 users of interest.

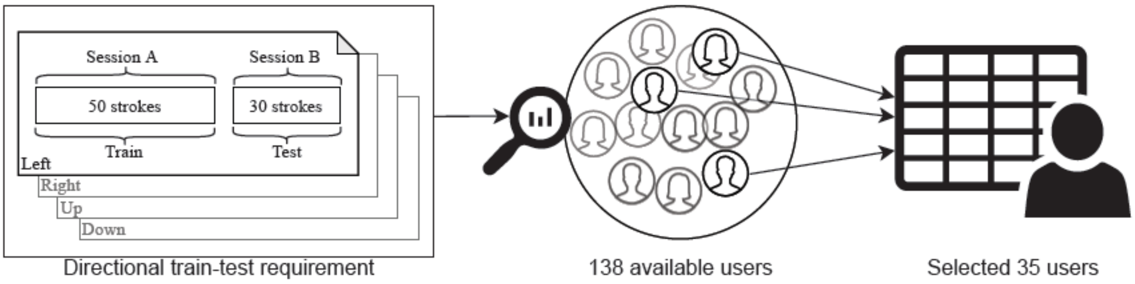


Figure 4.3: Datasubsetting and user selection according to training and testing data requirements.

4.3.6 Modelling pipeline and class balancing

In Figure 4.4, the proposed modelling pipeline is introduced, where data is labelled according to the OvR method for each user. CV is configured using a five-fold stratified loop, repeated five times, and is applied to reduce bias considering the limited number of training observations. When training the bi-directional model using 100 samples and the CV, each training fold is reduced to 80 gestures which adhere to the guidelines of [4]. The pipeline uses Python 3.8 Sci Kit learn [74] and ImbalancedLearn [127] as the sampler. The touch training data from Session A is provided to the pipeline for each user and the specific classifier per feature set from Table 4.2. Figure 4.4, Step 1, under-samples the majority class. Steps 2. and 3. standardise to zero mean and normalise values between 0-1 if the classifier requires it, e.g., SVM-RBF. Finally, Step 4. implements the parameters in Table 4.3.

Once a model is computed, the testing data from Session B. is evaluated by the model, generating predicted probabilities. Data from Session B. is collected at least

Table 4.3: Parameter search space for each classifier tested, as seen in the literature. The parameters are chosen based on the related work where applicable.

Classifier	Parameters
KNN	k={1, 3, 5, 7, 9}
SVM-RBF	C={0.01, 0.1, 1.0, 10, 20, 100}
GBC/RF/ET	Min samples split={0.005,0.01,0.1}, n_estimators={100,200,500,700,1000,1200}

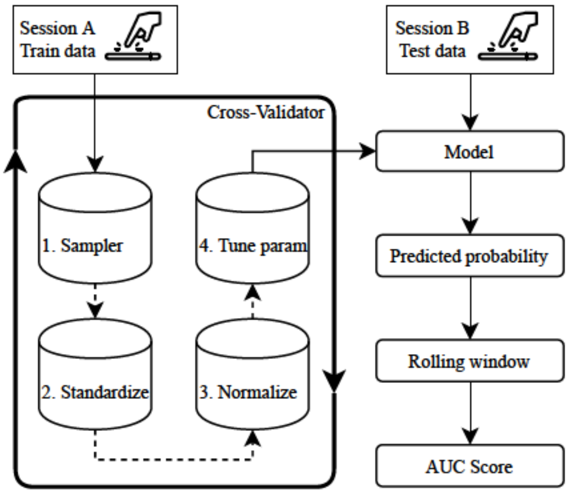


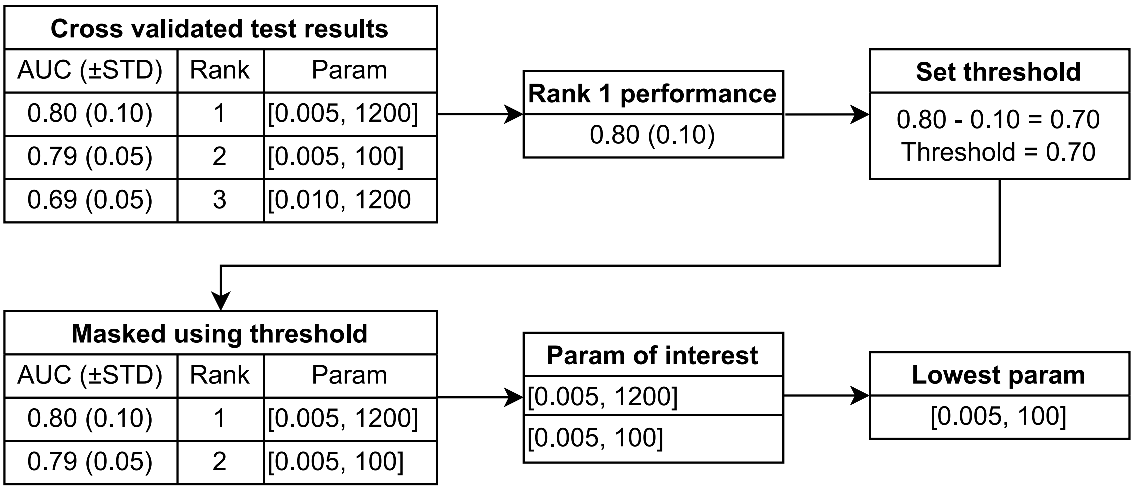
Figure 4.4: Modelling pipeline used for training and testing performance of each user

one day after Session A. Thus, the results measure the intersession performance and are possibly more conservative than works measuring intrasession performance [1], [2], [71]. Because of the CV, the pipeline defined in Figure 4.4 computes 25 (=5×5) models and searches for the optimal parameter for each feature set, classifier, and approach. Thus, the parameter grid in Table 4.3 results in a broad range of models to evaluate, such as an ET classifier, and the optimal parameter is selected by considering a total of 236,250 models. Since class imbalance can be challenging for some classifiers, the proposed method down-samples the majority class following an OvR approach similar to [2], [3]. At the same time, stratification ensures class balance in each fold.

4.3.7 Classifier parameters and complexity

This work assesses how the performance of the proposed single omni-directional model compares to that of the bi-directional model using different feature sets and classifiers. However, each classifier may perform differently depending on the configured parameters and the provided feature set. Classifiers such as KNN, SVM-RBF, RF, and ET are commonly seen in the literature with different parameters [1]–[5]. Still, in published work, it can be unclear which parameters are tested and thus the most effective across the different behavioural feature sets. We implement each classifier to address this while searching the parameters in Table 4.1. Most classifiers become more complex as their parameters increase in value, such as the number of trees in RF's and the regularisation parameter of SVM-RBF- which may lead to overfitting models. Thus, a consideration is made between balancing the best parameter and the model complexity. The parameters are optimised to maximise the AUC score as a function of all thresholds [72]. We select the classifier's parameter by subtracting one standard deviation of the AUC score from the model's best-performing AUC score, trading minimal performance gains for reduced complexity.

Figure 4.5 visualises the parameter selection approach. The example starts with the cross-validated output of three parameters tested for a given algorithm. Rank 1 provides the highest AUC score from the three test results. However, the standard deviation value is often high while providing minor performance over the other results. Thus, we take the best AUC score and subtract the associated standard deviation value to set a threshold of the test results, which defines a mask of acceptable parameters. This example has two parameter pairs as the mask, in which the lowest parameter is selected since it produces a less complex model while generally preserving good performance. Consequently, we sacrifice minor performance while lowering the deviation between users. Similarly, it reduces the model complexity, translating to faster training. While classifiers may have additional parameters that can affect performance, the scope of this chapter is to evaluate those tuned in the related work for comparability. The details and mechanisms of each classifier are well documented across the literature, but the following parameters are briefly covered. For SVM, the kernel used is a Radial Basis Function and the γ parameter scales according to the number of features and their variance [74]. For GBC, the sub-sample parameter is set to 0.95, which trains each base classifier on a fraction of the available data. Sub-sampling is a stochastic behaviour and typically enhances performance.



Cross validated test results

AUC (±STD)	Rank	Param
0.80 (0.10)	1	[0.005, 1200]
0.79 (0.05)	2	[0.005, 100]
0.69 (0.05)	3	[0.010, 1200

Rank 1 performance

0.80 (0.10)

Set threshold

0.80 - 0.10 = 0.70
Threshold = 0.70

Masked using threshold

AUC (±STD)	Rank	Param
0.80 (0.10)	1	[0.005, 1200]
0.79 (0.05)	2	[0.005, 100]

Param of interest

[0.005, 1200]
[0.005, 100]

Lowest param

[0.005, 100]

Figure 4.5: Example of the parameter selection method when balancing performance and complexity.

4.4 Results

We first present the classifiers and parameters selected using our approach and with a comparison to the related work. Next, we compare bi- and omni-directional models in the context of single-gesture authentication before considering the impact of combining gestures. Lastly, we highlight the benefits of combining gestures in the context of the five feature sets and classifiers.

4.4.1 Modelling parameters

We searched through the parameters in Table 4.3 for the Horizontal (Hs), Vertical (Vs), and Omni-directional models to better understand which parameters work for most users. For KNN and SVM-RBF, the optimal parameter changes depending on the applied feature set and directional modelling approach. For KNN, as seen in Figure 4.6, most models found three neighbours a suitable parameter when using the TA behaviour. In contrast, the other feature sets change between 1, 3, and 5 neighbours but rarely 7, regardless of direction. While a shared parameter cannon be suggested based on this result, we see a similar difference in the optimal parameter used across the literature, as seen in Table 4.1, where the optimal k is either 3, 7, 9, or 11, depending on the referenced work. For SVM-RBF and TA, as seen in Figure 4.7, the Hs models favour C=0.1 while the Vs models vary between C=0.1

and C=1.0. Similar to KNN, different feature sets prefer different parameter values. However, most models perform decently with a C parameter of 1.0.

Similar to the parameter search for SVM-RBF, it is challenging to suggest an optimal parameter for all users. Contrary to KNN and SVM-RBF, all tree-based classifiers use 100 trees and a sample split of 0.005 for all users, feature sets, and directions. An exception is the GBC classifier, with omni-directional models selecting 200 trees for approximately 5 out of 35 users depending on the feature set. Figure 4.8 shows an example of the trade-off between performance and time to train an omni-directional model depending on the parameter complexity selected for the tree-based classifiers. However, we can observe that the GBC improves more than the other classifiers when increasing the number of trees but still underperforms compared to the ET classifier. Performance may increase with the number of trees used in a tree-based classifier but requires a longer training time. We argue that the performance gain is insignificant, considering it can take up to four times longer to fit the models. Consequently, there is limited benefit in increasing the complexity beyond 100 trees compared to [2], [93], which uses up to 1000. Thus, our parameter selection approach may also benefit the time required to (re)train models.

4.4.2 Bi versus omni-directional single-gesture comparison

After setting the optimal model parameters, Figure 4.9 presents the mean AUC score for each classifier, grouped by each feature set when authenticating users using a

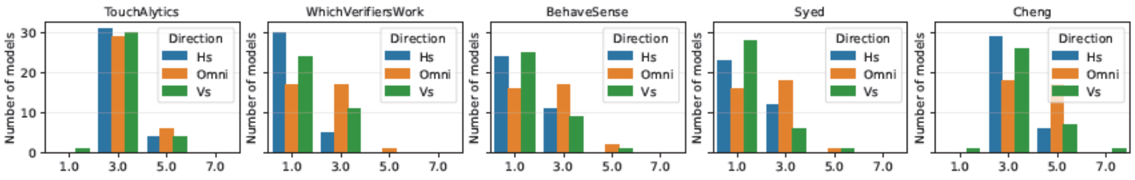


Figure 4.6: Frequency of parameters for KNN, per direction and feature set.

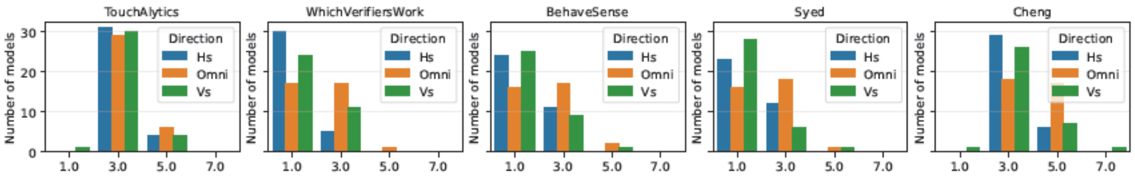


Figure 4.7: Frequency of parameters for SVM-RBF, per direction and feature set.

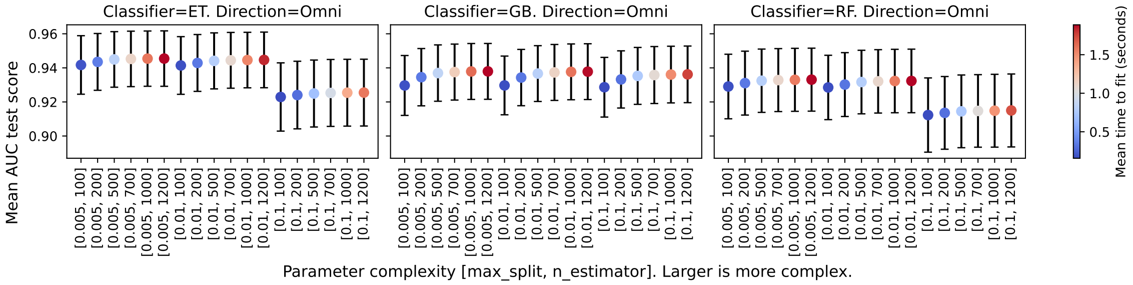
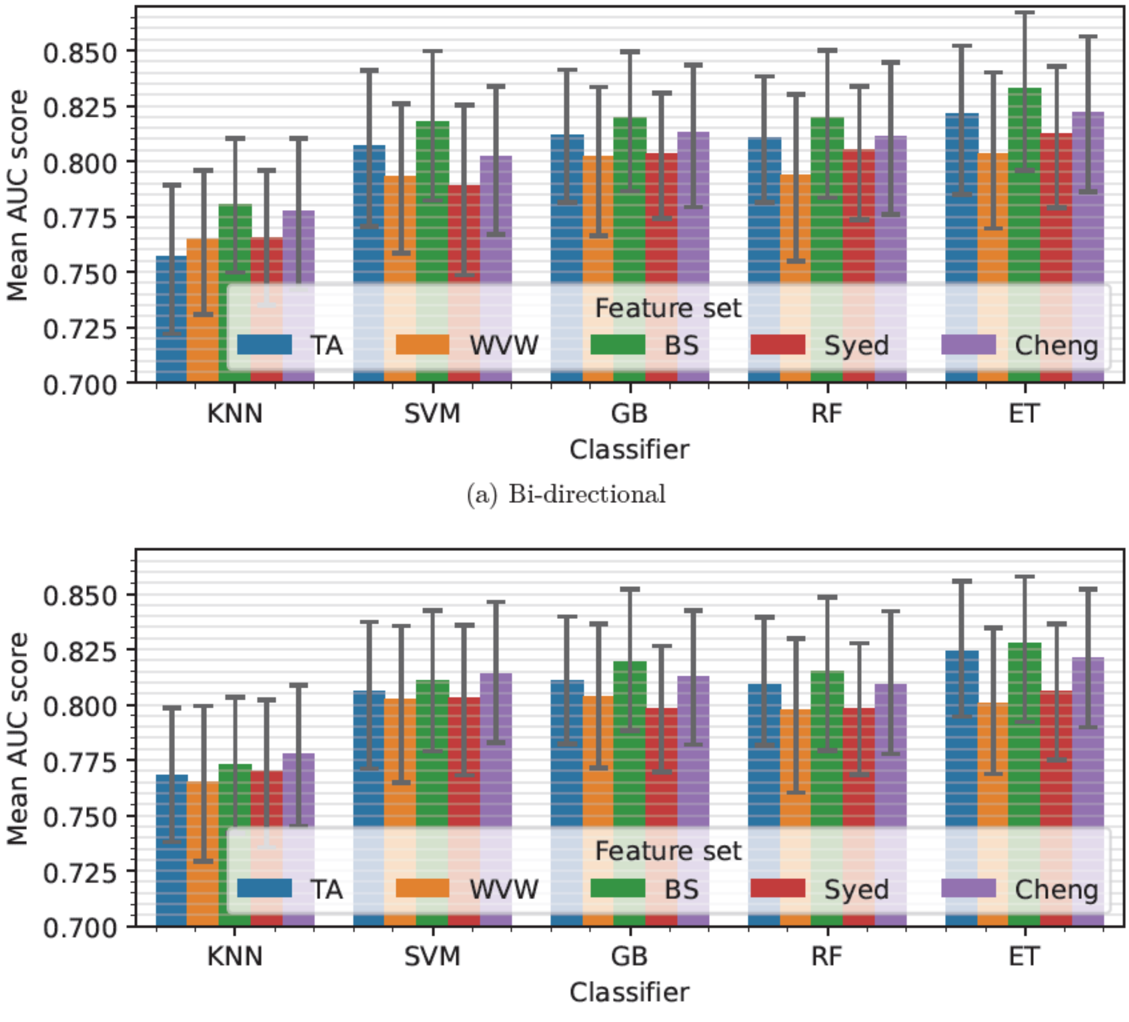


Figure 4.8: Mean AUC scores when fitting tree-based models with greater parameter complexity and coloured by the time to train models in seconds using the TA feature set. Similar patterns were found in the other feature sets.

Table 4.4: The top five single-gesture performances ranked by highest mean AUC score amongst Bi and Omni-directional classifiers and feature sets.

Classifier	Feature set	Approach	AUC (±STD)	EER (STD)
ET	BS	Bi	.833 (±.103)	.239 (±.098)
ET	BS	Omni	.827 (±.098)	.247 (±.094)
ET	TA	Omni	.824 (±.096)	.247 (±.087)
ET	Cheng	Bi	.822 (±.106)	.251 (±.104)
ET	TA	Bi	.821 (±.103)	.252 (±.096)

single gesture with each model. Unsurprisingly, the performance differs amongst the feature sets. BehaveSense (BS) [4] generally ranks top, whilst Syed [3] and WVW [2] often perform poorly. While the bi-directional approach has the highest mean AUC and EER score, the difference from the omni-directional counterpart is negligible; moreover, the standard deviation for omni-directional models is slightly lower for both AUC and EER compared to bi-directional models. In [2], they achieved an EER score of 13.8 and 17.2%, Hs and Vs, respectively, but required ten gestures. Similarly, we also notice that some users are more challenging to model than others, as indicated by the wide error bars in Figure 4.9. Regardless, the goal of this work is not to exclude or identify problematic users but to compare the modelling approach irrespective of these. Table 4.4 highlights the top five classifiers with the highest mean AUC score. The results for each model are compared against the top-performing model on the first line in Table 4.4 to further detail the answer to research question one.



(a) Bi-directional

(b) Omni-directional

Figure 4.9: Mean AUC scores for single-gesture performance across all users for each classifier and feature set while comparing the directional approaches with 95% confidence.

4.4.3 Wilcoxon signed-rank test

To measure whether the top-performing model is better than others, we apply the Wilcoxon signed-rank test using the best model as the reference. The test compares the AUC scores between the reference. It iteratively selects the other classifiers to evaluate the AUC distributions and whether the best AUC scores differ significantly from any other classifiers. More specifically, the null hypothesis assumes that the AUC scores predicted by classifier A are from the same distribution as classifier B's. We wish to reject the null hypothesis with a 5% confidence level. If the null hypothesis cannot be dismissed, the reference model is not significantly better than the comparison. Table 4.5 presents seven classifiers that fail to reject the null hypothesis, suggesting that the traditional Bi approach is not significantly better than the proposed omni-directional approach. As such, we can answer the first research question. In single-gesture authentication, there is an insignificant performance difference between the traditional and our proposed omni-directional methods. While Figure 4.9 shows that the BS feature set consistently outperforms the others irrespective of the classifier and modelling approach, we note this may not carry over when combining gestures, which the following section covers.

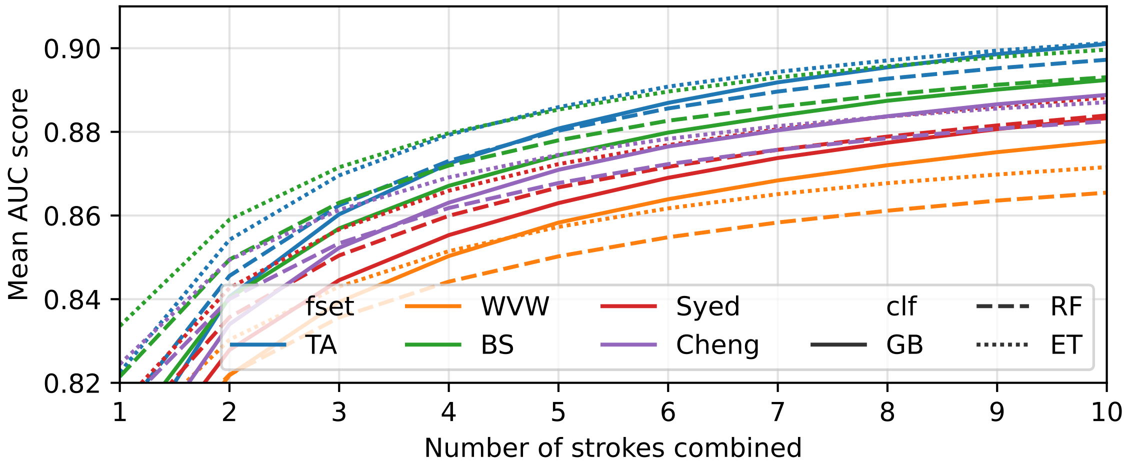
Table 4.5: Classifiers that are not significantly different from the single-gesture ET, BS, Bi-directional AUC distribution.

Classifier	Feature set	Direction	P-value
ET	Cheng	Bi	.3257
ET	BS	Omni	.3098
ET	Cheng	Omni	.2013
ET	TA	Omni	.1589
ET	TA	Bi	.1014
RF	Cheng	Bi	.0665
GB	Cheng	Bi	.0574

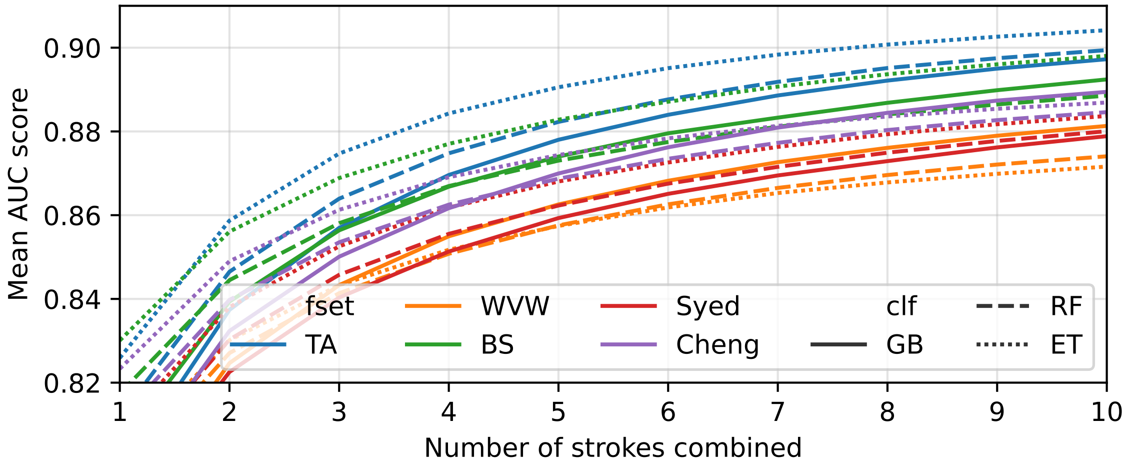
4.4.4 Combining gestures

While the best single-gesture classifier was an ET classifier using the BS feature set, we visualise the influence of combining gestures in Figure 4.10. What stands out is the steady incline in the mean AUC score for the omni-directional ET classifier using the TA features. Compared to [2], the proposed approach achieves equivalent results using five gestures compared to ten in the related work. Table 4.6 details the top five performing combinations across the bi and omni-directional methods when combining five gestures. When combining gestures, the best classifier remains an ET classifier, but the feature set changes to TA. Compared to single-gesture authentication results in Table 4.4, we improved the mean AUC score from 0.833 to 0.890 (+5.7%) and reduced the EER score from 0.239 to 0.179 (-6%). More importantly, the proposed omni-directional method outperforms the traditional bi-directional approach.

In the context of single-gesture authentication, our approach compares to the



(a) Bidirectional



(b) Omni-directional

Figure 4.10: Performance when combining gestures. The plot is limited to 1 to 10 gestures due to limited gains beyond ten and presents the different classifiers, Clf, and the feature sets, Fset.

Table 4.6: Top five performances, combining five gestures, ranked by highest mean AUC score amongst Bi and Omni-directional classifiers and feature sets.

Classifier	Feature set	Approach	AUC (±STD)	EER (±STD)
ET	TA	Omni	.890 (±.099)	.179 (±.112)
ET	BS	Bi	.886 (±.106)	.181 (±.112)
ET	TA	Bi	.886 (±.109)	.182 (±.117)
ET	BS	Omni	.881 (±.096)	.190 (±.104)
GB	TA	Bi	.881 (±.093)	.190 (±.103)

traditional one but requires just one model instead of two. Thus, modelling could be faster and easier to manage, deploy, and interpret. At the same time, our approach is superior when combining three gestures or more. We found limited improvements for any methods when combining more than ten gestures. Hence, Figure 4.10 is limited to combining ten gestures as the curve flattens without changing the rankings of classifiers. Compared to [2], we also combined ten gestures and achieved an average of 0.905 AUC and 0.159 EER score, which is +0.004 EER; however, we have a single model and a more stable standard deviation. We found minimal improvements using more than ten gestures, as seen in Figure 4.11, which shows omni-directional performance. The same is true for the bi-directional models combining more than ten gestures. Thus, to answer research questions two and three, three to five gestures are enough to provide satisfactory performance. Despite being the earliest feature set, we suggest using the TouchAlytics set since the results show better performance for the ET and amongst many of the classifiers used for bi and omni-directional methods.

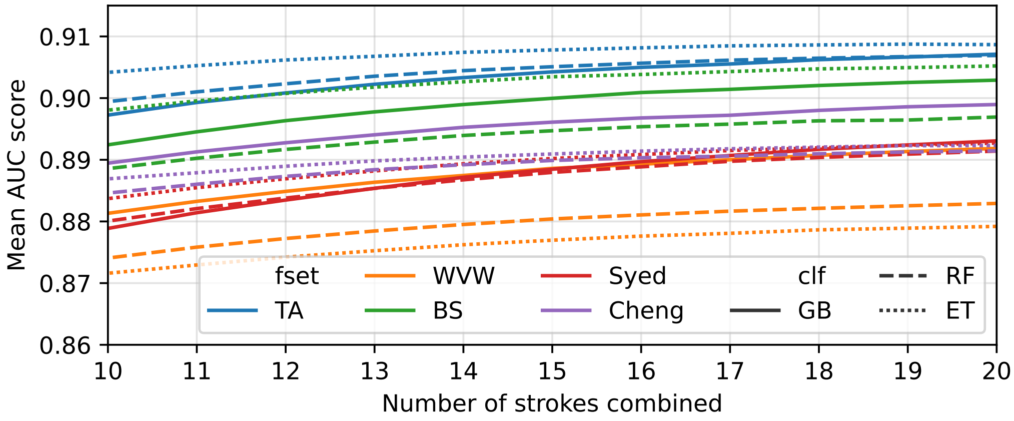


Figure 4.11: Mean AUC scores when combining 10 to 20 gestures grouped by classifiers, Clf, and the feature sets, Fset. The plot is exclusively for the omni-directional approach. Similar trends appear for the bi-directional method but with slightly lower scores

4.5 Limitations

4.5.1 Inconsistency of comparable metrics across the literature

AUC is threshold independent and aims to produce models that find the best trade-off between miss-classifying the genuine and non-genuine users. The EER is derived from AUC based on selecting a threshold that separates the two classes while balancing miss- classification equally. However, EER is not the best metric to compare since it depends on the chosen thresholds, which vary between users. Similarly, false acceptance and rejection rates suffer from the same issue. Accuracy is rarely seen in the related work, perhaps since it generalises both true positives and negatives over all data points; thus, a majority class with good performance may skew the results. In our work, we decided to optimise for more significant AUC scores while also providing EER scores to compare with other papers. However, direct comparison with related work is challenging since the metrics are derived using differences such as gesture combining methods, data sub-setting, data cleaning, and user selection.

4.5.2 Feature super and subsets

While this chapter focuses on five feature sets from the literature, an evaluation of feature importance can be made to define a superset which combines the best n-performing features from each related work into a new feature set. Similarly, subsets can be made to eliminate noisy or poorly performing features. However, we took the first steps to compare the feature sets and leave these potential improvements to future work. We highlight that it may be faster to evaluate feature importance using our omni-directional approach since simpler models are faster to train and more straightforward to interpret. For example, new features could be engineered, such as splitting the gesture at the 20 percentile to better focus on the beginning of touch interactions.

4.5.3 Coordinate specific features

Most feature sets used in touch-based biometrics incorporate at least the start and stop (x, y) coordinate pairs as features. However, models relying on coordinate pairs may have a contextual limitation since they can be affected by the screen content. E.g., the placement of a button or other screen content that a user needs to click or when users may avoid covering the screen with their finger while reading. Furthermore, the size of a device may further affect these features despite normalising the coordinates according to the Dots Per Inch, as seen in [1]. This work shows that the BS [4] feature set performed well on single-gesture authentication while suffering when combining gestures. Interestingly, the BS feature set also contains the most coordinate-specific features. It may be better to engineer coordinate-independent features or lean towards the TA [1] feature set.

4.5.4 Incompatibility between the gestures combining method

It is challenging to compare results between the state-of-the-art, as the methods to combine gestures differ, e.g., training a model by combining the feature vectors before training [2] or averaging the predicted probabilities [78]. Thus, single-gesture performance should be reported to allow comparisons based purely on model performance, where, under perfect conditions, each gesture could be accurately predicted. However, since models are trained to generalise, it is also essential to examine combining gestures. This work averages the predicted probabilities of classifiers trained on single gestures and a rolling window between 1 (no averaging) and 20 gestures. Thus, the comparison of merging feature vectors before training is left for future work.

4.5.5 Comparing Omni vs bi-directional paradox

While the omni-directional model outperforms the traditional method, a direct comparison may be unfair as the underlying data differs. Specifically, a horizontal model is exposed to 100 gestures, while the omni-directional must learn the horizontal behaviour collectively from all 200 observations. Hence, our approach may have an advantage in generalisation, which could cause a better performance when combining gestures.

4.6 Conclusion

While the bi-directional models based on an ET classifier work for single-gesture authentication, our approach is comparable and superior when combining three gestures. Interestingly, single-gesture authentication works better using the behaviour captured by the BS feature set, but the TA feature set improves performance when combining gestures. Despite KNN and SVM-RBF being commonly used, they are inferior to the tree-based classifiers. We conclude that the omni-directional approach is preferable when using an ET classifier using the TA feature set and combining at least three gestures. Further, we suggest our hyper-parameter tuning method, providing a lower AUC standard deviation.

Chapter 5

Touch Encodings

5.1 Introduction

From 2022 to 2027, identity theft and fraudulent banking transactions are projected to increase, with costs to merchants exceeding $343 billion [128]. Widely popularised approaches such as multi-factor authentication provide the opportunity to increase the protection of user accounts but are often inconvenient [27], [28]. However, the FIDO Alliance recently proposed a passwordless approach, where users can replace passwords with an internal or external authenticator, such as mobile tokens [129]. Mobile tokens could be an Android smartphone with embedded biometric authentication or other applications and lock screen protection.

This chapter proposes TouchEnc, a passive and CA mechanism on mobile tokens that can extract personal gestures from finger movement recorded on touchscreens beyond the point of entry. Thus, CA captures and verifies behavioural biometrics and ensures user authenticity over time. This chapter presents a method to authenticate users exclusively by behaviour extracted from *on-screen* gestures. State-of-the-art performance is achieved by encoding touchscreen records from a public dataset [2] into images and cropping the essential screen area for automatic feature extraction. An example of a single signature and corresponding important screen area can be seen in Figure 5.2. In the example, a user drew touch points during downwards-moving navigation. Each touch point encodes behaviour using the

Red, Green, and Blue colour channels, which enables powerful image classification methods for further user authentication.

5.1.1 Motivation and Challenges

CA can alleviate user frustration when authenticating on mobile devices and has seen increasing interest from the research community looking to harness information from sensors and modalities such as accelerometers, gyroscopes, and location, among others [10], [68]. However, we motivate this research by focusing exclusively on the touchscreen behaviour, which depends on several factors and challenges, including (i) adequately engineered features [10], [130], (ii) personally selected features [71], and (iii) faster detection, e.g., not relying on multiple gestures [68], [106]. As concluded in Section 4.1.2, the older feature set proposed by [1] in 2012 performed just as well as four other feature sets proposed in the more recent and related work [130]. Thus, a better and perhaps more personal or automatic way to engineer features is required [71] as discussed in Chapter 3. This observation is similar to the first survey paper from 2016 [10], discussing the need for better features. While neural networks have been studied, as shown in Section 2.4.10, the deeper architectures and ability to automatically extract features remain unexplored [68]. Thus, this chapter is motivated by exploring and illuminating how well NN, particularly DL, can help classify users based on their touch inputs using deeper and more modern architectures, such as computer vision used for facial recognition [112] amongst others.

5.1.2 Research Questions and Contributions

Following the challenges outlined, this chapter seeks to contribute answers to the following questions:

1. Rather than manually engineering a fixed feature set to describe behaviour, how can modern computer vision and deep learning be used to extract personal behaviour automatically?

2. How does the proposed approach compare to other ML approaches regarding EER and the number of gestures required to authenticate users on mobile devices, and what are the limitations and opportunities of using this technique on novel and unseen users?

To answer these questions, this chapter introduces and overcomes the previous challenges with static features by contributing the following novelties. (i) Minimal transformation of raw touch data into image encodings, (ii) exploration of six plotting variations for the pixel values, and (iii) measuring the ability of DL models to extract behaviour and classify users based on the transformed encodings without relying on grouping gestures.

The chapter is structured as follows. Section 5.2 presents the related work and baseline research performances using the same dataset. Section 5.3 describes the proposed approach to convert tabular signature data into images, and Section 5.4 demonstrates the implementation. Section 5.5 presents the results before discussing the limitations and future work in Section 5.6 while concluding the work in 5.7

5.2 Related Work

In 2021, Frank et al. [1] demonstrated that touchscreen inputs could be used for CA. Soon after, Serwadda et al. [2] published one of the most popular datasets for touch-based CA. Further, they investigated the best classifiers through a unique feature set while differentiating between landscape and portrait modes and individually modelling vertical and horizontal gestures for each screen orientation. In [4], the authors defined a new signature direction as oblique, which occurs when a signature curves during a horizontal or vertical gesture. They also apply different feature sets to various directional models, including clicks, horizontal, vertical, and oblique gestures. Like [1], [2], each model is trained according to the drawn direction, and analysis shows that the best performance is derived from oblique gestures. However, comparing these works remains challenging since they utilise different data and feature sets, and the directional modelling approach varies. [130] studies these differences in directional modelling using the public dataset from [2] and five feature sets from the literature [1]–[5]. They conclude that models can be trained as one, independent of the signature direction. Further, the TouchAlytics [1] feature set appears to outperform recent work despite age.

This chapter focuses on the data published by [2] and Table 5.1 present a comparison of papers using this data. The focus ensures fairer comparisons with our work and avoids bias towards private data sets, which often perform better but are challenging to verify [77]. Table 5.1 also showcase the differences in the number of features used, the number of required gestures for accurate authentication, and whether results rely on multiple models for good performance. It is also noted that an increase in the number of users appears to cause a decline in performance, which is consistent with the findings by Frank et al. [1], [69]. Despite focusing on related work using the same data [2], it proves challenging to ascertain the number of users in other studies and their inclusion or exclusion criteria.

Table 5.1: Comparison of related works based on data set availability or using image features. In the 'Single Model (SM)' column, a ✓indicates that the work can be implemented with a single model, and numeric values represent the number of models required for implementation. The 'Number of Features (NF)' and the 'Gestures Required (GS)' columns describe the number of features and gestures necessary for the respective works to achieve performance.

Study	Data	Users	NF	SM	GS	EER%
[1]	[1]	41	28	✓	1	13.00
[16]	[16]	30	Image	4	6	4.31
[2]	[2]	106	28	4	10	15.50
[17]	[16]	78	Image	6	6	4.70
[115]	[115]	25	Image	5	N/A	‡40.60
[95]	[2]	N/A	5	8	10	18.50
[96]	[2]	N/A	5+28	8	10	6.98
[99]	[2]	N/A	112	4	> 4	7.86
[4]	[4]	45	4-16	1	9	$*$95.85
[103]	[2]	N/A	28	†✓	1	22.50
[102]	[2]	N/A	33	4	10	24.16
[100]	[2]	N/A	33	4	33	15.04
[106]	[1]	N/A	125	N/A	1	21.00
[105]	[2]	N/A	28	2	10	16.48
[130]	[2]	35	28	✓	5	17.90

† Only modelling horizontal gestures
‡ F1 score
$*$ Accuracy score

The work presented here employs a unique approach to automatically extracting touch features using an image-based method. Despite not using the data provided by Serwadda in 2013 [2], the only three other papers utilising images for touch-based CA [106] are briefly summarised. First, [16] proposed a Graphic Touch Gesture Feature (GTGF) to extract identity traits and classify users using a SVM. Later, they extend and improve their work using a Statistical Feature Model [17]. More recently, [115] proposed and applied a modified Edge Orientation Histogram (EOH) to extract ten features. Their features are then used to classify users using an SVM. To distinguish ourselves from these works, we employ three methods: (i) propose three intuitive scalar values as colour encodings, (ii) reduce computational requirements by cropping and focusing on a limited section of the drawing canvas, and (iii) effectively apply computer vision and DL for automatic feature extraction and classification. Thus, the computation required to compute n-features is replaced with a maximum of three Red, Green, and Blue (RGB) values, based on the raw touch data, together with data reduction from processing the entire screen to only the area where the user touches the screen. Consequently, a matrix of pixel values that highly optimised GPU hardware can effortlessly train on is produced. The matrix of pixels can be stored to inspect and identify how the model works visually, such as shown in Section 5.5.4. Thus, our approach's simplicity and enhanced performance could make it an attractive option for researchers looking to approach touch-based CA from a DL perspective.

5.3 Proposed Approach

While most related work focuses on feature engineering and extraction, this chapter takes a fundamentally different approach by encoding raw touch data into graphical images. For each drawn gesture, a user generates several touchpoints. Traditionally, these touchpoints are grouped per gesture and computed into numeric features representing time, direction, speed, and force, amongst others, as seen in [1]–[4], [130]. However, a fixed feature set may not work for all users since behaviour is personal

[71]. Instead, we demonstrate how graphical gestures enable automatic feature extraction using modern computer vision and DL to overcome the challenges of manual feature engineering. However, the question is *how* to represent drawn gestures as images and which NN/DL is better suited for automatic feature extraction.

The following subsections describe the dataset, user selection, and how drawn gestures are encoded into colour images.

5.3.1 Data Selection and Preparation

In this work, we utilise the public data set published by [2] since it contains both areas covered by the finger and pressure information for each touch point. Several other data sets are available, as presented in [106], but they do not qualify due to missing area, pressure values, or too few samples per user. The data from [2] is provided in two sessions, each separated by at least one day between data captures for each user. The data is also captured in landscape and portrait, but we focus exclusively on portrait since it is the most commonly used orientation [96]. Figure 5.1 presents the first ten users and the number of gestures recorded when drawing gestures in horizontal and vertical directions for their first session.

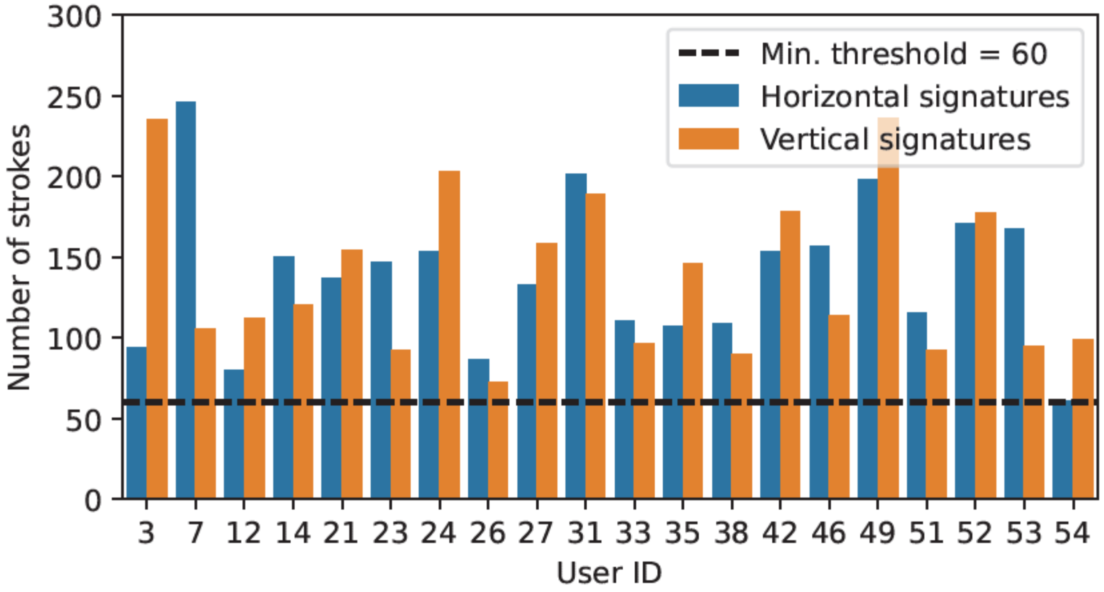


Figure 5.1: Comparing the number of gestures provided for the first ten users, given the direction of their drawn gestures. The threshold defines a lower boundary of required gestures to qualify for training.

Figure 5.1 illustrates that the number of gestures per horizontal and vertical direction varies among users. This may be due to personal preference or subconscious behaviour. Some users navigate shorter and more frequently, while others move quickly and may have longer pauses before drawing the next signature. Previous studies have typically flattened and restricted the number of gestures taken into account per user when training classifiers. [4], [71], [115]. We argue against this and instead opt for a minimum number of samples per user to qualify and ensure enough data is available to learn their gestures. Since related work finds 120 samples are required to perform well [4], we set a minimum threshold of 60 gestures in each horizontal and vertical direction. The criteria are applied for both sessions to allow data subsetting without affecting the minimum required number of gestures. Consequently, 74 users qualify out of the 106 in the data set.

We subset the data into training, validation, and testing to ensure no leakage between training and evaluation. The splits are grouped per user, session, and direction to respect the underlying distribution described in Figure 5.1. For each user, the last 20 gestures in each group are selected for testing, the previous 20 for validation, and the remaining for training. Thus, the validation and testing sets are balanced. Qualifying gestures must also have at least five touchpoints to be considered valid; otherwise, it is discarded as a click action [1], [2], [71], [106].

5.3.2 Data Cleaning and Cropping the Signature Canvas

Directional variations happen when users draw gestures on their device screen, e.g., swerving when scrolling down rather than drawing a straight line. These small directional changes expose the subconscious behaviour required to extract unique gestures. However, if a user changes their mind halfway through a gesture, a signature may become invalid since it deviates significantly from the intended direction. Thus, gestures where the moving average of the angle between five touchpoints' differs more than 90 deg are removed. Following data cleaning, a blank canvas with the maximum screen resolution is generated to accommodate drawing the signature.

Certain NN architectures can require significant memory when dealing with high-

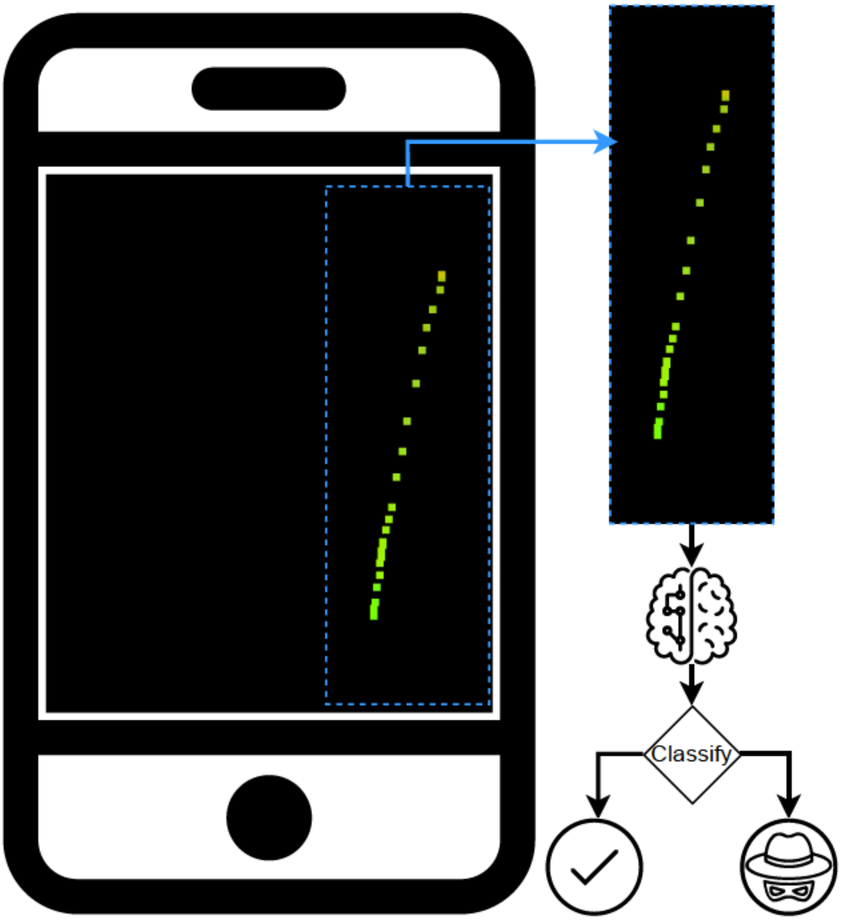


Figure 5.2: Example of extracting and cropping of visual touch to reduce image dimension for efficient training.

resolution images. Downsizing images to a lower resolution may seem like a solution, but it can result in signal loss and poor results. Instead, we propose cropping out the signature from the canvas as shown in Figure 5.2. Gestures are typically drawn within a small part of the screen, leaving a major part of the canvas empty. Therefore, we remove the empty parts and define a shared image dimension between horizontal and vertical gestures. We analyse the full dataset to determine the maximum screen area used in gestures and which cropping resolution captures the most. While cropping the signature may remove important location information, we address how to mitigate this issue in Section 5.3.3.

The visualisation in Figure 5.3 shows the distribution of maximum vertical gestures and the dimensions for x and y cropping. To define the cropping dimension, we use the 90^{th} percentile on each axis, which helps to eliminate outliers where

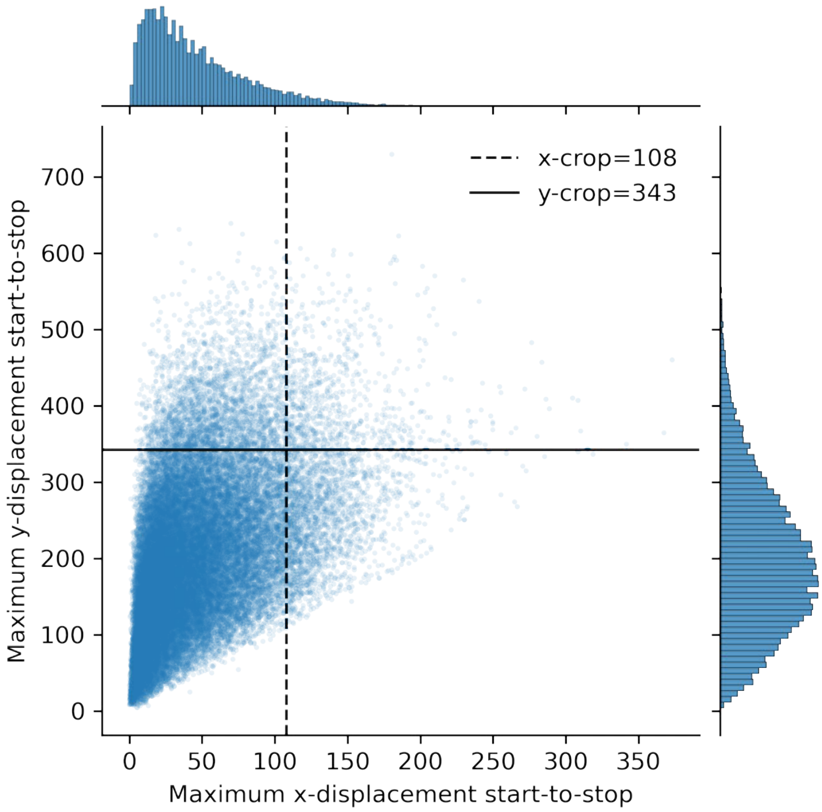


Figure 5.3: Scatter plot and distribution of the maximum touch displacement for each gesture for the vertical gestures. Two lines signify the 90^{th} percentile used to guide minimum cropping lines. Horizontal gestures have a similar distribution but are omitted for brevity.

users' gestures swerve excessively. We perform outlier analysis on horizontal and vertical gestures but exclude the horizontal figure for conciseness since the distributions are similar. It's important to mention that when working with horizontal gestures, the y-crop becomes the x-crop since the orientation and longest axis are swapped, and vice versa for the shorter axis. The outlier removal causes a minor data loss, resulting in 5,535 out of 31,432 gestures being dropped for horizontal and 7,333 out of 42,473 for vertical. Since image classification often requires the same image dimension, each horizontal signature is rotated 90 degrees counterclockwise to align vertical and horizontal gestures. Consequently, each extracted signature will produce the exact image resolution and orientation independent of the underlying signature direction.

5.3.3 Biometric Colour Encodings

As users interact with their device screens, the operating system records multiple touchpoints for each gesture, creating a unique signature. An example of such a signature is in Figure 5.2. The raw touch data, including the x and y coordinates, associated “pressure”, “area covered”, and “timestamp”, are obtained from [2]. While this information has traditionally been used to calculate ML features, we suggest reducing the demand to compute 20-50 features manually and instead encode the raw touch data into three colour encodings, producing a matrix of RGB values with a fixed dimension. Thus, each touchpoint is represented by three values in the RGB matrix, which can be processed by computer vision models and visualised for human interpretation. However, the choice of what each value represents can vary, and the following section further describes this aspect.

A square box is plotted on the canvas to represent each touchpoint using the coordinates of the x and y. This box is then scaled according to the raw value of the “area occluded by the finger” and coloured RGB in the range 0-1 or 0-255. The red colour encoding represents “pressure”, where lower values correspond to brighter red colours and higher values to harder pressure coloured full red. However, to allow cropping of the images, we suggest encoding the original canvas location by computing each touchpoint displacement using equation Equation (5.1) from the screen origin (0,0) and using the values for the green colour. The continuous “timestamp” value makes scaling and encoding within the colour range a challenging task. For encoding the timestamp, we recommend utilising either Equation (5.2) or Equation (5.3) to calculate the acceleration or velocity between touchpoints, which indirectly captures the time domain.

$$\text{displacement} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{5.1}$$

$$\text{acceleration} = \frac{\Delta\text{velocity}}{\Delta\text{time}} \tag{5.2}$$

$$\text{velocity} = \frac{\Delta\text{displacement}}{\Delta\text{time}} \tag{5.3}$$

Figure 5.4 presents the different distribution after zero mean scalings and normalises values between 0-1. Although acceleration and velocity may rely on the same raw data, we see in Figure 5.4 that the distributions differ from the normal nature of pressure and displacement. Consequently, the following section investigates the difference between two RGB combinations. The first contains Pressure, Displacement, and Acceleration (PDA), and the second contains Pressure, Displacement, and Velocity (PDV) while also proposing different plotting styles.

5.3.4 Plotting Styles

When plotting, we utilise the original screen size as the canvas and align the signature to the centre of the image while the padding is computed to reach the required cropping dimensions. The motivation behind the RGB colour channels involves capturing muscle information from pressure, screen area preference by displacement, and indirectly associating time movement through acceleration or velocity. Further, the distinct locations between each x and y coordinate pair of consecutive touchpoints can describe motorical identifiers regarding their alignment, distance, and time. Based on prototyping, we found that cropping gestures into smaller im-

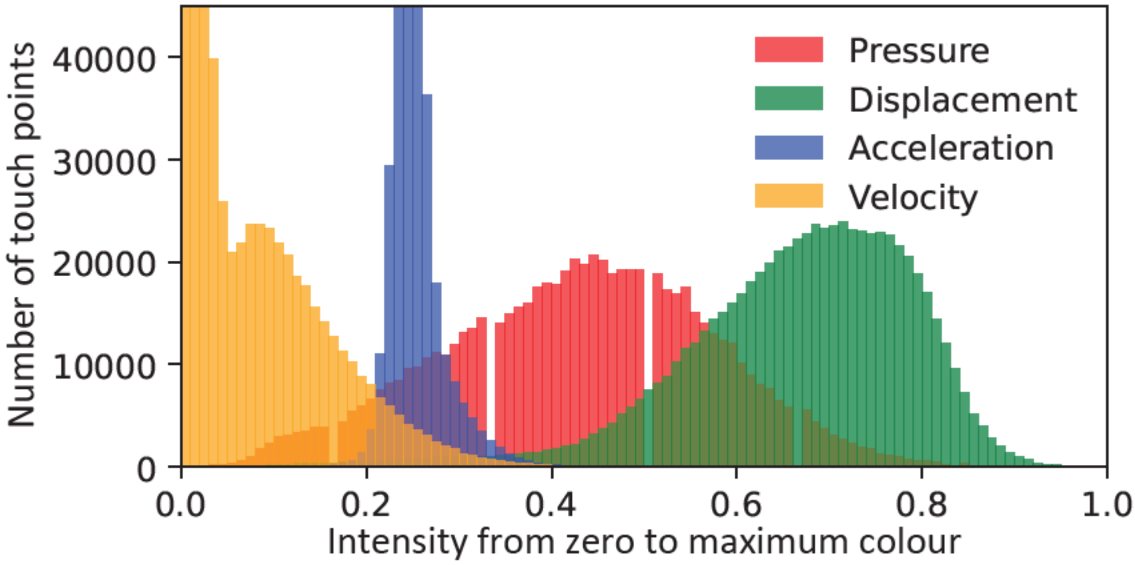


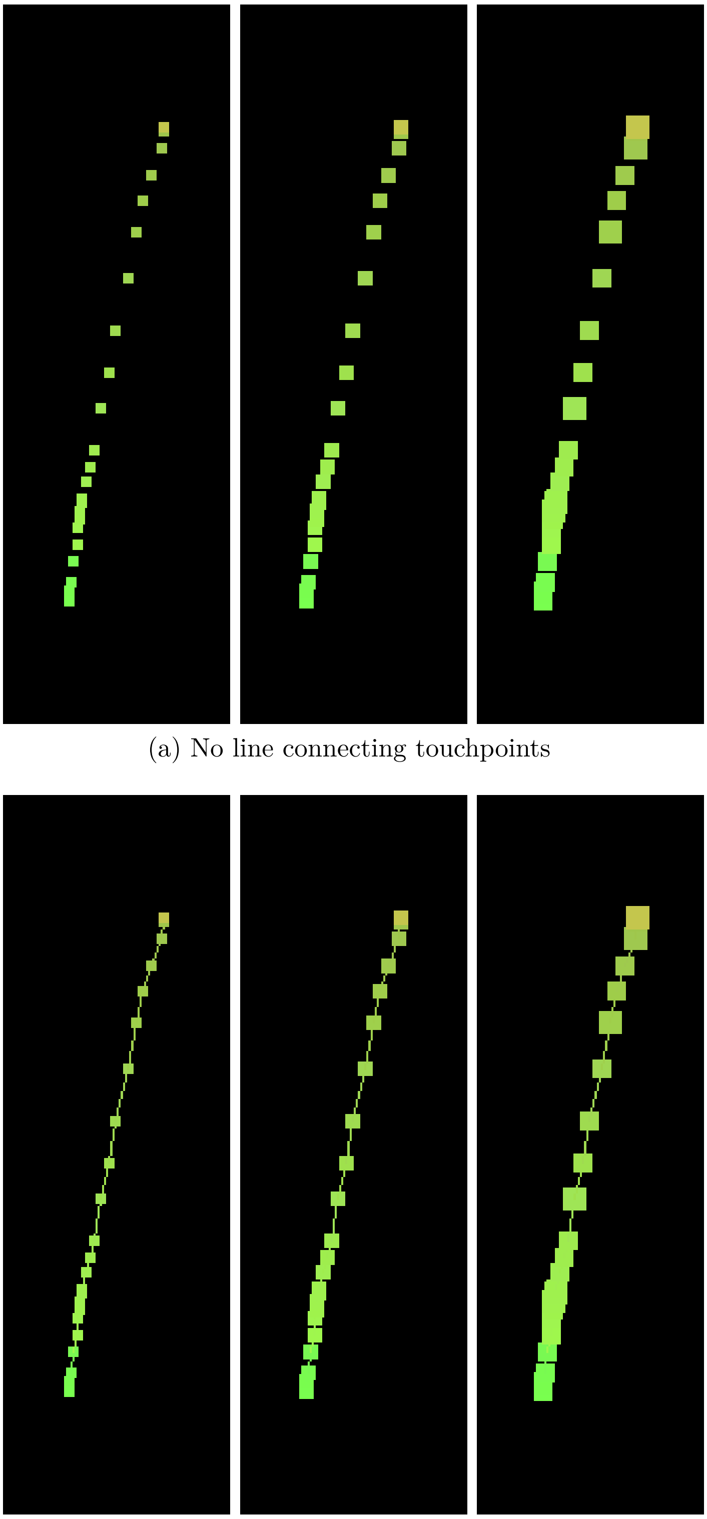
Figure 5.4: Distribution of colour encodings.

ages using a shared cropping size yielded the best performance with the benefits of a smaller image for the neural network to process, fewer empty pixels to consider, and reduced file size.

When assigning colours to each touchpoint, we must consider the occluded area of the screen caused by the finger. Our approach involves encoding RGB colours in a square box proportionate to the area. However, the raw area data is reported as a scalar value between 0 and 1. Thus, the area values must be scaled to whole pixel values to be visually detectable. We scale the area by multiplying the values by 5, 10, or 15. For example, if the phone reports an area of 0.2 scaled by 10, we create a square with 2x2 pixels and colour it according to the RGB touch encodings. Furthermore, this study also explored the impact of drawing connecting lines between touchpoints to determine the potential of increasing the neural networks' performance by extrapolating information between touchpoints. As such, we plot a variation for each area scale with and without connecting lines and train several image classifiers on the different plotting styles. Nonetheless, if the points are dense, a more significant scaling factor could cause the boxes to overlap and potentially lose some of the unique signature behaviour. Examples of this can be seen in Figure 5.5b, where dense touchpoints merge, and the lines connecting points become occluded by the boxes. Paradoxically, a network may find detecting and bringing attention to larger touchpoints easier, eliminating the benefits of the connecting line.

5.4 Implementation

To examine the effectiveness of image classification methods in extracting suitable features from TouchEnc encodings, we test the six proposed image variations illustrated in Figure 5.5. Our user classifiers are built using the PyTorch DL framework [131], which offers a range of well-researched neural network architectures. Given the focus on the encodings intended for mobile devices, we opted for image classifiers specifically designed for lower computational resources, such as the MobileNetV3



(a) No line connecting touchpoints

(b) With line connecting touchpoints

Figure 5.5: Example of the same gesture plotted using different variations of area scaling and line styling

(MNV3) [132] with 1.5mill parameters and a larger EfficientNetB0 (ENB0) [133] with 4.1mill parameters. We chose the smallest size for each architecture option in this study to conserve training time. The loss function for all models uses cross-entropy and is optimised with AdamW [134].

To ensure effective training, a modern approach inspired by [113], [135], [136] is adopted, which includes learning rate annealing after a short linear learning rate warm-up. This concept initially helps speed up convergence and mitigates large weight updates as the learning rate increases, while annealing combats issues where the optimiser may get stuck at a certain learning rate. Additionally, to prevent overfitting, Label Smoothing [137], Weight Decay [134], and Random Erasing [138] techniques are applied. Since we are interested in classifying each user, we utilise the Softmax activation function to produce multi-class probabilities that are suitable for AUC scoring as described in Section 2.2.5. For computing class-specific AUC scores, we employ the OvR [73] user classification method [73] and set a threshold individually to optimise EER scores. During training, we keep track of the macro-averaged AUC score for each epoch and store the model checkpoints for up to 50 improving epochs. However, training is stopped if the validation AUC score doesn't improve over ten epochs; at this point, the last model checkpoint is restored.

The last phase of the implementation involves conducting a hyperparameters search to determine the optimal touch encoding and plotting styles. Additionally, an exploration of whether using an ENB0 can improve performance on the selected encoding will be explored. Table 5.2 outlines a shared parameter grid used in the experiments to aid this process. While the static values in this parameter grid are inspired by the [113], [134]–[137], the variations in Learning Rate (LR) and Batch

Table 5.2: Hyper parameter used in the grid search

Parameter	Search space
Area Scale (AS)	5, 10, 15
Line Style (LS)	with, without lines
Learning Rates	1e-2, 1e-3, 1e-4
Linear Warm-up	5 epochs
Cosine Annealing [136]	Maximum Epochs
Weight Decay [134]	0.05
Label Smoothing [137]	0.1
Random Erase [138]	0.25
Pre-trained Weights [139]	False
Batch Size	32, 64
Maximum Epochs	50

Sizes (BS) are tuned in common ranges that can affect model convergence. The maximum BS allows any researcher with a GPU with 10 GB of memory to run the experiments, enabling result verification without access to expensive hardware. Detailed model performance can be inspected in [140]. The grid is initially implemented with the MBV3 model, and based on the results, the best encoding will be used with the ENB0 model to assess whether performance increases with a more complex architecture. When evaluating the validation set during training, the best parameters for any model are chosen based on the highest macro-averaged AUC score. Optimising for better AUC scores is effective since it improves overall performance independent of the classification decision threshold [72]. Consequently, we compute 72 MNV3 and 36 ENB0 models due to the search space. The following section presents the best five models for each grid search and extends the evaluation by combining gestures for better comparison against Table 5.1.

5.5 Evaluation and Results

Section 5.3.3 describes the PDA and PDV encodings used to train MNV3 models. Table 5.3 displays the top ten results, revealing that Acceleration outperforms Velocity due to the latter's bimodal nature. Therefore, PDA is selected as the superior encoding method. Our TouchEnc technology has yielded significant improvements in single-signature authentication, with a 23 per cent increase compared to the best single-signature result of 13 per cent [1]. This improvement is measured over 74 users, which is 33 more users than evaluated in [1]. Table 5.3 also presents the modelling time, which may be a crucial factor when deciding between the results in the following section, where the ENB0 results are presented.

5.5.1 EfficientNet Improvements

While the MNV3 performs well, the optimal encodings may improve performance in tandem with larger and more complex models such as EFB0. Table 5.4 presents the top five results when training an EFB0 model with the TouchEnc PDA encodings. The results show that the best models converge at common parameters, such as the Batch Size (BS) and Learning Rate (LR), with stable performance independent of the plotting style. We highlight these results are based on verifying users by analysing gestures individually. Hence, the results are conservative since most related works offer their performance by aggregating gestures. While the performance has increased from the MNV3, so has the time to model. This is a natural trade-off between complexity and performance, which could be an interesting area to further study. Regardless, the best ENB0 model increases the TouchEnc performance further compared to Table 5.3 with a 43 per cent improvement over [1] when authenticating individual gestures.

Each model is reported with the corresponding AUC score as part of the results. The values are related to the ROC, which explains the model performance as a function of different thresholds. Thus, Figure 5.6 visualise the ROC curve for the best ENB0 model and compares the validation and testing results for the model. A concern could emerge if the curves are significantly different with indications of over or under-fitting. Judging by the plot, the EFB0 model generalises well to the unseen testing data. However, the standard deviation suggests that certain users are more easily classified than others. Additionally, users can prioritise reducing false positives or negatives, but doing so may come at a cost in user experience,

Table 5.3: Top ten best performing MNV3 models when comparing Pressure, Displacement, and Acceleration (PDA) versus Pressure, Displacement, and Velocity (PDV) image-encodings. Line Style (LS), Area Scale (AS), Learning Rate (LR), Batch Size (BS), and Time in Seconds.

Enc	LS	AS	LR	BS	EER	AUC	Time
PDA	✓	15	.001	32	.103	.953	5852
PDA	✗	15	.001	64	.104	.953	2450
PDA	✗	15	.001	32	.104	.953	4715
PDA	✓	10	.001	32	.106	.952	4990
PDA	✓	15	.001	64	.108	.949	2857
PDA	✓	10	.001	64	.109	.946	4373
PDA	✓	5	.001	32	.110	.949	5297
PDV	✗	15	.001	32	.111	.948	5064
PDV	✓	15	.001	32	.112	.947	4783
PDA	✗	10	.001	32	.112	.948	4820

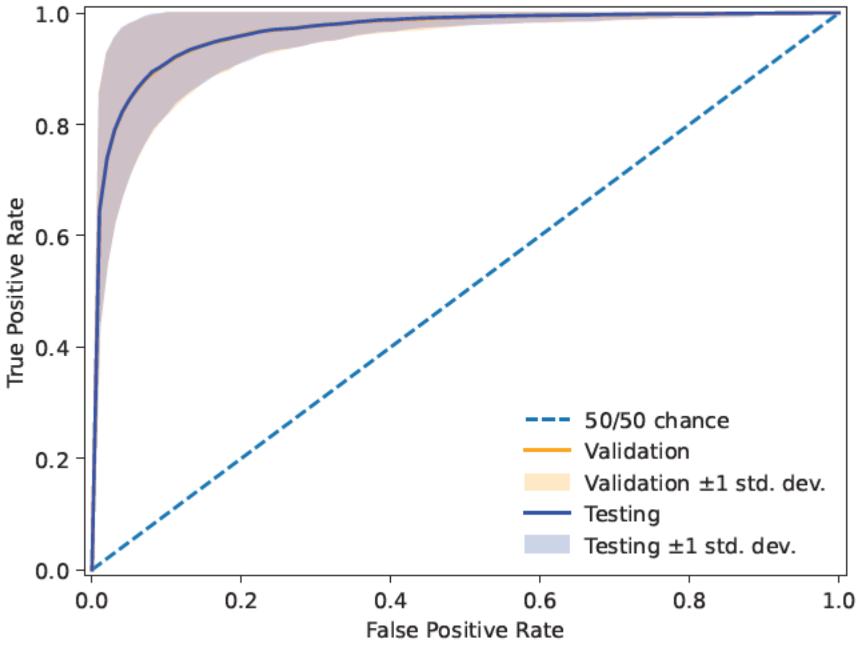


Figure 5.6: ROC plot showing the best performing Efficient Net according to the lowest EER score and highest macro-average AUC score

such as being mistakenly granted or denied access.

5.5.2 The 'Best' Plotting Variant

The best-performing model is an ENB0 since it trains reasonably fast and outperforms the MNV3. However, note that any of the six plotting variations presented in Figure 5.5 are applicable, although some perform better than others in specific contexts. For example, while the MNV3 are generally faster and cheaper to train, they are also more sensitive to lower AS, with a preference for scaling 10-15 times and

Table 5.4: Top ten best performing EFB0 models when training exclusively with the PDA encodings. Line Style (LS), Area Scale (AS), Learning Rate (LR), Batch Size (BS), and Time in Seconds.

LS	AS	LR	BS	EER	AUC	Time
x	15	.010	32	.084	.967	12027
✓	10	.010	32	.086	.965	16265
✓	15	.010	32	.087	.966	12922
x	5	.010	32	.088	.964	12002
✓	5	.010	32	.088	.964	15156
x	10	.010	32	.089	.964	12417
x	15	.001	32	.089	.963	7930
x	10	.010	64	.090	.962	8146
✓	15	.010	64	.091	.963	7665
x	10	.001	32	.093	.961	8335

benefiting from connecting the touchpoints. On the contrary, the ENB0 performs well in almost any plotting style but appears to converge faster with larger AS and no connecting touchpoints. Despite the difference in performance, single-signature authentication can be insufficient for some users, and the next section, therefore, presents the performance when combining n Number of Gestures (NS)

5.5.3 Single Versus Multi-Signature Authentication

The best model architecture can be defined in several ways, e.g., by the highest AUC, accuracy score, or lowest EER score. The AUC score provides a model that performs well regardless of thresholds, and the highest AUC score is achieved using the EFB0. However, the evaluation has focused on single-signature authentication, while most related work combines gestures, as seen in Table 5.1. As such, we implement an average moving window of 2 gestures, then 5, 9, and 10 gestures, to illuminate how well TouchEnc compares against the state-of-the-art performance against the

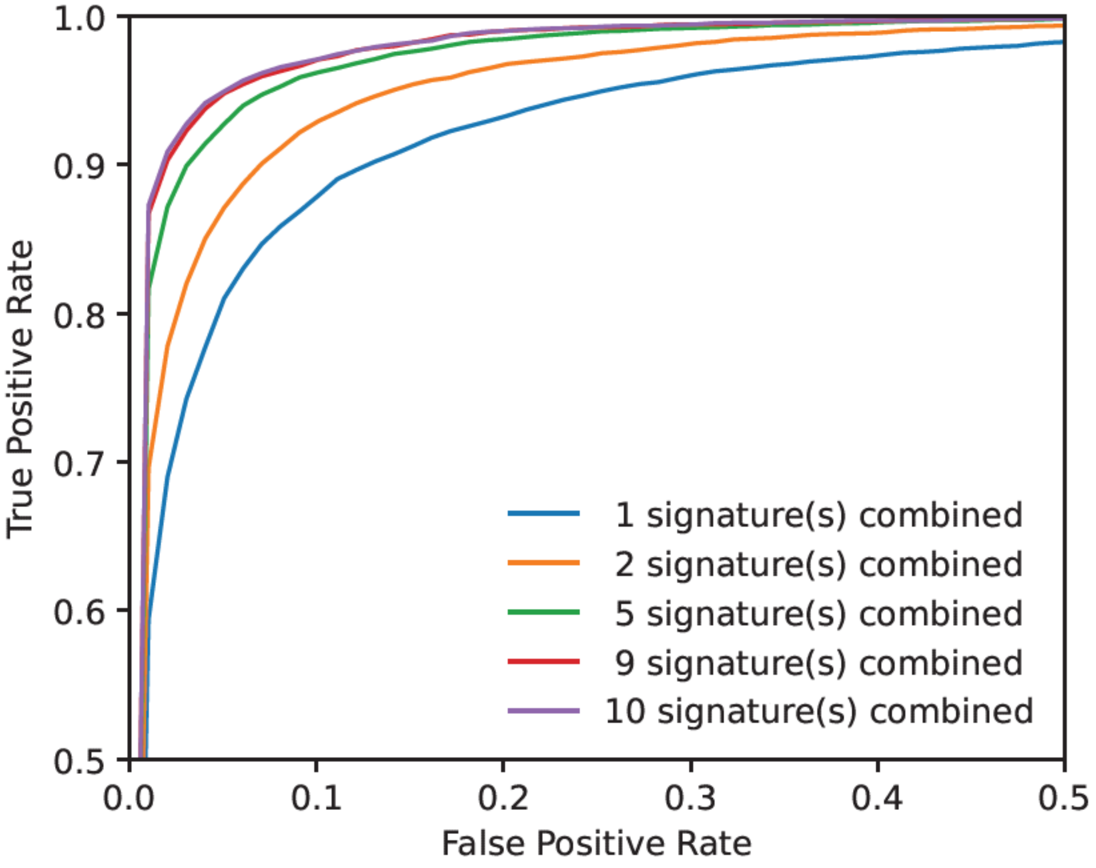


Figure 5.7: ROC when combining gestures

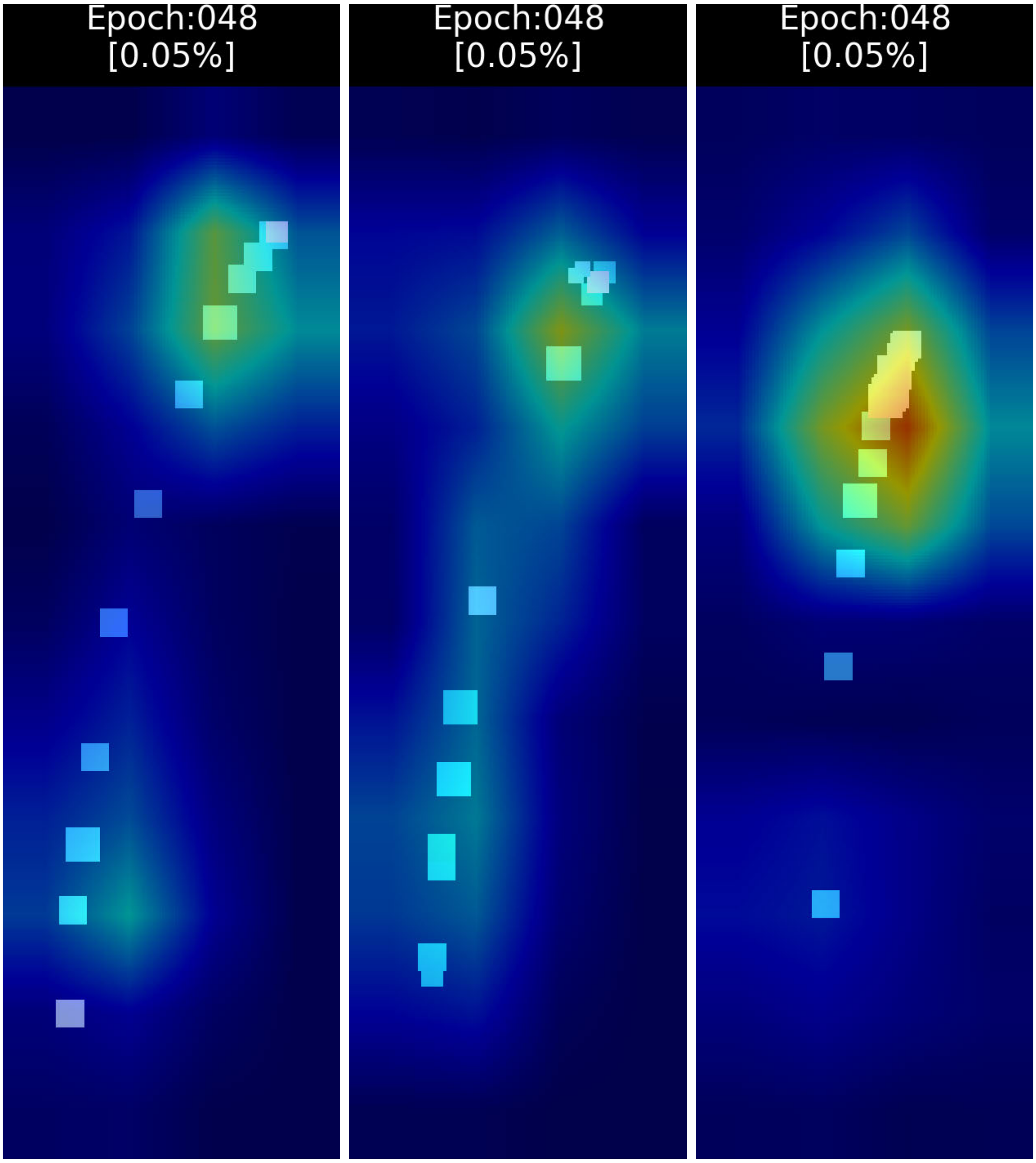
Table 5.5: Performance when combining n Number of Gestures (NG) using a rolling mean over the decision probabilities for testing data.

NG	AUC	EER	ACC	FAR	FRR
1	.97 (.03)	.08 (.05)	.92 (.05)	.08 (.05)	.08 (.05)
2	.98 (.02)	.06 (.04)	.94 (.04)	.05 (.04)	.06 (.04)
5	.99 (.01)	.04 (.03)	.96 (.03)	.03 (.03)	.04 (.03)
9	.99 (.01)	.03 (.03)	.97 (.03)	.02 (.03)	.03 (.03)
10	.99 (.01)	.03 (.03)	.97 (.03)	.02 (.03)	.03 (.03)

related work. The results are shown in Table 5.5 where NS is the Number of Gestures aggregated. In this work, we aggregate using moving average windows over the predicted probabilities, similar to others [1], [71]. When $NS = 1$, no gestures are aggregated, such as in Figure 5.6. Generally, a model with good single signature performance is also expected to perform well when combining gestures, and this behaviour is visually presented in Figure 5.7. The figure shows that our best ENB0 model and our automatic feature extraction approach are superior to the work of others. In the case of combining five gestures, we achieve 4% EER compared to [99], which achieves 7.86% EER. That is a 65% improvement, with diminishing improvements when aggregating more gestures. [130] found similar diminishing returns but needed more gestures before the performance converged.

5.5.4 Explaining the TouchEnc Attention

Since the Efficient Net performs well and trains fast, we recommend and use the ENB0 architecture to analyse and present Figure 5.8, which shows a GradCam [141] analysis of the activation maps for three upwards-moving gestures drawn by the same user, in sequence. As shown, the network has automatically extracted features and given attention to the touchpoints along the trajectory. As with many deep learning models, explaining why particular activations appear can be challenging. For example, it is peculiar to see Figure 5.8a appear to have skewed attention towards the right side of the first touchpoint. Still, a pattern can be observed relating to the increased attention given to the middle of the signature and further up compared to the lower part, where the finger would have started the upwards



(a) Gesture 1 (b) Gesture 2 (c) Gesture 3

Figure 5.8: GradCam [141] visualisation of activation maps using the best performing Efficient Net for automatic feature extraction

move.

5.6 Limitations and Future Work

This work uses two architectures with different parameter sizes of 1.6 Mill, 4.1 Mill, MNV3, and ENB0. While these architectures are commonly used, larger and more complex architectures could yield better results. It would also be interesting to experiment further by designing custom architectures or applying other off-the-shelf models, such as Swin Transformers [135] or ConvNeXt [113]. Thus, we recognise that the performance measures are conservative while still providing evidence of the feasibility of our method to encode gestures into images.

5.6.1 Deep Metric Learning

For this chapter, the feature outputs are optimised using Cross Entropy loss and evaluated using Softmax to demonstrate that automatic feature extraction is possible using our image transformation technique. Consequently, the probabilities are constrained for the learned users in our multi-class one-vs-rest scenario. However, such an approach is unrealistic for deployment, where gestures are available only for the valid owner of a device. Fortunately, deep metrics can also be mined from these images. Our next area of study is demonstrating the effectiveness of deep metric learning using our approach, which could enable one-class zero-shot learning of novel users.

5.6.2 Optimal Encoding and Transformation

While our images encode the drawn gestures effectively, different raw data could be encoded into the images to improve the feature extraction. In this work, we rule out *velocity* and replace it with *acceleration*, but other raw data points may be better. Furthermore, the image dimensions are fixed, but different sizes could allow further improvements. Perhaps a minimum x and y displacement are required. Lastly, different channel depths could improve the gestures by going hyper-spectral, E.g., encoding accelerometer force into a fourth or fifth colour channel.

5.7 Conclusion

Mobile devices are often used in friendly and hostile environments in small bursts. As such, most users rely on lock screen protection as a one-off point of entry check, including biometric fingerprints or facial identification. However, this work demonstrates a new method to conveniently and passively authenticate users by *how* they draw rather than what they draw over time using TouchEnc continuously. We have improved the state-of-the-art performance by shifting from a traditional ML-based approach and proposing converting touch gestures into images. We encode touch pressure, displacement, and acceleration into RGB colour channels, enabling off-the-shelf models such as Efficient Nets to extract behavioural features automatically. We achieved 96.7% AUC when authenticating users based on a single signature, which improves when aggregating gestures, with performance reaching 99% AUC. Lastly, our approach opens the door to exploiting other benefits of computer vision, such as deep metric and zero-shot learning.

Chapter 6

Conclusions and Future Work

This chapter draws together and presents an overview of the thesis contributions in Section 6.1. The following sections define the outcomes of each research objective based on Chapters 3 to 5, corresponding to the chapters describing each contribution. Finally, Section 6.2 discusses the future direction regarding each contribution and any overlap between the papers.

6.1 Overview of Contributions

To gain deeper insights into touch-based CA, Chapter 2 presents an extensive review of the related literature to establish the current state-of-the-art, uncover gaps, and define relevant research opportunities. While the research field is broad, this thesis maintains a narrow focus on exclusively modelling touch behaviour due to the presence of touch sensors on any modern devices, eliminating the need for extra hardware. With this focused approach, the literature review reveals three significant gaps: (i) the majority of existing studies concentrate on modelling users focusing on individuality rather than in a general sense; (ii) several directional models are typically required to achieve reasonable performance, along with complex parameter tuning strategies; and (iii), despite the advancement and benefits of DL, the majority of papers utilise traditional ML methods, often with limited and underperforming NN architectures. Addressing these challenges, Chapters 3 and 5 make significant contributions to the field, attempting to fill these research gaps and pave the way for new knowledge in touch-based CA.

6.1.1 Personalised Features

The first contribution in Chapter 3 begins by investigating the seminal work published by Frank et al. [1], the data, and the features released with their TA article. In the original and nearly all related work, authors typically model all users with equal weight to the behavioural features. However, the concept of touch-based CA is to model individuals according to patterns that can separate them from others. Thus, using the most descriptive features for each individual user should be an essential aspect of modelling unless all features are equally important. Consequently, Chapter 3 measures the performance of modelling users individually while utilising all behavioural features. Thereafter and more importantly, it explores the application of several feature selection algorithms applied to each user and classifier. The selection techniques include MI, SFS, SFFS, SBS, and SFBS.

While the MI selection technique is often used to evaluate features before modelling and implemented holistically over an entire dataset, including all users; thus, the methods do not assess the individual or the context of interaction among behavioural traits. For instance, an individual feature may become more descriptive in the context of another feature - commonly referred to as feature interactions. To ensure a fair evaluation of the original work, the investigation is carried out on the TA [1] public dataset with the original feature set. Results indicate that considering and selecting features for individual users improve overall authentication performance. Notably, the best selection technique was the SFS method, and Figure 6.1 visualises the selected features for each user using this approach combined with an SVM-RBF classifier. Two observations can be made from Figure 6.1. Firstly, features 27 and 28, “median pressure” and “median area covered by the finger,” are often chosen for most users. Secondly, most users can be distinguished using only a few features, whereas users 6 and 32 require nearly all features to achieve successful authentication.

6.1.2 Lesser Complex and Omni-Directional Modelling

Throughout the literature review, it has become apparent that researchers commonly employ various and different feature sets, often with small datasets containing few users, and which require multiple models to achieve reasonable performance. However, despite the demand for better and more feature engineering, new and distinct feature sets require rigorous evaluation against existing work to establish genuine progress. While some papers evaluate their approaches on multiple datasets, it remains uncommon to apply different feature sets, and there is no established default feature set for touch-based CA. In contrast to Chapter 3, this chapter is evaluated on a richer dataset after being granted access to the WVW dataset collected by Serwadda et al. [2]. This chapter's initial focus is an extensive investigation and comparison of five different feature sets within the context of the traditional modelling approaches, classifiers, and the proposed omni-directional method. Results show that several attempts to improve feature sets have not succeeded, as the best-performing feature set remains the earliest TA set released by Frank et al. [1].

Insights from the literature review indicate that tree-based classifiers demonstrated favourable performance. Thus, RF, GBC, and ET are implemented alongside KNN and SVM-RBF. Unsurprisingly, the three tree-based classifiers outperform the older, more traditional classifiers but require longer training time. Since descriptive

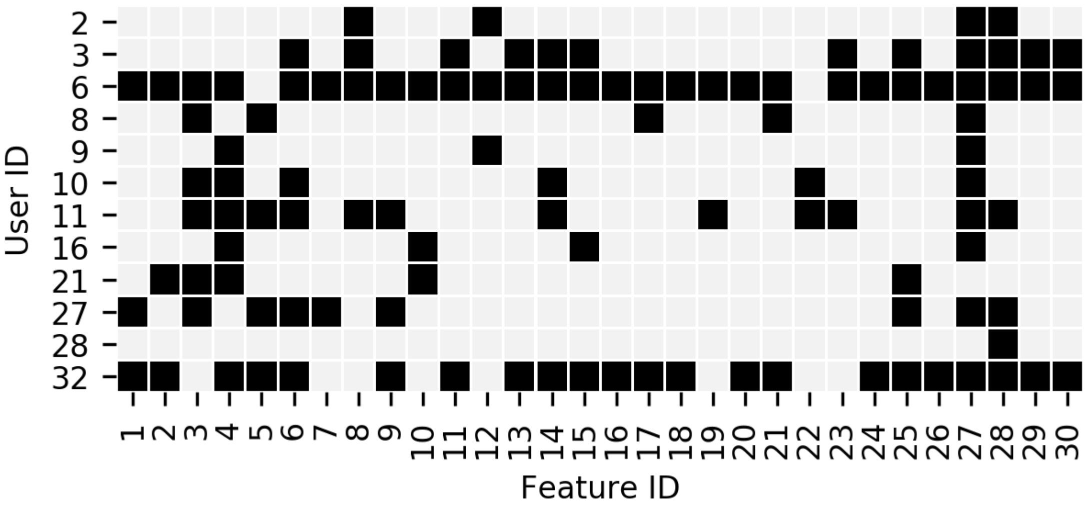


Figure 6.1: Selected features using SFS and SVM-RBF. Coloured boxed means the features are included.

features should accurately describe patterns, this chapter focuses on new features' ability to model users rather than relying on several direction-specific models. Thus, the analysis uses the traditional modelling approach, where horizontal and vertical gestures are grouped into different models and compared the performance to the proposed single omni-directional modelling method. The findings reveal that the performance of a single omni-directional model using a single gesture for authentication is similar to modelling two direction-specific models. Moreover, the omni-directional modelling approach outperforms the traditional bi-directional method when authenticating using multiple gestures.

As part of the proposed omni-directional modelling approach, the parameters are tuned in favour of lesser complex settings. The motivation is that tree-based classifiers train faster with fewer trees and are less prone to overfitting. As seen in Figure 6.2, the testing performance decays with the most complex parameters and the training time increase from 0.5 to almost 2 seconds in the worst case. In conclusion, examining various feature sets, modelling approaches, and classifiers has highlighted the opportunities for the omni-directional modelling approach to improve the performance of touch-based CA systems. The study emphasises the need for rigorous evaluation of newly engineered feature sets and the potential advantages of the omni-directional modelling approach. Besides comparable performance using the omni-directional model, examining and measuring the implementation of newly engineered features would be more straightforward as they only need to be evaluated using a single model, which is faster and more efficient to train.

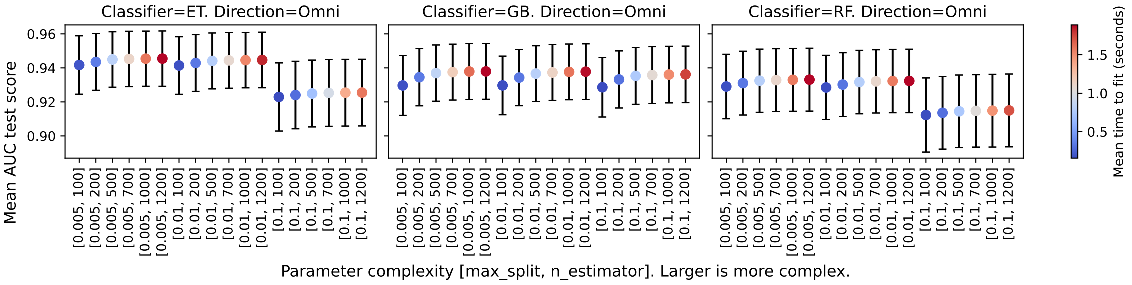


Figure 6.2: Mean AUC scores in the context of parameter complexity and coloured by the time to train models in seconds.

6.1.3 Superior Performance using Touch-Encodings

Finally, Chapter 5 shifts direction from MLby proposing the 'TouchEnc' method, a novel approach designed to facilitate the transformation of raw touch data into image encodings. This innovative technique leverages three colour channels, enabling computer vision to learn and extract automatic and personalised features automatically. Emphasis is placed on defining an efficient and optimal image dimension conducive to fast training and applicability on mobile devices while ensuring user authentication independent of user interactions. To achieve these objectives, an experimental design is defined to optimise and investigate model training through hyperparameter tuning, focusing on evaluating performance across various plotting styles, including scaling the touch area and connecting touchpoints with visual lines. The performance is measured and reported for single and multi-gesture authentication scenarios.

The chapter commences with an in-depth analysis of the related work, wherein attempts to implement shallow NNs with manually engineered features yield suboptimal outcomes. In light of this observation, the proposal to encode behaviour into images arises, capitalising on the innate feature extraction capabilities of CNN and Deeper architectures. The subsequent stages entail defining a canvas size based on smartphone screen dimensions, cropping touch gestures from the images to reduce the image size effectively, and establishing the input layer size of the initial convolutional layer. Notably, the transformation process incorporates three colour channels, red, green, and blue, to encode touch behaviour. The varying colour intensities reflect distinct levels of touch activity and are transformed into intensity values ranging from 0 to 255. Additionally, the investigation explores potential encodings, such as pressure, displacement, acceleration, and velocity, to ascertain their efficacy in enhancing feature representation. To assess the impact of different plotting styles on model performance, empirical evaluation entails the examination of (i) scaling touchpoints to represent the finger's area coverage accurately and (ii) drawing lines that connect touchpoints. Moreover, MobileNet and EfficientNet are adopted as off-the-shelf computer vision models for automatic feature extraction in their most compact variants.

From the experiment, several findings support the adoption of TouchEnc and further research using the encodings for automatic feature extraction. Notably, reducing image canvas dimensions from the entire device screen size to more manageable dimensions causes only 10 per cent of gestures to be dropped. Since the approach still outperforms the related work, this underscores the efficiency of the proposed method in preserving essential behaviour while concurrently reducing canvas size. Furthermore, the investigation reveals that scaling of touchpoints tends to diminish the significance of connecting lines between touchpoints, thereby presenting a noteworthy trade-off between scaling and preserving spatial relationships between touchpoints. Comparative analysis of the employed computer vision models demonstrates that while MobileNet exhibits quicker training, EfficientNet ultimately excels in accuracy and performance.

The models' credibility and informed decision-making process are corroborated through the insightful GradCam analysis, providing concrete evidence of the NN's focus on gestures rather than random guesses. Additionally, the combination of pressure, displacement, and acceleration as encodings prove superior, outperforming pressure, displacement, and velocity in enhancing the model's efficacy. Remarkably, the application of 'TouchEnc' for single gesture authentication yields an impressive EER of 8.4%, which compares favourably to related works that typically combine ten gestures, resulting in an EER of 3.1% using TouchEnc. Such compelling results highlight the superiority of TouchEnc in gesture-based authentication when contrasted with existing methods EERs ranging from 15.5% to 24.16% using ten gestures. The substantiated effectiveness and potential applications of TouchEnc underscore its invaluable contribution to enhancing touchscreen-based interactions and authentication mechanisms.

6.2 Limitations and Future work

While Chapters 3 to 5 each contributes to the knowledge and advancement of touch-based CA, they also carry limitations to help scope the work as part of their singular contributions. This section describes the major limitation of each work, while the last subsection presents a suggestion for a potential future research direction based on this thesis's findings.

6.2.1 Personalised Feature Sets

Access to the WVW [2] dataset was unavailable at the time of conducting the first experiment, causing the evaluation to be limited to a pilot study of few qualifying users from the TA [1] dataset. As such, the limitation of the first contribution relates to the number of available users and the potential for more significant insights when analysing more users. In a recent study, Georgiev et al. [106] also used the TA dataset. Similar to this contribution, they also model a subset of users. Specifically, they use 15 of 41 users. Moreover, they highlight the same challenge in their own dataset. Despite collecting data from 470 users, only 64 qualified, and the remaining 406 supplied insufficient data. As such, it is common to subset and define a protocol of minimum data to qualify for modelling.

6.2.2 Feature Engineering and Super Setting

Since features are personal, it may be appealing to start engineering new features. But, a more practical choice would be to first investigate the current feature sets with an aim to simplify modelling, such as with the proposed omni-directional approach. However, applying the personalised feature selection from Chapter 3 would be interesting to further understand whether improvements can be made using the single model and individually selecting features per user. Also, considering the increase in total features across the five sets, combining the best features from each could also work as a super-set, similar to the suggestion by Georgiev et al. [106]. However, this contribution is limited to justifying the omni-directional modelling approach, which can be further used to test new features quickly due to faster and cheaper modelling costs. Unfortunately, manually engineering features is impractical at scale; thus, the last contribution attempts to mitigate this limitation rather than continue feature engineering, selection, and analysis.

6.2.3 Image Encodings

By transforming the raw touchscreen input into images, the ability of computer vision and deep learning enables automatic feature extraction. However, the raw data still needs to be transformed and projected into images that are quick to process on low-end hardware such as smartphones. As part of this first step towards generating touch encodings, the choice of pressure, displacement, and acceleration is suggested as three colour channels. However, further information, such as accelerometer motion, can be encoded into high spectral images with more than three colour channels. Unfortunately, no datasets are available to conduct such experiments.

6.2.4 Future Direction using Embeddings

Because authentication should be designed to function with an unknown number of users, exploring the avenue of one-class classification using traditional ML techniques and manual features could be intriguing. However, developing new features that cater to an unknown number of users, each with unpredictable or different behaviour, may not be advisable. Instead, the Touch Encodings introduced in Chapter 5 offer a more practical approach, as they can be leveraged to extract embeddings for distance comparison.

As such, a promising research direction would involve the utilisation of Touch Encodings and few-shot learning techniques to extract embeddings. For instance, in [112], a 128-bit embedding is derived from facial images, and an overarching model is trained using a triplet loss and various mining strategies [142], [143]. During authentication, a comparison of distances between the known owner and new inputs can be established using a threshold similar to probabilities. In other words, the closer the distance, the higher the likelihood that the user is the same.

In addition to the triplet loss, other loss functions like the Circle loss [144] or Proxy-based losses [145], [146] could also be intriguing alternatives. Regardless of the chosen loss function, the Touch Encodings would serve as the input for extracting deeper metrics to authenticate users.

References

[1] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, Oct. 16, 2012. DOI: 10.1109/TIFS.2012.2225048.

[2] A. Serwadda, V. V. Phoha, and Z. Wang, "Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms," in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, Arlington, VA, USA: IEEE, Sep. 29, 2013, pp. 1–8. DOI: 10.1109/BTAS.2013.6712758.

[3] Z. Syed, J. Helmick, S. Banerjee, and B. Cukic, "Touch gesture-based authentication on mobile devices: The effects of user posture, device size, configuration, and inter-session variability," *Journal of Systems and Software*, vol. 149, pp. 158–173, Mar. 2019. DOI: 10.1016/j.jss.2018.11.017.

[4] Y. Yang, B. Guo, Z. Wang, M. Li, Z. Yu, and X. Zhou, "BehaveSense: Continuous authentication for security-sensitive mobile apps using behavioral biometrics," *Ad Hoc Networks*, vol. 84, pp. 9–18, Mar. 2019. DOI: 10.1016/j.adhoc.2018.09.015.

[5] Y. Cheng, X. Ji, X. Li, *et al.*, "Identifying child users via touchscreen interactions," *ACM Transactions on Sensor Networks*, vol. 16, no. 4, 35:1–35:25, Jul. 28, 2020. DOI: 10.1145/3403574.

[6] L. O'Gorman, "Comparing passwords, tokens, and biometrics for user authentication," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2021–2040, Dec. 2003. DOI: 10.1109/JPROC.2003.819611.

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

[8] I. M. Alsaadi, “Physiological biometric authentication systems advantages disadvantages and future development a review,” *International Journal of Scientific & Technology Research*, vol. 4, no. 8, pp. 285–289, 2015.

[9] J. Daugman. “How the afghan girl was identified by her iris patterns.” (2012), [Online]. Available: `http://www.cl.cam.ac.uk/~jgd1000/afghan.html`.

[10] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello, “Continuous user authentication on mobile devices: Recent progress and remaining challenges,” *IEEE Signal Processing Magazine*, vol. 33, no. 4, pp. 49–61, Jul. 2016. DOI: `10.1109/MSP.2016.2555335`.

[11] Y. Meng, D. S. Wong, R. Schlegel, and L.-f. Kwok, “Touch gestures based biometric authentication scheme for touchscreen mobile phones,” in *Information Security and Cryptology*, vol. 7763, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 331–350. DOI: `10.1007/978-3-642-38519-3_21`.

[12] W. Meng, D. S. Wong, S. Furnell, and J. Zhou, “Surveying the development of biometric user authentication on mobile phones,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1268–1293, 2015. DOI: `10.1109/COMST.2014.2386915`.

[13] A. Serwadda and V. V. Phoha, “When kids’ toys breach mobile phone security,” in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security - CCS ’13*, New York, New York, USA: ACM Press, 2013, pp. 599–610. DOI: `10.1145/2508859.2516659`.

[14] A. Serwadda, V. V. Phoha, Z. Wang, R. Kumar, and D. Shukla, “Toward robotic robbery on the touch screen,” *ACM Transactions on Information and System Security*, vol. 18, no. 4, 14:1–14:25, May 6, 2016. DOI: `10.1145/2898353`.

[15] Y. S. Lee, W. Hetchily, J. Shelton, *et al.*, “Touch based active user authentication using deep belief networks and random forests,” in *2016 6th International Conference on Information Communication and Management (ICICM)*, Oct. 2016, pp. 304–308. DOI: `10.1109/INFOCOMAN.2016.7784262`.

[16] X. Zhao, T. Feng, and W. Shi, “Continuous mobile authentication using a novel graphic touch gesture feature,” in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, Sep. 2013, pp. 1–6. DOI: `10.1109/BTAS.2013.6712747`.

[17] X. Zhao, T. Feng, W. Shi, and I. A. Kakadiaris, "Mobile user authentication using statistical touch dynamics images," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 11, pp. 1780–1789, Nov. 2014. DOI: 10.1109/TIFS.2014.2350916.

[18] Jacob Evans. "Phone reported stolen in london every six minutes," BBC News. (Apr. 11, 2023), [Online]. Available: https://www.bbc.com/news/uk-england-london-65105199.

[19] Dan Whitworth. "Mobile fraud: Thieves 'shoulder surfing' victims to steal phones," BBC News. (May 21, 2023), [Online]. Available: https://www.bbc.com/news/business-65456325.

[20] M. M. Koushki, B. Obada-Obieh, J. H. Huh, and K. Beznosov, "On smartphone users' difficulty with understanding implicit authentication," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA: ACM, May 6, 2021, pp. 1–14. DOI: 10.1145/3411764.3445386.

[21] B. Toulas. "PayPal accounts breached in large-scale credential stuffing attack," BleepingComputer. (Jan. 19, 2023), [Online]. Available: https://www.bleepingcomputer.com/news/security/paypal-accounts-breached-in-large-scale-credential-stuffing-attack/.

[22] B. Toulas. "NortonLifeLock warns that hackers breached password manager accounts," BleepingComputer. (Jan. 13, 2023), [Online]. Available: https://www.bleepingcomputer.com/news/security/nortonlifelock-warns-that-hackers-breached-password-manager-accounts/.

[23] C. Cross, K. Holt, and R. L. O'Malley, ""if u don't pay they will share the pics": Exploring sextortion in the context of romance fraud," *Victims & Offenders*, pp. 1–22, May 19, 2022. DOI: 10.1080/15564886.2022.2075064.

[24] A. Lawrence. "Uber hacked, internal systems breached and vulnerability reports stolen," BleepingComputer. (Sep. 16, 2022), [Online]. Available: https://www.bleepingcomputer.com/news/security/uber-hacked-internal-systems-breached-and-vulnerability-reports-stolen/.

[25] A. Greenberg. "Hacker lexicon: What is the signal encryption protocol?" Wired. (2020), [Online]. Available: https://www.wired.com/story/signal-encryption-protocol-hacker-lexicon/.

[26] T. Forman and A. Aviv, "Double patterns: A usable solution to increase the security of android unlock patterns," in *Annual Computer Security Applications Conference*, Austin USA: ACM, Dec. 7, 2020, pp. 219–233. DOI: 10.1145/3427228.3427252.

[27] M. Harbach, E. von Zezschwitz, A. Fichtner, A. D. Luca, and M. Smith, "It's a hard lock life: A field study of smartphone (un)locking behavior and risk perception," *SOUPS '14: Proceedings of the Tenth Symposium On Usable Privacy and Security*, pp. 213–230, 2014.

[28] A. Mahfouz, I. Muslukhov, and K. Beznosov, "Android users in the wild: Their authentication and usage behavior," *Pervasive and Mobile Computing*, vol. 32, pp. 50–61, Oct. 1, 2016. DOI: 10.1016/j.pmcj.2016.06.017.

[29] J. I. Wong. "Google plans to kill passwords with this tech, but scandinavia is way ahead of it," Quartz. (May 31, 2016), [Online]. Available: https://qz.com/695743/google-plans-to-kill-passwords-with-this-tech-but-scandinavia-is-way-ahead-of-it.

[30] Payment Card Industry. "Data security standard 4.0." (2022), [Online]. Available: https://docs-prv.pcisecuritystandards.org/PCI%20DSS/Standard/PCI-DSS-v4_0.pdf.

[31] S. Furnell, "Authenticating ourselves: Will we ever escape the password?" *Network Security*, vol. 2005, no. 3, pp. 8–13, Mar. 2005. DOI: 10.1016/S1353-4858(05)00212-6.

[32] National Cyber Security Centre. "Password policy: Updating your approach." (2018), [Online]. Available: https://www.ncsc.gov.uk/collection/passwords/updating-your-approach.

[33] E. Stobert and R. Biddle, "The password life cycle: User behaviour in managing passwords," *SOUPS '14: Proceedings of the Tenth Symposium On Usable Privacy and Security*, pp. 243–255, 2014.

[34] P. A. Grassi, M. E. Garcia, and J. L. Fenton, "Digital identity guidelines: Revision 3," National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-63-3, Jun. 22, 2017, NIST SP 800–63–3. DOI: 10.6028/NIST.SP.800-63-3.

[35] M. Kumar. "11 million ashley madison passwords cracked in just 10 days," The Hacker News. (Sep. 10, 2015), [Online]. Available: `http://thehackernews.com/2015/09/ashley-madison-password-cracked.html`.

[36] C. McGoogan. "The world's most common passwords revealed: Are you using them?" The Telegraph. (2017), [Online]. Available: `http://www.telegraph.co.uk/technology/2017/01/16/worlds-common-passwords-revealed-using/`.

[37] D. Apostal, K. Foerster, A. Chatterjee, and T. Desell, "Password recovery using MPI and CUDA," in *2012 19th International Conference on High Performance Computing*, IEEE, Dec. 2012, pp. 1–9. DOI: `10.1109/HiPC.2012.6507505`.

[38] F. Pires. "Eight RTX 4090s can break passwords in under an hour," Tom's Hardware. (Oct. 17, 2022), [Online]. Available: `https://www.tomshardware.com/news/eight-rtx-4090s-can-break-passwords-in-under-an-hour`.

[39] P. Oechslin, "Making a faster cryptanalytic time-memory trade-off," in *Advances in Cryptology - CRYPTO 2003*, D. Boneh, Ed., red. by G. Goos, J. Hartmanis, and J. van Leeuwen, vol. 2729, Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 617–630. DOI: `10.1007/978-3-540-45146-4_36`.

[40] P. G. Kelley, S. Komanduri, M. L. Mazurek, *et al.*, "Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms," in *2012 IEEE Symposium on Security and Privacy*, IEEE, May 2012, pp. 523–537. DOI: `10.1109/SP.2012.38`.

[41] W. E. Burr, D. F. Dodson, E. M. Newton, *et al.*, "Electronic authentication guideline," National Institute of Standards and Technology, Gaithersburg, MD, 2011. DOI: `10.6028/NIST.SP.800-63-1`.

[42] X. Wang, T. Kohno, and B. Blakley, "Polymorphism as a defense for automated attack of websites," in 2014, pp. 513–530. DOI: `10.1007/978-3-319-07536-5_30`.

[43] D. Silver, S. Jana, D. Boneh, E. Chen, and C. Jackson, "Password managers: Attacks and defenses," in *23rd USENIX Security Symposium (USENIX Security 14)*, San Diego: USENIX Association, 2014, pp. 449–464.

[44] P. Andriotis, G. Oikonomou, A. Mylonas, and T. Tryfonas, "A study on usability and security features of the android pattern lock screen," *Information & Computer Security*, vol. 24, no. 1, pp. 53–72, Mar. 14, 2016. DOI: `10.1108/ICS-01-2015-0001`.

[45] H. Khan, U. Hengartner, and D. Vogel, “Evaluating attack and defense strategies for smartphone PIN shoulder surfing,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, Montreal QC Canada: ACM, Apr. 19, 2018, pp. 1–10. DOI: 10.1145/3173574.3173738.

[46] EMC Corporation. “RSA SecurID hardware tokens.” (2016), [Online]. Available: https://www.tokenguard.com/datasheets/rsa-securid-hardware-tokens.pdf.

[47] D. M’Raihi, S. Machani, M. Pei, and J. Rydell, “TOTP: Time-based one-time password algorithm,” May 2011. DOI: 10.17487/rfc6238.

[48] Google. “The google authenticator project.” (2017), [Online]. Available: https://github.com/google/google-authenticator-android.

[49] D. M’Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen, “HOTP: An HMAC-based one-time password algorithm,” Dec. 2005. DOI: 10.17487/rfc4226.

[50] F. Sinigaglia, R. Carbone, G. Costa, and N. Zannone, “A survey on multi-factor authentication for online banking in the wild,” *Computers & Security*, vol. 95, p. 101 745, Aug. 2020. DOI: 10.1016/j.cose.2020.101745.

[51] A. Dmitrienko, C. Liebchen, C. Rossow, and A.-R. Sadeghi, “On the (in)security of mobile two-factor authentication,” in 2014, pp. 365–383. DOI: 10.1007/978-3-662-45472-5_24.

[52] K. Ullah, I. Rashid, H. Afzal, M. M. W. Iqbal, Y. A. Bangash, and H. Abbas, “SS7 vulnerabilities—a survey and implementation of machine learning vs rule based filtering for detection of SS7 network attacks,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1337–1371, 2020. DOI: 10.1109/COMST.2020.2971757.

[53] Patrick Heim. “An inside look at how we keep customer data safe.” (Nov. 2, 2016), [Online]. Available: https://blog.dropbox.com/topics/product-tips/dropbox-customer-data-safety.

[54] J. Colnago, S. Devlin, M. Oates, *et al.*, ““it’s not actually that horrible”: Exploring adoption of two-factor authentication at a university,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, Montreal QC Canada: ACM, Apr. 21, 2018, pp. 1–11. DOI: 10.1145/3173574.3174030.

[55] M. J. Schwartz. “Malware bypasses 2-factor authentication - BankInfoSecurity,” BankInfo Security. (2014), [Online]. Available: `https://www.bankinfosecurity.com/malware-bypasses-2-factor-authentication-a-7090`.

[56] Apple. “About touch ID advanced security technology,” Apple Support. (Sep. 11, 2017), [Online]. Available: `https://support.apple.com/en-us/HT204587`.

[57] Apple. “About face ID advanced technology,” Apple support. (Apr. 27, 2022), [Online]. Available: `https://support.apple.com/en-us/HT208108`.

[58] F. Rieger. “CCC — chaos computer club breaks apple TouchID,” Chaos Computer Club. (2013), [Online]. Available: `https://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid`.

[59] I. Buciu and A. Gacsadi, “Biometrics systems and technologies: A survey,” *International Journal of Computers Communications & Control*, vol. 11, no. 3, p. 315, Mar. 24, 2016. DOI: `10.15837/ijccc.2016.3.2556`.

[60] M. E. Garcia and P. A. Grassi, “Identity workshop on applying measurement science in the identity ecosystem: Summary and next steps,” National Institute of Standards and Technology, NIST IR 8103, Sep. 2016, NIST IR 8103. DOI: `10.6028/NIST.IR.8103`.

[61] D. Temoshok, “Digital identity guidelines,” National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-063-4 ipd, 2022, NIST SP 800–063–4 ipd. DOI: `10.6028/NIST.SP.800-63-4.ipd`.

[62] J. Daugman, “How iris recognition works,” in *The Essential Guide to Image Processing*, vol. 14, Elsevier, 2009, pp. 715–739. DOI: `10.1016/B978-0-12-374457-9.00025-1`.

[63] Y. Chen and H.-T. Ma, “Biometric authentication under threat : Liveness detection hacking,” 2019.

[64] S. K. Singla, M. Singh, and N. Kanwal, “Biometric system - challenges and future trends,” in *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, Mar. 2021, pp. 647–651.

[65] R. Bardou, R. Focardi, Y. Kawamoto, L. Simionato, G. Steel, and J.-K. Tsay, “Efficient padding oracle attacks on cryptographic hardware,” in *Advances in Cryptology - Crypto 2012*, vol. 7417, 2012, pp. 608–625. DOI: `10.1007/978-3-642-32009-5_36`.

[66] B. Krebs. "Dropbox smeared in week of megabreaches," Krebs on Security. (Jan. 6, 2016), [Online]. Available: `https://krebsonsecurity.com/2016/06/dropbox-smeared-in-week-of-megabreaches/`.

[67] C. McGoogan. "Dropbox hackers stole 68 million passwords," Telegraph online. (Aug. 31, 2016), [Online]. Available: `http://www.telegraph.co.uk/technology/2016/08/31/dropbox-hackers-stole-70-million-passwords-and-email-addresses/`.

[68] A. Z. Zaidi, C. Y. Chong, Z. Jin, R. Parthiban, and A. S. Sadiq, "Touch-based continuous mobile device authentication: State-of-the-art, challenges and opportunities," *Journal of Network and Computer Applications*, vol. 191, p. 103 162, Oct. 1, 2021. DOI: `10.1016/j.jnca.2021.103162`.

[69] M. Georgiev, S. Eberz, H. Turner, G. Lovisotto, and I. Martinovic, "Common evaluation pitfalls in touch-based authentication systems," in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, New York, NY, USA: ACM, May 30, 2022, pp. 1049–1063. DOI: `10.1145/3488932.3517388`.

[70] A. Buriro, B. Crispo, and M. Conti, "AnswerAuth: A bimodal behavioral biometric-based user authentication scheme for smartphones," *Journal of Information Security and Applications*, vol. 44, pp. 89–103, Feb. 1, 2019. DOI: `10.1016/j.jisa.2018.11.008`.

[71] P. Aaby, M. Valerio Giuffrida, W. J. Buchanan, and Z. Tan, "Towards continuous user authentication using personalised touch-based behaviour," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, Calgary, AB, Canada: IEEE, Aug. 2020, pp. 41–48. DOI: `10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00023`.

[72] Z. Wang and Y.-C. I. Chang, "Marker selection via maximizing the partial area under the ROC curve of linear risk scores," *Biostatistics*, vol. 12, no. 2, pp. 369–385, Apr. 1, 2011.

[73] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *Journal of Machine Learning Research*, vol. 5, pp. 101–141, Jan 2004.

[74] F. Pedregosa, G. Varoquaux, A. Michel, *et al.*, “Scikit-learn machine learning in python,” *Journal of machine learning research*, vol. 12, pp. 2825–2830, Oct 2011.

[75] F. Provost, “Machine learning from imbalanced data sets 101,” 2008.

[76] B. Hamdan and K. Mokhtar, “The detection of spoofing by 3d mask in a 2d identity recognition system,” *Egyptian Informatics Journal*, vol. 19, no. 2, pp. 75–82, Jul. 2018. DOI: 10.1016/j.eij.2017.10.001.

[77] P. Kałużny, “Touchscreen behavioural biometrics authentication in self-contained mobile applications design,” in *Business Information Systems Workshops*, W. Abramowicz and R. Corchuelo, Eds., ser. Lecture Notes in Business Information Processing, Cham: Springer International Publishing, 2019, pp. 672–685. DOI: 10.1007/978-3-030-36691-9_56.

[78] M. Antal, Z. Bokor, and L. Z. Szabó, “Information revealed from scrolling interactions on mobile devices,” *Pattern Recognition Letters*, vol. 56, pp. 7–13, Apr. 15, 2015. DOI: 10.1016/j.patrec.2015.01.011.

[79] Z. Syed, J. Helmick, S. Banerjee, and B. Cukic, “Effect of user posture and device size on the performance of touch-based authentication systems,” in *2015 IEEE 16th International Symposium on High Assurance Systems Engineering*, Jan. 2015, pp. 10–17. DOI: 10.1109/HASE.2015.10.

[80] U. Mahbub, S. Sarkar, V. M. Patel, and R. Chellappa, “Active user authentication for smartphones: A challenge data set and benchmark results,” in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, Sep. 2016, pp. 1–8. DOI: 10.1109/BTAS.2016.7791155.

[81] M. D. Papamichail, K. C. Chatzidimitriou, T. Karanikiotis, N.-C. I. Oikonomou, A. L. Symeonidis, and S. K. Saripalle, “BrainRun: A behavioral biometrics dataset towards continuous implicit authentication,” *Data*, vol. 4, no. 2, p. 60, Jun. 2019. DOI: 10.3390/data4020060.

[82] R. Tolosana, J. Gismero-Trujillo, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia, “MobileTouchDB: Mobile touch character database in the wild and biometric benchmark,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2019, pp. 2306–2314. DOI: 10.1109/CVPRW.2019.00284.

[83] A. Roy, T. Halevi, and N. Memon, "An HMM-based behavior modeling approach for continuous mobile authentication," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy: IEEE, May 2014, pp. 3789–3793. DOI: 10.1109/ICASSP.2014.6854310.

[84] P. Saravanan, S. Clarke, D. H. (Chau, and H. Zha, "LatentGesture," in *Proceedings of the Second International Symposium of Chinese CHI*, New York, NY, USA: ACM, Apr. 26, 2014, pp. 110–113. DOI: 10.1145/2592235.2592252.

[85] T. Feng, J. Yang, Z. Yan, E. M. Tapia, and W. Shi, "TIPS," in *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, New York, NY, USA: ACM, Feb. 26, 2014, pp. 1–6. DOI: 10.1145/2565585.2565592.

[86] S. Mondal and P. Bours, "Continuous authentication and identification for mobile devices: Combining security and forensics," in *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, Nov. 2015, pp. 1–6. DOI: 10.1109/WIFS.2015.7368591.

[87] Ş. Budulan, E. Burceanu, T. Rebedea, and C. Chiru, "Continuous user authentication using machine learning on touch dynamics," in S. Arik, T. Huang, W. K. Lai, and Q. Liu, Eds., vol. 9489, Cham: Springer International Publishing, 2015, pp. 591–598. DOI: 10.1007/978-3-319-26532-2_65.

[88] J. Nader, A. Alsadoon, P. Prasad, A. Singh, and A. Elchouemi, "Designing touch-based hybrid authentication method for smartphones," *Procedia Computer Science*, vol. 70, pp. 198–204, 2015. DOI: 10.1016/j.procs.2015.10.072.

[89] L. Lu and Y. Liu, "Safeguard: User reauthentication on smartphones via behavioral biometrics," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 3, pp. 53–64, Sep. 2015. DOI: 10.1109/TCSS.2016.2517648.

[90] H. Zhang, V. M. Patel, M. Fathy, and R. Chellappa, "Touch gesture-based active user authentication using dictionaries," in *2015 IEEE Winter Conference on Applications of Computer Vision*, IEEE, Jan. 2015, pp. 207–214. DOI: 10.1109/WACV.2015.35.

[91] N. Palaskar, Z. Syed, S. Banerjee, and C. Tang, "Empirical techniques to detect and mitigate the effects of irrevocably evolving user profiles in touch-based authentication systems," in *2016 IEEE 17th International Symposium on High Assurance*

Systems Engineering (HASE), IEEE, Jan. 2016, pp. 9–16. DOI: 10.1109/HASE.2016.39.

[92] A. A. Alariki, A. A. Manaf, and S. M. Mousavi, “Features extraction scheme for behavioural biometric authentication in touchscreen mobile devices,” *International Journal of Applied Engineering Research*, vol. 11, no. 18, pp. 9331–9344, 2016.

[93] C. Shen, Y. Zhang, X. Guan, and R. A. Maxion, “Performance analysis of touch-interaction behavior for active smartphone authentication,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 3, pp. 498–513, Mar. 2016. DOI: 10.1109/TIFS.2015.2503258.

[94] C. J. Kroeze and K. M. Malan, “User authentication based on continuous touch biometrics,” *South African Computer Journal*, vol. 28, no. 2, Dec. 16, 2016. DOI: 10.18489/sacj.v28i2.374.

[95] A. Pozo, J. Fierrez, M. Martinez-Diaz, J. Galbally, and A. Morales, “Exploring a statistical method for touchscreen swipe biometrics,” in *2017 International Carnahan Conference on Security Technology (ICCST)*, Oct. 2017, pp. 1–4. DOI: 10.1109/CCST.2017.8167823.

[96] J. Fierrez, A. Pozo, M. Martinez-Diaz, J. Galbally, and A. Morales, “Benchmarking touchscreen biometrics for mobile authentication,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2720–2733, Nov. 2018. DOI: 10.1109/TIFS.2018.2833042.

[97] S. J. Alghamdi and L. A. Elrefaei, “Dynamic authentication of smartphone users based on touchscreen gestures,” *Arabian Journal for Science and Engineering*, vol. 43, no. 2, pp. 789–810, Feb. 4, 2018. DOI: 10.1007/s13369-017-2758-x.

[98] W. Meng, Y. Wang, D. S. Wong, S. Wen, and Y. Xiang, “TouchWB : Touch behavioral user authentication based on web browsing on smartphones,” *Journal of Network and Computer Applications*, vol. 117, pp. 1–9, Sep. 2018. DOI: 10.1016/j.jnca.2018.05.010.

[99] S. Y. Ooi and A. B.-J. Teoh, “Touch-stroke dynamics authentication using temporal regression forest,” *IEEE Signal Processing Letters*, vol. 26, no. 7, pp. 1001–1005, Jul. 2019. DOI: 10.1109/LSP.2019.2916420.

[100] M. Santopietro, R. Vera-Rodriguez, R. Guest, A. Morales, and A. Acien, “Assessing the quality of swipe interactions for mobile biometric systems,” in *2020 IEEE*

International Joint Conference on Biometrics (IJCB), Sep. 2020, pp. 1–8. DOI: 10.1109/IJCB48548.2020.9304858.

[101] A. Sarhan and A. Ramadan, "Continuous user authentication on touchscreen using behavioral biometrics utilizing machine learning approaches," in 2020, pp. 243–281. DOI: 10.4018/978-1-7998-2701-6.ch013.

[102] N. Pokhriyal and V. Govindaraju, "Learning discriminative factorized subspaces with application to touchscreen biometrics," *IEEE Access*, vol. 8, pp. 152 500–152 511, 2020. DOI: 10.1109/ACCESS.2020.3014188.

[103] S. Keykhaie and S. Pierre, "Mobile match on card active authentication using touchscreen biometric," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 4, pp. 376–385, Nov. 2020. DOI: 10.1109/TCE.2020.3029955.

[104] M. Agrawal, P. Mehrotra, R. Kumar, and R. R. Shah, "Defending touch-based continuous authentication systems from active adversaries using generative adversarial networks," in *2021 IEEE International Joint Conference on Biometrics (IJCB)*, IEEE, Aug. 4, 2021, pp. 1–8. DOI: 10.1109/IJCB52358.2021.9484366.

[105] A. Z. Zaidi, C. Y. Chong, R. Parthiban, and A. S. Sadiq, "A framework of dynamic selection method for user classification in touch-based continuous mobile device authentication," *Journal of Information Security and Applications*, vol. 67, p. 103 217, Jun. 1, 2022. DOI: 10.1016/j.jisa.2022.103217.

[106] M. Georgiev, S. Eberz, and I. Martinovic, "Techniques for continuous touch-based authentication," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13620 LNCS, Springer Science and Business Media Deutschland GmbH, 2022, pp. 409–431. DOI: 10.1007/978-3-031-21280-2_23.

[107] M. Martinez-Diaz, J. Fierrez, R. P. Krish, and J. Galbally, "Mobile signature verification: Feature robustness and performance comparison," *IET Biometrics*, vol. 3, no. 4, pp. 267–277, 2014. DOI: 10.1049/iet-bmt.2013.0081.

[108] N. Z. Gong, M. Payer, R. Moazzezi, and M. Frank, "Forgery-resistant touch-based authentication on mobile devices," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, New York, NY, USA: ACM, May 30, 2016, pp. 499–510. DOI: 10.1145/2897845.2897908.

[109] B. C. Ross, “Mutual information between discrete and continuous data sets,” *PLOS ONE*, vol. 9, no. 2, e87357, Feb. 19, 2014. DOI: `10.1371/journal.pone.0087357`.

[110] A. K. Belman, L. Wang, S. S. Iyengar, *et al.*, *SU-AIS BB-MAS (syracuse university and assured information security - behavioral biometrics multi-device and multi-activity data from same users) dataset*, Nov. 20, 2019. DOI: `10.21227/RPAZ-0H66`.

[111] Haibo He, Yang Bai, E. A. Garcia, and Shutao Li, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, Hong Kong, China: IEEE, Jun. 2008, pp. 1322–1328. DOI: `10.1109/IJCNN.2008.4633969`.

[112] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, pp. 815–823, Oct. 14, 2015. DOI: `10.1109/CVPR.2015.7298682`.

[113] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 27, 2022, pp. 11 966–11 976. DOI: `10.1109/CVPR52688.2022.01167`.

[114] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2016-December, IEEE, Jun. 9, 2016, pp. 770–778. DOI: `10.1109/CVPR.2016.90`.

[115] J. Ahmad, M. Sajjad, Z. Jan, I. Mehmood, S. Rho, and S. W. Baik, “Analysis of interaction trace maps for active authentication on smart devices,” *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4069–4087, Feb. 15, 2017. DOI: `10.1007/s11042-016-3450-y`.

[116] Bkav. “Bkav’s new mask beats face ID in ”twin way”.” (2017), [Online]. Available: `bkav-s-new-mask-beats-face-id-in-twin-way-severity-level-raised-do-not-use-face-id-in-business-transactions`.

[117] Z. Sitova, J. Sedenka, Q. Yang, *et al.*, “HMOG: New behavioral biometric features for continuous authentication of smartphone users,” *IEEE Transactions on*

Information Forensics and Security, vol. 11, no. 5, pp. 877–892, May 2016. DOI: 10.1109/TIFS.2015.2506542.

[118] C. Shen, Y. Li, Y. Chen, X. Guan, and R. A. Maxion, "Performance analysis of multi-motion sensor behavior for active smartphone authentication," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 48–62, Jan. 2018. DOI: 10.1109/TIFS.2017.2737969.

[119] W. Meng, W. Li, D. S. Wong, and J. Zhou, "TMGuard: A touch movement-based security mechanism for screen unlock patterns on smartphones," in *Applied Cryptography and Network Security*, Springer International Publishing, 2016, pp. 629–647. DOI: 10.1007/978-3-319-39555-5_34.

[120] Q. Gu, Z. Li, and J. Han, "Generalized fisher score for feature selection," *arXiv:1202.3725 [cs, stat]*, Feb. 14, 2012.

[121] S. Raschka, "MLxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack," *Journal of Open Source Software*, vol. 3, no. 24, p. 638, Apr. 22, 2018. DOI: 10.21105/joss.00638.

[122] S. O'Dea. "Penetration rate of smartphones in selected countries 2020," Statista. (Sep. 2020), [Online]. Available: https://www.statista.com/statistics/539395/smartphone-penetration-worldwide-by-country/.

[123] BBC News, "Face ID iPhone x 'hack' demoed live," *BBC News*, 2017.

[124] W. R. Almeida, F. A. Andaló, R. Padilha, *et al.*, "Detecting face presentation attacks in mobile devices with a patch-based CNN and a sensor-aware loss function," *PLOS ONE*, vol. 15, no. 9, H. Debiao, Ed., e0238058, Sep. 4, 2020. DOI: 10.1371/journal.pone.0238058.

[125] S. Eberz, G. Lovisotto, A. Patane, M. Kwiatkowska, V. Lenders, and I. Martinovic, "When your fitness tracker betrays you: Quantifying the predictability of biometric features across contexts," in *2018 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA: IEEE, May 2018, pp. 889–905. DOI: 10.1109/SP.2018.00053.

[126] N. Shekoufa, J. Rahimipour Anaraki, and S. Samet, *Replication data for: Continuous authentication using touch dynamics and its application in personal health records*, in collab. with J. Rahimipour Anaraki, 2019. DOI: 10.7910/DVN/VVXWZO.

[127] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017.

[128] Juniper research. "Fighting online payment fraud in 2022 & beyond." (2022), [Online]. Available: `https://www.juniperresearch.com/whitepapers/fighting-online-payment-fraud-in-2022-beyond`.

[129] S. Ghorbani Lyastani, M. Schilling, M. Neumayr, M. Backes, and S. Bugiel, "Is FIDO2 the kingslayer of user authentication? a comparative usability study of FIDO2 passwordless authentication," in *2020 IEEE Symposium on Security and Privacy (SP)*, vol. 2020-May, IEEE, May 1, 2020, pp. 268–285. DOI: `10.1109/SP40000.2020.00047`.

[130] P. Aaby, M. V. Giuffrida, W. J. Buchanan, and Z. Tan, "An omnidirectional approach to touch-based continuous authentication," *Computers & Security*, vol. 128, p. 103 146, May 2023. DOI: `10.1016/j.cose.2023.103146`.

[131] A. Paszke, S. Gross, F. Massa, *et al.*, "PyTorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, Dec. 3, 2019.

[132] A. Howard, M. Sandler, B. Chen, *et al.*, "Searching for MobileNetV3," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 1314–1324. DOI: `10.1109/ICCV.2019.00140`.

[133] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 10 691–10 700, May 28, 2019. DOI: `10.48550/arxiv.1905.11946`.

[134] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *7th International Conference on Learning Representations, ICLR 2019*, Nov. 14, 2017. DOI: `10.48550/arxiv.1711.05101`.

[135] Z. Liu, H. Hu, Y. Lin, *et al.*, "Swin transformer v2: Scaling up capacity and resolution," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Institute of Electrical and Electronics Engineers (IEEE), Nov. 18, 2021, pp. 11 999–12 009. DOI: `10.1109/CVPR52688.2022.01170`.

[136] I. Loshchilov and F. Hutter, “SGDR: Stochastic gradient descent with warm restarts,” in *International Conference on Learning Representations*, 2017.

[137] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 2818–2826, Dec. 2, 2015. DOI: `10.48550/arxiv.1512.00567`.

[138] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, “Random erasing data augmentation,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 13 001–13 008, Apr. 3, 2020. DOI: `10.1609/aaai.v34i07.7000`.

[139] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 1, 2009, pp. 248–255. DOI: `10.1109/CVPR.2009.5206848`.

[140] P. Aaby, *TouchEnc: Repository and runs*, version V1.0.0, Sep. 10, 2023. DOI: `10.5281/ZENODO.8332523`.

[141] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, vol. 2017-October, IEEE, Oct. 22, 2017, pp. 618–626. DOI: `10.1109/ICCV.2017.74`.

[142] H. Xuan, A. Stylianou, X. Liu, and R. Pless, “Hard negative examples are hard, but useful,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12359 LNCS, pp. 126–142, 2020. DOI: `10.1007/978-3-030-58568-6_8`.

[143] H. Xuan, A. Stylianou, and R. Pless, “Improved embeddings with easy positive triplet mining,” in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, Mar. 8, 2020, pp. 2463–2471. DOI: `10.1109/WACV45572.2020.9093432`.

[144] Y. Sun, C. Cheng, Y. Zhang, *et al.*, “Circle loss: A unified perspective of pair similarity optimization,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2020, pp. 6397–6406. DOI: `10.1109/CVPR42600.2020.00643`.

[145] M. Ramadiansyah and L. Rahadianti, “Proxy-based losses and pair-based losses for face image retrieval,” *2020 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2020*, pp. 177–186, Oct. 17, 2020. DOI: `10.1109/ICACSIS51025.2020.9263132`.

[146] S. Kim, D. Kim, M. Cho, and S. Kwak, “Proxy anchor loss for deep metric learning,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 3235–3244. DOI: `10.1109/CVPR42600.2020.00330`.