An exploration of dynamic biometric performance using device interaction and wearable technologies

Marco Santopietro

A Thesis submitted to the University of Kent for the Degree of Doctor of Philosophy in Electronic Engineering

School of Engineering
University of Kent

October 2022.

Abstract

With the growth of mobile technologies and internet transactions, privacy issues and identity check became a hot topic in the past decades. Mobile biometrics provided a new level of security in addition to passwords and PIN, with a multitude of modalities to authenticate subjects. This thesis explores the verification performance of *behavioural biometric* modalities, as previous studies in literature proved them to be effective in identifying individual behaviours and guarantee robust continuous authentication. In addition, it addresses open issues such as single sample authentication, quality measurements for behavioural data, and fast electrocardiogram capture and biometric verification. The scope of this project is to assess the performance and stability of authentication models for mobile and wearable devices, with ceremony based tasks and a framework that includes behavioural and electrocardiogram biometrics.

The results from the experiments suggest that a fast verification, appliable on real life scenarios (e.g. login or transaction request), with a single sample request and the considered modalities (Swipe gestures, PIN dynamics and electrocardiogram recording) can be performed with a stable performance. In addition, the novel fusion method implemented greatly reduced the authentication error.

As additional contribution, this thesis introduces to a novel pre-processing algorithm for faulty Swipe data removal. Lastly, a theoretical framework comprised of three different modalities is proposed, based on the results of the various experiments conducted in this study. It's reasonable to state that the findings presented in this thesis will contribute to the enhancement of identity verification on mobile and wearable technologies.

Acknowledgements

This research was supported and funded by Callsign®, a leading company in the field of web identification and privacy. In particular, I would like to express my gratitude to Dr. Oscar Miguel-Hurtado, Dr. Ramon Fuentes, Dr. Johan Swård, and Dr. Daniel Maldonado for their presence and supervision over the past years.

I would also like to thank my supervisor Prof. Richard Guest for the guidance and the support during each stage of my work, from the beginning of the project to the submission of the thesis.

A special mention to the Biometric group of Universidad Autonoma de Madrid (UAM), for their collaboration during the study of Swipe quality interactions.

Indirect contributors to this thesis are also the fellow PhD students from the School of Engineering and Digital Arts (EDA) at the University of Kent: Hazal Bicakci, Matthew Boakes, Ethan Cheung, Paula Delgado, Dr. Elakkiya Ellavarason, Rania Kolaghassi, Michael Liao, Mayank Loonker, Ayda Majd, Pavlos Nicolau, Sophia Ppali, Chantal Zorzi. They provided help, insight, and new ideas in addition to amazing company and joyful time spent together.

Last but not least, I sincerely thank my family and my friends from Italy, who never failed to support and encourage me, regardless of the distance, even during the periods of lockdown.

Published content

During this research, the following publications were submitted and accepted to peer-reviewed conferences:

- "*Assessing the Quality of Swipe Interactions for Mobile Biometric Systems*",Marco Santopietro, Ruben Vera-Rodriguez, Richard Guest; Aythami Morales and Alejandro Acien, *2020 IEEE International Joint Conference on Biometrics (IJCB)*, 2020, pp. 1-8.

  This publication [1] is an assessment of Swipe quality from User and Sample perspective. Methodology, experimental protocol and results are presented in Chapter 3.2.

- "*Evaluation of Electrocardiogram Biometric Verification Models Based on Short Enrollment Time on Medical and Wearable Recorders*",Hazal Su Bıçakcı, Marco Santopietro, Matthew Boakes and Richard Guest, *2021 International Carnahan Conference on Security Technology (ICCST)*, 2021, pp. 1-6.

  This paper [2] explores electrocardiogram biometric verification, comparing different classification models and proposing a novel deep learning architecture able to authenticate the user with short enrollment time and fast recording. All the procedures and methodologies are explained in Chapter 5[1].

Portions of the above papers appear verbatim within this thesis.

[1]Part of this study, mentioned in Chapter 5, was not conducted by the author of this thesis. In details, Linear Discriminant Analysis evaluation and results are the only external contribution appearing in this thesis.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Introduction to the thesis

This research was motivated by the increasing demands of security measures in the last decades, especially considering the growth of mobile technologies, online transactions, and biometric modalities. We wanted to explore biometric verification performance through behavioural modalities with a fast and reliable approach. We addressed unsolved challenges in literature and proposed our own framework for a novel approach on mobile verification.

Each Chapter will describe and resolve a separate issue, starting from a general introduction to biometrics and moving forward to each modality, to finally combine all the information, experiments, and findings to a complete assessment of the aforementioned framework. We present novelties in terms of the pre-processing, pipeline, fusion methods, and device experimentation. We are confident in thinking that this thesis can provide an insight on the direction and development that mobile and wearable biometrics will follow in the near future.

1.2 Introduction to biometrics

Over the centuries, identity has always been a matter of study from various perspective (e.g. ontological, social, philosophical, and legal); personal identity is not just something that defines *us* as people and person, but from a non self-deterministic point of view it also has severe implications on a daily basis in society. For every document signed, every award, every action claimed, there's a connection with personal identity. Therefore, considering how valuable identity is, the existence of identity theft is not surprising. Especially in the recent decades, with the growth of technology, media, and globalisation, more measures are required to secure and verify identity.

In this context, biometrics provide a solid way to verify identity. It is defined by ISO as *"the automated recognition of humans based on their biological and behavioural characteristics"*[7] and, as the definition suggests, it comprises of systems, sensors, and algorithms meant to extract and evaluate unique features descriptive of an individual, either physical or behavioural. With such features, biometric systems are built to either *identify* a specific subject amongst a database comparing the given sample(s) to a list of templates (1 vs all) or to *verify* the claimed identity of an individual comparing the given sample(s) to their own enrolled template (1 vs 1).

Biometric systems are historically implemented for security reasons and forensic applications (e.g. face recognition or fingerprints) but, with the growth in the last decades of mobile devices, online trading, IoT, and mobile applications [8], *mobile biometrics* [9] have started to be a common trend and are gradually accepted, to the point that now every mobile device is expected to have at least one biometric system embedded.

In parallel with the diffusion of biometric modalities to safeguard identity or secure transactions/activity on a device, the number of techniques to exploit biometric systems vulnerabilities has also risen. To improve biometric system performances

and reduce errors due to attacks and forgeries, especially on mobile devices, research continues to explore solid models and modalities that can provide safety for the user, occasionally by combining multiple modalities at once.

This study focuses on exploring biometric performance on mobile and wearable devices, proposing a novel framework with three different modalities (PIN and Swipe as behavioural modality, plus electrocardiogram biometric as third modality) for verification purposes, with the the objective to minimise the amount of recorded data requested to perform an authentication.

1.3 Aims and objectives

The aim of this study is to address some open challenges regarding mobile authentication through behavioural biometrics, such as reducing to the minimum the amount of data required for the enrollment and the authentication (for the latter, ideally one single sample). In particular, for the considered modalities, previous studies have either considered mostly continuous authentication, or proposed models that required a large amount of training data.

Another objective of this thesis was to provide generalised models, that would not need to be retrained on every subject. Further considerations about gap in literature, open issues, reasoning for this study , and research questions are presented in Chapter 2.

1.4 Research methodology

All the experiments of the project have been designed following the same methodology. For each biometric modality, the related studies and state of the art were reviewed, addressing the open challenges and gaps in literature. Then, it's presented a description of the datasets used, comprised of demographic information (when possible) and details on data capture.

This was followed by an in-depth description of the models and related algorithms proposed for biometric authentication, and the reasoning behind the choice of different methodologies. The training and testing criteria were defined, as well as parameters optimisation.

Lastly, results on experiments were presented and commented, with comparison to previous studies and relating to the research questions.

1.5 Key contributions

The major contribution of this thesis can be listed in the following points:

- The Introduction of new, fast, and light (in terms of computational cost) deep learning authentication models for the three biometric modalities explored (Swipe, PIN and ECG).
- The introduction of an algorithm to measure quality (for both user and samples) of a Swipe.
- The development of a pre-processing algorithm for data cleaning and faulty sample removal.
- A fusion algorithm on score Level for multi-modality purposes, which relies on contextual information.
- The proposal of a theoretical framework for three-factors authentication, based on the finding of the experiments conducted on the aforementioned modalities (Swipe, PIN and ECG).

In particular, the work in this thesis addresses the issues of quick data recording and authentication with behavioural data. Especially considering Swipe and ECG biometrics, for which the presented models perform authentication on a single sample basis.

Another strong contribution is given by the quality experiments, being the first study conducted on Swipe quality. The novelty of the fusion method resides in its specific design based on the contextual information of the framework; more specifically, it takes into consideration the dependency of the two modalities, and the authentication error on the first modality.

1.6 Thesis Structure

This thesis is composed of six chapters and one appendix that follow our proposed framework for multimodal biometric authentication. *Chapter 2* explores the state of the art of all the methodologies, algorithms, and devices used in this study or that appear in the related works. It provides insights on artificial intelligence in general, with a focus on models applied to biometric authentication, pattern recognition, and clustering. In addition, we discuss the hardware requirements for electrocardiogram recording devices in order to justify the decision making when exploring wearable or mobile possibilities for biometric authentication. We summarise the recent works in the field for all the considered modalities and we provide an overview on multimodal fusion. Lastly, we define the research questions and the framework. The appendix (see Appendix A) reports the section of the contract with Callsign involving use of Materials (comprised of production data).

The following three chapters describe in details the methodology, experiments, and results for each modality assessed in this study. *Chapter 3* introduces behavioural interactions on mobile devices with hand gestures and Swipe dynamics; it gives a description of the recorded data from used datasets and adds in depth information on the theory behind the evaluated models and algorithms. This Chapter is further divided in two sub-chapters, the first part assessing Swipe authentication and the second part discussing Swipe quality and its effects on authentication performance.

Chapter 4 is also dedicated to behavioural biometrics from a PIN perspective. As in the previous Chapter, a general introduction to PIN and keystroke on mobile devices is provided, followed by data description and model assessment. The last

section of the Chapter describes our proposed fusion model for Swipe and PIN authentication.

Chapter 5 explores electrocardiogram as a biometric modality. We first define the idea behind user authentication with ECG recordings, then we proceed describing the data from a physiological and a biometric perspective. We describe the datasets used, our experimental setup and the models that we evaluated for biometric verification. Results and conclusion are presented in the last section of the Chapter.

Chapter 6 summarises the content of the thesis, with respect to the aim of this study, addressing each research question from Chapter 2. It highlights the novelty and the contribution of this research, while also assessing the open challenges and the possible follow-up studies that can be inspired by our work. This Chapter also cover theoretical formulations that could not be assessed in this research due to COVID-19 impediments.

1.7 SARS COVID-19 related issues

On December 2019 several cases of pneumonia were identified in Wuhan, China which resulted later as direct effects of a new coronavirus branch that has not been previously identified in humans (later named as COVID-19). In the early months of 2020 the virus spread globally, starting from central Europe, with high infection and mortality rates. By March 2020, the World Health Organisation (WHO) acknowledged the outbreak as a pandemic and gradually all countries, including U.K., adopted emergency measures such as work from home and lockdown.

Over the next two years, the NHS and the government were forced to apply strict regulations for workplaces, educational institutions and general life, in order to mitigate the spread of the virus while waiting for an effective vaccine. Access to facilities was prohibited, as well as social interactions with the exception of

social bubbles, people living in the same households, and people with special circumstances (health related, childcare, etc.).

Restrictions became gradually lighter with the spread decay and the increase of the number of vaccinated people, occasionally becoming strict again with new virus variants and peaks in spread (usually after social breakouts during summer). Currently, even if the virus is still active in the majority of the countries, in U.K. and other European countries almost all regulations decayed.

Nevertheless, during 2020 and 2021 COVID-19 had a strong impact on people lives [10], both job and mental health related. In the specific case of this study, due to the inability to fully access facilities and devices nor to interact with other individuals, a considerable part of the research could not be performed. In particular, all the evaluations on electrocardiogram data and models were slowed down, a data collection could not be performed due to the impossibility of interacting with the participants. Moreover, the sensors for electrocardiogram recording need to be in direct contact with the skin and this was considered a threat for contamination during the pandemic.

The absence of a data collection for electrocardiogram data not only forced us to rely on public dataset, but also precluded the experiments on three factors fusion, due to the lack of paired data. However, we still provide an insight of the original idea and theoretical solutions for multimodal authentication with PIN, Swipe and electrocardiogram.

Chapter 2

Mobile and wearable biometrics: state of the art

2.1 Introduction

This Chapter is a review of the latest biometrical studies and machine learning advancements over the years, related to the work in this thesis. We will first introduce the reader to mobile biometrics and wearable technologies, with an insight on biometric performance and how to measure it. Then we will review different methodologies of machine learning and deep learning architectures that are used or led to decision making in our framework. Then we will compare state of the art authentication models for swipe, PIN, and electrocardiogram biometrics and highlight the issues in the previous studies or the follow-up analysis that we conducted based on other studies. We included a brief review of sensors used to collect the data. We then explore the common fusion methods for multimodal biometrics. Finally, we highlight open challenges and trade offs of the current authentication systems, and declare the research question and the proposed framework.

2.2 Mobile biometrics and wearable technologies

Mobile biometrics refers to any implemented technology for authentication or identification purpose through biometric modalities on a mobile device (e.g. smartphone, smartwatch, tablet). In the last decades there's been a wide shift from biometrics used in supervised environment for security control (e.g. airport border control) to mobile biometric on personal devices in unsupervised environments. Mobile transactions, login to personal accounts or even unlocking the device itself were the main reasons for this growth. The most common biometric modalities found on mobile devices are *face*, *iris*, *fingerprint*,and *voice* recognition, with later appearances of *hand gestures* and *keystroke*. In addition, with the increase in popularity of wearable devices such as wrist or chest bands for health monitoring during physical exercises, other biometric modalities began to appear (not necessarily connected to user authentication). But before reviewing the past studies on mobile biometrics, it's necessary to define the biometric performance of a model and how to measure it.

2.2.1 Biometric performance

The biometric performance of an authentication model is the likelihood to provide the correct prediction on a verification or identification task, therefore the ability to predict the correct class given unseen data. A verification task, in which a subject is asked to confirm their identity, can be considered a binary classification problem: accept if the system evaluates the data as belonging to the genuine subject or reject in the opposite case. An identification task is a multiclass problem, in which the biometric system compares the given sample with a list of existing templates and returns the class with the maximum likelihood. In both cases, if the prediction of the class is correct, we will talk about *True Positive* (TP) if the data was provided by the genuine subject, or *True Negative* (TN) in case of forgeries. As opposite, in case of misclassification, we talk about *False Positives* (FP) and *False Negatives* (FN).

Given these premises, there are various metrics to assess the performance of a biometric system, such as *Accuracy*,*False Positive Rate* (FPR), *True Positive Rate* (TPR),*False Negative Rate*(FNR) and *Equal Error Rate*. If we consider P and N the total number of data from genuine and forgeries respectively in the test set, we can calculate the accuracy, the FPR, the TPR,and the FNR of the system as follows:

$$\begin{aligned} Accuracy[\%] &= 100 \times \frac{TP + TN}{P + N} \\ FPR[\%] &= 100 \times \frac{FP}{N} \\ TPR[\%] &= 100 \times \frac{TP}{P} \\ FNR[\%] &= 100 \times \frac{FN}{P} \end{aligned} \tag{2.1}$$

These metrics are based on a direct prediction, but biometric systems returns a probability of the given sample to belong to a class. This probability is usually expressed as a similarity (or dissimilarity score) between the sample and a template. The prediction of the class is based on the aforementioned score and a threshold that should minimise the classification error.

Figure 2.1 shows an example with two synthetic distributions of scores from genuine samples and forgeries. In an ideal scenario, the classifier should be able to perfectly separate the two distributions, but in reality there will be overlapping. Depending on the selected threshold, the tails of the distributions crossing it will provide classification errors: type 1 error (false positive) for the forgeries distribution, type 2 error (false negative) for the genuine distribution.

The optimal threshold should minimise the errors and it's found where FPR and FNR are equal. That point can be usually found at the intersection of the two distributions and it's defined as Equal Error Rate.

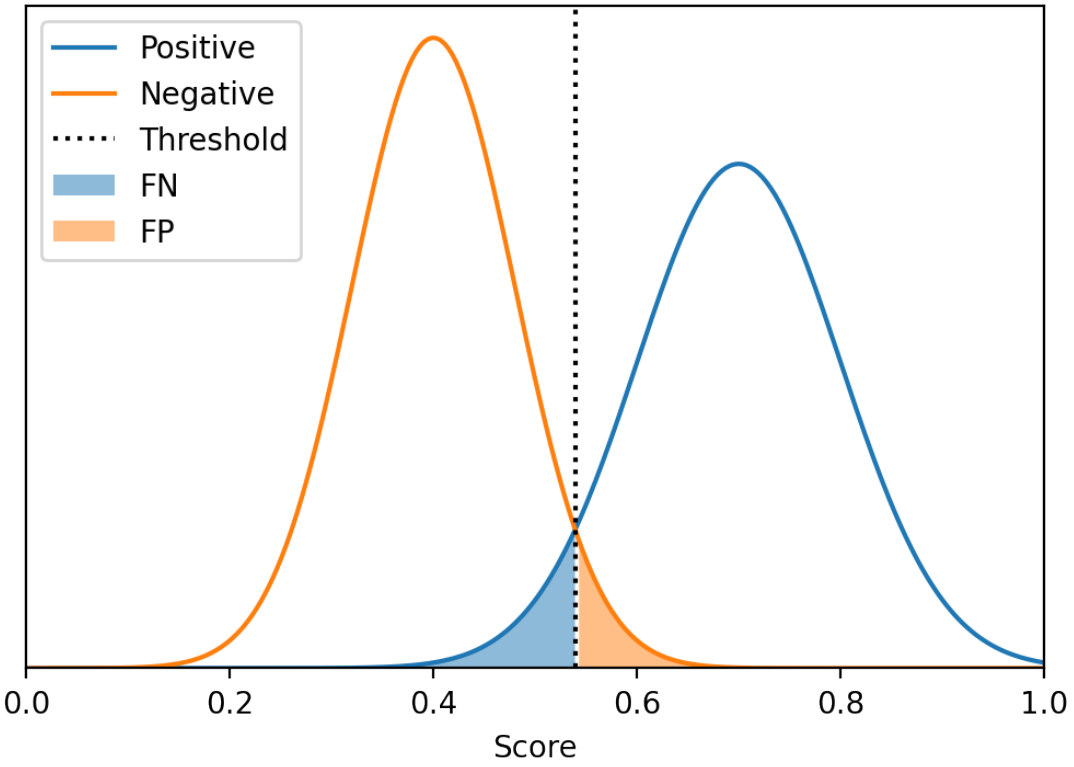


FIGURE 2.1: An example of scores distribution from genuine and forgery samples. Blue and orange filled areas represent the classification errors of the system, given the optimal threshold.

2.3 State of the art for machine learning

2.3.1 Brief history of machine learning

Machine learning is a term to describe algorithms that, in a very simplistic definition, after training on a set of data can provide predictions on unseen data. Such prediction or scores can be used for classification, comparison, reconstruction, and more. The origin of machine learning is not clearly defined and it depends on the definition or the context, but it's usually associated to the studies on the human nervous system and the development of a device that imitates the behaviour of a neuron cell to compute a decision [11]. Such a device was the first *perceptron* machine, developed by Frank Rosenblatt et al. in 1957 [12].

From that study, artificial intelligence and machine learning followed an exponential progression in terms of proposed models and applications in many fields; in the

60s, thanks to the first appearance of the nearest neighbor algorithm, there was the beginning of pattern recognition and the studies on artificial intelligence began to separate and focus on specific tasks or training techniques. Support vector machines and multilayer perceptrons appeared in the 70s alongside new optimisation criteria and cost functions. At the same time, researches involving neural networks slowed down for a decade when Minsky et al. addressed its limitations in 1969 [13], diverging over probabilistic approaches.

In 1985 with Ackley et al.'s backpropagation algorithm [14] and later in 1998 with LeCun's convolutional neural network *LeNet* [15], interest in neural networks arose again and a new subset of artificial intelligence referred to as deep learning was born[16]. Amongst the first applications, it's relevant to cite speech recognition through recurrent neural networks [17] and in 2012 face recognition with *AlexNet* [18].

2.3.2 Machine learning categories

As stated previously, Artificial Intelligence can be divided in macro categories. Machine learning and deep learning are two subsets (see Figure 2.2), but it's useful to mention four additional subsets of learning approaches categorised as follows: *supervised*, *semi-supervised*, *unsupervised*, and *reinforcement*. These four categories describe the approaches of ML techniques depending on the training data and/or the environment in which they are deployed.

- *Supervised learning* comprises of all the models, algorithms and scenarios that require training with labeled data [19]. Such labels describe both the input features and the target to forecast (also known as *ground truth*), even when there's a mismatch (in case of event prediction, segmentation or contextual classification).

- *Unsupervised learning*, as the name suggests, is the opposite of supervised learning. This approach is employed when data are unlabeled or unclassified, and it's not possible to manually provide such information. This

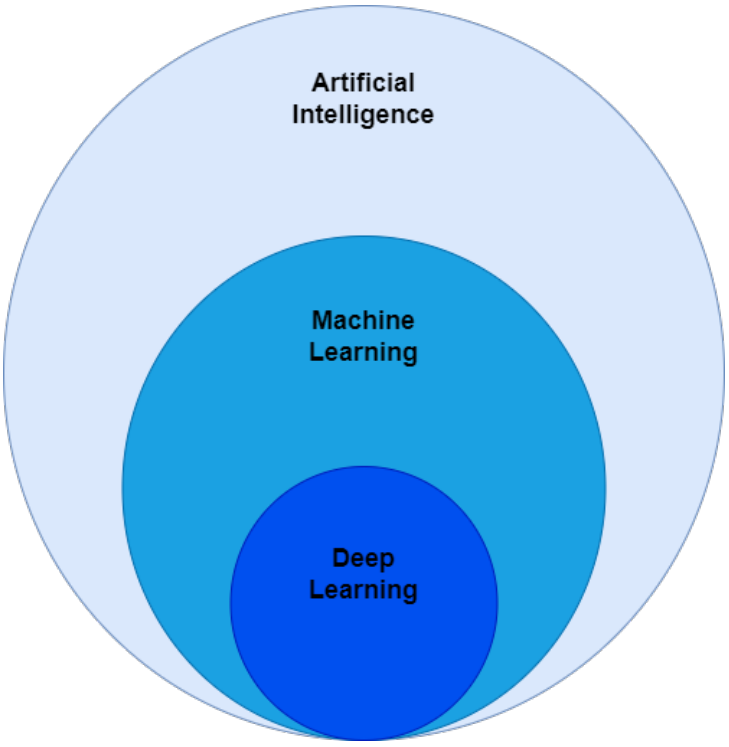


FIGURE 2.2: Hierarchy of AI.

is usually the case for behavioural data or quality estimation. The model needs to understand data by context and discover similar patterns in the raw data. Usually clustering algorithms and reconstruction models that directly compute the differences between raw inputs and reconstructed outputs are employed in this category.

- *Semi-supervised learning* explores the case of label and unlabeled data, either when just a small percentage of the available data are provided with labels and when the labels are not descriptive of the actual class, but define some common conditions for a set of samples [20]. Compared to a fully unsupervised scenario, the ground truth information provided from the small amount of data can be helpful to fine tune unsupervised algorithms, tweaking the decision boundaries for the generated clusters.

- *Reinforcement learning* (RL) [21] relies entirely on the environment and the task. Compared to the previous approaches, the final goal is not a single static prediction, but a change in the environment (or the machine interacting with it). Usually RL models are based on the Markovian decision

process [22], which requires an *Agent* performing an *action* that will change the environmental *state*, based on the previous state and a *reward* system. Those models (e.g. Hidden Markov Model) and the various deep learning implementations of the last decade are used to replicate the human decision process for sequence of actions (e.g. chess games, car driving, etc.).

In this study we explored biometric authentication only, with various modalities, therefore we will provide a review exclusively on the latest supervised and unsupervised approaches.

2.3.3 Supervised and semi-supervised models

Classic supervised machine learning models are widely used in a multitude of situations for their ease-to-implement and generally fast training times, such as the *decision tree* [23] and the *Naive Bayes* classifier [24].

The decision tree is a logic-based algorithm that splits the data in nodes and leaves, providing a partial decision at each node. To overcame issues such as uncertainty about the size of the model and likelihood to over-fit to training data, two follow up algorithms were proposed: the *random forest* in 2001 by Leo Breiman [25], and the *isolation forest* by Liu et al. [26] in 2008. While random forest reduces generalisation and prediction error compared to a single decision tree, the isolation forest is used to clean data, catch outliers, identify anomalies, or rearrange the data in subsets based on the isolation criteria. Due to its functionality, it's closer to a semi-supervised or unsupervised approach.

The Gaussian Naive Bayes (GNB) classifier is based on the maximum likelihood estimation theorem and assumes that all the features from the observed data are independent. This classifier can be used both for binary and multiclass prediction, but needs to be retrained each time a new class is presented. An extension of the GNB is the Gaussian infinite mixture model, which estimates a mixture of Gaussian distributions for data clustering [27][28]. Mixture weights prior distribution

can be estimated through a Dirichlet process. Considering the clustering potential on unseen classes, it can be considered a semi-supervised model.

Regarding Deep learning architectures, most models can be interchanged between supervised and unsupervised depending on the last layers and cost functions (e.g. a same structure for a convolutional network may have a softmax/sigmoid output and cross-entropy loss or a reconstructive layer with a distance metric based loss).

2.3.4 Unsupervised models

As previously mentioned, unsupervised learning models find their use with unlabeled data; such models employ clustering algorithms in order to group and classify the aforementioned data, usually basing the decision on distance criteria [29]. The easiest model to implement is the *k-means* algorithm, with k corresponding to the number of classes, or centroids, in which the model will learn to separate the training data. The k value is a hyperparameter representing the number of cluster. It usually correnponds the number of classes in which the data need to be separated; if such number is not known *a-priori*, there are criteria for determining the optimal value (examples are *X-means* through *Bayesian Information Criterion*, *minimum descriptor length* framework to reduce k, *G-means* algorithm to grow the number of clusters starting from a low value of k) [30] [31].

Other unsupervised algorithms are *singular value decomposition (SVD)*[32], *independent component analysis (ICA)*[33], and *principal component analysis (PCA)*[34]; the three of them have a strict relation to one another [35]. The underlying idea for these three methodologies is to apply a transformation to a set of data S defined in $\mathbb{R}^n$, with $n \geq 2$, in order to obtain a new set S' that contains all the information from the previous set but projected in a space that maximises the differences. The dimension of the new set can be equal or smaller than the original set. The transformation and the new components/vectors are estimated just by looking at the data distributions in the original set and with prior assumptions (e.g. in the case of ICA, the set should contain n mixtures from n independent components).

Unsupervised deep learning architectures are usually distance-metric-based models, comparing the input with the output in case of autoencoders or variational autoencoders [36]. In other cases, the loss metrics can be learned by context, as in the case of the Loc2Vec study [37] which employed a triplet loss in a convolutional model with assigned labels on mined triplets on distance criteria during the training process.

2.4 Behavioral biometrics

2.4.1 Behavioural systems on mobile devices

Biometric systems can be classified based on the characteristics observed to compute the recognition; such characteristics can be physical, behavioural, or miscellaneous traits as shown in Figure 2.3. Compared to physical traits, behavioural biometric modalities are far less intrusive and in many cases the user is unaware of the capturing and recognition processes happening during the authentication attempt.

The technology improvements of the recent years, with the spread of wearable and mobile devices with embedded sensors, facilitate the growth of behavioural biometric systems implementation, mostly due to the usability and the ease-to-collect behavioural data. Behavioural biometrics on mobile devices also proved to attain the characteristics of standard biometric systems [38][39], such as *universality, uniqueness, usability, acceptance, collectability, invariance* and *aversity to circumvention*. The aforementioned characteristics are defined as follows:

- *Universality*: Ideally, a biometric feature should be obtainable by every individual. Biometric systems should utilize data that can be collected by the widest population as possible.

- *Uniqueness*: Biometric properties should be distinctive of each individual.

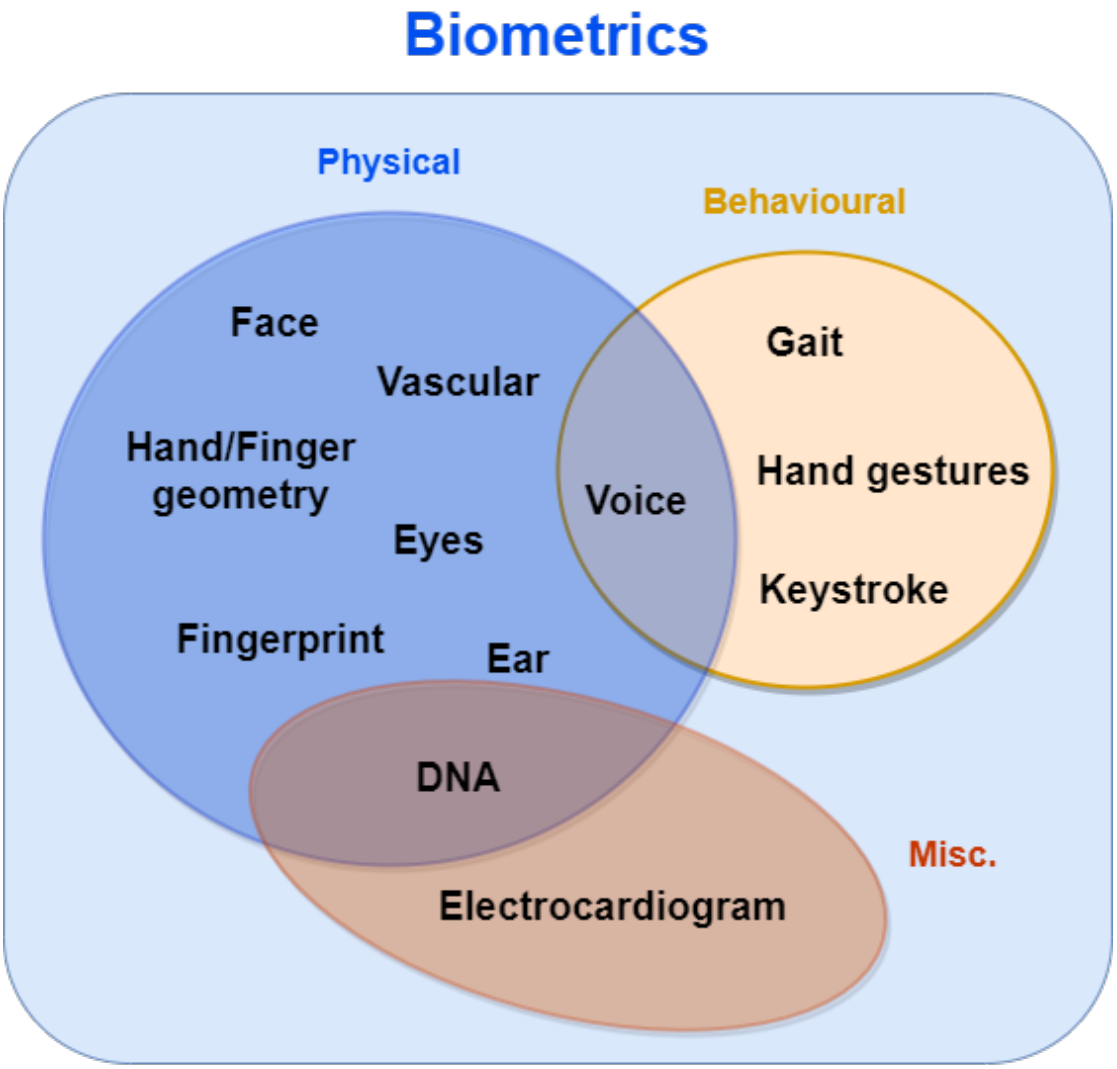


FIGURE 2.3: Different kinds of biometric modalities.

- *Usability*: The capture of biometric data should not generate discomfort to the user, nor be intrusive, nor slow down the flow of action.

- *Acceptance*: The extent of which the population (or diverse sets of populations) is willing to adopt the biometric system.

- *Collectability*: The ability to collect and measure biometric data.

- *Invariance*: Consistence of the biometric features and/or the system performance over extended periods of time.

- *Aversity to circumvention*: The ability of the system to not be vulnerable to spoofing or forgeries.

2.4.2 Capture and interaction: ceremony vs continuous authentication

Due to the dynamic nature of behavioural data, the capture process is not immediate or static, as it would be for face or fingerprint data. This implies the existence of sampling time and the possibility to record more than one sample per session or to have multiple samples of variable length. The sampling frequency and sampling time is directly connected to the sensor and the required task (e.g. for voice recognition it depends by the length of the sentence), but it's also affected by the kind of authentication the system is set to perform: continuous or ceremony-based authentication.

In the first case, the device is constantly monitoring the behaviour of the user in the background, collecting behavioural data and analysing them sample by sample or in batches. From this perspective, the biometric system can be portrayed as an anomaly detector.

In the case of a ceremony-based authentication, the user is prompted to provide a fixed number of samples (it could be one or more) to confirm their identity. This kind of authentication is usuallyperformed during a login attempt to a service, or as part of a two-factors authentication system for any kind of transactions. In these cases, the user may not even be aware of the underlying data capturing and biometric authentication (e.g. the task prompted could be a PIN request, during which behavioural data could be collected). Compared to the continuous authentication method, this requires the least possible amount of data and the verification model must provide a fast and sharp decision.

This study addresses the issue and challenges related to ceremony-based authentication for Swipe, PIN and Electrocardiogram.

2.4.3 Swipe biometrics

With the advent of touch screen mobile devices, many biometric studies have been conducted in relation to touch behaviors, considering different implementation factors and analyzing performance from different classifiers. Most of the swipe-related publications explored continuous authentication only, or evaluated models that require a large amount of training data (quantifiable in many dozens of samples) from the enrolling user, or else algorithms that solve an identification task (which can be applied on a narrow set of individuals).

In 2013 Frank et al. [40] conducted a study on continuous authentication with swipe dynamics, training a SVM and a k-NN classifier, reaching 13% Equal Error Rate (EER) for single strokes and 3% for combined strokes. Also Serwadda et al. [41] studied the performance of various algorithms for active authentication with a large dataset collected during two sessions from 190 subjects, reaching a 15% EER performance with random forest, SVM and logistic regression. Li et al. [42] assessed the performance of an SVM classifier with Gaussian radial basis kernel for continuous authentication on smartphones with 75 participants, achieving a maximum of 95.78% accuracy. Xu et al.[43] recorded touch data (slides, pinch and keystroke) from 30 participants, using a binary non-linear SVM for continuous authentication, varying the number of imposters in the train and test set. Best results were obtained with just one imposter with an average EER of 0.75% for slide and 3.33% for pinch. Feng et al. in 2014 [44] also conducted a study on continuous authentication through touch gestures in different scenarios using mobile devices. With a model based on dynamic time warping (DTW), they achieved 90% accuracy over 123 participants. In 2015 Antal et al. [45] collected swipe gestures constrained on horizontal movement and evaluated the performance of the classifiers based on the number of swipes used for authentication, reaching 0.2% EER with 2- class random forest classifier and 5 swipes.

Miguel-Hurtado et al. [46] in 2016 conducted an analysis on sex prediction based on swipe gestures, extracting 14 features from every swipe and using linear SVM,

logistic, Naïve Bayes and decision tree as classifiers; results showed a 78% accuracy when fusing the different methods. Swipe authentication through pattern recognition with discriminative and statistical methods using SVMs and GMMs has been studied by Fierrez et al. [47], evaluating over different datasets and obtaining a best value of EER equal to 2.6% fusing the 2 modalities. In 2020 Lamb et al. [48] evaluated the biometric performance of three different models (shrunk covariance, Bayesian Gaussian and infinite mixture of Gaussians) considering two kind of attackers. Results are shown in terms of EER over all the genuine and imposter verification attempts across the 32 subject templates, reaching a 4.54% EER considering only blind imposter and 15.7% EER in the case of over-the-shoulder (OTS) attackers.

Table 2.1 summarises the reported studies. As previously mentioned, most of the studies have some underlying issues regarding amount of data required to train the algorithms, authentication time, and generalisation power of the model. The study presented in this thesis addresses these issues and propose new models and perspectives for swipe authentication on mobile devices.

TABLE 2.1: Recent studies on Swipe for biometric authentication on mobile devices, with focus on number of subjects in the datasets and classification methods.

Studies	# Subjects	Classification	Performance Results
Frank et al.[2013]	41	SVM and K-NN	13% EER (single stroke) 3% EER (combined strokes)
Li et al.[2013]	75	Gaussian radial SVM	95.78%
Xu et al.[2013]	30	SVM RBF	0.75% EER (slide) - 3.33% EER (pinch)
Serwadda et al.[2013]	190	SVM and logistic regression	15% EER
Feng et al.[2014]	123	DTW	90% Accuracy
Antal et al.[2015]	71	Random forest	0.2% EER
Miguel-Hurtado et al.[2016]	116	NB, logistic regression, decision tree, and SVM	78% Accuracy
Fierrez et al.[2018]	vary	SVM and GMM	2.6% EER
Lamb et al.[2020]	32	Shrunk covariance, Bayesian Gaussian, infinite mixture model	4.54% EER (blind) 15.70% EER (OTS)

2.4.4 PIN biometrics

Personal Identification Number (PIN) authentication is relatively new as a biometric modality, not in terms of concept but in terms of an hardware-led approach. The PIN is a substitute of a password, with a fixed number of numerical digits.

As such, it appeared initially as a form of plain authentication or login without providing any biometric characteristic. This was due to the digit insert mechanic, relying on physical keyboards. The only features that could be extracted, aside from the raw PIN and considering the setup, were time features on key press and release (this is discussed in details in Chapter 4). For this reason, by a biometric perspective, PIN dynamics are very similar to keystroke dynamics, which have been proven to provide good results in terms of security in the past studies, either for continuous authentication or for ceremony based tasks with passwords [49][50][51][52].

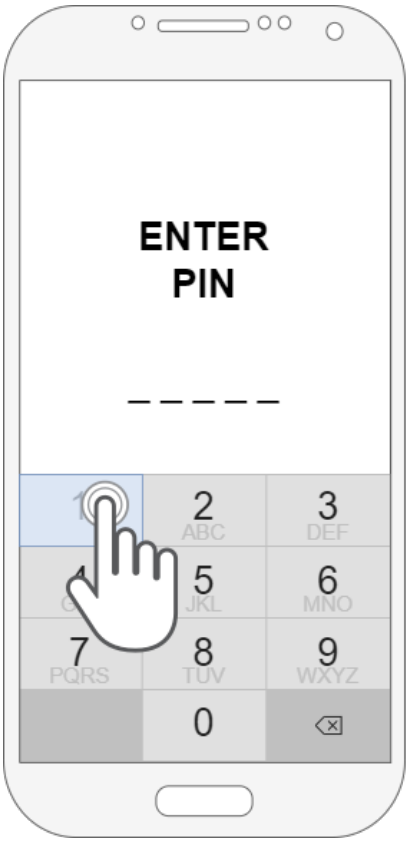


FIGURE 2.4: Example of touchscreen interaction on a PIN request. Finger relative position is recorded during the touch interaction.

As for Swipe, with the spread in the past two decades of the use of touchscreen sensors on mobile devices, PIN data have also been able to record an increased number of features, making it possible to evaluate it as a biometric modality. Figure 2.4 shows an example of PIN interaction on mobile device.

This section describes, as follows, a list of recent studies involving keystroke and PIN dynamics for biometric verification purposes. The first attempt to use keystroke dynamics on mobile devices was in 2003 by Clarke et al. [53]: they tried to identify 16 subjects employing a multilayer perceptron. In 2014 Mendizabal-Vazquez et al. [54] explored biometric verification with a 4-digits PIN training a multi-layer perceptron and using distance metrics on extracted features, obtaining 20% EER with 9 enrollment samples per subject. The same year, Sen and Muralidharan [4] also explored the biometric performance of touch dynamics on a 4-digits PIN, reaching 84.2% accuracy at the cost of a large amount of training samples required from each separate subject (compared to the 6 or 9 from the previous study). Tasia et al. [5] claimed in their study (conducted on 100 subjects) that pressure and finger area on touchscreen devices are two additional features that can improve performance on PIN-based authentication models on mobile devices. In 2015 Wu and Chen [55]recorded keystroke data from a mobile device on unlock-by-password task from 100 individuals and, after feature extraction, they assessed the performance of a SVM varying the training size and considering device movements as extra features. They achieved a lowest EER of 1.25%, but the results are partially biased by the length of the password, the expertise of the participants, the reduced number of genuine subjects from the participants pool, and by ignoring the environmental effects. In 2019, Maiorana et al. [6] adopted a deep learning approach with a CNN architecture to extract deep features from PIN entries and verify the subjects, achieving a minimum EER of 4.5% for a 4-digits PIN testing on a dataset collected by Teh et al. [56]

In Table 2.2 are summarised the aforementioned studies.

TABLE 2.2: Recent studies on PIN dynamics on mobile devices, with focus on enrollment data and number of subjects in the dataset.

Studies	# Subjects	Modality	Enrollment samples	Performance Results
Clarke et al. [2003]	16	4-digits PIN	20/subject	∼15% EER
Mendizabal-Vazquez et al. [2014]	80	4-digits PIN	9/subject	20% EER
Sen and Muralidharan [2014]	10	4-digits PIN	100/subject	84.2% Accuracy
Tasia et al. [2014]	100	4-digits PIN	5/subject	8.4% EER
Wu and Chen [2015]	100 [20 genuines]	Password	vary	1.25% EER
Maiorana et al. [2019]	150 [20 testing]	4-digits PIN	7/subject	4.5%

2.5 Electrocardiogram biometrics

As anticipated in the introduction, during the past decades electrocardiogram data received more attention amongst the biometric community as a possible modality for authentication or verification purposes. Compared to other modalities already affirmed (e.g. face and fingerprint), biometric ECG recognition received an increased interest in the latest years, as a consequence of technology improvements and sensors miniaturisation.

The further sections will resume the progression of ECG devices, including the most recent mobile implementation, and the studies that explored ECG biometrics, evaluating models and parameters affecting the authentication performance.

2.5.1 Devices

ECG devices are comprised of various parts, each with specific requirements [57]. The signal can be recorded with invasive (i.e. needles) and non-invasive (i.e. foam Ag/AgCl electrodes) sensors. In both cases, the sensors need to be in direct contact with (at least) the skin of the subjects. This has implications not only regarding the materials used for the sensors, but also in terms of safety measurements for the whole system (sensors, cables, power supply, etc.). Nevertheless, compared to the first electrocardiogram prototype machine from 1906 [58], sophisticated and miniaturised sensors have been released in the past few decades. Recent advances in technology allowed to record the cardiac signal using wearable devices, in compliance with regulations.

There are four main components to consider when choosing an ECG recorder, depending on the usage: *electrodes*, *analog front-end*, *power supply* and *digital signal processing*.

- *Electrodes* count a wide variety of kinds, most of them developed in the last few years. Conventional metal electrodes are comprised of a metal mixture disc (generally silver-silver chloride) in direct contact with the skin through

an electrolyte gel to reduce impedance. Figure 2.5 shows the schematic of the equivalent circuit of a standard electrode[59][60].

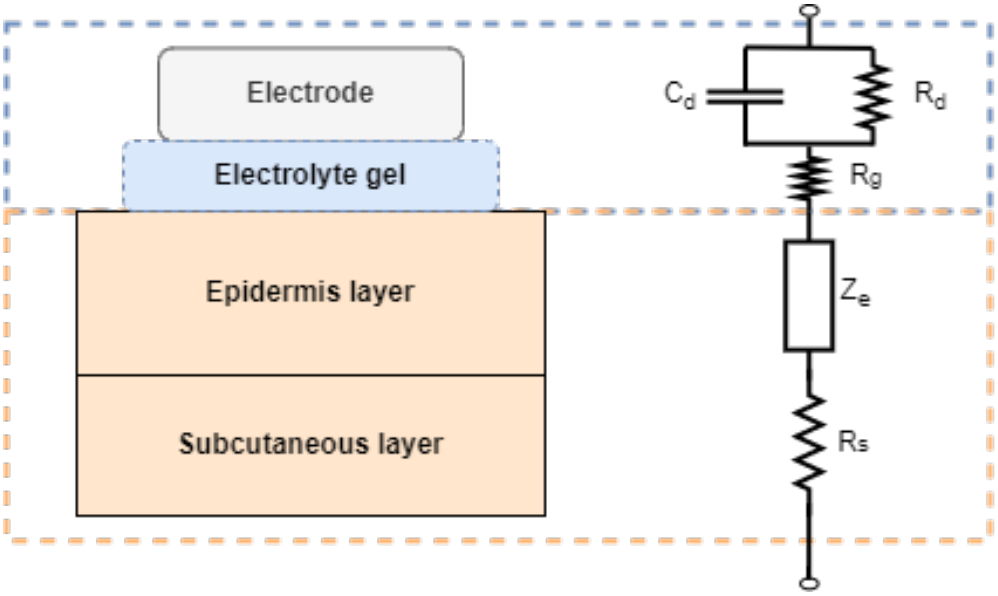


FIGURE 2.5: Equivalent circuit for a metal electrode.

R_s and R_g refer to the corresponding subcutaneous layer and electrolyte gel resistances, both with very low impedance compared to the epidermis (which is a mixture layer of connective tissue and sweat glands). Z_e is the total impedance of the epidermis layer, comprising capacitance and resistances from the various materials. At the interface, the ions generated by the heart electrical activity react with the metal and produce a potential difference [61]. C_d and R_d represent respectively the capacitance and the resistance between the skin and the metal electrode.

These electrodes are the most common and widely used, especially for diagnostic purposes, but are impractical for continuous recording over a long period of time. In this scenario, the gel would dry too quickly and the electrode itself would be affected by change of impedance; moreover, classical electrodes are poorly flexible and require cable connection. Latest solutions comprise invasive micro-needle electrodes array [62], dry electrodes [63] with various shapes (e.g plate or disc shaped) and textiles electrodes integrated in wearable garments [64]. From a wearable perspective, the latter options are the most likely to be applied in a real life scenario at consumer level:

dry electrodes are already in use with small strap bands and smartwatches, while textile electrodes can be embedded in smart clothing.

- The *analog front-end* components of an ECG recorder are dedicated to buffer and amplify the feeble signal and apply an initial noise reduction strategy. For safety reasons, the circuits are not directly connected to the power supply. The circuits usually comprise an instrumentation amplifier (Figure 2.6) and filter batches for initial noise removal and signal smoothing.

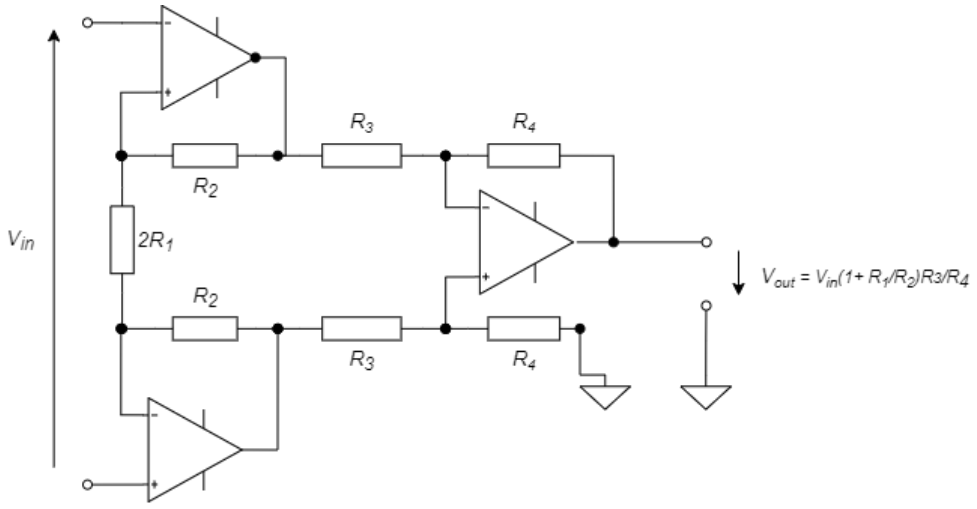


FIGURE 2.6: Schematics of a generic instrumentation amplifier. The first two buffers are connected to the electrodes.

For recording systems with 2 electrodes a different configuration might be implemented, with just one amplifier and a voltage divider to attenuate the common-mode voltage. In general the resistors connected to the source (in this case, the electrodes) should be very large (e.g. $1G\Omega$) in order to minimise motion artifacts [65].

- The *power supply* requirements have changed a lot across the past decades, with the spread of integrated circuit technology that allowed the production of portable devices. The former power source was the domestic power-line (220V, 50-60 Hz), decoupled and with reduced voltage through an isolation transformer. But for portable or integrated solutions, new options have been available (e.g. lithium batteries and low-voltage suppliers)[66][67]. In this scenario, the specifications shifted from the power supply itself to circuits

able to operate for prolonged times under a low voltage supply (e.g 1V to 8V batteries)[68][69]. In addition, rechargeable solutions have also been explored, like solar panel ECG recorders [70].

- *Digital signal processing* units for ECG data, in addition to digitisation of the signal, have two main objectives: clean the data and transfer it to another device (e.g. a storage, a cloud service, a server). Classical methodologies involved band pass [71] and Notch filters [72] followed by signal modulation [73]. More recent devices incorporate filtering, peak detection, and signal transmission through wavelet transform [74][75][76]. Latest wearable ECG recorders have also an integrated WiFi or Bluetooth board and can digitise the signal at 16 or 18 bits[77][78].

Considering all of the above key features of an ECG recorder, in 2021 Cosoli et al.[79] published a review of commercial wearable wireless devices for cardiac monitoring. The study provided an insight of commercial recorders shortlisted on a basis of electrode type (most of them being dry non-invasive electrodes), pricing and quality of the recording. Devices providing a chest-strap band with embedded electrodes (e.g. *Quasar*[80] and *Qardio*[81]) can ensure continuous monitoring and ideally can work in the background, provided a stable WiFi or Bluetooth connection with the receiving interface. This is very useful in a biometric scenario, especially considering a continuous authentication task, and it's preferred to devices connected to the interface through cables and wires.

2.5.2 ECG biometric authentication

Many studies have explored biometric performances with ECG in the past years, but mostly considering an identification task or continuous authentication. The amount of data for training and testing is also to be taken into consideration, as the previous studies didn't address the recording time in a realistic scenario; another related issue from previous works is the lack of diversity in the data to

proper generalise. Page et al. [82] in 2015 used deep neural networks for ECG-based biometric identification systems through an ECG waves detection method, isolating the QRS complex from the recorded ECG signal and using it as feature vector for the CNN. This research involved 90 participants and collected data using 1-lead ECG recorder for 20 seconds with a 500 Hz sampling frequency. The study resulted in a 99.96% identification rate and 0.0582% equal error rate (EER), considering 70/15/15% training, validation and testing data. This study was significant because it showed that prediction with QRS techniques gives better results compared to non-QRS techniques. Although their results were promising, the lack of samples per subject may have resulted in over-fitting.

Biel et al. [83] who used 85 samples for training and 50 samples for testing, found 100% accuracy but they used a 12-lead ECG recorder to identify 20 participants. As such, it doesn't provide any solid statement for mobile device authentication, due to the entity of the device used and the small number of participants involved in the study. Israel et al. [84] used a Linear discriminant analysis (LDA) method to identify 29 subjects with 98% accuracy. They used fiducial points to extract a vector of 16 features from each heartbeat, averaging over 20 seconds of ECG recordings from the original 14 minutes per subject, sampled at 1000 Hz.

Wang et al. [85] used ECG recordings from 13 participant to perform authentication with K-nearest neighbours (K-NN) and LDA methods. The study reached 96% accuracy rate with 50% training samples size from all samples. Chiu et al. [86] used discrete wavelet transform and LDA analysis on 35 subjects of ECG signals and found 100% verification rates. 2 minutes of signals were chosen for training and testing separately. Chan et al. [87] recorded 270 seconds of data during 3 different sessions from 50 participants, then employed wavelet distance measure algorithm to assess biometric verification. It resulted in 89% authentication rates with 90 seconds of training and 180 seconds of testing data. Sriram et al. [88] used wavelet and LDA classifiers with approximately 4 seconds of test data and 400 seconds of training data per person.ECG signals were collected from 17 participants during different activities. The study achieved 88% verification rate. Shen et al. [89] used data from 20 participants for authentication, with 400 training

and 200 testing heartbeat samples using template matching and decision-based neural network (DBNN) systems. The study reached 95% authentication rates for template matching, 80% for DBNN and 100% for their combinations. Wieclaw et. al. [90] used 3 finger ECG measurement with a deep neural network system with 30% testing and 70% training samples for biometric identification. In [91], a deep learning (DL) algorithm gave 98.4% identification rates from wearable single-arm ECG. Bajpai et. al. [92] evaluated WESAD dataset with different training time and different K size to classify different stress levels. They found their best results as 90% accuracy on K=5 with 30 minutes training sample size. Bobade et. al. [93] compared the results of DL and machine learning (ML) systems to find stress level on the WESAD dataset. They found between 81.65% and 93.20% accuracy rates on ML and between 84.32% and 95.21% accuracy rates on DL. In Table 2.3 are listed the aforementioned studies.

TABLE 2.3: List of recent biometric studies on ECG authentication.

Studies	Features	# Subjects	Recording Duration	Accuracy Results
Page et al.[2015]	QRS	90	20 seconds	99.96%
Biel et al.[2001]	Fiducial points	20	4, 5 , and 10 minutes	100%
Israel et al.[2005]	Fiducial points	29	14 minutes	98%
Wang et al.[2008]	Temporal points	26 (13, 13)	vary	96%
Chiu et al.[2008]	QRS	45	15 minutes	100%
Chan et al.[2008]	Non-fiducial	50	270 seconds	89%
Sriram et al.[2009]	Fiducial and non-fiducial	17	12-15 minutes (5-7 testing)	88%
Shen et al.[2002]	Fiducial points	20	48 hours	95%
Wieclaw et al.[2017]	DL features	18	147 x 10 seconds	88.97%
Zhang et al.[2017]	DL features	10	26 minutes	98.4%

All the results from the previous works are promising, but each study lacks a realistic scenario, and in the majority of the cases a very small pool of subjects participated in the study, leading to over-fitting and unreliable data.
To consider a real-life application, it's necessary to assume that only a single ECG channel would be recorded from a mobile device in a short time window.

For a matter of usability, it's unlikely that the ECG track would be used for continuous authentication. The most probable situation is a verification task when login to a profile or for identity confirmation.

2.6 Multimodal biometrics

In many cases, to improve security during an authentication task, more than one biometric modality is employed by the verification system. This is the case when two or three (or more) factors authentication are requested. The different modalities can prompt at the same time (recording multiple data types from different sources), in background or in cascade. This is finalised to have a better performance in terms of match and non-match rates[94], and to do so different fusion methods can be applied at different Levels [95] (this will be discussed more in details in Chapter 4.3.2).

Applying fusion on sensor level and therefore on raw data is complex to perform, especially with different modalities, since it requires to blend together different data types. More common fusion methods involve feature Level fusion [96][97], which can be performed by either classical machine learning algorithm or tailored deep learning architectures, especially in case of variable length features [98]. Many studies have been conducted on score Level fusion, applying linear combinations, considering mean, maximum, or minimum score from different classifiers as final score[99][100][101], or applying probabilistic techniques [102][103]. On decision Level, the most common fusion methodology is the majority vote, which returns a decision based on the highest occurrence amongst the labels from all the classifiers employed [104]. Other methodologies at decision Level comprise hierarchical and parallel fusion modes [105][106].

2.7 Trade-offs and open challenges

The aforementioned biometric studies regarding the three focused modalities (swipe, PIN, and ECG) present many promising performance results, hence it's important to highlight the trade offs and issues that emerged from the various experiments. Such issues are common to all the related studies (see Tables 2.1, 2.2, 2.3) and will represent the core factors of this thesis.

2.7.1 Training data

The amount of training data required for each model is very large, especially when using statistical models that don't provide a subject template but need to retrain for each new subject. This is a critical issue, considering that a poor training (in terms of amount, quality, and variability of the data) can lead to an overfit model unable to properly generalise. Moreover, in a realistic scenario it's unreasonable to ask the users to provide a large amount of training data for enrollment (this statement can be extended to any modality), especially when the data recording is prompted and not executed in the background.

2.7.2 Sampling time

This issue is related mostly to electrocardiogram biometrics, considering that for PIN the sampling time is equal to the input from the subject and for swipe it depends exclusively from the number of samples considered for the authentication. In this perspective, ignoring server connections and response time, the authentication time is strictly connected to the sampling time, which means that the least data are recorded, the faster the authentication will be. In the previous studies this issue has been pointed out but never properly addressed. Recording times for ECG are always very long and not fit for ceremony based user verification.

2.7.3 Ease of use

This issue is equally related to the modality itself, in terms of tasks and dedicated softwares, and to the device in use. In case of Swipe and PIN biometrics, as mentioned before when discussing the training data, requesting many samples to perform authentication discourages the usage of the modality. In the case of ECG data, this is even more demanding if the device is connected with many wires or if it's too big in size. The recording time needs also to be taken into consideration. In favor of usability, the device should not be a source of discomfort and the task

should be as quick and simple as possible (and this excludes all the experiments comprising repeated actions or long recording time).

2.7.4 Stability

A good biometric system should maintain a steady performance, regardless of the circumstances. Model stability should be invariant to environmental factors, to change of devices, to time from enrollment, and to the subject itself. In a realistic scenario, this is not possible and all these factors will affect the performance. Previous studies (see Tables 2.1, 2.2, 2.3), even when considering these issues, didn't always address them properly: in some cases multiple sessions or different sources were not taken into consideration, or the enrollment was not chronologically happening strictly before the verification, or the data were lacking diversity (reducing the generalisation power of the classifier).

2.8 Summary and research questions

In the recent years, many studies focused on mobile authentication through behavioural biometrics. Most of them assessed continuous authentication only, and in many cases the enrollment time or number of samples required for authentication has not been considered (or just marginally taken into account). Considering the literature alongside the aforementioned challenges, this research addresses and solves the following questions:

Is it possible to obtain a steady performance on mobile biometric authentication with behavioural (PIN and Swipe) and electrocardiogram data ? This implies a model able to generalise and maintain a stable performance with unseen data from new subjects even after prolonged periods of time.

Is it possible to perform authentication on a ceremony based task with just one sample request? With this, we are considering a real life scenario reducing at the very least the amount of time spent by the user to verify their identity, which

would result in a single swipe or PIN attempt and the lowest time to record a valid ECG signal.

Can the performance be improved by combining the aforementioned modalities? Fusion methods and multimodal biometrics provided an added level of security and reduced authentication mismatches. In this research we are evaluating optimal fusion methodologies to blend the modalities, taking into consideration the entity of the data, the position in the framework and the implementation cost.

To address all these challenges, we are proposing a framework for ceremony-based authentication, with constrained and semi-constrained tasks for Swipe and PIN modalities and on-demand ECG recording in the background.

2.9 Proposed Framework

In this thesis we propose a framework for a ceremony-based biometric authentication with the combination of three different modalities: Swipe, PIN and ECG. Figure 2.7 shows the general idea behind this project on a real life scenario.

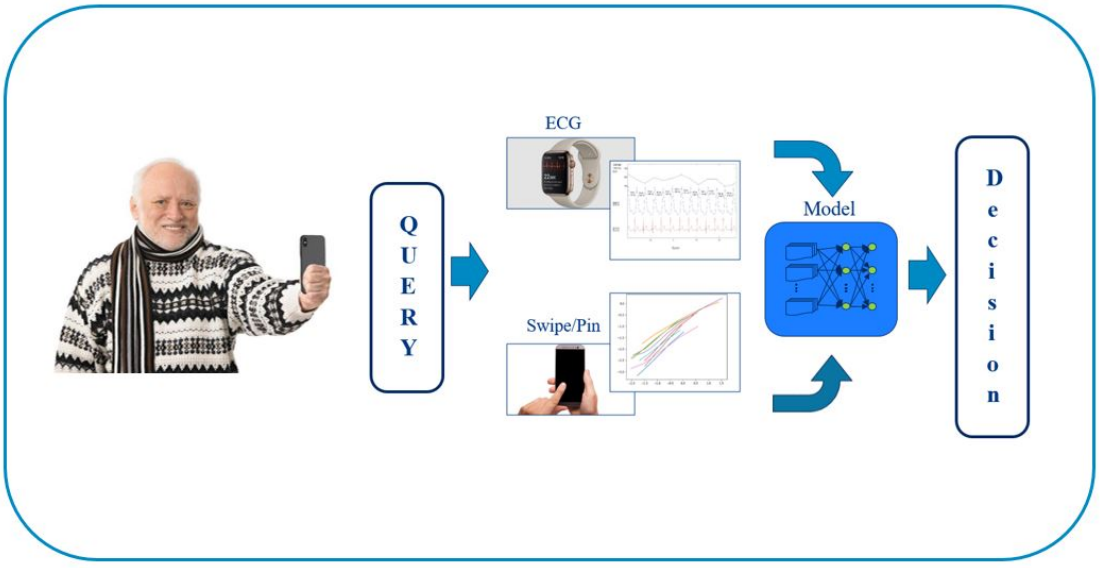


FIGURE 2.7: Multi-modal authentication, combining the three modalities (Swipe,Pin and ECG) in this study. The user tries to login to a service provider (e.g. a bank account) and the server requests a multi-modal biometric authentication.

The study will first explore the modalities separately, assessing the single biometric performance with the least possible amount of data for enrollment and verification. We then evaluate fusion methods to blend the modalities and improve the overall performance. We also considerate the specific order in which the different modalities are prompted. Figure 2.8 shows the proposed framework, comprised of all the stages. To achieve this, we divided the study in sub-phases:

- *Phase 1*: Swipe authentication model assessment. Performance evaluation of Swipe model: with verification performed on one sample.

- *Phase 2*: PIN authentication model assessment. As per *Phase 1*, the performance of the authentication model is evaluated on PIN data alone. This modality is considered to be prompted after an hypothetical non-match from the Swipe model.

- *Phase 3*: 2 factors authentication with Swipe and PIN fusion. Evaluation of a fusion model based on the fusion of the two modalities considering their cascade structure.

- *Phase 4*: ECG authentication model assessment. Performance of ECG authentication models, considering short input signals that would be ideally recorded in background during the Swipe and/or PIN tasks.

- *Phase 5 (theoretical)*: 3 factors authentication with fusion models evaluated at various points in the workflow. Due to Covid-19 and the impossibility to conduct a data collection with paired sensors, this part of the study is entirely theoretical, but it's based on the results furtherly presented in this thesis, related to both single performances of the models and fusion performance. A series of approach finalised to fuse the three modalities are proposed for future work.

From the following Chapter, we will start evaluating Swipe biometrics.

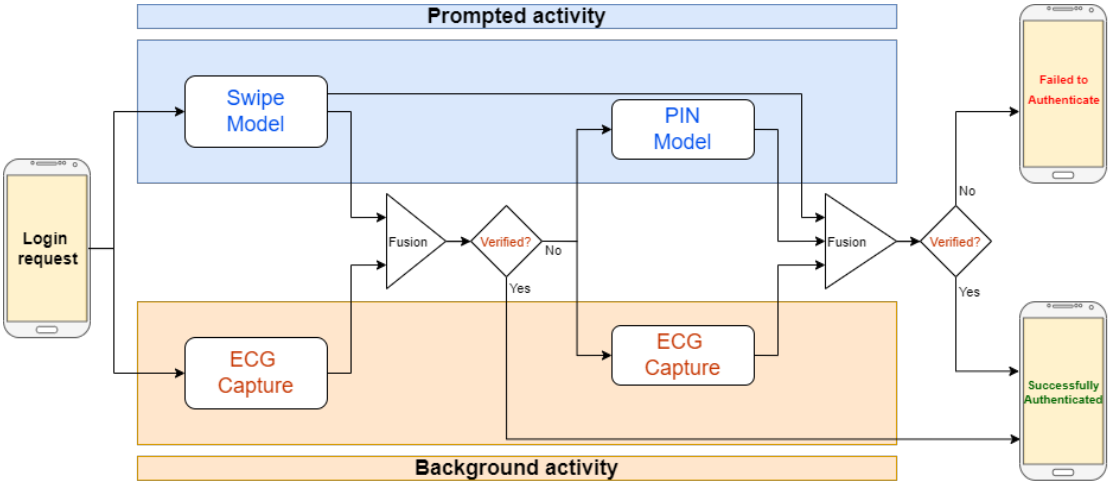


FIGURE 2.8: All the stages of the framework. Pin and the second ECG recording are prompted only after a failure from the previous modalities.

Chapter 3

Swipe biometrics

3.1 Swipe Authentication

3.1.1 Introduction and Framework

In the last years, swipe dynamics has been a recurrent topic, regarding continuous authentication on mobile devices. The reason for this can be addressed by the exponential growth of touch screen devices and by the fact that swipe gestures are the most common and habituary actions performed by a subject when using such a device. On a data perspective, there's an obvious analogy between swipe gestures and signature: both contains data points mapping over time on a xy plane (the screen), with the main differences being that the signature is more constrained and usually consists of more sample points, making it easier to generate a subject pattern. Since swipe gestures are less constrained and can be affected by a multitude of factors (environmental and interaction based, such as subject swapping hand or interacting with the device in different conditions) [107], previous studies have focused on continuous authentication over prolonged periods, in some cases considering a penalty score (defined as a decaying score over repeated unusual behaviours from the subject) [108].

The aim of this particular investigation described in this chapter was to provide verification of a query, using just one single swipe; for this purpose, we relied on a semi-constrained scenario, with the subject only able to slide the screen on a particular side, reducing the variability of the gesture. The steps of our framework are the followings:

1. The user provides few samples (10) to the trained classifiers to enroll and create their own template.
2. Data are processed and the new template is generated.
3. When the authentication request is prompted, the subject provides one single swipe.
4. The swipe undergoes processed and compared with the template.
5. A verification result is returned.

This describes the first step of our multi modal fast verification framework, with minimum amount of data required. The remaining pieces of the complete framework will be shown in further sections.

3.1.2 Data overview

Swipe recordings are usually collected from touch screen devices, such as mobile phones or touchpads. The recording starts when the finger presses on the screen until it is released. The least amount of data that can be obtained are x and y position on screen, paired with the corresponding timestamp; we will consider each pair of x-y position points and timesteps. For each timestep, a pen down boolean value can be assigned during the recording, to check when the swipe starts and when it ends.

Another data pair that can be obtained is the finger pressure on screen (usually normalised), and the finger size (usually express as pixel area or length on major

finger axis). The last two values are particularly relevant, since they refer directly to the subject's body and not their general behaviour (although the values might be misleading if the subject does not operate the device with the same finger consistently).

Those values refers directly to the swipe itself, but there is other information that can be stored and assessed when processing swipe data for biometric recognition. Additional data can be obtained from accelerometer, magnetometer and gyroscope sensors which will give information on the device state and position; it's important to point out that the reliability of these data is dependant on the sensor capability and by environmental conditions (electromagnetic sources, vibrations, etc.). Moreover, these sensors usually have different sampling frequencies, both between themselves and the touchscreen sensor itself, and this must be accounted when processing the data.

Other useful data are from GPS sensors, to confirm that the device is actually where "it's supposed to be" at the time of the authentication and not exploited with a virtual machine. Also, the device model can be extracted, which can be not only used as an extra security measurement but also an useful information to draw statistics.

3.1.3 Public and Callsign datasets

Across our experimentation we used a range of swipe datasets, public and private. We mainly focused on data provided by an industrial collaborate, Callsign, since it will be descriptive of the real-life scenario in which the biometric authentication would be applied. In this subsection the datasets (*Serwadda, Antal, Frank and Callsign*) will be described and compared, providing demographic information when possible.

Serwadda Dataset:

This is a public database collected at Louisiana Tech University by *Serwadda et*

al. [41] from 190 different subjects of different ages with one smartphone (Google Nexus S.). The data were collected through two applications, asking the subject multiple questions and allowing free interaction with the touch screen. Touch data were recorded considering only one finger touch and ignoring other interactions with multi-touch like zoom gestures. Features collected were *x and y coordinates, timestamp, pressure, and finger area*. Data collection was split in two sessions at least one day apart, the overall number of strokes per user was around 80. Amongst the three public datasets used, the Serwadda database is the largest in terms of number of samples and subjects.

Frank Dataset:

This database was collected by *Frank et al.*[40], and it is composed of swipe data collected in two sessions (1 week apart) from 41 subjects. Data have been acquired from several different Android devices and two applications have been developed for the purpose. Users were free to interact with the screen. The applications captured *x and y coordinates, pressure, finger area, timestamp, device orientation and finger orientation*. It is also important to note that not all swipe directions result in the same number of samples. For example, the *Down* direction contains the most samples.

Antal Dataset:

The final public databases used, it is composed of horizontal and vertical swipe data collected from 71 users on eight different mobile devices [45]. An application was developed for this purpose and the strokes were task related (vertical to read text, horizontal to choose pictures). The data was collected in one single session with each user interacting with multiple devices. The same features of the previous databases were collected. It is important to note that the majority of swipes in this dataset are horizontal, while the least amount is found in the *Up* direction.

Callsign Dataset:
Callsign provided various sets of swipe data with a similar structure. The only differences are the number of unique subjects and the labelling, with later dataset containing information about device model and screen size. Our study focused on the final dataset provided by Callsign for training and testing.

The data are stored in nested dictionary format, separated according to user IDs and transactions; each transaction can be considered a single swipe. The recorded features for each recorded point are: *timestamp* (for each recorded point), *X and Y positions* on screen, *Accelerometer and Gyroscope*, and *Major Axis* of the touching finger. The recording is performed on mobile devices on verification query as a ceremony task . It's a semi-constrained task, where the user is requested to perform a horizontal swipe (either from right to left or left to right) in order to move the rectangular screen. A guided path is not provided (like, for example, the "*slide to unlock*" of Apple devices). The dataset contains 150 users of mixed ages and genders.

3.1.4 Pre-processing

Regardless of the authentication system, all raw data went through basic pre-processing, to provide a first level of cleaning after the capture. We used as a reference the last Callsign dataset to define the basic criteria, and to identify the features useful for biometric authentication.

The first step is to remove *NaN* (not a number) values from each row; this usually happens when the touch sensor does not record anything due a non capturing event, capture lag, or if there's some other sensor (i.e., gyroscope) with a higher sampling rate (which will add extra timesteps rows).
In our study, we decided to directly remove gyroscope and accelerometer data,as our study wanted to asses touch information explicitly.

After clean up, we then separated individual swipes actions related to on individual capture records. If the start and the end of each swipe sequence were not flagged,

we considered the time difference in milliseconds between rows according to the timestamp: if it was higher than a threshold value, we assumed that the data rows belonged to different swipe sequences. Ideally, the value should be equal to the inverse of the sensor sampling rate, but this could lead to errors in the case of missed recordings; for this reason, we decided to empirically set the threshold to 50 *ms*.

The next step was a first removal of swipe outliers (either too short or too long compared to the majority, in terms of data points). As suggested by previous studies in literature [109, 110] , we removed all swipes with less than 4 data points. For the upper bound, the decision was more complicated: we avoided capping the length of the swipe at the mean or the max value across the length of all the swipes in the dataset, since it would have removed half of the samples in the first case and samples would have been retained with a large number of data points in the second case. We decided to rely on the 95^{th} quantile of all the swipe lengths as a threshold for sample removal. The reason was to reject only the 5% of outliers with the biggest gap in terms of sequence length.

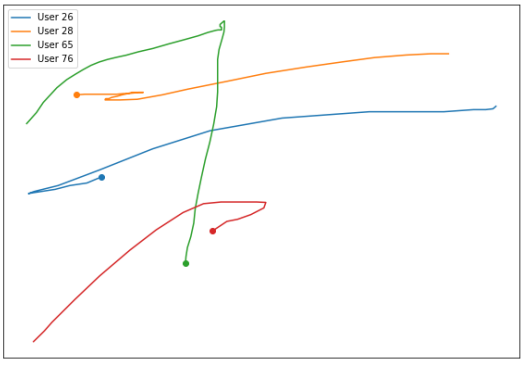


FIGURE 3.1: Examples of rejected back-and-forth swipes from few subjects. Different colors represent different swipe gestures. The dot indicates the beginning of the swipe.

The only swipes left to reject were the ones not conforming to the requested task for authentication, hence either vertical swipes or back-and-forth ones (see Figure 3.1). The first case (vertical swipes) was easy to reject, simply by considering

the position variances in the two directions (X position data should vary way more than Y position for horizontal). For the second case (back-and-fort swipes), we developed an algorithm that could be extended to detect anomalies in any preferred direction. We will describe the operation for the specific case of the X direction and then extend it to the general case.

Considering a swipe S defined as a set of pairs with cardinality N, each pair describing the position on the screen at point i, defined as:

$$\begin{gathered} i \in [1, N] : S = (x, y) \in \mathbb{R}^2 \\ N = \#(S) \end{gathered} \tag{3.1}$$

Also, assuming that the swipe is developed primarily on the x direction, ignoring y, we can calculate the maximum distance travelled on the screen d , defined as $d = max(x) - min(x)$.

Fixing the parameter α as a percentage value of tolerance on the travelled distance d, we can calculate the two threshold points (t_1 and t_2) on the swipe than cannot be crossed for the swipe to be accepted as follows:

$$\begin{aligned} t_1 &= min(x) + d \cdot \alpha \\ t_2 &= min(x) + d \cdot (1 - \alpha) \end{aligned} \tag{3.2}$$

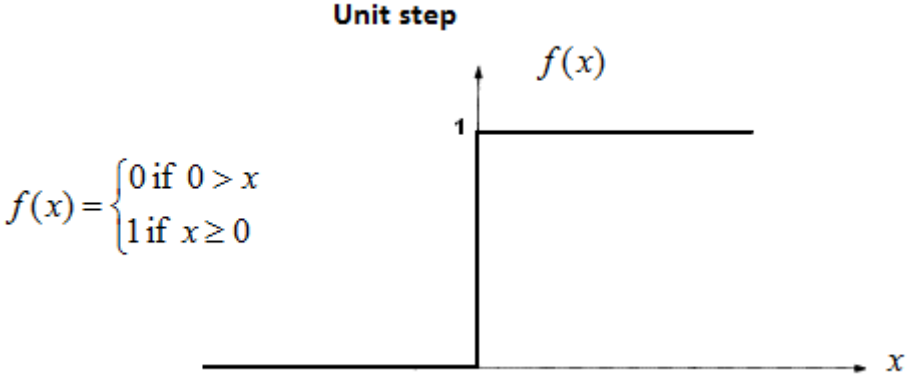


FIGURE 3.2: Heaviside function response.

With all those values, we can calculate the number of crossings C on the two threshold points (considering H the Heaviside function, see Figure 3.2):

$$C = \sum_{i=1}^{N-1} [H(x_{i+1} - t_1) - H(x_i - t_1) -] + [H(x_{i+1} - t_2) - H(x_i - t_2)] \tag{3.3}$$

If $C > 2$, the sample is rejected. We called this algorithm the *shifted zero crossing*.

For the general case, we need first to project the swipe on the x direction. To do so, we need to estimate the directrix of the swipe and calculate its versor v. The next step is to calculate the angle θ between our objective direction and the swipe directrix, as:

$$\theta = \arccos(x \cdot v) \tag{3.4}$$

With this, we can generate the inverse of the rotation matrix R:

$$R = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{3.5}$$

and multiply it by the swipe points to get a new swipe developed on the x direction. Then, algorithm 3.3 is applied. Figure 3.3 shows an example of the linear transformation applied before the shifted zero crossing.

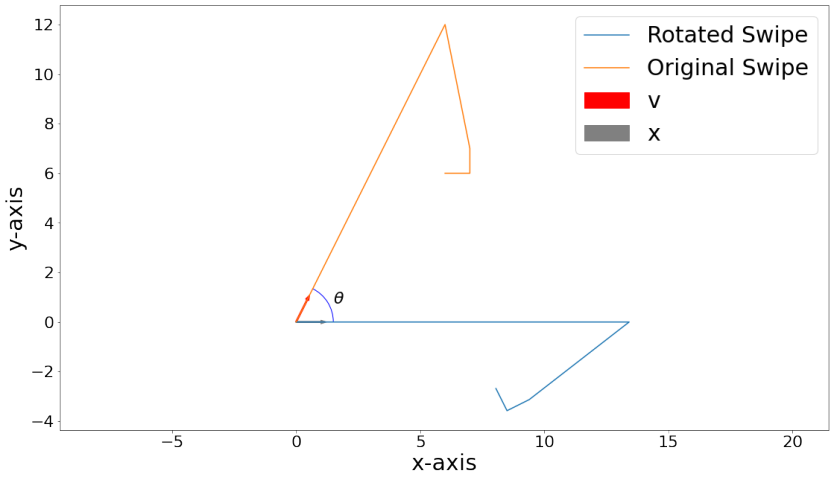


FIGURE 3.3: Example of Swipe projected on the x-axis before using the shifted zero crossing. x and v are the two versors. Without the transformation, the original swipe wouldn't be rejected.

For our purpose, we rejected samples with more than one crossing at a distance from the tail major than 15% of the travelled distance ($\alpha = 0.15$). This value was heuristically determined based on visual clues (plotting the swipes) and by comparing the improvements in performance at different chosen thresholds.

3.1.5 Feature extraction

After cleaning the database, we extracted two sets of features from each sequence and we rearranged the swipes in two more data structures, one for the training and testing of global feature-based algorithms and another for a recurrent neural network architecture. The first set of features (mostly based on [38, 47]), which we will refer to as Global Features, is representative of behavioural characteristics of a swipe sequence on a global scale (with global we mean over the whole duration of the recorded gesture).

These features are shown in Table 3.1.

TABLE 3.1: Global features extracted from swipe gestures.

Identifier	Feature name	Description
1	Max X	Maximum value of the X-coordinate.
2	Min X	Minimum value of the X-coordinate.
3	Mean X	Mean value of the X-coordinate.
4	Max Y	Maximum value of the Y-coordinate.
5	Min Y	Minimum value of the Y-coordinate.
6	Mean Y	Mean value of the Y-coordinate.
7	Max Finger Area	Maximum value of the area described by the finger recorded during the swipe.
8	Travel Distance	Cumulative Euclidean distance between the start and the end of the Swipe.
9	Swipe Length	Number of recorded points during the swipe.
10	Global Slope	Slope angle between start and end of the Swipe.
11	Cumulative Slope	Sum of local slopes occurring during the Swipe.
12	Max X Velocity	Maximum velocity recorded over the X-coordinate.
13	Min X Velocity	Minimum velocity recorded over the X-coordinate.
14	Mean X Velocity	Mean velocity recorded over the X-coordinate.
15	Max Y Velocity	Maximum velocity recorded over the Y-coordinate.
16	Min Y Velocity	Minimum velocity recorded over the Y-coordinate.
17	Mean Y Velocity	Mean velocity recorded over the Y-coordinate.
18	Max Swipe Velocity	Max velocity recorded over travelled distance.
19	Mean Swipe Velocity	Mean velocity recorded over travelled distance.
20	Total Swipe Time	Difference in Time between start and end of the Swipe.
21	Swipe Direction	Direction in which the swipe was performed according to Start and End (1 if from left to right, -1 if from right to left)

Maximum, minimum and mean values are calculated over the whole Swipe after shifting and scaling the values based on the screen size. Travel distance is calculated summing all the Euclidean distances between *Delta X* and *Delta Y* (difference between consecutive X and Y values) coordinates from first touch point (Swipe START) and last touch point (Swipe END). Global slope was calculated through the first derivative of Y-Coordinate with respect to the X-Coordinate considering only the first and last touch points. Cumulative Slope was calculated summing the first derivatives of Y with respect to X for every pair of touch points during the entire swipe movement. Min, Max and Mean Velocity of coordinates and directions (for Swipe velocity) were calculated on the point-to-point first derivative of the considered coordinate or direction respect to the delta timestamp. Swipe direction was calculated considering the sign of the difference between starting point and end point of the swipe. Max finger area (or in some cases, max major axis value of that area) is a novel feature that was calculated as the highest value of touch area covered by the finger recorded during the swipe; minimum value has not been taken into consideration because it would be affected by initial or accidental interactions and con-strained by screen resolution, therefore not relevant for authentication.

The extracted 21 features corresponded to a single swipe and a single row in the data structure that would later be fed as an input to a conventional classifier and not a sequence based deep learning model.

TABLE 3.2: Raw features from swipe gestures.

Identifier	Feature name	Description
1	Timestamp	Touch event timestamp.
2	X-position	X-Coordinate values at touch point.
3	Y-position	Y-Coordinate values at touch point.
4	Finger area/major axis	Area (or major axis) described by the finger on the screen.
5	Pressure	Pressure on screen at touch point.
6	Sequence length	Number of touch points for the given swipe gesture.

The second set of features, shown in Table 3.2 referred to as Raw Features, was saved in another three-dimensional data structure and used only for training and

testing of a recurrent neural network. This was due to the different input requirements in terms of input shape.

- *Timestamp* is expressed in milliseconds and the first value of the sequence is subtracted from each data entry point, so the time for each swipe will start from 0.
- X and Y positions are the raw values of x and y coordinates for each touch point. We also used Delta positions, this was due to the fact that the initial position on screen can vary easily, but ultimately we concluded that it could be a valid feature description, not only for subject behaviour but also with respect to their physiology; Delta values could also be calculated by the deep learning model, so it was worth maintaining original values.
- *Finger area/major axis*, as described before, refers to the portion of the screen touched by the finger at each touch point.
- *Pressure* is a value capped to a maximum defined by the sensor, describing the intensity of the pressure applied by the finger on the touchscreen.
- *Sequence* length is the original number of touch points from the start to the end of the swipe, not considering the padding (defined as adding zero values to complete a sequence). This is particularly useful, since the data are saved in a structure with fixed size.

The first dimension of the structure equals to the number of rows for each swipe gesture of the dataset ,fixed after preprocessing considering the 95^{th} quantile of the lengths of all swipes in the dataset. Sequences with fewer data points are padded during the creation of the data structure, but we wanted to preserve the original length to avoid introducing the bias of the padding in the training procedure.

As a last step of pre-processing, we normalised the two sets of features applying a z-score normalisation to all samples. This normalisation is a suitable choice when the data lacks globally defined boundaries, which prevent min/max scaling, but

on the downside, it is influenced by the size of the dataset. When upper level boundaries were available, we normalised the raw features as follows:

- x and y coordinates scaled and shifted based on the screen size and axis origins.
- *Finger area* scaled by the screen area; considering H = Height of the screen and W = Width of the screen, we divide it by $H \cdot W$.
- *Major axis* scaled by the diagonal D of the screen, with $D = \sqrt{(H^2 + W^2)}$

It's important to mention that an anonymous unique ID was assigned to each subject in the dataset and appended as the last column of each sample provided by the same subject. All the samples were stored chronologically in the data structure, according to the date and time of the session when they were recorded.

3.1.6 Evaluation of previous methodologies

Before introducing new deep learning architectures, we evaluated previous models for swipe authentication with some exclusion criteria based on existing publications, project scope and data structure. At the beginning of this project, other studies had been published analysing the issue of the number of swipes required for authentication [45] or the impact of certain soft biometrics on classification performance [46]. Nevertheless, in many cases the validity of the studies was biased by the databases and their recording procedures. Moreover, the performance of models like k-nearest neighbours (K-NN) is proportional to the size of the training dataset and how well they represent the database population. In a realistic scenario, these kinds of models would require continuous retraining for each new subject with no guarantee of improvement in performance.

For those reasons, K-NN was excluded from evaluation alongside one-class classifiers. Even if one-class models may seem to be the best option in a verification task, since they only rely on genuine samples for training, in reality they only

perform well as “cleaning” classifiers, removing outliers from an existing database or during a verification query if the subject provided a very unusual sample. The drawback is that the hyperplane describing the new feature space can be very descriptive of user behaviours but has no cost function that constrain it to be descriptive only to the genuine user, resulting in features common for a vast majority of the user population which would facilitate impostor attacks.
As an analogy, we could consider a one-class classifier for pictures of cats: it would most likely reject different species, but would probably misclassify other felines.

We also rejected expensive models, in terms of memory and execution time: very large models with a huge number of parameters that are computationally slow such as dynamic time warping (DTW) were excluded. The aim of the project is to provide fast authentication on mobile and wearable devices, which with these models could not be possible.
Considering all the exclusion criteria above, we ran a relatively small evaluation on classical methodologies. This because we used them just as a baseline for comparison with our proposed methodology. In the next section, we will describe in detail the two deep learning models constructed and evaluated for swipe authentication.

3.1.7 LSTM and GFNN architectures

3.1.7.1 Introduction

As opposed to the previous methodologies, and most deep learning models that directly perform a classification, returning a score ranging between 0 and 1 for binary classification (in case of sigmoid output) or a score distributions for the classes (in case of softmax), our aim was to train a model able to generate embeddings for an input swipe sequence in a Euclidean hyperplane. Classification would then be performed evaluating the Euclidean distance between a genuine subject embedding pattern from a previous enrolment and new embeddings from a query. Success or failure of authentication are determined by a threshold.

We developed two architectures: a light weight architecture focused to compute deep features starting from extracted features, the Global Feature Neural Network (GFNN), and a second architecture based on a Long Short Term Memory layer (LSTM) that would receive as an input a whole time-sequence and return the embeddings. Before describing the architectures of each model, we will provide basic knowledge on layers, cost functions, optimisers and back propagation, focusing on the specific elements used for our networks. These concepts are more extensively explained in [111, 112].

3.1.7.2 Neural networks

The simplest form of a neural network is comprised by an input layer, a hidden layer, and an output layer. Each layer is comprised of one or more nodes, or perceptrons, connected to the previous and following layer. A single perceptron, takes an input vector and applies to it a linear transformation with weights and bias.

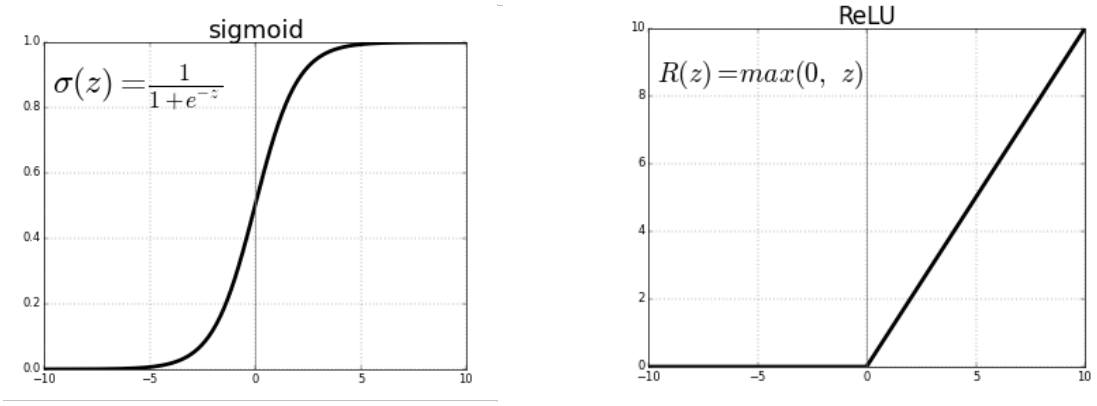
The results pass through an *activation function* which will return an output based on a threshold; the output can have upper and/or lower boundaries depending on the activation function 3.6.

$$\begin{aligned} u &= \sum_{i}^{N} w_i \times x_i + b_0 \\ y &= f(u) \end{aligned} \tag{3.6}$$

With f = activation function, w = trainable weights and x = input vector.
The activation function is also the source of non-linearity in the layer, with several being able to be implemented. We will describe just two that are relevant to our study: the sigmoid (equation 3.7) and the rectified linear unit (ReLU, equation 3.8). Figure 3.4 shows the responses of these functions.

$$\sigma(x) = \frac{1}{(1 + e^{-x})} \tag{3.7}$$

$$ReLU(x) = \max(0, x) \tag{3.8}$$



(A) Sigmoid (B) ReLU

FIGURE 3.4: Activation functions. Sigmoid (A) and ReLU (B)

The *weights* of the neural network are updated during the training phase through linear regression, layer by layer. The error is calculated starting from the output layer and computed using a *cost function*. Depending on the output of the last layer and the desired task, specific cost functions are implemented. For example, in case of a single sigmoid output on the last layer for a binary classification problem, binary cross-entropy loss (BCE) is used (in case of a multiclass output with a softmax, normal cross entropy is implemented) 3.9.

$$BCE(p) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \tag{3.9}$$

With N being the number of samples, y_i the ground truth for the *i-th* point and $p(y_i)$ the predicted score for the *i-th* point.

In our case, with the goal of training the network to cluster the data, we implemented another cost function, the *triplet margin loss*, described in detail in the next chapters.

The aim of training is to minimise the error, therefore the cost function, at the last layer of the architecture and update the weights and bias accordingly. To do so, there are various algorithms called optimisers. The most common is gradient descent, which simply computes the first-order derivatives of the cost function and

back propagates 3.10.

$$w_{it+1} = w_{it} - \alpha \frac{\delta E(x)}{\delta w_{it}}$$
$$b_{it+1} = b_{it} - \alpha \frac{\delta E(x)}{\delta b_{it}} \tag{3.10}$$

with it = training iteration. The disadvantages are the high computational cost and the risk to reach local minima or no convergence. There are many other options to solve this issue [113], but we will cover a specific one that we implemented in all our models: the Adaptive momentum (ADAM).

This optimiser adapts the learning rate over the training for each parameter, by storing an exponentially decaying average of past momentum and square momentum of the gradients. Considering m_t and v_t the first and second moment of the gradients at iteration it, are calculated as follows:

$$m_{it} = \beta_1 m_{it-1} + (1 - \beta_1) g_{it}$$
$$v_{it} = \beta_2 v_{it-1} + (1 - \beta_2) g_{it}^2 \tag{3.11}$$

The weights w are updated following this rule:

$$w_{it+1} = w_{it} - \frac{\eta}{\sqrt{v_{it}} + \epsilon} m_{it} \tag{3.12}$$

With β_1, β_2 and ϵ as hyperparameters of the optimiser conventionally initialised at 0.9 , 0.999 and 10^{-8} respectively.

3.1.7.3 Proposed architectures

The GFNN is a very simple feed forward model [114] with two fully connected layers, a dropout layer, a L2 normalisation layer and the triplet margin loss function.

The LSTM network is comprised of a LSTM layer followed by a fully connected, a dropout layer, a L2 normalisation and again the triplet margin loss function.
The LSTM [115][116] is a particular kind of recurrent cell that solves the issue of classic RNN networks: while both relies on hidden states from the previous iteration that contributes in the dependency of the time sequence, the LSTM

cell contains more gates that help manage the long-term dependencies, without overfitting on the single sequence through the forget gates (Figure 3.5).

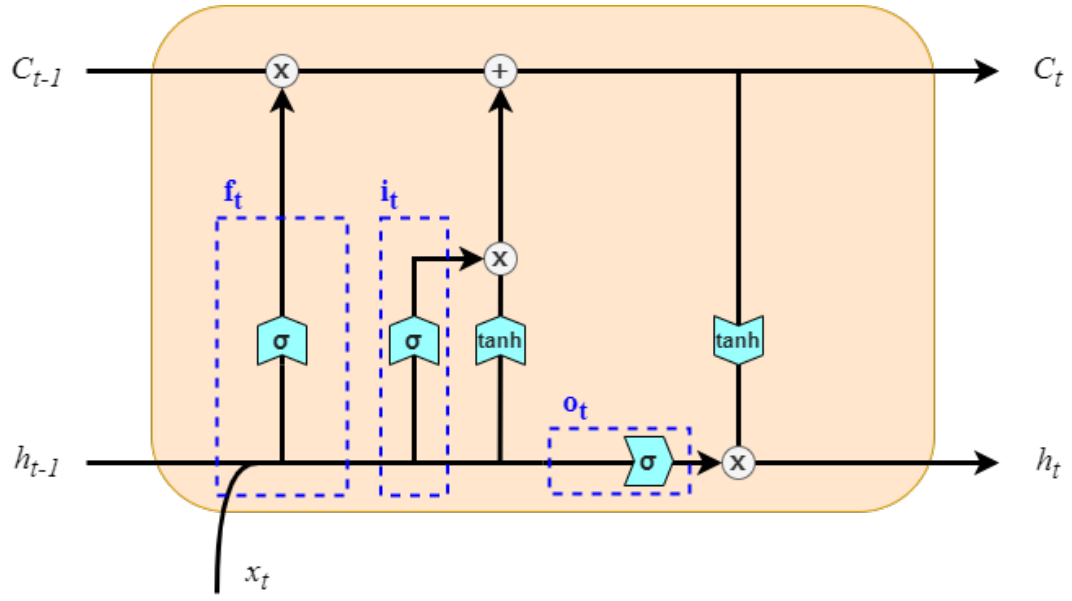


FIGURE 3.5: LSTM cell diagram, describing the various gates and input-output connections.

The LSTM cell consists of three gates (forget gate f_t, input gate i_t and output gate o_t), the cell state C_t and the cell output h_t, defined as follows:

$$\begin{aligned} f_t &= \sigma(W_f \times [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \times [h_{t-1}, x_t] + b_i) \\ \hat{C}_t &= \tanh(W_c \times [h_{t-1}, x_t] + b_c) \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \hat{C}_t \\ o_t &= \sigma(W_o \times [h_{t-1}, x_t] + b_o) \\ h_t &= o_t \cdot \tanh(C_t) \end{aligned} \tag{3.13}$$

Considering x_t the input sequence at timestep t.

Both models are very small because we wanted to achieve our goal with the least number of trainable parameters, making it possible to upload the model on any mobile device. In the case of the GFNN we relied mostly on the importance of the extracted features, therefore we didn't add many more layers which would be either redundant or generating overfitting. In the case of the LSTM, other studies

stack multiple recurrent layers to increase the complexity of the extracted features, but this has several flaws: LSTM layers are very slow to train and very memory expensive. The advantages of such layers decrease exponentially with the depth of the network, best performing in the proximity of the original sequence (in some cases, stacking more than three LSTMs worsen the performance) [117].

For both networks, we had a set of hyperparameters that we changed time by time until an optimal one was found:

- ***Model Depth***: The depth is defined as the number of multiple non-linear layers between input and output. Each layer can have multiple nodes (or filters). Pooling and dropout layers are usually associated with convolutional or fully connected layers as a single block.
- ***Number of filters***: The number of nodes or filters of the layers (not necessarily the same number for each layer). In the case of convolutional layers, kernel must also be defined by other hyperparameters, such as width, length, stride and padding.
- ***Learning rate***: this defines the rate of the weight update with respect to the gradient during back propagation. It is a decaying value to avoid large oscillations at the end of the training that would prevent smooth convergence.
- ***Batch size***: The number of samples per iteration fed to the network during training. When all the samples in the training set are evaluated, an Epoch is passed (Iterations per Epoch = Number of samples in training set / Batch size)
- ***Epochs***: Number of Epochs to run during the training.
- ***Normalisation***: Flag value for data normalisation.
- ***Output dimension***: the number of embeddings returned from the last layer.

To find optimal hyperparameters, it was used a trial-and-error methodology, varying one parameter at the time and exploring the effects on the model. The methodology can be compared to a manually executed grid search. The selection of the *Number of filters* was always connected to the *Output dimension*, being a pseudo-average between *Input* and *Output dimension*.

In terms of input/output, the GFNN took as input a tensor of shape GFNN_Input_shape = (*batchc size,Number of features*) and returned an embedding vector of shape GFNN_Output_shape = *(batch size, Number of embeddings*). The LSTM network took as input a tensor of shape LSTM_Input_shape = (*timesteps,batch size, Number of features*) and returned another embedding vector of shape LSTM_Output_shape = (*batch size, Number of embeddings*).

It's important to mention that the LSTM layer evaluates the whole input sequence regardless of padding, which propagates error during the training. For this reason, during the pre-processing we stored the original length of the swipe sequence and we implemented it as an extra parameter in the LSTM cell to pick the output from the real endpoint and pass it to the fully connected layer.

The two implementations utilised Python's TensorFlow library [118], but we used customised iterator and loss functions to perform the training due to the complexity of the task. The core of the model was in fact the clustering of the classes to generate the embeddings, performed by the triplet loss.

3.1.7.4 Triplet loss and embedding space

There are several ways to generate embeddings from data depending on the information that the new features must contain, and their purpose. In the case of clustering a virtually infinite number of classes, we need to project our data to a hyperspace that at the same time groups data of the same class and separates them from the others, maximising the distance. This is something not only inherent of the architecture itself, but mostly computed by the cost function.

One option could be a Siamese network [119], as it's generally defined as follows: the same basic architecture (tower) is duplicated. At the bottom, the two towers blend their outputs in a merging layer. The model receives pairs of data and the cost function is a contrastive loss (equation 3.14) that tries to minimise the distance between pairs from the same class and maximises it for data from different classes.

$$Contrastive\_Loss(f_1, f_2, m) = y \cdot d(f_1, f_2) + (1 - y) \cdot max(m - d(f_1, f_2), 0)^2$$
$$y = \begin{cases} 1, & \text{if } C_2 = C_1 \\ 0, & \text{if } C_1 \neq C_2 \end{cases} \tag{3.14}$$

With $m = margin$.

During the backpropagation, the weights of the towers are updated simultaneously. Once trained, the model can perform authentication directly by feeding enrolment data and query data as pair of inputs. The drawback of this methodology is that a Siamese network is very hard to train, generating pairs needs some extra preprocessing and it increases the size of the database (O^2) without adding any variance.

Random weights initialisation could also lead to early convergence to local minimum resulting in a random classifier. For those reasons we instead used a single tower network with triplet margin loss [120] defined as follows in equation 3.15:

$$Triplet\_Loss(f_a, f_p, f_n m) = max(\|f_a - f_p\|^2 - \|f_a - f_n\|^2 + m, 0) \tag{3.15}$$

with $m = margin$, $f = embedding\ features$.

The idea is again to maximise the distance between the same classes and different classes with respect to a margin, with the added value to check that the distance between anchor and positive should always be smaller than the distance between anchor and negative (see Figure 3.6).

This implementation also has the advantage that it only requires a one-dimensional set of labels and no pair creation, since distances and triplet mining will only be performed at the bottom of the model, once all the embeddings are created. It solves the problem of enlarging the input datasets, but the dimensionality of the output increases. The Tensorflow library provides an implementation of triplet loss called triplet_semihard_loss, which mines for semihard triplets in every batch to speed up the training phase.

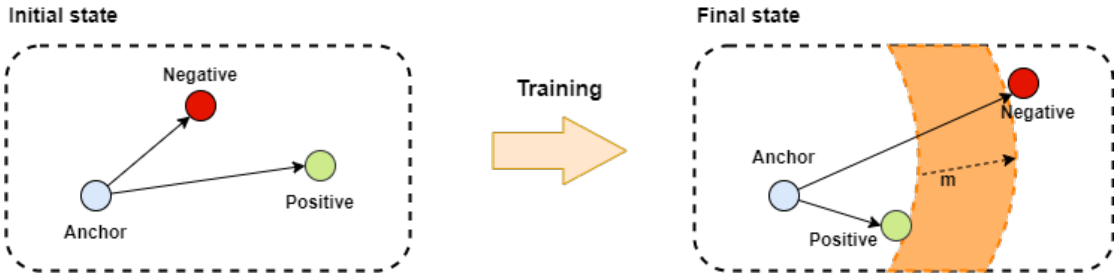


FIGURE 3.6: Visual representation of the idea behind the triplet loss for data clustering.

With the goal to improve the model at the cost of longer training time (and the risk of not converging), we decided to use a custom version of triplet loss with two small adjustments: we selected the hard triplets, evaluated on pairwise distances of the embeddings in the batch. Moreover, instead of considering just the anchor-positive and anchor-negative distances, we also introduced in the algorithm the positive-negative distances.

This last distance would have been set instead of the anchor-negative if smaller, avoiding loss of information given by the step function and better clustering of the classes

Other issues to be wary of when implementing triplet loss are the number of different classes in the database, the distribution of samples per class and therefore the batch size. With a large number of unique classes, it's not possible to use the built-in iterator of TensorFlow: in most of the cases there would not be enough sample per class and in the worst-case scenario a batch could have one sample per class, making it impossible to evaluate the loss and therefore resulting in a system crash, or an array of NaN values and exploding gradients.

The issue was solved by fixing the minimum number of samples per class on a single batch, meaning that not every class from the database appeared at each iteration (constrained by batch size). The class index was randomised at every epoch, and at each iteration we overlapped a percentage of the previous classes with the new classes to avoid overfitting on subclusters of classes. We also had to increase the number of epochs, since this iteration system not only slowed the training process, but also had some data loss caused by the randomisation process, and sometimes not all the available samples were loaded.
The training steps with the custom randomiser are summarised as follows:

- Step 1: a dictionary with indexes of the samples according to class ID is generated, alongside of a list of unique classes.
- Step 2: At the beginning of each epoch, the class list is shuffled randomly.
- Step 3: From the class list, $N\_classes = \frac{Batch\_size}{N\_samples\_per\_class}$ are selected. This will iterate at each loop until all the classes in the list have been selected. From one iteration to another, it will move by a fixed number (Stride)
- Step 3: random sample N_samples_per_class for each class in N_classes and append to the batch.
- Step 4: train the model on the current batch.
- Step 5: update the weights with gradient clipping to avoid large values.
- Step 6: repeat from Step 2 for the selected number of epochs.

Once the training is complete, the testing is performed on unseen subjects from the same database through a verification task. For each subject, a fixed number N of swipes are used for enrolment and fed to the trained model to generate the user template.

Considering X_i^s the i-th enrolment swipe of a subject s and M our embedding model, the template T^s for the subject is calculated as it follows:

$$T^s = \frac{1}{N} \sum_{i=1}^{N} M(X_i^s) \tag{3.16}$$

Following this process, random samples from the same user class and different classes in the testing set are presented to the model and the Euclidean distances between the verification embeddings and the user templates are calculated. If the distance is lower than a set threshold, the verification response is positive, otherwise it is negative. The threshold is calculated through evaluating false match (positive) rate (FPR) and false non-match (negative) rate (FNR) and selecting the threshold that would give the least combined error (equal error rate) or, from another point of view, the threshold that would maximise the area under the curve (AUC) in the related receiver operator curve (ROC).

The dissimilarity score is calculated through a Euclidean distance for two main reasons: the loss function of the model also uses the Euclidean distance as a distance metric when evaluating the triplets. Utilising another distance would give biased results that could not relate to the training process. Moreover, considering also the last normalising layer, the Euclidean distance can return values constrained between 0 and 2, making it easier to understand the results in the embedding space (or in a reduced space for visual representation).

The embedding space S^c is the multidimension hypersphere of extracted features from the deep learning model. It's a subspace of $\mathbb{R}^{od}$ with $od = output\_dimension$ of codimension $c = od - 1$ and is defined as follows:

$$S^c = \{x \in \mathbb{R}^{od} : \|x\| = 1\} \tag{3.17}$$

This reduction is due to the L2 normalisation layer, and implies (as can be deducted from the equation) that the maximum possible absolute value of a projected point is 1, while in the od-dimensional Euclidean space that value would be $\sqrt{od}$ for the corresponding vertexes. This means that the reduction to the sphere

does not just imply a loss of a dimension itself but also a different perspective when evaluating optimal centres for clusters. Using the entire space instead of the reduced space would increase the maximum distance between clusters, allowing different classes to spread wider and ensuring a slower decay of the authentication threshold with the increase of unique classes.

The reason for choosing an apparently worse feature space is model-related: constraining the output values improves the gradient calculations and backpropagation, avoiding large numbers or an imbalance between features. During the calculation of the distance and the score, we did not observe large differences between values, which would be good with a small dataset but would affect the distribution on a larger dataset, resulting in outliers and producing oscillating thresholds.

To observe the spread of the data in the clusters and the positions of the centroids, batches of data have been projected on a reduced space for visual representation. For this purpose, we used *Principal Component Analysis* (PCA) to reduce the dimensionality of the embedding vector and obtain the first three principal components according to the variance of the subset of data. The PCA algorithm calculates the principal components applying a linear transformation to the input data (in our case, the embeddings) considering the Eigenvalues of the covariance matrix of the data and rank ordering them from largest to smallest.

For a n-dimensional dataset with m elements (rows), the covariance matrix will be a $n \times n$ symmetric matrix, with the diagonal equal to variances for each dimension. The concept of PCA is to transform a dataset so that the new features will maximise the variance between data and minimise the correlation between dimensions. In this way the new components will be ideally independent between each others (in reality, the dependency will be minimal). It's important to point out that this is a statistical method and is heavily affected by the size of the dataset and by data normalisation.

In our case, we selected the first three principal components of batches of embeddings (the PCA was trained on a training batch and applied both to the last batch of the training set and to the sample embed-dings of the test set) to be projected in

a three-dimensional space. Different classes have been assigned different colours. In case of the test set, the template of the subject has been represented as a small diamond with black corners to highlight it in the cluster cloud. Ideally it should appear near the centroid of the cluster.
In the following section, results are presented alongside figures of projected embeddings in the reduced features space.

3.1.8 Results and comparisons

Several experiments were conducted to train the two deep learning models, varying parameters such as margin, number of training epochs, train and testing set percentages, hidden and output dimensions, and normalisation. We could not apply a grid search due to the large amount of memory usage by the Tensorflow library, hence we run each experiment with different parametrisation by itself (more than once for repeatability). Both models were very slow to train (especially the LSTM network due to the recurrent layers), and to reach the flattening of loss, approximately 900 epochs were required.

In each case of the LSTM network, the model was trained and tested removing a subset of raw features from the training set, to explore if any of them were more impactful with respect to the others. We noticed a consistent change in accuracy when excluding the major axis/finger area features; this was expected, since it's the only feature directly connected to a physical trait of the subject and by itself it's more unique and less changing over time.

The testing was conducted on a percentage of the whole dataset (the test set) with unseen subjects. For each unique subject, 10 consecutive swipes were used to generate the user template. Then, 300 verification attempts were evaluated between remaining samples from the genuine user and random samples from the remaining users in the testing set considered impostor attempts. For genuine verification, we used only samples collected after the enrolment samples, to avoid feeding data from the past to the model. As described above, the Euclidean

distance was calculated between the new embeddings of the query and the user template. We calculated the optimal threshold and the related equal error rate for all the scores from the verification attempts.

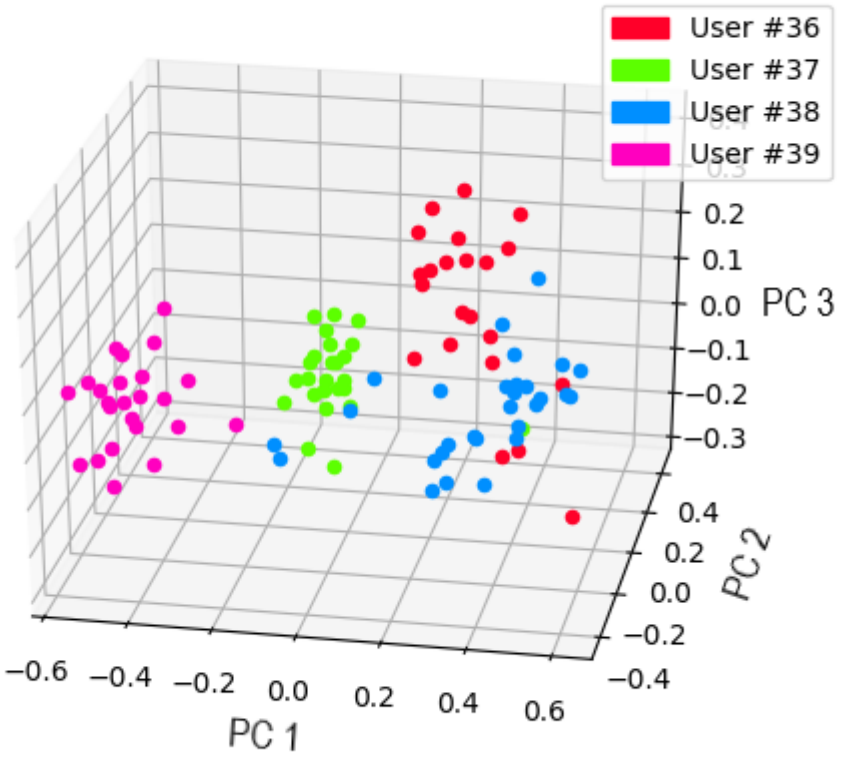


FIGURE 3.7: Projection of the three principal components of the swipes' embeddings for four different subjects. Points of same colors are samples belonging to the same subject.

The initial tests were run on a very small subset of the Callsign database (36 subject for training of the network and 4 unseen subjects for testing), to debug the code and to analyse the performance of the triplet loss on an easy clustering task. Whilst the results were not relevant, this process provided a better visualisation of how the clusters were displayed (see Figure 3.7). The final tests with optimised hyperparameters were run on the entire CallSign dataset and on the public dataset, splitting train and test set as 90% and 10% respectively to the total number of subjects. The hyperparameters are listed for Callsign dataset (Table 3.4) and Serwadda Dataset 3.3 for both architectures. In the case of Serwadda dataset, we

used a reduced number of subjects for training and testing, due to memory issues. In Table 3.5 are shown the results for both datasets in terms of equal error rate.

TABLE 3.3: Models hyperparameters for Serwadda dataset.

	LSTM	**GFNN**
Input dimension	5	21
Hidden dimension	64	64
Output dimension	128	128
Batch size	128	128
Epochs	100	100
# Subject in training set	90	90
# Unseen users in test set	10	10
Normalisation	Z-score	Z-score
Margin	1.0	1.0

TABLE 3.4: Models hyperparameters for Callsign dataset.

	LSTM	**GFNN**
Input dimension	5	21
Hidden dimension	64	64
Output dimension	128	128
Batch size	256	256
Epochs	900	1000
# Subject in training set	148	148
# Unseen users in test set	14	14
Normalisation	Z-score	Z-score
Margin	1.0	1.0

TABLE 3.5: Equal error rate for LSTM and GFNN models evaluated on Serwadda and Callsign dataset

	EER [%]	
	LSTM	**GFNN**
Serwadda Dataset	25.6	22.4
Callsign Dataset	12.9	15.0

In Figure 3.8, it can be seen the clustering for the final batch in the training set, after applying PCA for dimensionality reduction.

The classification error for both models is relatively high, but it's important to consider the context: this evaluation considers a single swipe authentication, not a continuous authentication or an average over several samples. It is reasonable to expect an overlapping in the behaviours between subjects or inter sample inconsistency over time. It's also important to note that these results are provided

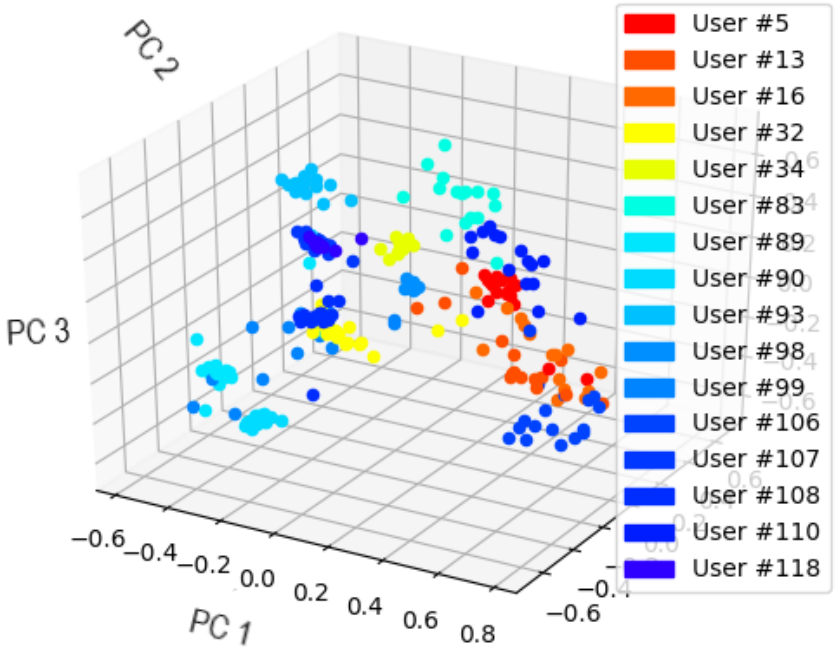


FIGURE 3.8: 3-D projections of the first three principal components of the swipes embeddings in the last batch of training.

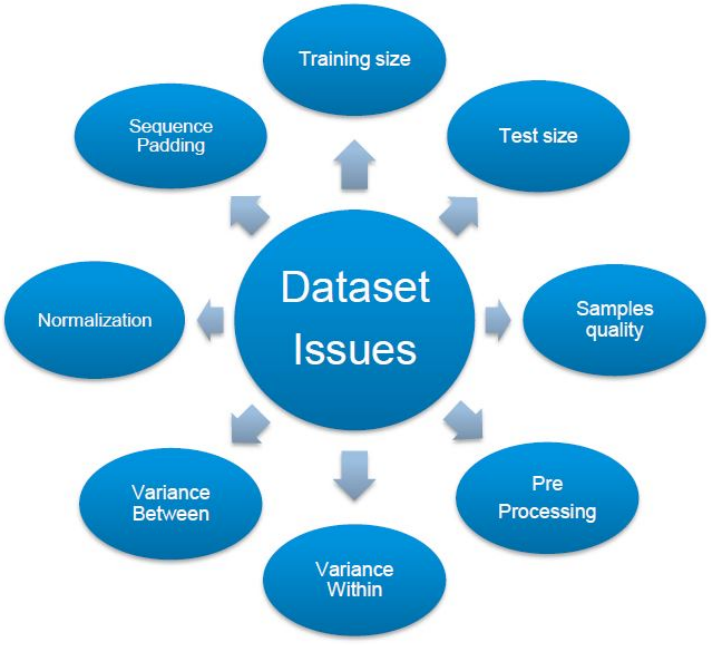


FIGURE 3.9: Common dataset issues

from a very light model using a single modality. Having a multimodal system or other features could improve the results, but this will be covered in Chapter 4. Nevertheless, a 13% error on verification for a single swipe model makes it competitive with the state of the art; results are better compared to studies relying on single swipe authentication [40], in addition the proposed model doesn't need retraining on new subjects. The fact that it performed better than GFNN means that the LSTM layer was able to extract more relevant features compared to the pre-processed features for the feed forward network.

Regarding the huge difference in performance between the production database from Callsign and the publicly available dataset collected by *Serwadda et al.*, the most likely explanation is that the models work better on a semi-constrained task. Subjects from Callsign were inherently more consistent with themselves and provided better samples, not only in terms of behavioural data but also in terms of swipes quality. Pre-processing procedures and normalisation were consistently applied across both datasets and a search for best hyperparameters has been conducted separately for each.

Figure 3.9 highlights the common dataset related issues that can affect model performance. Most of these issues have been addressed in this chapter, except for the quality which will be the focus of section 3.2 .

3.2 Swipe quality

3.2.1 Introduction to Biometric quality

Quality is a term with different meanings depending on the context. It is not easy to define; in a very shallow way, quality is a metric that defines the inherently *goodness* of an entity. To better define the metric, it's necessary to understand what makes something good and from what perspective. "Goodness" itself is very subjective.

From this initial statement, quality would be a metric impossible to define or generalise, since we would need to know the ground truth of goodness to establish parameters and metrics, but to do so we need first to assign those labels based on observer's decisions. It's a vicious circle. The way to solve this issue is to adopt a definition of quality that only describes the correlation between the metric itself and the effect, without the in-between process.

In this study we will refer to the NIST definition of biometric quality [121], in which quality is defined as:

'A sample should be of good quality if it is suitable for automated matching. [...] A quality measure could be tuned to predict the performance of one matcher or the more difficult case of one that generalizes to other matchers or classes of matchers.'

By this definition, the quality of a sample should give information on the predictive performance of one or more models, despite their specific implementation; a good sample will be suited for the specific classification task and will not result in an outlier on a decision scale. For those reasons, quality estimation is a key study in biometrics, allowing optimisation and improvement of existing authentication systems by giving a prediction on the model performance based on the goodness of the sample or the user.

The previous definition covers *what* makes samples of good quality, but leaves open to interpretations the *why*. For the same reason, a general rule or metric system or range for the quality measure is not specified. For a small number of specific modalities there exists a standard protocol with quality ranges, based on specific features.
One example is face image quality: the ground truth is still defined by the observer on a subjective decision. The observer defined a range of global quality features that can be extracted from every image (blurriness, distortion, resolution, entropy, etc.) from which a quality score is generated. The testing steps evaluate the correlation between the quality score and the ground truth, and furthermore the quality with the model(s) performance.

Those criteria cannot be applied for behavioural biometrics, since there is no standardisation nor regulation to estimate the quality of behavioural data such as a keystroke or swipe sequence.
The following section will describe related studies and how their approach can be relevant for quality on behavioural biometrics.

3.2.2 Related works

Over the past years, biometric quality has been a topic of interest for many research groups and has seen its definition changed multiple times. Most studies were focused on image quality for fingerprint, iris and face recognition; quality was assessed in terms of extractability of features or suitability of the sample or even as an estimation of degrading factors known to affect the classification.

In 2014 *Bharadwaj et al.* [122] reviewed the methodologies for quality assessment and explored factors that could affect quality for different modalities. Regarding fingerprint quality assessment, in 2005 NIST released the *"Fingerprint Image Quality (NFIQ) Compliance Test"* [123]. More recently, in 2016, *Yao et al.* published a review of quality assessments for fingerprints [124].

Regarding the face, different studies assessed quality for face images considering different kinds of approaches and issues. *Corsetti et al.* [125] investigated how accessibility influences the quality of the captured image and therefore the authentication process, revealing how users with accessibility issues struggle in providing good samples compared to control population.

Chen et al. [126] proposed a flexible ranking method to evaluate the quality of face images depending on the dataset and the authentication system in use, allowing to select the best performing images during the authentication process (when more than one is provided, for example in a video recording).

Hernandez-Ortega et al. [127] developed a quality assessment approach for face recognition based on deep learning (FaceQnet) for face recognition purposes; the model would assess the suitability based on the image itself, without prior extraction of image features. The training process of the model followed the same framework previously described and even in that case they used ground truth for quality assessment provided by ICAO compliance level. Very recently, NIST has released the "Face Image Quality Assessment" on the Ongoing Face Recognition Vendor Test (FRVT) [128].

Despite the common interest to develop a solid and consistent quality score for each modality, few studies have been conducted on behavioural biometrics, except for signature recognition. A big issue is the absence of ground truth when it comes to behavioural biometrics (such as swipe or keystroke dynamics). Manual labelling is also not possible in these cases due to lack of understanding, unlike in the case of pictures where visual samples are provided.

A number of studies have explored the impact of quality in signatures trying to find metrics or predictors to estimate sample quality and correlate to the classification performance.

Müller et al. [129] described the a priori and a posteriori approaches to evaluate quality on handwritten signatures, identifying specific quality features descriptive of signature stability; *Galbally et al.* [130] applied the Sigma Longnormal model as

a quality estimator for handwritten signature, while *Sae Bae et al.* [131] proposed a quality metric for online signatures that measures the separation between intra-user and inter-users distributions.

Considering the existing issues with quality for behavioural data, the approach of Sae Bae et al. [131] seemed the best to fit the case of swipe biometrics for both sample and user quality.

3.2.3 User quality and Sample quality

Before describing the methodology used to estimate the quality for swipe data, a differentiation between sample and user quality must be done. With respect to a single sample, which could be of bad quality for a multitude of reasons (environmental or otherwise), the user provides a more systematic behaviour when it comes to quality and therefore to verification outcome; moreover, the evaluation of user quality is never estimated on just one pulled request to the system and even the *a posteriori* methodologies consider the percentages of false match and non-match for a single user, which implies a list of queries and comparison with the rest of the population.

In particular, the first classes of the biometric zoo [132] describe the ability of certain subjects to easily impersonate others or to be often unable to authenticate themselves regardless of the biometric authentication system used. At this stage, where the classification is executed on behalf of a system performance, quality features and metrics are not defined yet, but everything points towards two aspects: the variance within a subject, and the variance between subjects and the rest of the population.

A *"goat"*, hereby defined as a subject unlikely to pass a verification test, hence having high false non-match rate, will probably be very inconsistent with themselves; this high variance will reflect both with the enrolment and the template matching on query.

Similarly for a *"lamb"*, a subject easily to be impersonated and that increases the false match rates of the system, the major cause of errors is due to the resemblance of the user with others in the population; the samples provided and their features won't be discriminative enough to increase the uniqueness of the subject, which can be seen as a very low variance between subject and population.

For sample quality, the same assumptions can be applied, but from a slightly different perspective. We are not considering anymore a constant good or bad behaviour from a subject when recording data, but the single capture itself; to be more precise, a *"high quality user"* may happen to occasionally provide a very bad sample and a *"low quality user"*, as opposite, could provide sometimes a very good sample.

The reasons for this could be an accidental capture, hardware issues or delayed/prolonged action. In this context, the variability between the single swipe and the average swipes from the observed population can still be comparable with the previous example of variance between subjects; but in the case of sample quality, the variance within a single person sample is more like a measure of stability of the swipe gesture.

For this reason, it is necessary to extract a few features that define the swipe as an operation and not as a discriminant for an individual behaviours. They need to be descriptive of local changes and address software or hardware issues while still being screen independent, device invariant and position independent.

The following section will describe the features and metrics used to estimate both sample and subject quality for swipe biometrics, along with the experimental protocol for the various datasets used in this study.

3.2.4 Quality metrics and experimental protocol

3.2.4.1 User template quality

User quality has been estimated considering the enrolment template and the related extracted features. Being a value descriptive of the subject, it also depends on the authentication model used and therefore by the extracted features used by the model for enrolment and verification.

In the case of the LSTM model, the embedding has been used to calculate the score. For the GFNN and classical machine learning models, the input feature vector was used. Both feature vectors (input vector for GFNN and embedding from LSTM) were discussed in sections 3.1.5 and 3.1.7.4

Considering F the number of features (or embedding dimension) extracted from the swipes in the considered dataset, N_i the number of enrolment samples for the i-th subject in the database, and S the total number of samples in the database, the quality score for subject i was calculated with the following algorithm based on a study by Sae-Bal et al.[131]:

$$Q_i = \frac{1}{F} \sum_{f=1}^{F} \frac{\|\mu_{l,f} - \mu_{g,f}\|}{\sqrt{\frac{\sigma_{l,f}^2 + \sigma_{g,f}^2}{2}}} \tag{3.18}$$

With $\mu_{l,f}, \mu_{g,f}, \sigma_{l,f}, \sigma_{g,f}$ being the mean and standard deviation of feature f for the enrolment set (local) and the whole dataset (global). We will refer again as local and global in swipe quality for the single sample and the whole population estimator respectively.

If the data were previously z-score normalised, the quality score would be entirely determined by the subject's enrolment samples since the global mean for each feature would be equal to 0 and the global standard deviation equal to 1. For each subjects in the considered dataset, a single quality score is evaluated and the corresponding EER is obtained with the selected classifier for biometric authentication.

3.2.4.2 Sample quality

Compared to user quality and referring to the previous assumptions regarding the quality feature of a swipe sample, further pre-processing was required before calculating the quality score. In Table 3.6 the extracted features have been listed, six derivatives and two physiological:

TABLE 3.6: Extracted features for sample quality.

Identifier	Feature name	Description
1	P2P X Velocity	Instant velocity on x-axis.
2	P2P Y Velocity	Instant velocity on y-axis.
3	P2P Swipe Velocity	Instant velocity on distance travelled.
4	P2P X Acceleration	Instant acceleration on x-axis.
5	P2P Y Acceleration	Instant acceleration on y-axis.
6	P2P Swipe Acceleration	Instant acceleration on distance travelled.
7	Pressure	Finger pressure value during the swipe gesture.
8	Finger Area	Area (or major axis) described by the finger.

The first and second derivative features describe the trend along the swipe without screen size dependency (on a certain extent); such features are similar to the derivative features considered for biometric authentication (see Table 3.1), with the difference that it's been considered the instant derivative for sample stability. Ideally, a good sample would mantain stability with no major changes as it was previously mentioned.

Pressure and finger area are also useful to detect accidental touch over the swipe or uncommon interactions that would affect the recording. In addition, from a user quality point of view, these two physiological features are good candidates as they are expected not to considerably vary in case of a consistent user (high quality).

Local mean and variance are computed for each feature across the sample recording, while global mean (eq.3.19) and variance (eq. 3.20) are computed over the means in the dataset as follows:

$$\mu_{g,f} = \frac{1}{S}\sum_{i=1}^{S}\mu_{l,f}^{i} \tag{3.19}$$

$$\sigma_{g,f} = \sqrt{\frac{\sum_{i=1}^{S}(\mu_{l,f}^{i} - \mu_{g,f})^2}{S-1}} \tag{3.20}$$

The quality score is assigned at a given sample following eq. 3.18 taking into account these new global mean and global variance features.

It's important to note that the quality score used in this study has no upper bound and, as previously stated, there is no ground truth for quality. Therefore, to define quality ranges and a suitable threshold, two methodologies are proposed: K-means and quantile normalisation.

One approach is to use the K-means algorithm [133] with K as the number of quality ranges set to 3 (*low, medium and high*) to cluster the data in an unsupervised mode. The algorithm considers only the score distributions for samples and users without further information provided. Once the centroids and the thresholds have been obtained, the mean and the variance of the EER (or similarity score for sample quality) are computed for every range and subsequently compared. This approach works well when considering a large number of points, however when data points are sparse, the estimation of the cluster means was mostly random and not reliable.

The second proposed solution was a max score normalisation with fixed thresholds at 0.33 and 0.66 and outliers removal. A standard scaler was not a good option as it would result in negative values without solving the issue of the missing constraints, and a normal min-max or max scaler would be biased by the presence of outliers.

With the proposed approach, data were normalised based not on the maximum value of the quality scores in the dataset, but on the 95^{th} quantile (considering 5% of the samples as outliers). This methodology works well with small datasets containing outliers, but the downside is that it also imposes an equal width for all the quality ranges (which might not necessarily be the real case).

This method is still a valid option when the K-means algorithm cannot be applied and consistently highlights the correlation between quality and classifier performance.

3.2.4.3 Protocols

For the Callsign dataset, the GFNN and LSTM models were used to provide authentication performance records, while for the public datasets a state of the art model was selected (the fusion model comprised of a SVM and a GMM proposed by Fierrez et al. [47]). Protocols for the various datasets are similar, the main difference is that for the Callsign dataset, as the data provided were task related and not split in multiple supervised sessions, we only produced a single output of data for sample and user quality. For the public datasets, we explored the correlation within and between sessions and considered the directions of the swipes separately.

For *user template quality* we first estimated the quality scores on the extracted features used by the models; for the Callsign dataset we used the features for the GFNN as described in Table 3.1 whilst for the public datasets we calculated the quality score of each subject based on the extracted features used with the State of the Art model previously described. Such features are distributed in two vectors of 28 and 5 dimensions each, according to [47]. The first 28 extracted features are representative of the state-of-the-art swipe biometric recognition:

- *mean, standard deviation, first quartile, second quartile and third quartile* of velocity, acceleration, pressure and finger area.
- x and y coordinates of extreme points in the stroke.
- Distance between start and end of the stroke.
- Stroke duration.
- Distance traveled.

The last 5 features were proposed for on-line signature verification. These features are selected from a larger set of 100 features as explained in [47]:

- θ (finger down to finger up)
- σ_{a_x} (std of the acceleration in x)
- $(x_{max} - x_{min})/x_{maxrange}$
- $(\overline{x} - x_{min})/\overline{x}$
- $(y_{max} - y_{min})/y_{maxrange}$

From the second set, the last four features are considered for vertical strokes. In the case of horizontal strokes, x and y are alternated in the formulas.

A quality score is calculated from the extracted features using equation 3.18, but instead of considering a small subset of impostors, we calculated the global mean μ_g and standard deviation σ_g from all users and samples in the database with respect to each task and only on the first session, if there was more than one. μ_l and σ_l are calculated for each feature over the enrollment samples of the *i-th* subject.

The enrollment samples of each user are also included in the computation of the global mean and variance, to avoid them being automatically considered outliers from the population and, as a consequence, having a higher quality score than expected.

In the case of the public datasets, for each subject a single quality score is evaluated for each direction (meaning that the same user could provide higher quality data for specific tasks) and the corresponding EER is obtained with the classifier for both inter and intra sessions.
The quality score evaluation was performed for each subject during the training phase of the classifier, using only the enrollment samples. Quantile normalisation or K-means were used depending on the number of testing subjects.

For the public datasets, after calculating the quality score for each subject the evaluation was repeated three more times, varying the number of enrollment samples for each user (5 , 15 and 20 samples from genuine, and equal numbers from imposter samples) and comparing the mean EERs for the different quality groups for each enrollment size.

Regarding *Sample quality*, the scores are calculated for all the samples in the testing set using the extracted features from Table 3.6 and according to equations 3.19 and 3.20.

Instead of the EER, for each sample used during the testing phase a similarity score is stored and only genuine samples are considered. The reason is that the same genuine sample could be an impostor if compared to the profile of another subject during the evaluation, but it would still maintain the same quality score. Thus, in this case it is better to just consider the correlation between quality and similarity score for genuine samples.

The K-means algorithm is then applied to find quality thresholds on genuine samples considering the corresponding similarity score. Mean and variance of the similarity scores for the samples in the three quality ranges are calculated, expecting a correlation between quality and classifier performance (similarity score should be higher for high quality samples).

All the experiments on the Callsign dataset were run on Python. For the public datasets, MatLab r2018 was used for model compatibility.
This decision did not impact the outcome of the research and did not provide a bias for the results either; instead, it explains the cosmetic differences in visual representations of the results.

3.2.5 Experimental results and conclusions

Experimental results are provided for public and Callsign datasets. The differences in design for the figures are due to different environments used to run the codes.

For *Sample Quality*, Table 3.7, Table 3.8, and Table 3.9 show the *mean* and *standard deviation* (in brackets) of the similarity score for genuine testing samples, considering quality ranges, stroke directions, sessions and datasets. Higher scores represent a better classifier performance. An example is shown in Fig. 3.10.

Intra-session classification performs better in every circumstance, due to the increased consistency of the subjects. Overall, we can see an increase of the similarity score for higher quality samples, with some exceptions (especially in the Frank database) caused in all probability by a smaller number of samples or increased inconsistency towards certain stroke directions.

For *User Quality*, Tables 3.10, 3.11, 3.12, 3.13 and 3.14 show EER values (*mean* and *standard deviation*) for the three public datasets, considering directions, sessions and number of enrollment samples. Examples of quality clustering using quantile normalisation and corresponding distribution of EER per range are shown in Figure 3.11a and Figure 3.11b .

The Tables highlight not only the decreasing EER over the quality ranges, but also how the different number of training samples affect the performance for different quality subjects. In general, high quality users are quite consistent with their own samples and increasing or decreasing the number of enrollment samples does not impact on the classifier performance.

In contrast, for low quality users, increasing the number of training samples helps to provide correct classification during testing, as shown in Figure 3.12.

Finally, Table 3.15 shows results for sample quality and user quality on the Callsign dataset in terms of EER per quality ranges. The same criteria has been used to calculate the quality ranges. In Figures 3.13, 3.14, 3.15, 3.16, 3.17, 3.18, 3.19, 3.20 and 3.21 are shown the similarity score distribution for genuine and impostors for all the quality ranges when evaluating sample and user quality. It's interesting to notice how for sample quality the scores are skewed on the left, which results in closer thresholds for low and medium quality samples.
Observing the results, it can be seen that the estimation of quality is impactful

and consistent over different datasets and models. Predicting the user quality behaviour or rejecting low quality sample can improve the performance.

The bottleneck of this study resides in the models themselves: even with effective quality estimation, if the biometric authentication models perform badly by themselves the improvement will be low to none.

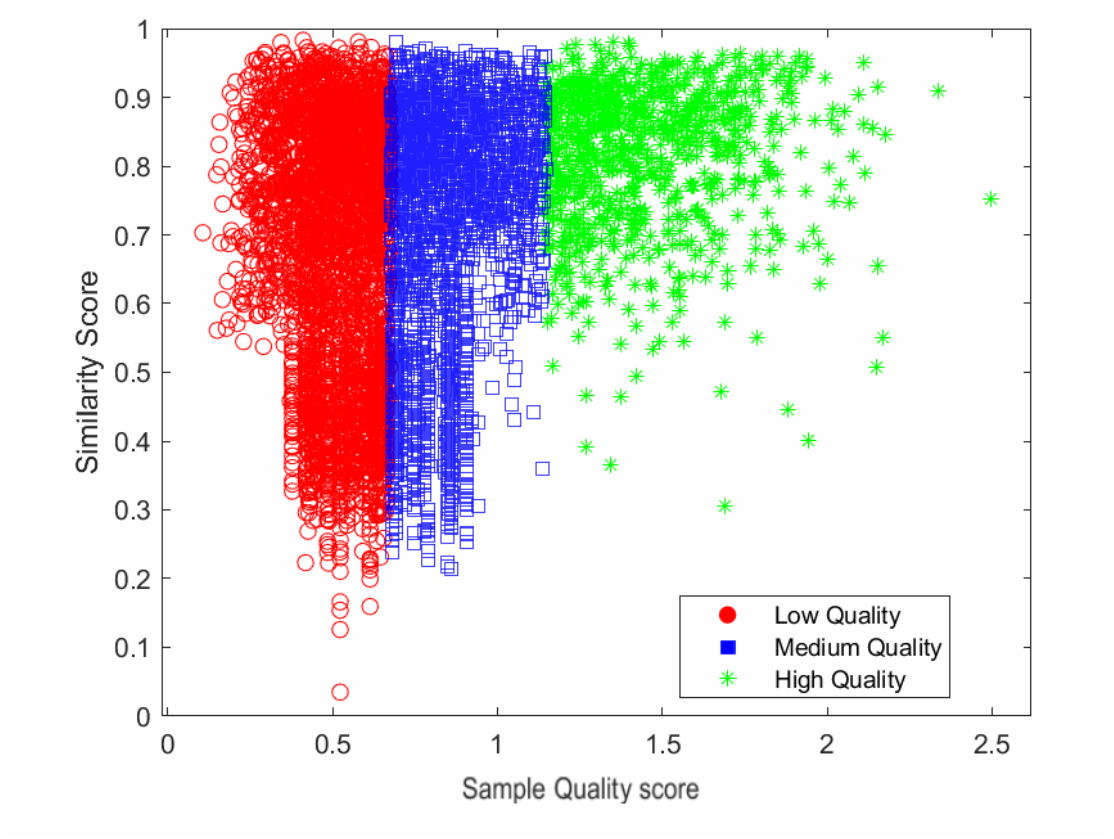
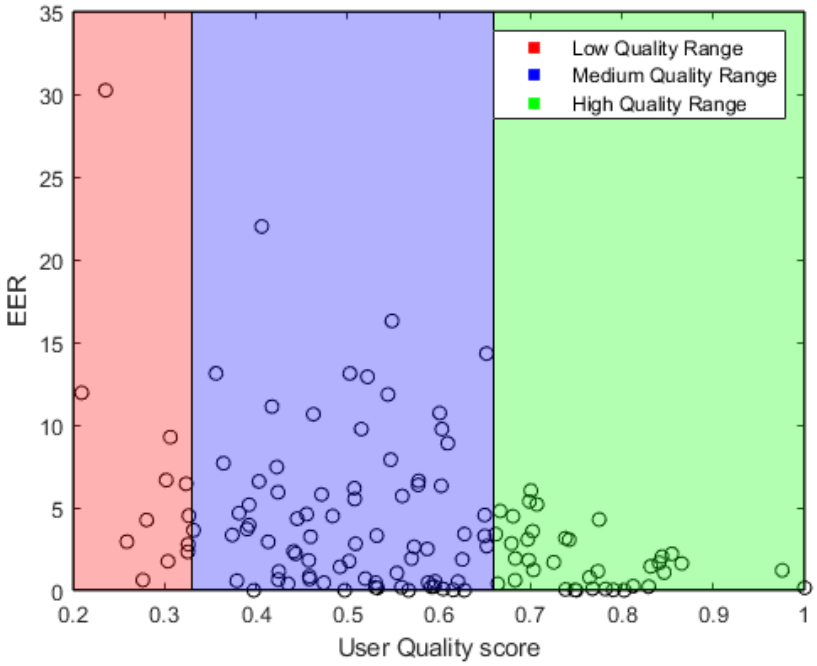


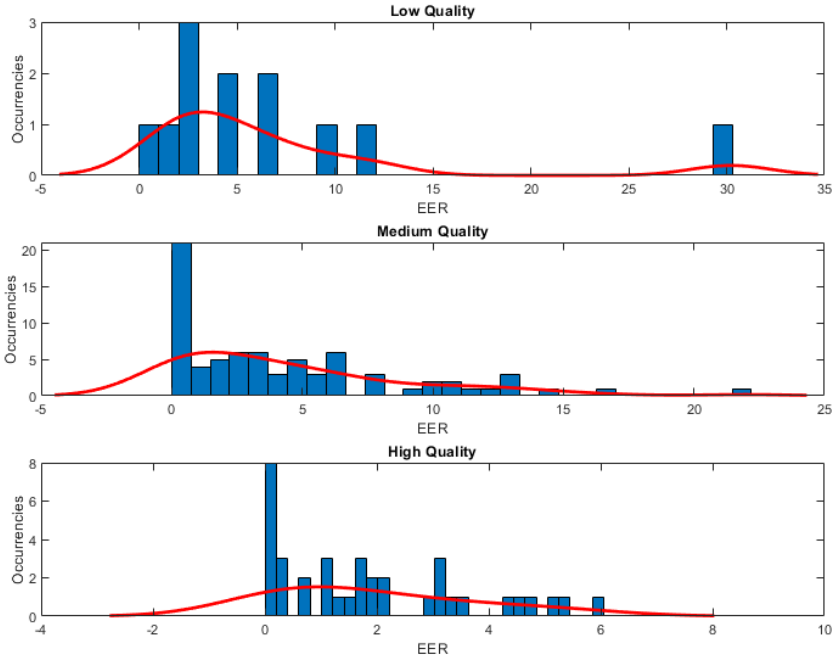
FIGURE 3.10: Sample quality score vs Similarity score, Serwadda database, intra session, right swipe. K-means algorithm has been used to cluster the three ranges.

TABLE 3.7: Results for Sample Quality Analysis on Serwadda dataset. Mean Similarity score (standard deviation in brackets) for genuine samples in different quality ranges, evaluated for each direction on both intra and inter sessions. values are left blank when missing.

Serwadda Database		**Quality Range**		
		Low	**Medium**	**High**
INTRA SESSION	**Down**	0.73 (0.12)	0.74 (0.13)	**0.78 (0.14)**
	Up	0.71(0.14)	0.75 (0.13)	**0.79 (0.13)**
	Left	0.63 (0.23)	0.78 (0.10)	**0.78 (0.15)**
	Right	0.67 (0.19)	0.70 (0.18)	**0.82 (0.1)**
INTER SESSIONS	**Down**	0.63 (0.14)	0.65 (0.14)	**0.67 (0.15)**
	Up	0.61(0.14)	0.63 (0.14)	**0.66 (0.15)**
	Left	0.64 (0.15)	0.67 (0.13)	**0.69 (0.15)**
	Right	0.60 (0.14)	0.63 (0.15)	**0.68 (0.15)**



(A) EER vs Quality.



(B) EER distributions over quality ranges.

FIGURE 3.11: User quality in Serwadda database (Direction: down, Intra session). Here we used quantile normalisation to separate quality groups. In figure (A) it's shown EER vs quality score, in figure (B) the histograms of EER distributions over quality ranges (note: x-axis' scales differ for the three histograms).

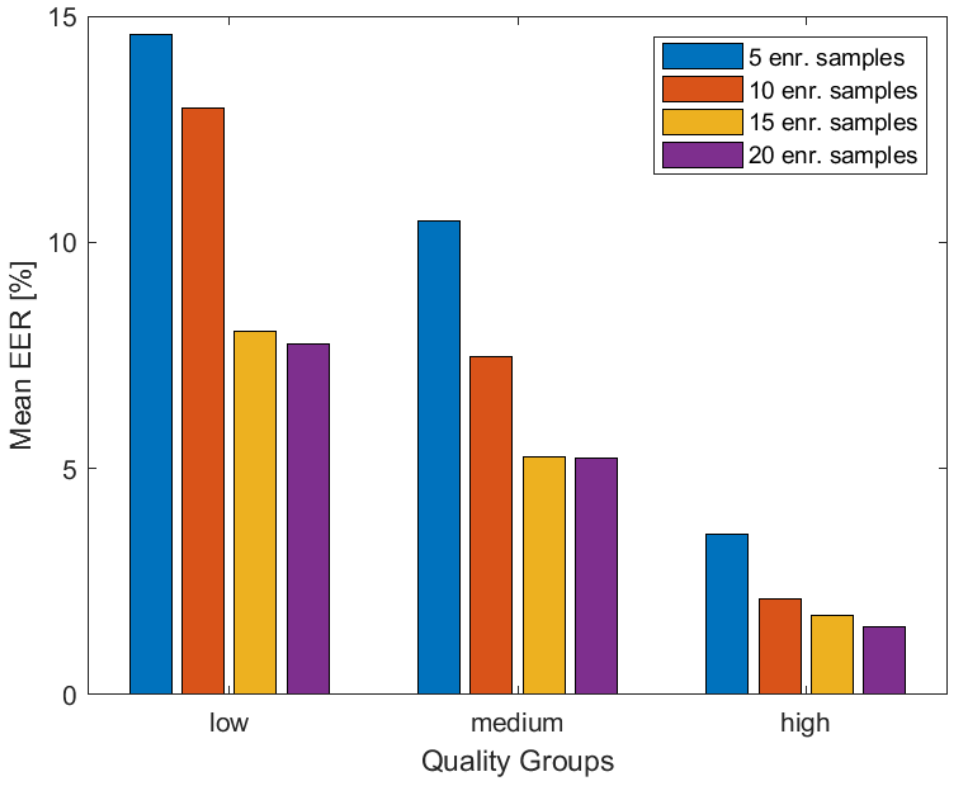


FIGURE 3.12: Mean subjects' EERs with varying enrollment sizes for each quality group. The quality score has been evaluated considering 10 enrollment samples.

TABLE 3.8: Results for Sample Quality Analysis on Frank. Mean Similarity score (standard deviation in brackets) for genuine samples in different quality ranges, evaluated for each direction on both intra and inter sessions. values are left blank when missing.

Frank Database		**Quality Range**		
		Low	**Medium**	**High**
INTRA SESSION	**Down**	0.69 (0.12)	0.60 (0.19)	**0.68 (0.2)**
	Up	-	-	-
	Left	0.73 (0.14)	**0.77 (0.12)**	0.74 (0.15)
	Right	0.71 (0.12)	0.76 (0.12)	**0.77 (0.15)**
INTER SESSIONS	**Down**	**0.62 (0.15)**	0.49 (0.16)	0.52 (0.18)
	Up	-	-	-
	Left	0.63 (0.15)	**0.69 (0.14)**	0.65 (0.14)
	Right	0.66 (0.14)	**0.70 (0.14)**	0.65 (0.17)

TABLE 3.9: Results for Sample Quality Analysis on Antal dataset. Mean Similarity score (standard deviation in brackets) for genuine samples in different quality ranges, evaluated for each direction. Values are left blank when missing.

Antal Database		**Quality Range**		
		Low	**Medium**	**High**
INTRA SESSION	**Down**	0.62 (0.12)	0.74 (0.10)	**0.83 (0.01)**
	Up	0.70 (0.15)	0.68 (0.13)	**0.74 (0.14)**
	Left	0.65 (0.18)	0.71 (0.17)	**0.74 (0.17)**
	Right	**0.74 (0.12)**	0.71 (0.16)	0.52 (0.19)

TABLE 3.10: Results for User Quality Analysis for Serwadda database (Intra session). Mean EER in % (standard deviation in brackets) for subjects in quality ranges. In addition to directions, different number of training samples are considered in the evaluation.

		SERWADDA DATABASE INTRA SESSION Enrollment samples			
	Quality Ranges	*5 samples*	*10 samples*	*15 samples*	*20 samples*
Down	*low*	14.61 (9.6)	12.97 (9.5)	8.02 (5.4)	**7.75 (6.0)**
	medium	10.48 (7.9)	7.45 (6.5)	**5.25 (5.3)**	5.28 (5.0)
	high	3.55 (4.4)	2.11 (2.5)	1.74 (2.2)	**1.50 (1.6)**
Up	*low*	13.86 (8.8)	10.45 (7.9)	7.77 (5.8)	**7.19 (6.4)**
	medium	7.93 (6.3)	4.38 (4.3)	3.95 (4.1)	**3.18 (3.5)**
	high	1.83 (2.1)	0.51 (0.7)	0.18 (0.3)	**0.11 (0.2)**
Right	*low*	11.26 (7.2)	9.25 (6.9)	6.09 (5.2)	**5.14 (3.8)**
	medium	5.63 (5.7)	3.85 (3.8)	2.90 (3.3)	**2.74 (2.6)**
	high	2.19 (2.1)	1.35 (1.4)	**0.85 (1.2)**	1.08 (1.2)
Left	*low*	10.36 (9.4)	8.25 (7.0)	**5.84 (6.2)**	6.21 (5.5)
	medium	6.51 (6.9)	5.20 (5.2)	2.78 (2.8)	**2.12 (2.5)**
	high	3.24 (4.6)	2.47 (3.5)	2.16 (2.8)	**1.43 (2.8)**

TABLE 3.11: Results for User Quality Analysis for Serwadda database (Inter sessions). Mean EER in % (standard deviation in brackets) for subjects in quality ranges. In addition to directions, different number of training samples are considered in the evaluation.

		SERWADDA DATABASE INTER SESSIONS Enrollment samples			
	Quality Ranges	*5 samples*	*10 samples*	*15 samples*	*20 samples*
Down	*low*	25.31 (14.1)	23.56 (14.7)	22.22 (13.7)	**19.79 (13.9)**
	medium	22.41 (16.6)	17.88 (14.0)	20.10 (14.4)	**16.89 (13.7)**
	high	12.57 (12.5)	11.27 (10.4)	**10.16 (12.5)**	15.76 (22.6)
Up	*low*	30.12 (14.9)	25.97 (15.1)	24.71 (18.2)	**22.86 (14.6)**
	medium	21.85 (13.8)	21.85 (15.9)	**17.56 (16.1)**	18.51 (14.7)
	high	15.30 (11.6)	12.14 (12.8)	**10.24 (9.1)**	11.97 (10.4)
Right	*low*	25.88 (16.4)	24.70 (15.7)	21.59 (13.4)	**20.43 (12.8)**
	medium	16.42 (14.4)	16.24 (13.8)	15.47 (14.4)	**13.78 (12.4)**
	high	21.27 (21.4)	**18.45 (17.3)**	19.35 (17.4)	18.49 (23.2)
Left	*low*	19.29 (14.5)	19.67 (13.6)	17.86 (15.6)	**17.49 (12.3)**
	medium	18.34 (17.3)	17.00 (16.9)	17.41 (18.4)	**13.87 (15.1)**
	high	17.82 (16.8)	18.30 (19.5)	**17.35 (20.7)**	18.49 (23.2)

TABLE 3.12: Results for User Quality Analysis for Frank database (Intra session). Mean EER in % (standard deviation in brackets) for subjects in quality ranges. In addition to directions, different number of training samples are considered in the evaluation.

		FRANK DATABASE INTRA SESSION Enrollment samples			
	Quality Ranges	*5 samples*	*10 samples*	*15 samples*	*20 samples*
Down	*low*	40.06 (0.0)	15.13 (0.0)	**13.96 (0.0)**	16.90 (0.0)
	medium	17.69 (11.3)	9.19 (2.4)	**4.67 (4.5)**	6.6 (4.9)
	high	13.3 (8.6)	8.71 (6.9)	8.83 (6.4)	**7.95 (7.3)**
Right	*low*	17.46 (0.0)	12.42 (0.0)	19.58 (0.0)	**8.28 (0.)**
	medium	15.27 (17.7)	10.06 (7.8)	**6.46 (3.9)**	7.61 (4.6)
	high	2.09 (1.7)	**1.00 (1.1)**	1.94 (2.4)	1.14 (2.5)
Left	*low*	-	-	-	-
	medium	9.31 (10.1)	5.55 (4.9)	9.46 (10.5)	**5.16 (7.1)**
	high	7.36 (6.1)	4.29 (3.4)	4.71 (4.3)	**3.29 (3.1)**

TABLE 3.13: Results for User Quality Analysis for Frank database (Inter sessions). Mean EER in % (standard deviation in brackets) for subjects in quality ranges. In addition to directions, different number of training samples are considered in the evaluation.

		FRANK DATABASE INTER SESSIONS Enrollment samples			
	Quality Ranges	*5 samples*	*10 samples*	*15 samples*	*20 samples*
Down	*low*	-	-	-	**-**
	medium	31.61 (27.9)	**19.24 (5.4)**	2.95 (4.8)	23.08 (25.2)
	high	5.66 (5.1)	2.95 (4.8)	4.82 (8.4)	**2.83 (3.9)**
Right	*low*	-	-		**-**
	medium	11.31 (9.1)	9.25 (10.9)	9.48 (11.1)	**4.08 (4.1)**
	high	11.24 (12.1)	12.00 (12.1)	**11.29 (11.8)**	12.43 (12.5)
Left	*low*	-	-	-	-
	medium	19.91 (10.1)	18.88 (10.3)	13.33 (12.2)	**11.28 (5.1)**
	high	17.33 (20.5)	11.80 (9.3)	**8.58 (11.1)**	9.46 (9.5)

TABLE 3.14: Results for User Quality Analysis for Antal database. Mean EER in % (standard deviation in brackets) for subjects in quality ranges. In addition to directions, different number of training samples are considered in the evaluation.

		ANTAL DATABASE INTRA SESSION enrollment samples			
	Quality Ranges	*5 samples*	*10 samples*	*15 samples*	*20 samples*
Down	*Low*	-	-	-	-
	Medium	13.77 (7.2)	12.75 (8.8)	**5.61 (4.9)**	7.01 (3.7)
	High	5.66 (5.0)	7.83 (6.1)	3.97 (4.7)	**0.26 (0.4)**
Up	*Low*	-	-	-	-
	Medium	-	-	-	-
	High	9.68 (15.8)	5.03 (6.2)	5.51 (6.2)	**0.85 (0.5)**
Right	*Low*	26.17 (5.0)	23.31 (11.3)	17.83 (5.4)	**17.75 (8.3)**
	Medium	18.15 (10.3)	14.59 (8.9)	9.95 (6.6)	**9.49 (5.8)**
	High	11.82 (8.3)	10.81 (8.5)	6.84 (5.8)	**6.25 (5.0)**
Left	*Low*	16.91 (4.6)	29.70 (6.7)	15.88 (6.5)	**8.64 (8.5)**
	Medium	18.90 (10.9)	14.59 (8.3)	11.70 (8.6)	**10.05 (7.6)**
	High	11.00 (6.1)	8.82 (6.5)	6.48 (5.3)	**4.72 (4.6)**

TABLE 3.15: Sample and user quality for the Callsign dataset with the deep learning architectures

	Callsign Database					
	Sample quality Quality ranges				**User quality Quality ranges**	
	Low	*Medium*	*High*		*Low*	*Medium*	*High*
GFNN	-	-	-		13.33	**9.27**	10.5
LSTM	13.67	11.42	**9.09**		18.88	9.07	**8.31**

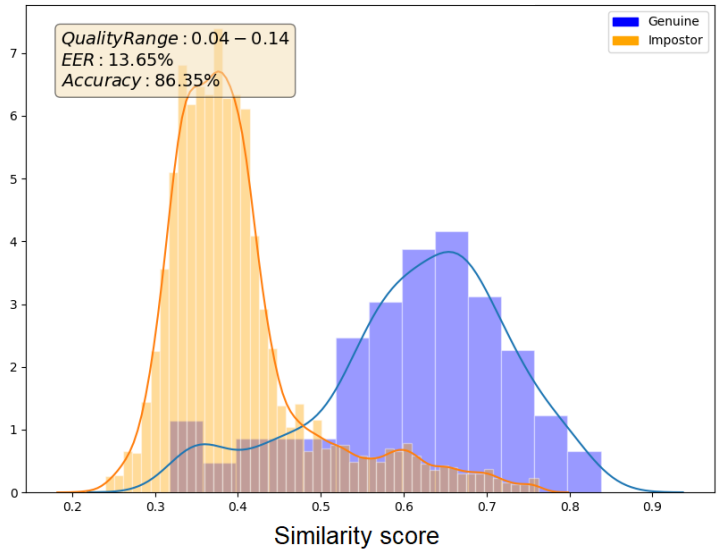


FIGURE 3.13: Sample quality in Callsign database with LSTM model. Similarity score distributions for genuine and impostor samples, low quality range.

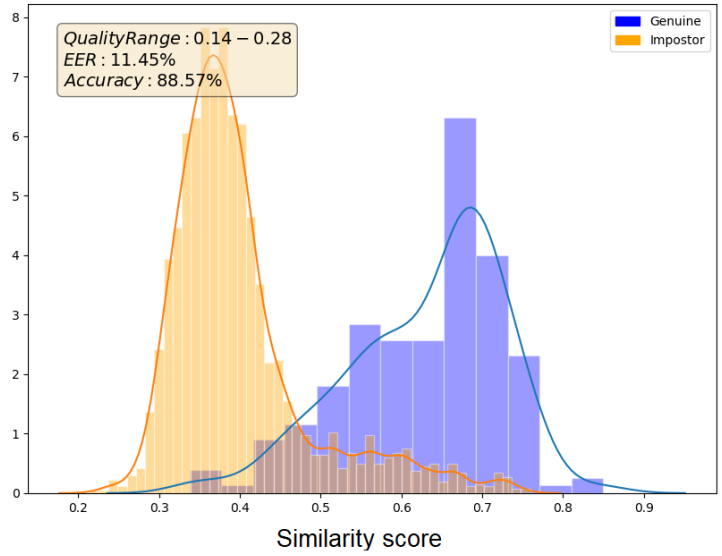


FIGURE 3.14: Sample quality in Callsign database with LSTM model. Similarity score distributions for genuine and impostor samples, medium quality range.

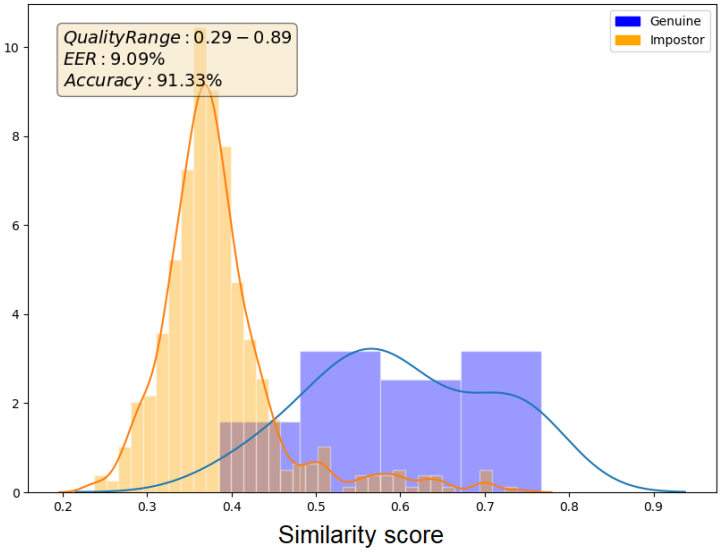


FIGURE 3.15: Sample quality in Callsign database with LSTM model. Similarity score distributions for genuine and impostor samples, high quality range.

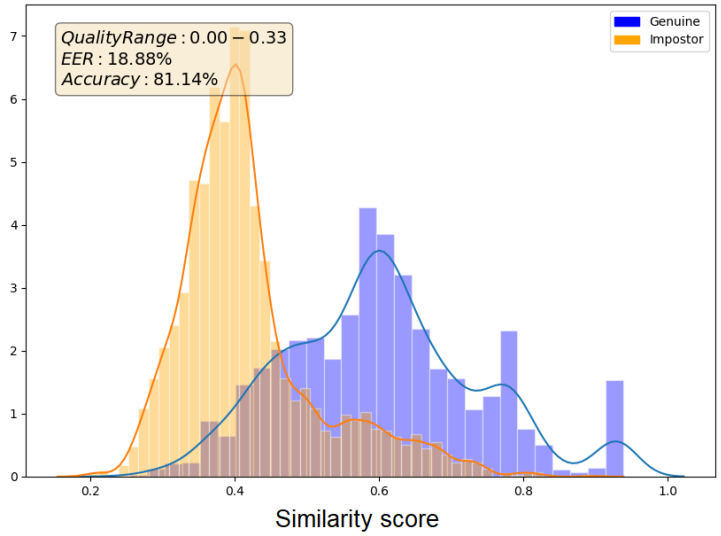


FIGURE 3.16: User quality in Callsign database with LSTM model. Similarity score distributions for genuine and impostor samples, low quality range.

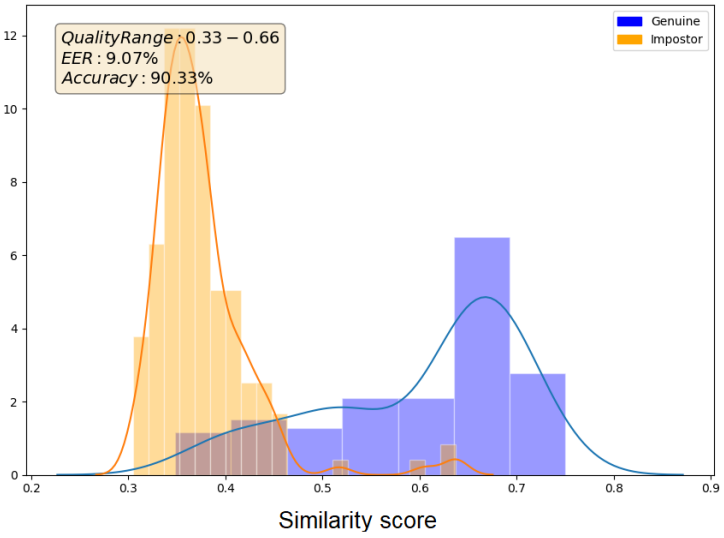


FIGURE 3.17: User quality in Callsign database with LSTM model. Similarity score distributions for genuine and impostor samples, medium quality range.

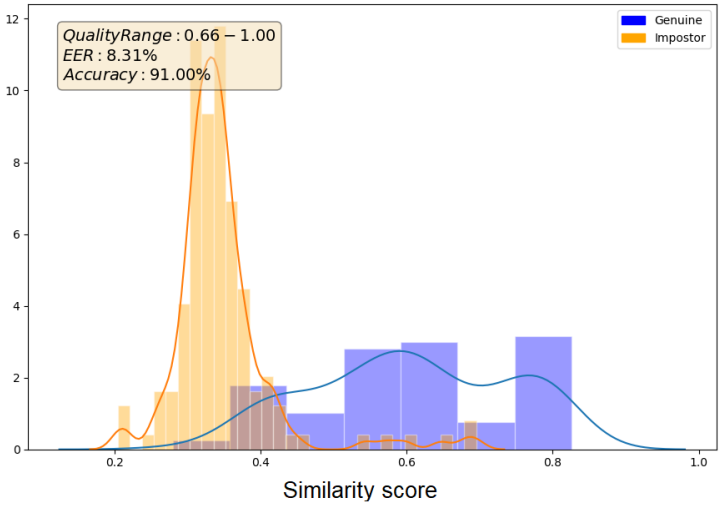


FIGURE 3.18: User quality in Callsign database with LSTM model. Similarity score distributions for genuine and impostor samples, high quality range.

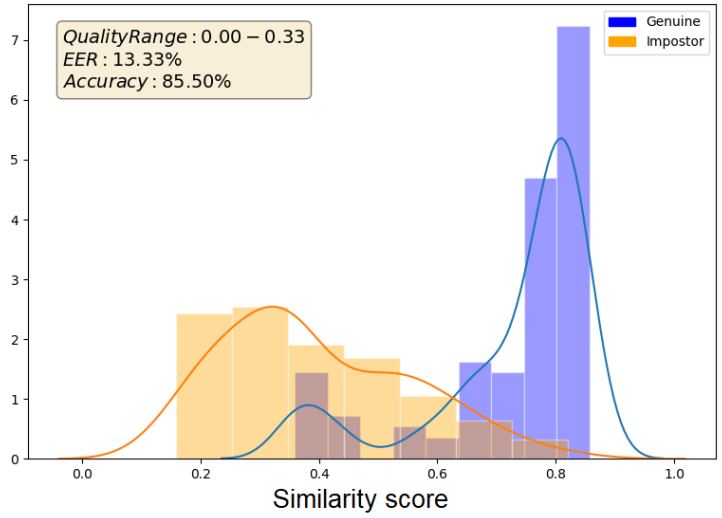


FIGURE 3.19: User quality in Callsign database with GFNN model. Similarity score distributions for genuine and impostor samples, low quality range.

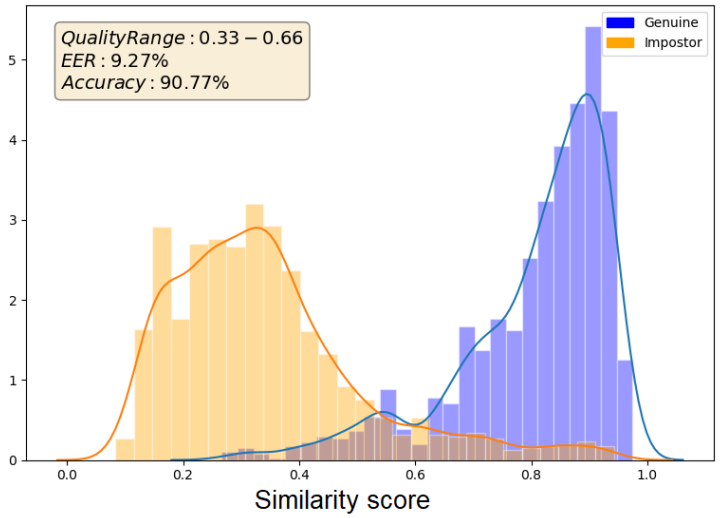


FIGURE 3.20: User quality in Callsign database with GFNN model. Similarity score distributions for genuine and impostor samples, medium quality range.

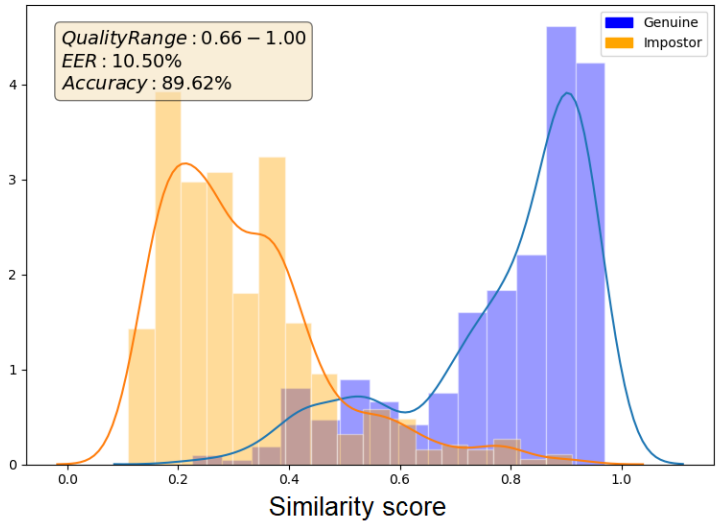


FIGURE 3.21: User quality in Callsign database with GFNN model. Similarity score distributions for genuine and impostor samples, high quality range.

3.3 Discussion

This Chapter explored extensively Swipe dynamics from a biometric authentication perspective. Two deep learning models for single Swipe authentication were introduced (GFNN and LSTM) and compared to State of the Art systems. The strengths of the proposed deep learning architectures reside in their ability to generalise over unseen subjects (without the necessity of re-training the model) and the ability to perform authentication with just one sample.

In addition, to improve model performance and explore furtherly user behaviours, it was introduced a novel quality metric for Swipe quality. Such metric is appliable to both user quality and samples quality. Results show that the estimation of the quality of the sample or the user gives a solid prediction on the performance of the system and the consistency of the user. It also suggests the amount of samples required to enroll a user depending on their quality rank (see Figure 3.12), or if during an authentication task it would be reasonable to reject a low quality sample due to the expected uncertainty of the prediction.

The next Chapter will explore PIN biometrics and present a fusion algorithm to combine the two modalities.

Chapter 4

PIN biometrics

4.1 Introduction

This Chapter will cover the second part of the framework, as a direct follow-up of Swipe biometrics. We will describe PIN data as a biometric modality related to user behaviours, highlighting a parallelism with keystroke dynamics. In this context, we will explore device interactions when providing a sequence of 4 or 5 digits on a mobile device through a touchscreen, similarly to swipe.

In this Chapter we will describe our dataset, the methodology and experiments to asses biometric verification performance through PIN and, finally, our proposed fusion method to combine the two modalities.

4.1.1 PIN gestures as behavioural data

PIN is a security measure used in a multitude of situations, as a main authentication or as part of two factors authentication systems, and can be used as a substitute for a password. Usually it consists of 4 or 5 digits between 0 and 9, with the option to replicate the same numbers. It has less variance compared to an alphanumerical password (which can be longer and contain special characters), but

unhence it's more difficult to guess. Moreover, passwords and PINs serve different purposes.

The possible combinations are theoretically "just" 10^{digits}, but the actual number can be reduced dramatically if we take into consideration a scenario with an expert attacker and a mobile device with a touchscreen. If fingerprints are partially visible in proximity of the numbers on the screen, the possible combinations are reduced to $digits!$ and it further decreases if one or more entries are repeated. For example, considering this scenario with a 4 digits PIN where two numbers are repeated and known, the possible combinations to guess drop from 10000 to 14. This is a very specific case, but in general there are many ways to guess a PIN code. For this reason, it's useful to consider device interaction when inserting the PIN as biometric data, to provide an added layer of protection. As for keystroke dynamics, PIN data can be considered time sequences with a known start and end. In addition, for mobile devices provided with a touchscreen sensor, positional data can also be retrieved too.

4.1.2 Differences between swipe and PIN gestures

Swipe and PIN gestures have similarities in terms of data collected, especially related to mobile devices provided with touch screens. Both can provide positional data (from the touch screen or the accelerometer/gyroscope), temporal data (from the timestamp), flags (start and end of a gesture, either flagged by a finger down/finger up identifier or by a PIN digit flag). Aside from those parallelisms, there are many differences that result in different challenges to overcome for each biometric modality. The first difference to mention is that swipe gestures can be generally analysed from a biometric perspective as a continuous authentication modality and not as a ceremony based or identification modality. Even for the aforementioned verification task, discussed in the previous Chapter, there could be the possibility to use more than one swipe sequence for authentication purposes.

In the case of PIN, a single attempt (if successful in providing the correct sequence of numbers) will be the only available data for biometric verification. It's not possible to provide continuous authentication on PIN due to the ceremonial nature of the task. We could however consider it as a special case of keystroke dynamics (in which data could be analysed in the background for authentication purposes), prompted by a request and constrained by the number of digits required for the PIN.

Another difference involves the finger position capture on the screen. While swipe data describe a trajectory, in the case of PIN data the sensor records sparse tappings located in the relative position of each digit. This means that those data cannot be interpolated, nor certain features such as velocity and acceleration related to the position be calculated. Instead, the majority of features are related to the timestamp, in contrast with Swipe features mostly relating to finger position and x and y derivatives.

4.2 PIN Authentication

For our study we designed a protocol to train models and evaluate their performance on PIN authentication, adopting a similar procedure as used for the swipe experimentation. For our experiments we exclusively utilised the database provided by Callsign, described in the next section.

4.2.1 Callsign dataset

The Callsign PIN dataset was collected using a range of different mobile devices with data entry from 385 different subjects. Similarly to the swipe dataset, PIN data are stored in a nested dictionary format, according to subject ID and transaction. Each transaction corresponds to a sequence of touch interactions performed by the subject to enter PIN digits. Subjects were requested to enter a 4 digit PIN on a standard ten digits entry screen.

The recorded features for each timestep in the sequence are: *timestamp*, *x and y relative positions* on the screen, *Accelerometer and Gyroscope*, *Major axis* of the touching finger area, and *event* flag which can assume the value of *pressed* or *released* according to the finger action. This was used to determine the press time, the flight time and other features related to the digits dynamics. Table 4.1 summarises the Callsign dataset information, compared to the swipe dataset.

The recording was performed on mobile devices on a verification query as a ceremony task, but the PIN request would only occur after a failed Swipe authentication attempt. This peculiarity of the framework has been taken into consideration when designing the multimodal fusion system.

TABLE 4.1: Comparison between Callsign datasets for Swipe and PIN data.

		PIN dataset	**Swipe dataset**
Database informations	*Source*	Callsign	Callsign
	Task	4 Entries	Horizontal swipe
	# Users	385	148
	95^{th} quantile sequence length	29	16
Features	*x position*	Relative position	Absolute position
	y position	Relative position	Absolute position
	Timestamp	✓	✓
	Major axis	✓	✓

4.2.2 Pre-processing and feature extraction

We decided to follow the same procedure applied to swipe data to pre-process the PIN data, when possible. This decision was taken after considering the previously mentioned similarities between the two dataset structures. In addition, we wanted to explore the same methodology for model evaluation as described in Section 3.1.

The first step was to remove *NaN* values from the rows, occurring during signal capturing for the same reasons explained in Chapter 3. Furthermore, we removed accelerometer and gyroscope data, as we wanted our model to rely on touch interactions only.

After this initial pre-processing of the data, we separated individual PIN sequences and stored them for each individual subject, according to the subject ID and the transaction ID. While doing this, we replaced the subject ID and transaction ID with anonymised numbers to ensure the full anonymisation of the dataset.

We also stored the length of each PIN sequence in terms of number of *timesteps*. With this, we were able to remove outliers based on the 95^{th} quantile, similarly to the procedure used previously for Swipe data.

Unfortunately we could not apply further cleaning of the dataset, due to the fact that PIN gestures, even if related to a more constrained task compared to Swipe gestures, are more complex to understand and evaluate in terms of behaviours. In the case of Swipe dynamics, we could recognise and reject samples not conforming to the given task (i.e. back-and-forth swipes), but this was not possible with PIN data; multi-tapping a single digits could be due to a delay of the touchscreen sensors or the sampling frequency, but it was not a criteria for rejection. In general, it was not possible (neither with visual clues nor descriptive statistics) to define what samples could have been considered faulty or sources of errors.

For PIN data we created two new data structures. One structure preserved the raw data points for each *timestep* in the PIN sequence to train a recurrent deep learning model. The other structure contained extracted features stored in a row for each PIN sequence.

The first data structure is a list of 3D matrixes, each element in the list contains a matrix with all the PIN sequences from a given user. The matrix dimensions correspond to *timesteps*, *PIN transaction*, and *raw features*.

The number of *timesteps* as previously mentioned is equal to the 95^{th} quantile of sequence lengths calculated over all the sequences in the database, with padded values for shorter sequences. For each sequence, the starting value of *timestamp* is subtracted from all the remaining *timestamp* values. With this method, every sample started at 0 seconds.

TABLE 4.2: Raw features from PIN sequences.

Identifier	Feature name	Description
1	*Timestamp*	Touch event timestamp
2	*X relative position*	X coordinate value at touch point relative to digit area.
3	*Y relative position*	Y coordinate value at touch point relative to digit area.
4	*Major axis*	Major axis described by the finger touching the screen at touch point
5	*Sequence length*	Number of actual touch points for the given PIN gesture.

The raw features are listed in Table 4.2. We applied a z-score normalisation to each individual feature vector, calculating global mean and standard deviation from the training set data.

The second data structure is comprised of extracted features based on keystroke studies. This structure comprises a list of 2D matrixes, each one containing the data of a different subject from the original database, with each row corresponding to one *transaction* and with the columns representing the *extracted features*.

To describe the extracted features, it's necessary to mention the common time intervals occurring during keystroke dynamics.

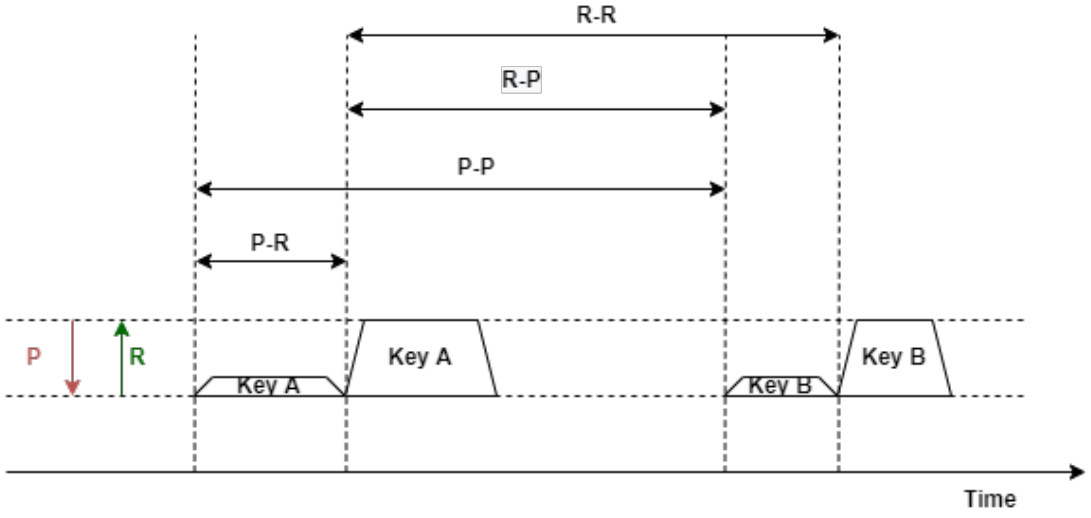


FIGURE 4.1: Keystroke time intervals based on Press (P) and Release (R) events.

Taking as reference Figure 4.1 and considering P and R respectively as *Press* and *Release* events, we can define the following intervals:

- ***P-R***: the time elapsed between the press and release of a key. It's also referred to as *dwell time*.
- ***R-P***: the time between the release of a key and the press of the following key. Also known as *flight time*.
- ***P-P***: the time between the press of a key and the press of the following key. Also known as *inter-key timing*.
- ***R-R***: the time elapsed between two consecutive key releases.

These intervals are descriptive of keystroke behaviours on mechanical keyboards, but can be extended to virtual keyboards on touchscreen devices and to PIN dynamics on mobile devices. Considering that tyPINg on a keyboard can be done with both hands (i.e. two digits can be pressed at the same time) and that for a virtual keyboard there might be input latency, it's reasonable to assume that occasionally the *flight time* and the *R-R* time could have negative values.

TABLE 4.3: Second set of PIN extracted features based on previous studies [4–6]

Identifier	**Feature**	**Description**
1	*Max m.a.*	Maximum value recorded of the major axis of the finger.
2-5	*Mean X digits*	Average value of x coordinate for each PIN digit.
6-9	*Mean Y digits*	Average value of y coordinate for each PIN digit.
10-12	*P-P*	Inter-key timing for each pair of digits.
13-15	*P-R*	Dwell time for each PIN digit.
16-19	*R-P*	Flight time for each pair of digits.
20-22	R-R	Inter-key release time for each pair of digits.

The positional-based features are calculated by averaging the coordinate values recorded by the touchscreen from the beginning to the end of a digit entry event (so from press to release). The time-based features correspond to the previously defined intervals for each digit (or pair of digits) during the PIN recording.

We discarded the *Sequence length* because, considering the task, it's reasonable to expect that each tap on the touchscreen could provide multiple and variable number of touch points for each digit and it would not be a consistent feature.

As for the positional values, being recorded relative to the digits, it seemed more reasonable to separate them between events instead of calculating a global value that could have similarities inter subjects.

The features were z-score normalised, with the global mean and standard deviation calculated on the training set and used to normalise both training and test set. After this, we proceeded with evaluating the models.

4.2.3 Experiments and results

We evaluated the performance of two deep learning authentication models with architectures similar to the swipe models in Chapter 3. For each model, it was followed the same experimental procedure:

- Hyperparameters fitting through manual grid search.
- Training of the embedding model over 90% of the users from the dataset.
- Testing the model on the remaining 10% of the unseen subjects in the database.

Each step is furtherly described in details.

We manually searched the best hyperparameters amongst the number of hidden layers, output dimension, learning rates, and training epochs. As explained in Section 3.1.7.3, we iteratively changed each hyperparameter and explored training performance until we found the best set of parameters based on trial-and-error criteria; similarly to the swipe models, hidden layers and output dimension parameters were intrinsically connected.

The first model, trained on the first 3D data structure, was an LSTM-based architecture with the number of timesteps for the input vector equal to 25 (95^{th} quantile over the lengths of all data). For PIN data, median and mean of lengths distribution were equal.The second model, trained on the 2D data structure with extracted features, was a feed-forward architecture with two fully connected layers.

Both models returned an L2 normalised output vector of 128 elements and the cost function was again the *triplet margin loss* that already proved to be optimal for data clustering of swipe data. We used 90% of the data to train the models and the remaining data for testing. Table 4.4 shows the hyperparameters selected for the evaluation.

TABLE 4.4: Hyperparameters for PIN

	PINLSTM	**PINGFNN**
Input features	5	22
Hidden dimension	64	64
Output dimension	128	128
Batch size	256	256
Training epochs	3000	3000
Training set (#subjects)	90% (347)	90% (347)
Test set (#subjects)	10% (38)	10% (38)
Normalisation	z-score	z-score

For testing, we used the first 10 samples in chronological order for each unique subject in the test set to generate the subject template. The template is a single vector of 128 elements, obtained by averaging the 10 deep feature vectors obtained by feeding the enrollment samples through the models. We calculated the Euclidean distances between each template and the remaining samples in the test set, then we calculated the optimal threshold and the global EER for each model. In Figure 4.2 is shown the last batch of embeddings projected on the first three principal components after applying PCA.

We repeated the experiments several times for consistency of the methodology. Results are shown in Table 4.5 for the two models assessed in terms of EER.

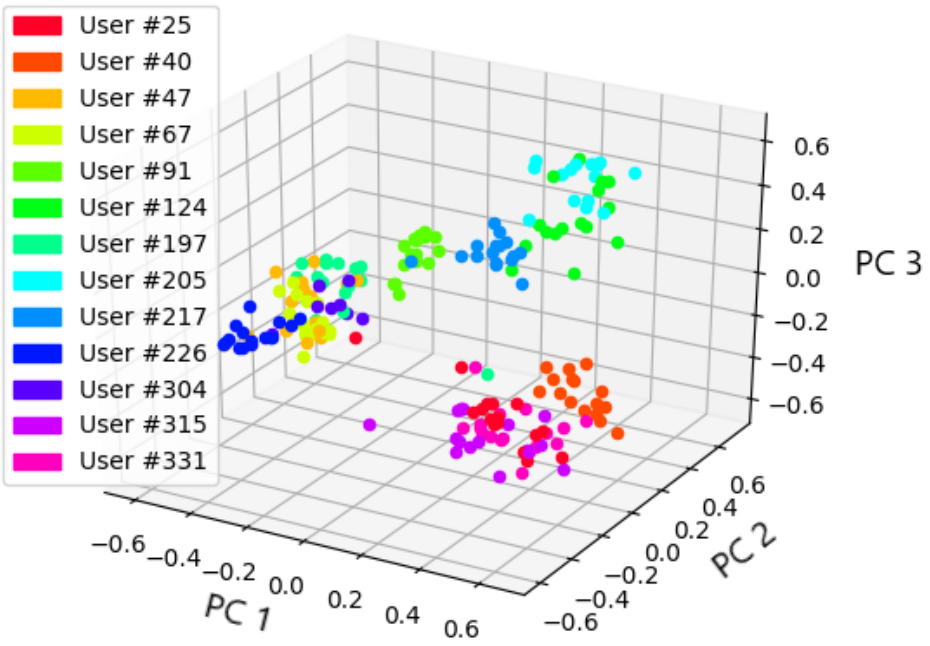


FIGURE 4.2: PCA 3-D projections of extracted features from last training batch PIN data. Different colours corresponds to different subjects.

TABLE 4.5: Results for PIN models.

	PINLSTM	**PINGFNN**
EER [%]	13.72	20

Compared to swipe authentication, PIN provides slightly better results with the LSTM, probably due to the larger amount of data for training, but there's still a considerable margin of error. To overcome this issue, we developed a fusion algorithm to increase the performance of the authentication system at this stage of the framework (see Figure 4.3).

4.3 Swipe and PIN fusion

4.3.1 Introduction

After assessing the authentication models for PIN and swipe, the next step in our framework is to combine these two modalities to improve the biometric system performance. Fusion models or in general multimodal systems are common solutions when multiple authentication modalities are implemented, sequentially or at the same time, to reduce the classification error.

In this Section, the most common fusion methods will be described in terms of mathematical and practical implementation. Our own fusion method will be presented and justified according to the current framework, and the results of our experimentation will be shown.

4.3.2 Multimodal systems

Multi-biometrics is a complex topic that can be studied from a multitude of perspectives with systems typically designed for a specific task in which biometric recognition needs to be performed. Multi-biometrics presents in a multitude of scenarios which need to be identified at first to understand the optimal fusion method. For example, for the same modality there might be different scenarios in which multiple algorithms, instances, or sensors are used. In the case of this study, we are considering a specific scenario with multiple modalities with one sensor each. The next question is: at which *Level* should the fusion be enabled? To answer this question, it's important to define the different Levels. In order of "appearance" in an authentication system, these are *Feature Level, Model Level, Score Level,* and *Decision Level* fusion.

- ***Feature Level:*** The fusion happens during the feature extraction process. Either the extracted features from the different sensors are combined in a

single feature vector or brand new features are computed from the combination of the previous data. The new feature vector is fed into a classification model.

- ***Model Level:*** Feature vectors are calculated separately and fed to different models that are blended together at some point into a new model. It's possible that a part of the originals model weights is left untouched.

- ***Score Level:*** Prediction scores from dedicated models for each of the modalities are fused together, returning a new score (or set of scores). The fusion could be performed by an algorithm or a brand new model.

- ***Decision Level:*** The decision scores (binary value for authentication models or categorical for an identification model) from the different models are evaluated, usually based on statistical criteria, and a final decision is generated.

For all the Levels, there is a variety of algorithms to perform the fusion. At feature Level, a possibility is to reduce the features from the various modalities using PCA or alternative and then concatenate them in a new feature vector. At Model Level it becomes more complicated, since the mathematical operations need to be embedded in the models; this requires access to the raw architecture and the complexity may vary depending on the number and types of predictors. In general there is no specific technique for Feature Level and Model Level, this is due to the fact that it's relative to the modalities, features and models in use.

For Score Level and Decision Level there are common techniques used for fusion, for example a linear combination of the scores (or weighted scores), a nearest neighbour algorithm or other distance metrics, or in the case of the Decision Level a majority vote (if the modalities are more than two). Statistical models, like Naive Bayes based systems, can also be used to generate a new score or decision, depending on the distribution of the labels or the scores from the predictive models.

Choosing the optimal technique depends on the data that are provided, not only in terms of features and distributions, but also regarding the amount of data

available for training. As for the Level, choosing where to execute the fusion (or if implementing multiple fusion Levels) depends on the task and the flow of data. In some cases it is not possible to implement a multi-modal system at certain Levels. For example, in the case of a two factor authentication when modalities are in cascade, it's possible but not necessarily optimal to implement the fusion at Feature Level. Or, as stated before, if there are only two modalities it's not possible to implement the majority vote at Decision Level.

In the case of this study, there was a very specific framework when considering swipe and PIN authentication. The latter modality would be prompted by the system only after a recorded failure from the former authentication, as shown in the updated framework in Figure 4.3.

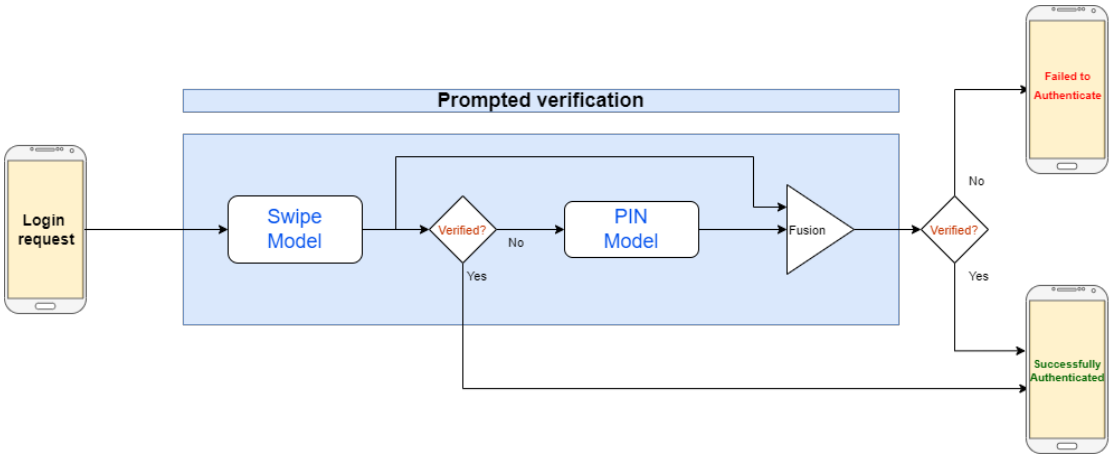


FIGURE 4.3: Adapted framework of the verification pipeline for Swipe and PIN. PIN verification is prompted only after a swipe failure, which means that the fusion cannot happen before that event.

Moreover, PIN and swipe data do not have a fixed number of timesteps, which would make it very difficult to fuse the data at Feature Level, especially when using the LSTM models (better performing compared to the GFNN models).

The first intuition was to implement the fusion at *Score Level*, since all the proposed architectures from both modalities have the same output vector and same kind of similarity score, calculated using the Euclidean distance between the template (defined in equation 3.16) and the target sample. Most importantly, the

similarity scores can only assume fixed values in a range, which did not change across the models.

With these premises, various algorithms could be used without the necessity of a further normalisation of the score (which was already provided). But the major issue of score fusion methods, like linear combination or support vector machines, is that they ignore the contextual information related to this specific framework, defined as when and how the PIN authentication is prompted. In our case, the context is a "*failed verification from previous modality*". For this reason, a dedicated fusion method was implemented: the *penalty fusion*.

4.3.3 Penalty fusion

The fusion method proposed in this study is novel has not been applied previously for biometric authentication. The fusion is performed on the dissimilarity score obtained from swipe and PIN authentication calculating the Euclidean distances between templates and samples from each modality.

Considering T_S and T_P the subject templates for swipe and PIN respectively, $q_{S,i}$ and $q_{P,i}$ the embeddings of the *i-th* attempts of authentication from a subject against these templates on the two modalities, the dissimilarity scores d_S and d_P are the Euclidean distances between the templates and the embeddings.

Considering th_S and th_P the authentication thresholds for the two modalities, in a way such that if $d > th$ then the authentication fails, the aim of the fusion method is to modify the PIN score d_P based on the error magnitude during the swipe authentication.

Therefore, the fusion method provides a penalty on the follow-up modality based on the performance on the first modality; the assumption is that a genuine user, when failing to authenticate themselves, performs better still against their template compared to a random imposter. The mathematical formulation for this methodology is:

$$d_P^{fusion} = d_P \cdot (\gamma + e_S) \tag{4.1}$$

with γ being the learning weight and e_S the error on swipe authentication, defined as d_S - th_S, always strictly positive in the interval $(0, 1 - th_S]$.

The optimal value for γ is found through regression, under the condition that the penalty fusion should increase the PIN's dissimilarity score if the error on swipe authentication was large, while decreasing it if the error was small. Considering the range of the error, γ ideally should converge to a value smaller than 1.

To perform the regression on γ, the cost *function*, the *weight updater* and the *gradients* were calculated and implemented in Python. We performed training and testing with fixed thresholds from the previous models. Such thresholds correspond to the discriminant value of the models dissimilarity scores below which the subject is authenticated. If we considered the thresholds as extra parameters, the system would have not converged to a single solution. Nevertheless, after the training, we also estimated a new possible threshold for the new set of fusion scores for comparison.

Since to train and test this fusion method paired data were required from the same subject. To avoid overfitting towards the most populated class (genuine subjects or impostor subjects) we considered the scaling parameter C for each different class during the *weight update* process. This parameter is defined as follows:

$$C_i = N\_samples/(N\_classes \cdot N\_occurrences_i) \tag{4.2}$$

With C_i being the weight for the *i-th* class, *N_classes* being the number of classes considered (in our case, two), and $N\_occurrences_i$ being the number of occurrences for the *i-th* class in the training set.

4.3.4 Experiments and results

To train and test the fusion method it was necessary to find paired data from PIN and swipe and evaluate the dissimilarity scores from each. To find the correct pairs from the same subject we matched the transaction IDs; we could not generate

artificial pairs since it would bias dramatically the results. For both genuine attempts and impostor attempts, both swipe and PIN data had to be from the same session.

For this reason, the amount of usable data from the original datasets decreased consistently and we could only rely on around three thousand pairs. Due to the dataset reduction and the fact that the new subset listed real impostor attempts and not zero effort impostor authentication, the EER of the PIN model was slightly higher compared to the previous evaluations.

We trained the fusion system on 80% of the available pairs and tested on the remaining 20%. We used the same set to also train an SVM with dissimilarity scores as input vector. We repeated the process several times for consistency, randomising the train and test set and changing the initial value of the weight γ. Since the training set was very small, we didn't create mini-subsets and at each iteration we used all the data, computing the gradients on the average error from all samples. We also trained the model several times varying the initial value of γ to ensure its convergence to a singular optimal value, as can be seen in Figure 4.4

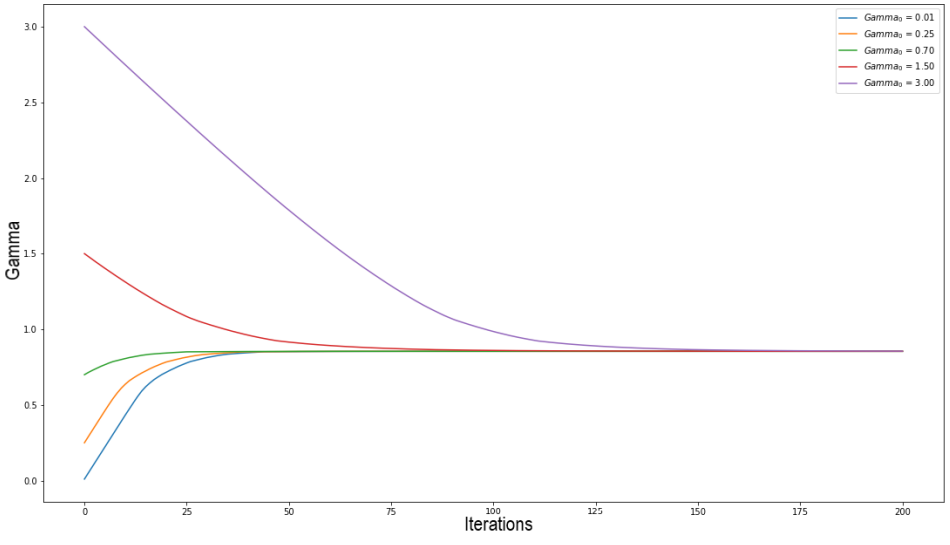


FIGURE 4.4: Weight update over training from different initial values. For higher values, more iterations are needed to reach convergence to the optimal value.

Figure 4.5 shows the mean absolute error at each iteration. Table 4.6 and Table 4.7 show the experiment parameters and the results respectively. In addition, in

Figure 4.6 are shown the ROC curves over a 6 fold cross-validation for the proposed methodology. The model performance proves to be consistent, as can be seen by the area under curve (AUC) values. As further information, the thresholds for the three authentication systems (original PIN model and the two fusion methods) are also provided as well as the corresponding ROC curves in Figure 4.7.

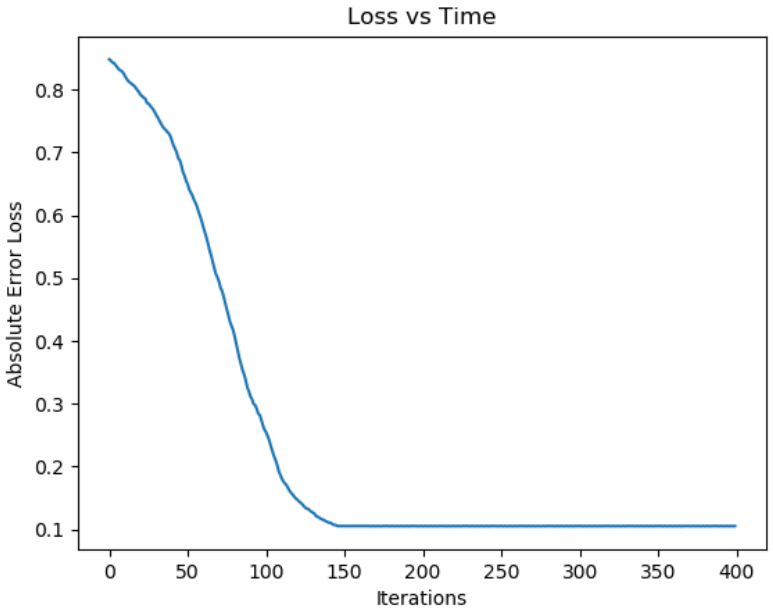


FIGURE 4.5: Mean absolute error per iteration during the training of the fusion system.

TABLE 4.6: Parameters for the penalty fusion experiments. γ_0 is the initialised value of γ.

Parameter	**Value**
Training set	2698
Test set	675
Iterations	400
Learning rate	0.01
Highestγ_0	3.0

TABLE 4.7: Results and thresholds for the score fusion methods compared with raw scores.

	PIN scores	**SVM fusion**	**Penalty fusion**
EER [%]	15.32	13.55	7.84
Threshold	0.25	1.04	0.23

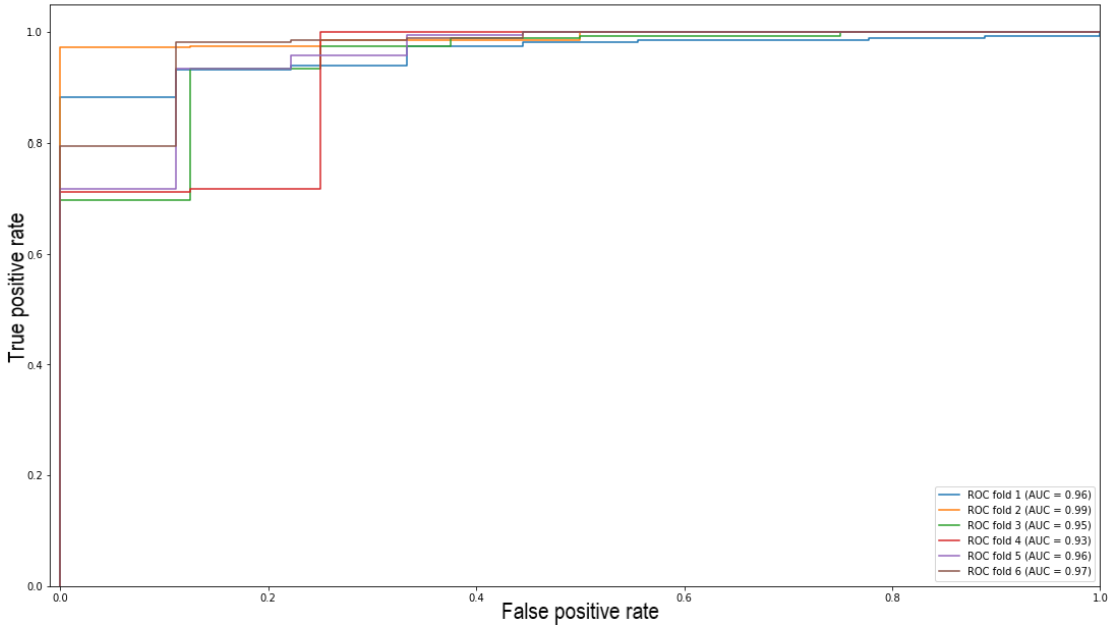


FIGURE 4.6: 6 folds cross-validation for the penalty fusion model.

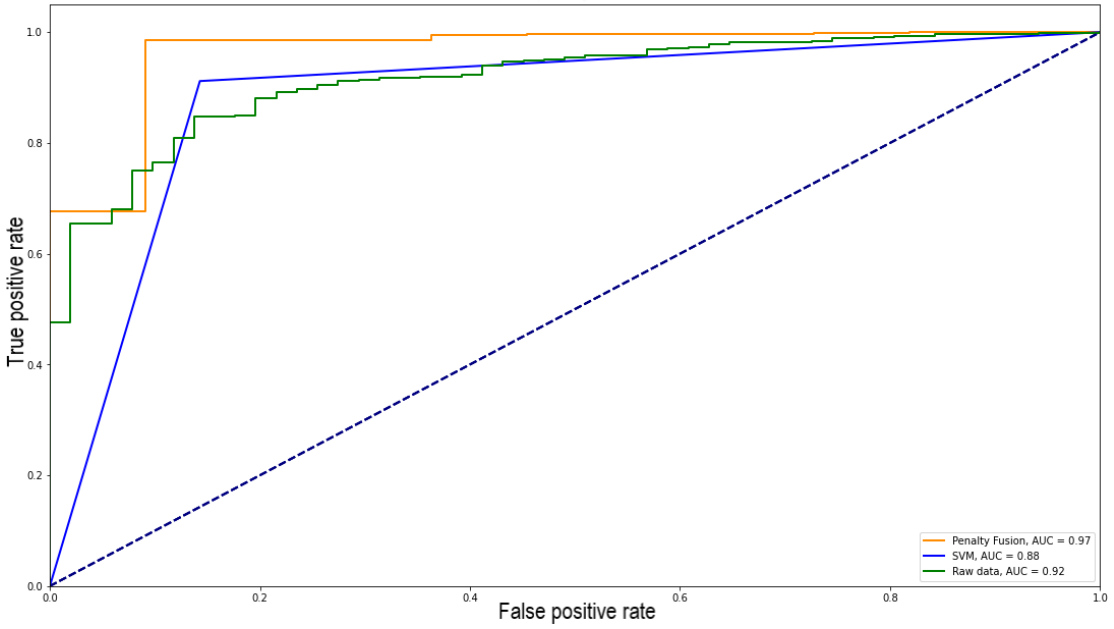


FIGURE 4.7: ROC curves for SVM fusion, penalty fusion and original PIN scores. The dotted line represents a random classifier.

As predicted, the SVM is not improving the performance enough compared to our fusion system. We introduced a non-linearity that considers the specific task and adheres to the classification criteria.

It can be seen that the optimal threshold after the Penalty fusion doesn't deviate

considerably from the original threshold. This is an added value to this methodology, making it consistent and more difficult to spot by external attackers. It's also very fast to re-train even with high amount of data, due to the one and only trainable parameter γ.

4.4 Discussion

In this Chapter we evaluated PIN dynamics on mobile devices as behavioural biometrics for user verification. As the premises and the data structure of PIN data are similar to Swipe data, we assessed the performance of two deep learning models proposing the same architecture and cost function used to assess Swipe biometrics. Our findings show that the RNN with the LSTM cell outperforms the fully connected model, with 13.72% EER on the testing set. We followed the same protocol for enrollment and testing, such as only one sample for each verification task and enrollment samples recorded before the testing samples.

These results show that the model can generalise and generate deep features specific to the subject, but it's still subject to misclassification due to the spread of user's sample, with few overlapping as can be seen in the 3D projections in figure 4.2. This can be due to variation in behaviours over time, use of different devices or, in the worst possible case, changes in writing hand.

To improve the model performance without increasing the number of samples requested for the authentication, we implemented a fusion model based on the Framework contextual structure, which penalises or eases the verification decision based on the magnitude of the failure from the previous modality.

Our algorithm proved to greatly improve classification performance, with lowest computational expenses (calculated based on the number of weights of the fusion model, in this case just one), low training cost (each input vector is only comprised of a pair of scores) and avoiding risks of data leaks, since the model receives as input just the dissimilarity scores and no sensible data from the mobile device (hence,

no need of location data, nor accelerometer data, nor to record and transfer any personal data of the user).

Nevertheless, the whole authentication process can still be improved. To accomplish this goal, we added a new modality to the current framework: Electrocardiogram authentication.

Chapter 5

Electrocardiogram biometrics

5.1 Introduction

5.1.1 Problem statement

In the past years, electrocardiogram (ECG) data and heart activity have been studied not only from a clinical point of view, but also from a biometric perspective. Past studies have demonstrated how features that characterise the individual can be extracted from the electrocardiogram signal and how these features can be used for encryption or identification. The drawback consists in the difficulty to record the signal, pre-process it and implement the system in a real life scenario (due to recording times and cumbersome devices required). However, the fast growth in technology led to the development of miniaturised systems and wearable devices, making it possible to record the electrocardiogram signal without the need for large devices or leads and wires. Furthermore, the Internet of Things, alongside the most recent smartwatches and smartphones being capable of fast computation and comprised of large data storage, suggest that electrocardiogram biometrics for identity verification with wearable devices could be feasible.

In this Chapter, we will explore this possibility by analysing the electrocardiogram signal, the available datasets, wearable devices for signal capture and a few models for user authentication with the lowest recording time.

5.1.2 ECG overview

The heart is the biggest autonomous muscle of the human body and it's the only one that never ceases its activity during the whole life of a person. Some characteristics like the heart rate or intensity of the activity vary over time due to growth, emotional or environmental conditions, and illness, but in general it provides a unique description of the person.

The heart can be divided into four different sections: the atrial and ventricular, two per side. To function properly, the activity of the four sections must be perfectly timed and synchronised or else blood would not be pumped correctly and there would be turbulence and reflux in the heart itself.

The timing of the activity is regulated by a small network of nerves which electrical activity triggers depolarisation and therefore the contraction of the appropriate muscles. There are four important parts of this network: the sinoatrial and atrioventricular nodes that act as pacemakers, the bundle of Hiss that delivers the signal to the ventricular nerves and lastly the Purkinje fibres innervating the large surface of the two ventricles.

An electrocardiogram is a recording of the aforementioned heart's electrical activity, captured using electrodes attached to specific body locations. These electrical activities are the sum of all the electrical waves occurring during the depolarisation of the cardiac muscles and can be visualised as a moving vector, the cardiac vector, that travels through the heart. In a normal cardiac cycle, there are three phases (Fig. 5.1):

- Initially, the atrial depolarization is triggered by the sinoatrial node (P wave). Atrial repolarisation follows this depolarization. Then a new trigger by the atrioventricular node occurs after a short delay.

- The signal travels through the bundle of Hiss to the Purkinje fibres, activating the ventricular depolarization and contraction (QRS complex). During this phase, the cardiac vector forms a triangular wave, hence the resulting characteristic complex.

- During the last phase, ventricular relaxation and repolarisation occur (T wave).

For a diagnostic ECG recording, 10 electrodes are positioned on the body of the patient, as shown in Figure 5.2. *RA, LA, LL* and *RL* refer to the anatomical positions for the electrodes, respectively: *right arm, left arm, left leg* and *right leg*. The first four electrodes contribute to the first six limb channels of the ECG, defined as:

- *Channel I: from RA to LA.*

- *Channel II: from RA to LL.*

- *Channel III: from LA to LL.*

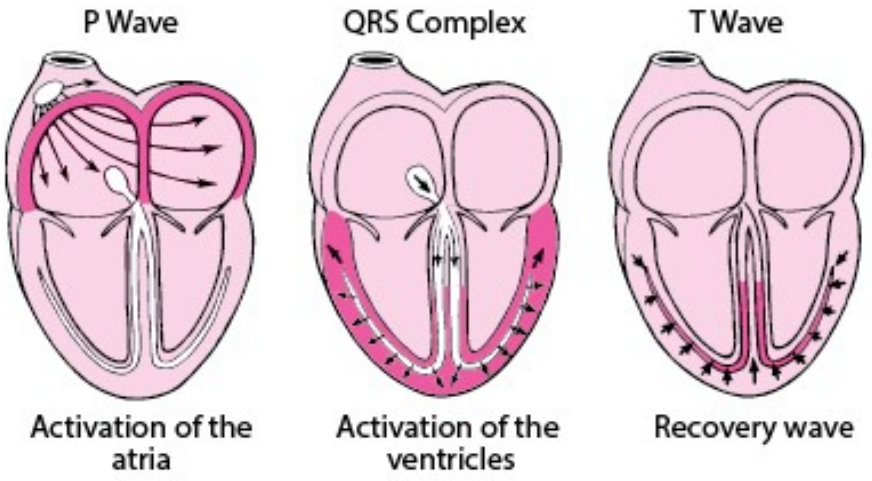
- *aVR: from the average value between LA and LL to RA.*

- *aVL: from the average value between RA and LL to LA.*

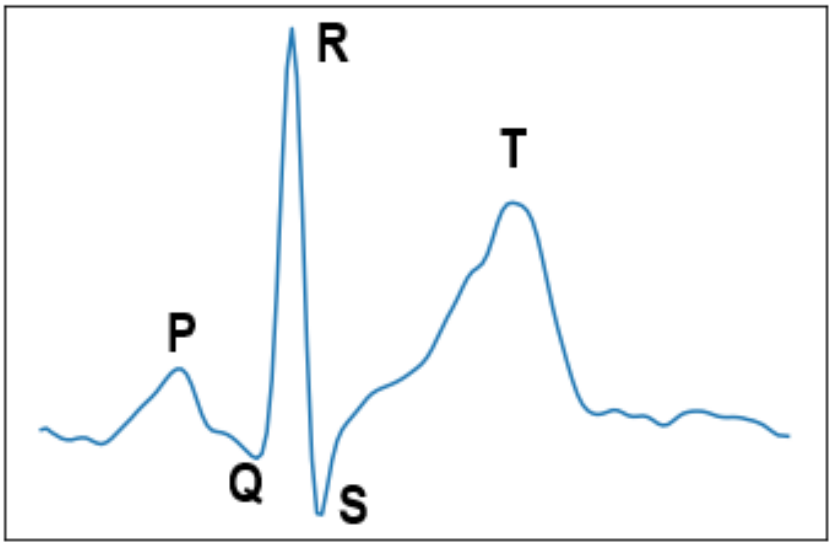
- *aVF: from the average value between RA and LA to LL.*

The *right leg* electrode is used as a *driven right leg circuit* to denoise the signal and reduce common-mode interference by actively removing it.

The last six electrodes provide six more channels, the *precordial leads*. The signals are recorded through the differences between each one of the 6 electrodes and the average values of the first 3.



(A) Heart activity during one cycle [134].



(B) Electrocardiogram signal for I lead.

FIGURE 5.1: ECG example for one heartbeat. In Figure (A) it's showed the propagation of the electrical signal across the myofibers. P wave, QRS complex and T wave are labeled in proximity of the peaks in Figure (B).

The bandwidth of the cardiac cycle is substantial. It may overlap with other signals (not necessarily biological), but it is consistent with itself (considering healthy subjects) and is unique to each subject. The entire cycle follows the same rules, and timing is triggered by the sinoatrial node. This leads to two important conclusions:

1. Another cycle cannot occur unless there is repolarisation of the muscle fibres.

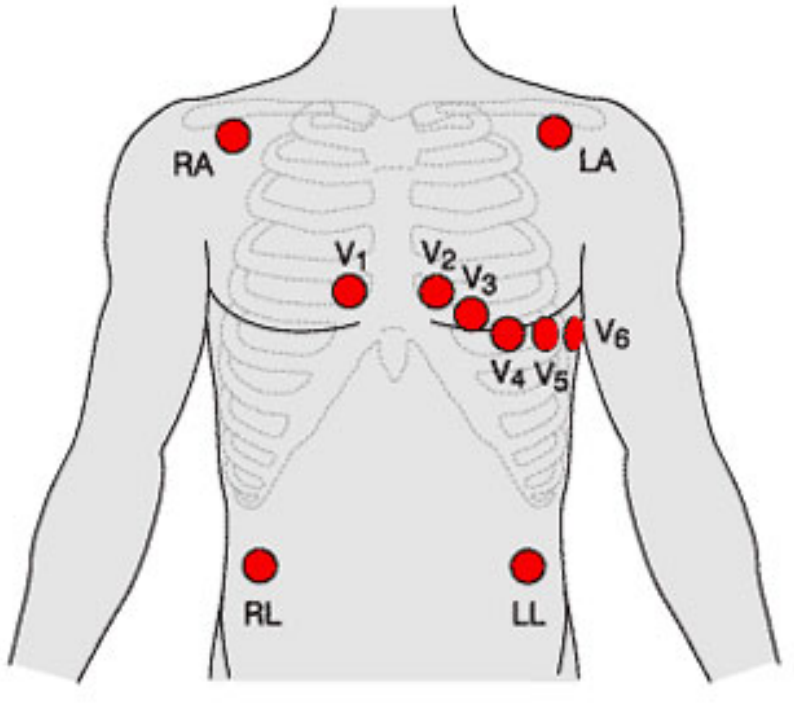


FIGURE 5.2: Anatomical locations of the electrodes for ECG recording [3].

2. Only the heart rate can be affected by external factors and cannot exceed a particular value.

Therefore, the heart rate is irrelevant for biometric purposes, and the only useful data we can extract are from the single cycle during the various phases.

5.1.3 Framework

The framework for ECG recognition is very similar to swipe and PIN, however with the latter two modalities we recorded a single sample representing one single action, well defined by the start and end of the sequence (the horizontal slide or the PIN entries respectively). In the case of the ECG, we need to arbitrary decide the start and the end of the recording and select an appropriate time window.

Figure 5.3 illustrates the entire framework for ECG verification. The flowchart refers to the data processing for a single subject.

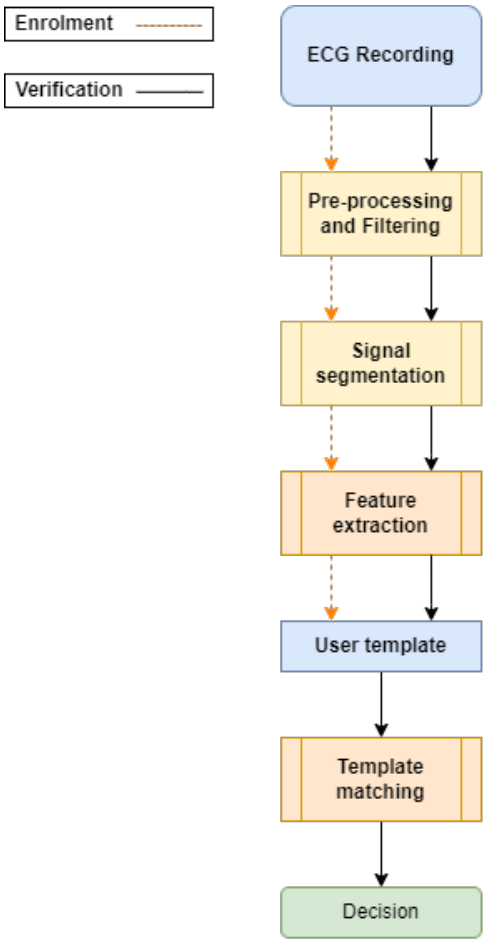


FIGURE 5.3: Generic framework for ECG verification, regardless of the model used for decision making. This block diagram is valid for each subject on a non-continuous authenticaton.

Signal segmentation is the process of selecting a specific window from the whole recording that would be fit for the feature extraction. The *User template* and the *Template matching* are managed by the classifier, but this will be explained in detail in the following sections.

This framework provides an evaluation of ECG verification alone, but the aim of this research is to combine all the three modalities for mobile authentication. In our original framework, ECG authentication should happen in the background during Swipe and/or PIN verification; this suggests the ideal time window for capturing and processing the ECG data. We considered this when exploring the methodology and the possible devices.

Unfortunately, as mentioned before, we were impaired in our ability to conduct a

data collection using the paired sensors to provide further evidence of the feasibility of this framework due to the ongoing COVID-19 pandemic and the related restrictions. Nevertheless, we utilised our time by conducting experiments on public datasets while minimising the recording and authentication time for ECG data, in order to be fit for mobile use. We found an open-source wearable device able to connect in real time with a mobile device and stream data, with device specification and sensor resolution comparable in performance to the devices used to collect the publicly available datasets. Finally, we formulated a theoretical fusion protocol for further follow-up studies in Chapter 6.

5.2 Databases and devices

5.2.1 Public databases

The *WeSAD* dataset [135] consists of ECG recordings from 15 subjects collected from a RespiBAN device for 36 minutes per subject. Data were collected from the subjects while they were sitting, speaking, and watching video clips in the sitting position. RespiBAN data were collected at 16-bit sampling resolution and 700 Hz sampling frequency from a chest band. Subject genders did not distribute equally (3 females, 12 males). In this study, we used RespiBAN data from all 15 subjects.

E-HOL-03-0202-003 [136] consists of 24 hours of continuous digital Holter recordings from 202 healthy subjects collected from three electrodes positioned to obtain the pseudo-orthogonal lead configuration. The dataset indicated that participants had no cardiovascular diseases or disorders, high blood pressure, or chronic illness. The population consists of equally distributed genders (100 males, 100 females and two undefined). The data were captured after a 20 minutes resting (supine) period, and 200 Hz sampling frequency and 10mV of amplitude resolution recorded the ECG signals.

The *ECG_ID* dataset was recorded especially for biometric purposes by *Lugovaya et al.* [137] in 2005 and it's widely used in related studies to assess biometric

performances on ECG signals. It consists of 310 ECG recordings from 90 different subjects over six month period of time. Participation to the data collection was voluntary. The subjects are split into 44 men and 46 women aged between 13 and 75 years. The number of recordings collected from the same subject varies from 2 (collected on the same day) to 20 (collected periodically over the 6 months). Each recording was a 20 seconds-long ECG Channe I, sampled at 500 Hz with 12 bit digital resolution, and $\pm 10\ mV$ range.

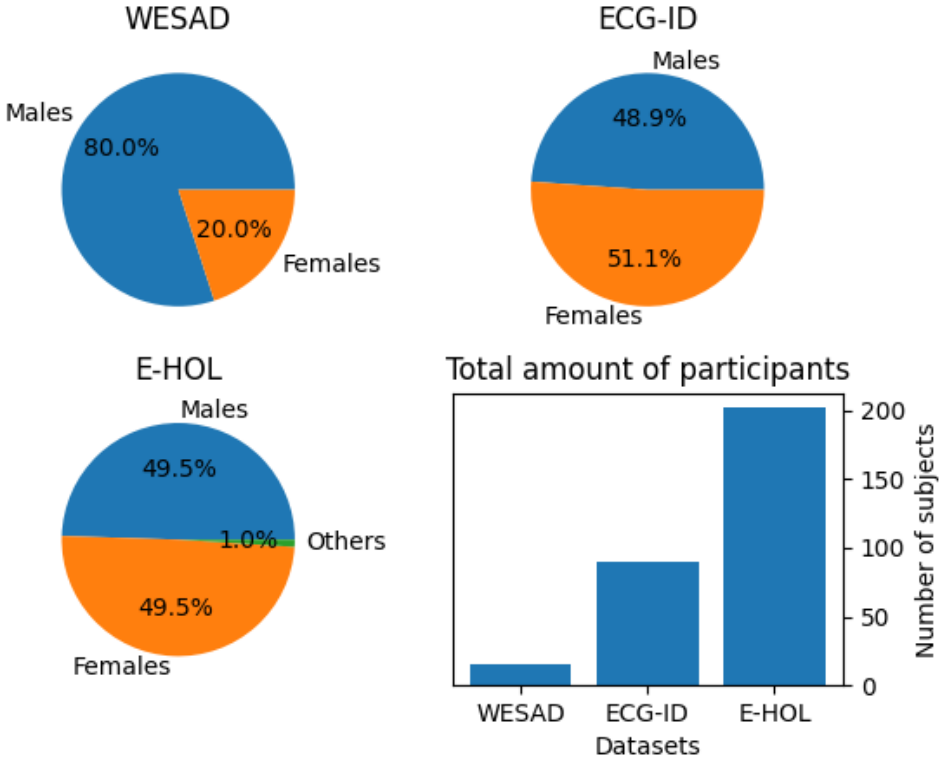


FIGURE 5.4: Demographic distributions over the three ECG public datasets.

In Figure 5.4 are summarised the demographic information of the three datasets. We chose these three datasets for the large amount of data provided from healthy subjects and the differences in recording devices, the wearable-based chest bands versus medical-grade Holter ECG recorders. Furthermore, the lack of existing studies about biometric verification performance assessments using the first two datasets and the many participants in the *E-HOL* database played a vital role in dataset selection. An aim of this work is to prove that the evaluated models

were consistent and not biased by device specification, providing reliable biometric verification with wearable devices.

5.2.2 Wearable devices

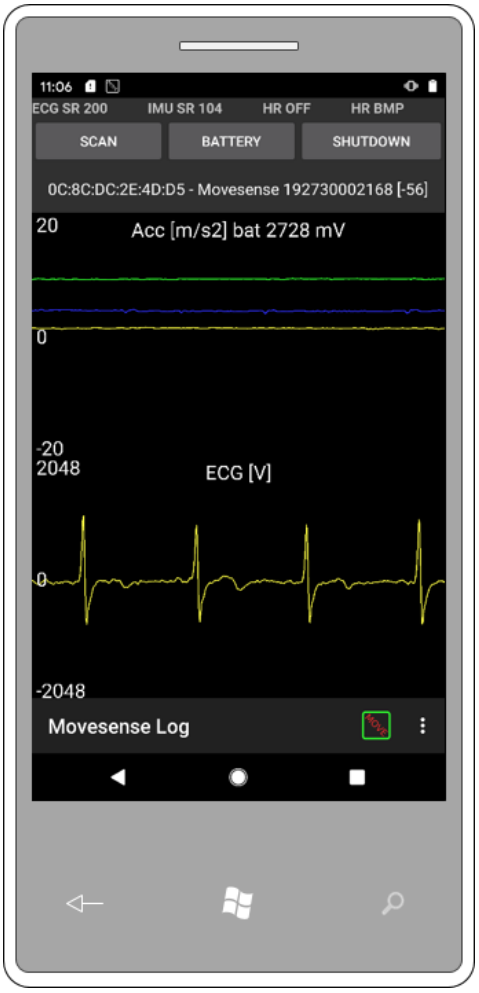
The public datasets were chosen to provide an understanding of the ECG signal captured with different devices under different conditions, but to apply biometric authentication in real life small wearable sensors need to be assessed. The selection criteria for such a sensor have to take into consideration the cost, the size, and the connectivity of the device, without sacrificing signal resolution. We shortlisted a number of devices that provide the required signal with built-in circuitry that performs initial signal denoising, signal amplification, and that can be connected to other devices with a Bluetooth connection.

In terms of smartwatches or smart bracelets, the best options were the *Apple Watch 4*, the *Amazfit Verge 2* and the *Shenzhen Smart Bracelet*, all of which provide ECG and photoplethysmography (PPG) signals. Digital resolution and sampling rate were comparable with the public datasets devices, with the *Apple Watch* having the highest sampling rate (600 Hz). The downside was that the signal was recorded on demand of the user, by placing the finger on the electrode positioned on the top or on the side of the smartwatch/smart bracelet to close the circuit with the other electrode beneath the device (recording in this way the first lead). The recording time was fixed and could not be shortened. Moreover, the *Apple Watch* raw data are difficult to access due to the design of the software. The cost of the devices was also taken into consideration.

For the aforementioned reasons, we decided to focus on another wearable device, the *Max ECG monitor* (see Figure 5.5a). Compared to the previous devices, which provided many other functions, the ECG monitor's only purpose was to record ECG signals alongside accelerometer data. Nevertheless, it was the best option to be implemented in the framework, due to the reasonable cost, the device specification (see Table 5.1), the small size and the connectivity with other devices

via Bluetooth. The recording could be performed in the background without the

(A) Max ECG monitor. Image belonging to Maxim Integrated$_{TM}$, all rights reserved.



(B) ECG Recording from the device.

FIGURE 5.5: Max ECG monitor (A) and an example of signal recording (B). The values on the ECG track are yet to be normalised based on the digital resolution. On the top-left of the screen it's shown the sampling rate.

need for an active input from the subject, which would perfectly fit in our study. The sampling rate for the signal could also be set, varying in a range from 200 Hz to 500 Hz.

Unfortunately, as mentioned before, we could not conduct a data collection, but this would have been the main choice. We presented a recording example in Figure 5.5b, the resolution and the structure were comparable to the data from the public datasets.

TABLE 5.1: Specification table for the MAX ECG monitor.

Max ECG	
Features	***Values***
Weight	9.4 g
Diameter	36.5 mm
Operating temperature	−20°C to +60°C
Battery type	CR2025 3V lithium cell
Transmission frequency	2.4 GHz
Sampling ranges	200-512 Hz
Digital resolution	18 bit
ECG sensor	MAX30003
Manufacturer	Maxim Integrated

5.3 Pre-processing and filtering

The pre-processing of the ECG signal is one of the most important parts of this study, due to the fact that it relies on all the previous assumptions and information about the cardiac cycle and aims to provide a new input structure for the classifiers that had both a high signal-to-noise ratio (SNR) and a small sample time window.

In the following subsections, all the procedures will be described. As previously done with Swipe and Pin data, a series of methodologies were applied to generate feature vectors depending on the classifiers used for evaluation.

5.3.1 Signal filtering

Signal filtering was the only pre-processing step common to both data structures feeding into different classifiers that we aimed to create, as the only purpose of the filtering was to remove the noise from the raw signal.

To provide appropriate filtering, it's imperative to have a full understanding of the full frequency range of the signal, the useful range, the signal magnitude range and the main noise sources.

With *useful range* we refer to the only part of the signal useful to the task, which can easily change depending on the desired goal. For example, in hospitals a monitoring ECG is used to check exclusively the correct activity of the heart in hospitalised patients, usually after a surgery; no accurate information is required, except for the heart rate and an approximate view of the various waves on a single channel. This means that the cutoff frequency for the low-pass filter can be set at a very low value (50 Hz or lower).

In the case of a diagnostic signal, every component of information in the signal is useful, including all twelve channels. It requires sharper filters with low ripple and distortions. In most extreme cases, noise from high frequencies overlapping with ECG natural frequencies should be removed with direct subtraction, recording the noise signal with electromyography. This however is not a case of interest for this study.

For our purposes, we applied a low-frequency noise removal filter and powerline band removal. To do so, all recordings were filtered with a high-pass Butterworth filter with cutoff frequency at 0.5 Hz and a notch filter centred at the powerline frequency (50 or 60 Hz depending on the power source). Applying the high-pass Butterworth we also removed the zero drift.

5.3.2 Windowing and feature extraction

After cleaning the data, we created two different data structures, one to train the classical machine learning algorithms and one for the deep learning architectures, since the two methodologies work with different data structures. This procedure was applied to 3 databases, resulting in a total of six new data structures.

To generate the first data structure, N samples per subject were trimmed from the whole recording. Each sample was a time window of size T and delay between windows s, resulting in a matrix of dimensions $[(N\_subjects \cdot N) \times (T + 1)]$. The window size considered for each sample is ten seconds, with a one-second shift. Each sample is further processed to extract a feature vector, resulting in a final structure of dimensions $[(N\_subjects \cdot N) \times (N\_features + 1)]$.

It's a set of 15 features, comprised of 9 fiducial features common in literature and 9 based on the interval between heartbeats. Such features are:

- QS distance.
- PQ distance.
- ST distance.
- P peak amplitude.
- Q peak amplitude.
- R peak amplitude.
- S peak amplitude.
- S peak amplitude.
- RR minimum distance over the sample.
- RR maximum distance over the sample.
- RR median distance over the sample.

- RR standard deviation.
- Number of RR intervals shorter than 50 points.
- Ratio between RR intervals shorter than 50 points and all RR intervals in the sample.

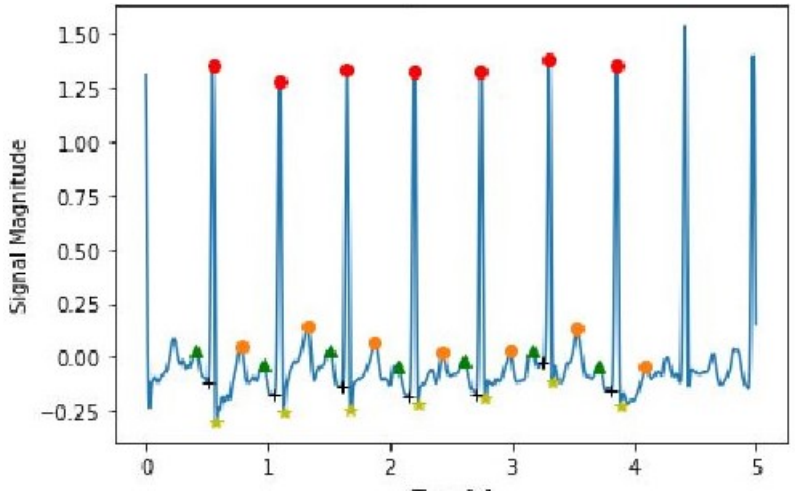


FIGURE 5.6: Peak detection on ECG recording. For each peak a different marker has been used.

With P, Q, R, S, T waves of the signal.

Distances between peaks (QS, PQ and ST) are calculated considering the position of the maximum value of the two peaks and averaged over the recording. *Peak amplitude* refers to the mean of the maximum values of the related peaks over the recording. To calculate each peak amplitude in each cycle, we took as reference the R peak position (detected with the Pan-Tompkins algorithm [138]). We evaluated the maximum (in the case of P and T) or minimum (in the case of Q and S) value in a confident interval of milliseconds before or after the R peak. An example of peak detection is provided in Figure 5.6.

The extracted features are further described in Table 5.2. In Figure 5.7 the paired distributions of some of these features are shown for each subject from the WESAD dataset. It can be seen that, especially for some pairs of features (e.g. Ramp and QS distance), samples from same subjects were clustered together. This representation helped to visually understand the validity of the chosen features and

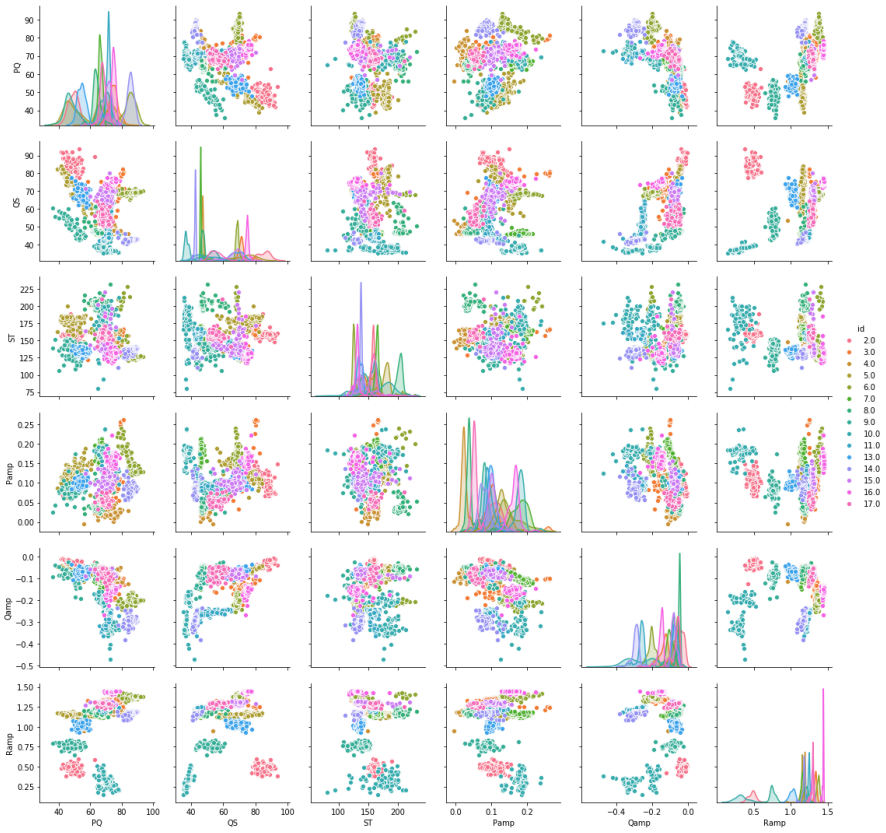


FIGURE 5.7: Pair distributions of features across WESAD subjects. Each color corresponds to a different subject.

to estimate how easily separable were the different subjects based on the current features. The diagonal shows the distributions of the subjects samples for the single features.

The second data structure, generated for the deep learning models, is a 2D matrix with each row representing a single heartbeat cycle. We stored the first 10000 cycles centred on the R peak as single samples for each subject in the datasets. Each sample consists of 150 points, equal to 0.75 seconds of recording. We take the R peak as a reference (from the peak, the previous 0.25 seconds and the following 0.5 seconds are selected).

For each dataset collected with a sampling rate higher than 200 Hz, after trimming with the 0.75 seconds window, all samples are downsampled to match the 151-point size, resulting in a matrix of dimensions $[(N\_subjects \cdot 10000) \times 151]$. The last column corresponds to the unique ID assigned to the subjects. To detect R peaks,

TABLE 5.2: Description of the extracted features. t is the time vector of the sample, N the number of heartbeats in a sample. P, Q , R S, T refer to wave peaks. Distance and amplitude features are commonly referred as fiducial features [83, 139]

Features	***Description***	
QS	$\frac{1}{N}\sum_{i=1}^{N} t_{S_i} - t_{Q_i}$	Distances between peaks (averaged over sample)
PQ	$\frac{1}{N}\sum_{i=1}^{N} t_{Q_i} - t_{P_i}$	
ST	$\frac{1}{N}\sum_{i=1}^{N} t_{T_i} - t_{S_i}$	
Pamp	$\mu\{P_i\}_{i=1}^{N}$	Peak amplitude (averaged over sample)
Qamp	$\mu\{Q_i\}_{i=1}^{N}$	
Ramp	$\mu\{R_i\}_{i=1}^{N}$	
Samp	$\mu\{S_i\}_{i=1}^{N}$	
Tamp	$\mu\{T_i\}_{i=1}^{N}$	
minRR	$\min\{t_{R_i} - t_{R_{i-1}}\}_{i=2}^{N}$	R-R distances statistics
maxRR	$\max\{t_{R_i} - t_{R_{i-1}}\}_{i=2}^{N}$	
medRR	$m\{t_{R_i} - t_{R_{i-1}}\}_{i=2}^{N}$	
meanRR	$\mu\{t_{R_i} - t_{R_{i-1}}\}_{i=2}^{N}$	
stdRR	$\sigma\{t_{R_i} - t_{R_{i-1}}\}_{i=2}^{N}$	
RR50p	$\dim(\{RR50p \mid RR50p = RR_i < 50\})$	
RR50pRatio	$\frac{RR50p}{\dim(RR)}$	

we used the *neurokit2* libraries with the implemented Pan-Tompkins algorithm [140].

5.4 Models and methodology

To assess the performance of the ECG on a biometric verification task, we explored classical feature-based machine learning models and three deep learning architectures. We did not compare all the existing models proposed in the previous studies (see table 2.3), but we restricted the selection based on the following exclusion criteria. Models too slow or too memory consuming were discarded, since the authentication task should ideally be happening on a mobile device within a

very short time. In our framework, this process would occur in the background during the swipe authentication.

Models requiring further processing or more complex data structures (i.e. images or spectrogram) as inputs were discarded for the same issues mentioned above.

Finally, we discarded KNN and DTW models, the former for the large numbers of samples required to generate the template and the latter for computational cost alongside unreliability of the performance.

In the following sections we will describe what models we used and the parameters selected.

5.4.1 Feature-based machine learning models

Considering the distributions of the extracted features (see Figure 5.7) and how well they clustered different classes for some specific pair distributions, we decided to evaluate the performance of three statistical classification models: *Linear Discriminant Analysis* (LDA) [141], *Naive Bayes* classifier (NB) [24] and *Decision Tree* (DT) [23].

All the models were implemented using *Python's Scikit Learn module* [142]. The models and their behaviours are described below.

5.4.1.1 Linear Discriminant Analysis

LDA works based on the Fisher's linear discriminant for normally distributed data. The main assumption is that all the features have a Gaussian distribution and each input variable has the same variance for its class. The computation of the prediction is performed on a linear projection of the features, considering the mean of the distribution and the covariance matrix for all the classes. For this reason, the size of the training set, the number of outliers and the normalisation of the data are possible sources of errors. Another downside of the LDA and a direct consequence of its simplicity is the likelihood to overfit on the training data.

5.4.1.2 Naive Bayes

The Naive Bayes classifier (GNB) works on the same assumptions of LDA, such as Gaussian distribution of the data and independence between features (this last strong statement is what gives the name "Naive" to this model). The implementation is simple and the computational cost is low, due to the model relying entirely on the conditional probability (and the corresponding probability density functions) of all the features respect to the class.

During the training process, the model calculates the *a-priori* probabilities for each unique class (in the case of a binary classifier, this probability is determined by the class imbalance or by the training criteria for data splitting) and the conditional probability of each feature with respect to each class. The decision rule for this classifier is to pick the *maximum likelihood a-posteriori*. Considering K as the number of classes, C_k the *k-th* class, F the number of features and x_f the *f-th* feature in the feature vector x, the decision y is given as:

$$y = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}} p(C_k) \prod_{f=1}^{F} p(x_f | C_k) \tag{5.1}$$

This model is very fast to train, but it is also very dependant on the training size and the randomisation of the imposter samples. The performance is directly correlated to how representative the training set is of the population and how are linearly separable the features.

5.4.1.3 Decision Tree

The *Decision Tree* (DT) is a non-parametric model (meaning that it only learns probabilities and conditions, and the only hyperparameter is the depth of the tree) that, given a feature vector, recursively partitions the features in branches (each one being an "if" statement) to group samples with the same labels and/or similar target values. For each branch, a new threshold is estimated based on the

input feature from the previous branch. The cost function tries to minimise the propagated error on each branch.

The advantages of the DT are that it's an easy-to-understand white-box model, with computational cost growth logarithmically proportional to the number of samples used for training and doesn't require normalisation of the data. The downsides are the sensibility to oscillations and outliers and the risk of overfitting, especially for datasets with a large class imbalance. Also, as for the previous models, branches parameters need to be retrained for each new subject, resulting in *N_subjects* number of models instead of a *N_subjects* number of templates, which is ideal for deep learning models.

5.4.2 Deep learning models

We evaluated the performance of three deep learning models. As anticipated before for *swipe* and *pin* gestures, the models should be small enough to be used on a mobile device and should not be excessively computationally expensive, or require further processing on the ECG data that would increase the complexity. For these reasons, we avoided using recurrent networks or convolutional networks with 2-dimensional kernels, typically used with image data.

The models we chose were a convolutional network based on the *DeepECG* model [143], a *Siamese network* and a *Variational autoencoder*.

5.4.2.1 DeepECG-like model

The first deep learning model is a unidimensional convolutional network based on a previous implementation by *Labati et al.*[143], but with a smaller structure and a different set of weights and parameters, in order to make it fit for a mobile device in terms of memory space and computation speed. The architecture consists of four successions of *convolutional and pooling* layers, a *dropout* layer, a *flatten* layer and a *fully connected* layer (see Figure 5.8).

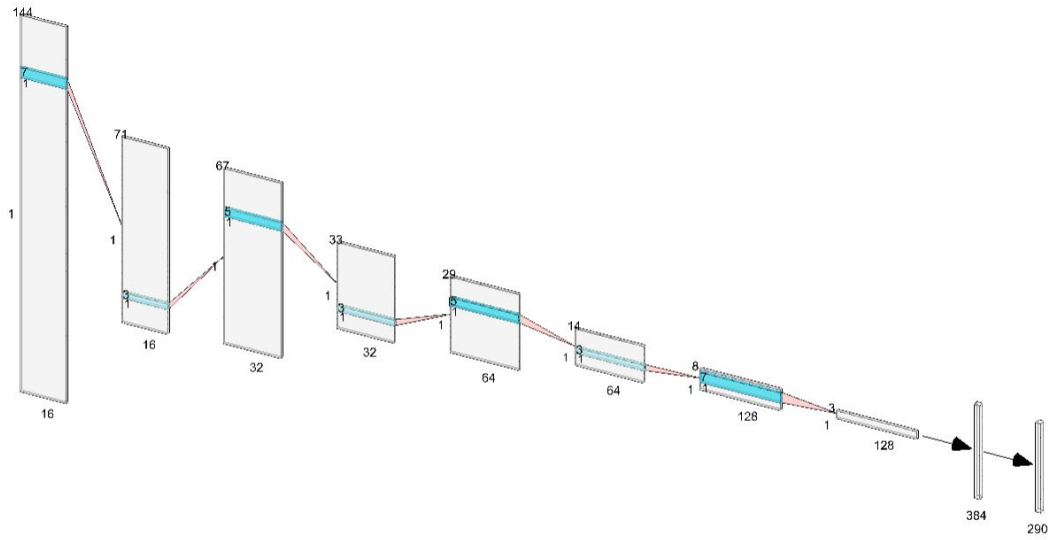


FIGURE 5.8: Visualization of the DeepECG-like deep learning model. Blue sections inside layers represents the filters. Numbers describe layers and filters dimensions (height, width and depth).

A *convolutional* layer is defined by its *filter* f with *height* H, *width* W and *depth* D (in the case of 1-dimensional convolution, *height* will be 1). *Depth* can also be visualised as the number of filters. Another parameter for the convolution is the *stride* s, defined as the shift between successive convolutional operations. Given an input vector x of length N, the output y for a single filter can be calculated as follows:

$$y_{(n)} = \sum_{i=0}^{W} x_{(n+i+s-1)} \cdot f_{(i)} \tag{5.2}$$

For all the N values in the input vector. The new length o of the output vector will be:

$$o = \frac{N-k}{s} + 1 \tag{5.3}$$

This is repeated for all the D filters of the layer.

During the training phase, we used a softmax layer and a cross-entropy loss function, as also used for an identification task. When the training was complete, we tested the model on unseen subjects, using the output of the fully connected layer as a feature extractor. We created the user pattern averaging the extracted

features from five samples after an L2 normalization, and later, we performed authentication calculating the Euclidean distances between the pattern and testing samples the same way we did for swipe and PIN in Chapters 3 and 4.

The model was trained as a feature extractor with data from 160 subjects of the *E-HOL* dataset. We did not repeat the training procedure for the other two public datasets. The main assumption was that once the model had learned to extract features from a dataset, it should be able to generalise on other datasets if the structure of the input data is the same.

We decided to perform the training on the E-HOL dataset for two main reasons: it is the largest dataset amongst the three and has the lowest sampling rate. This directly implies that to have the same input shape, data collected with different sampling rates need to be reshaped to match the required input (after applying the selected time window). This reshaping operation, depending on the sampling rates, can be seen as a smoothing/moving average filter (if the data were collected at a higher sampling rate) or an upsampling/interpolation (if the data were collected at a lower sampling rate).

The first case is always a better choice, because even if the smoothing filter implies some data loss, the output is still not distorted. Upsampling implies a function estimation and the direct introduction of artificial data that may not correspond to the real signal, resulting in an introduction of a bias.

5.4.2.2 Siamese network

As anticipated in the Swipe Chapter, a Siamese Network is a deep learning architecture with two towers sharing weights and a merging layer that combines the outputs. After the merging layer, there might be other layers performing additional operations, but the output will still be a *sigmoid* that will give the prediction. The merging layer is structured to calculate a distance metric between the encodings of the two tower structures. In Figure 5.9 an example of a generic siamese network is shown.

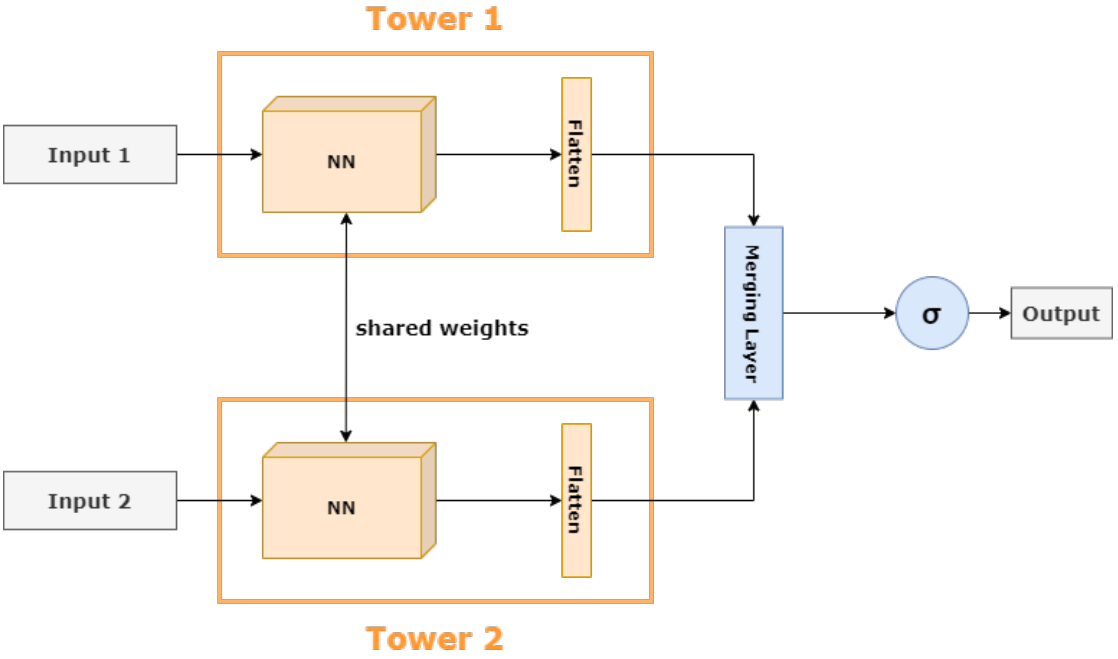


FIGURE 5.9: Generic Siamese network. In the merging layer, a distance metric function is executed to calculate distances between pairs.

A Siamese Network is used to predict if the two inputs correspond to the same class or to different classes, hence the output will be a value between 0 and 1, and the ground truth will be a binary value. The loss function, as described previously in equation 3.14, tries to maximise the distance between different classes and minimise the distance for data from the same class. The distance vector can also be the monodimensional prediction value, the important factor is that at the merging layer a distance was calculated.

Ibtehazx et al. provided an implementation of a Siamese Network for biometric identification in closed environments and template verification for ECG recordings, but their model was not fit for mobile devices, due to the high number of parameters and memory usage. We adopted a simpler solution, using the Siamese Network framework just to improve the discrimination between classes compared to our first deep learning model. We used the trained deepECG-like model as a backbone for the two towers, freezing the weights, and merged the two output layers, followed by a fully connected layer and a sigmoid output. The training step of the Siamese Network was finalised to fine tune the distance criteria and the clustering.

With these premises, we did not evaluate the Siamese Network on all the datasets, due to the complexity of generating pairs and the different protocol for testing (averaging the enrolment heartbeat data to obtaining a mean heart cycle to compare with the remaining data).

5.4.2.3 Variational autoencoder

The autoencoder is a hourglass network composed of an encoder and a decoder. The encoder part is an extractor that embeds the input data into a latent space Z of deep features. The decoder part expands the features from the latent space and tries to reconstruct the original input. Due to its architecture and the cost function relying on the reconstruction error between the input to the encoder and the output of the decoder, an autoencoder is considered an unsupervised model and doesn't require labels during the training procedure. A visual representation of a generic autoencoder is shown in Figure 5.10.

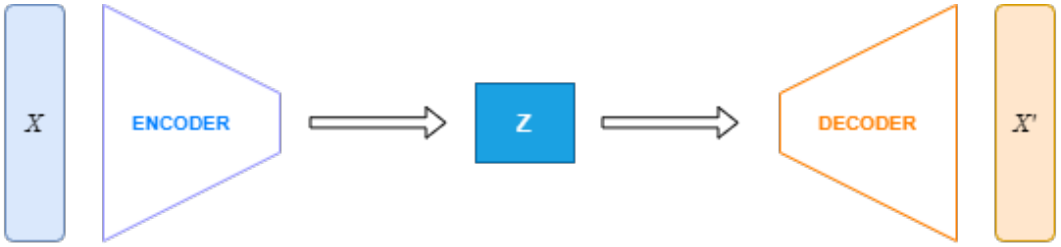


FIGURE 5.10: Generic structure of an autoencoder. X is the output, Z the latent space and X' the reconstructed output.

The purpose of an autoencoder is to reduce the dimensionality of an input vector, retaining the features that better describe the data, and to act as a generator for synthetic data; this is feasible after the model is trained, utilising only the decoder part and feeding a custom latent vector as an input. A good encoding would ideally generate a regularised set of features in the latent space, each one with specific information content. The main issue of autoencoders is that they are very dependant on the shape and the distribution of the training dataset and they can overfit very easily (especially if the decoder and encoder architectures are very complex and the latent space has few dimensions, resulting in too much information loss).

To improve the generalisation of the model and the regularisation of the latent space, variational autoencoders (VAE) have been introduced. Compared to a normal autoencoder, the output of the encoder is not a latent space but a latent normal distribution, sampled to generate a new Gaussian distributed representation. In this way, the new features are better regularised and distributed, providing values that are equally descriptive of the input data.

The new sampled distribution Z is defined as follows:

$$Z = h_{(x)}\aleph_{(0,I)} + g_{(x)} \tag{5.4}$$

with *h(x)* and *g(x)* functions of the encoding network describing the mean and variance of the distributions, and $\aleph_{(0,I)}$ representing the Gaussian distribution centred in 0 and with variance of 1. Using this reparametrisation, errors can be backpropagated ignoring the issue of the sampling from normal distribution.

To optimise the parameters of h and g, another loss is added to the reconstruction loss based on the Kullback-Leiber (KL) divergence [144] between estimated sampled distributions that project data into the latent space and the actual distribution. The KL loss is defined as follows:

$$KL\_Loss = 1 + log(\sigma_z^2) - \mu_z^2 - \sigma_z^2 \tag{5.5}$$

Considering that the latent space of the VAE provided mean and variance of the distributions, we decided to use two distance metrics to evaluate the biometric performance on verification of this model: the Euclidean distance and the Mahalanobis distance [145]. The latter seemed a better fit for the task, since it calculates the distance between a point (in our case, the subject template) and a distribution. For a distribution with diagonal covariance, which we assumed to be our case, the Mahalanobis distance is defined as follows:

$$D_M(x) = \sqrt{\sum_i \frac{(x_i - \mu_i)^2}{\sigma_i^2}} \tag{5.6}$$

5.4.3 Experimental protocol

We designed our experiments to evaluate not only the performance of the models, but also the improvements when varying the enrolment sets for the subjects. For classical ML classifiers, we used a 10 seconds time window for each sample during the preprocessing. We calculated the mean EER for each model. During training, we used 50, 150, 250 and 500 seconds of total time from the sample selected amongst genuine and imposter subjects; in the case of the DL algorithm, we also included a smaller training window of 5 seconds that classical models could not achieve.

Imposter samples for training were selected randomly across all the subjects in the dataset different from the enrolled subject. The biometric verification time corresponds to the single sample window (10 seconds for classical machine learning models and 2 seconds/1 heartbeat for deep learning models).

To train the DL algorithms, we used all the pre-processed samples from the first 160 subjects of the E-HOL dataset; the data were split into training and validation sets, to evaluate the ongoing performance during the training phase. Then, for verification, we used the first samples (according to the various aforementioned enrollment times) in chronological order from each unseen subject (40 remaining in the E-HOL dataset and all the others from WESAD and ECG-ID datasets) to create the user template.

We applied the same concept from equation 3.16 to create the templates with the DeepECG-like model and the VAE. Euclidean distance between the template and the embeddings was used to perform the verification (Mahalanobis distance in the case of the VAE). We used all the remaining samples from the enrolled subject and the rest of the data for this evaluation.

Results are expressed in terms of EER for each enrollment set, for each model and for each dataset. In the case of the ECG-ID dataset, we also analysed the inference of gender during the authentication, calculating similarity scores for random pairs

of genuine and impostor samples and comparing them with the scores of same/opposite gender pairs only. This was under the assumption that the heart structure and size change not only through growth but also differs based on biological sex.

In the following section, we present the results for all the experiments and we draw the conclusions.

5.5 Results and discussion

Results are shown for each dataset, comparing model performances on different enrollment sizes (in terms of recording time). As explained above, all the deep learning models extract features from a single heartbeat (2 seconds of recording) and perform the verification using a distance metric from the template. The statistical machine learning models, instead, perform direct binary classification on a 10 seconds window of recording. We provided results separately for the Siamese model, since it directly compares two samples and we could not apply the same protocol as used for the other models.

In Table 5.3 results are shown for the E-HOL dataset. It's the largest dataset in terms of subjects and recording time. This affected the machine learning model performance, that failed to generalise, with the exception of the Naive Bayes classifier; and even in this case, results could vary depending on the randomisation seed for training samples and more importantly depending on the impostor samples selected for training.

We could not provide results for the VAE for the remaining three training sizes due to internal computational memory issues, but we can predict the trend. As distance metric, we used the Euclidean distance for the DeepECG-like model and Mahalanobis distance for the VAE (which had a 10% increased error rate with Euclidean distance).

In Table 5.4 the results for the WESAD dataset can be seen. As described before, the WESAD dataset contains the least number of subjects and all the recordings

were unsupervised (which resulted occasionally in noisy data and/or anomalies). The reduced number of different classes and samples in all probability allowed the *decision tree* to reach the best performance, with the 500 seconds of enrollment: it's reasonable to assume that data from all the different subjects were randomly picked during the training procedure, becoming an identification problem more than a verification task (which, as can be seen, is easily solved with the extracted features). By this, it's meant that, during the training procedure, the model could most likely gain some information from *every* single subject in the dataset and generate the best boundary for each one of them at every interaction. Such boundaries would have the same validity if it was an identification problem, since data from all the classes contributed to the training, not just an estimation from few of them.

LDA and GNB classifiers remain consistent with the previous results. The deep learning models perform slightly worse, but this is probably due to the aforementioned anomalies during the recordings. It's important to remember that, unlike the shallow models pretrained on the specific datasets, the DL architectures were pre-trained only on a different dataset (E-HOL dataset, with a lower sampling rate) and they still maintain consistency, proving that these models are good at generalising. This is also the reason why the GNB model performs better on the WESAD dataset compared to the DL models; the training allows a better fit, at the cost of an ungeneralised model that needs re-training (and therefore a whole database of data) for every new subject.

Overall, the DL models provide consistent results over different datasets (without the necessity of retraining the weights) from data collected under different conditions, with different devices, at different sampling rates, and with different recording times.

An additional proof is provided in Table 5.5, with the results from the ECG-ID dataset. Due to the small amount of data and very short recordings, we could only evaluate the deep learning models and only for two enrollment sizes. Nevertheless, the average EER was comparable with the previous results.

Figures 5.11a and 5.11b show the clusters for the subjects from the E-HOL dataset and the WESAD dataset respectively, using T-SNE to reduce the dimensionality of the embeddings and project just the first 3 principal components. For both datasets, it's shown how well the samples recorded from the same subject are clustered together with few to none overlap with other subjects samples.

In Figure 5.12 are shown a small number of reconstructed samples from the E-HOL testing set using the VAE. Even if the performance as a biometric classifier was not comparable to the DeepECG-like model, it would be a valid option as a synthetic data generator to expand datasets.

As mentioned before, we evaluated the Siamese network model on one dataset only due to the further preprocessing and complexity of creating pairs, which also modified the testing protocol. We used the E-HOL dataset and the same criteria for training testing. We repeated the training phase multiple times, noting that, depending on the initial random value of the weights, it could either converge or become a random classifier. Nevertheless, the EER was never lower than 5% with the lowest value of **5.42%**.

We reached the conclusion that, despite the increased complexity, the Siamese could not provide a better separator for template and matching sample. In Figure 5.13 are shown pairs of samples and the predictions from the network.

Table 5.3: Results on E-HOL dataset. All results are expressed in term of Equal Error Rate [%].

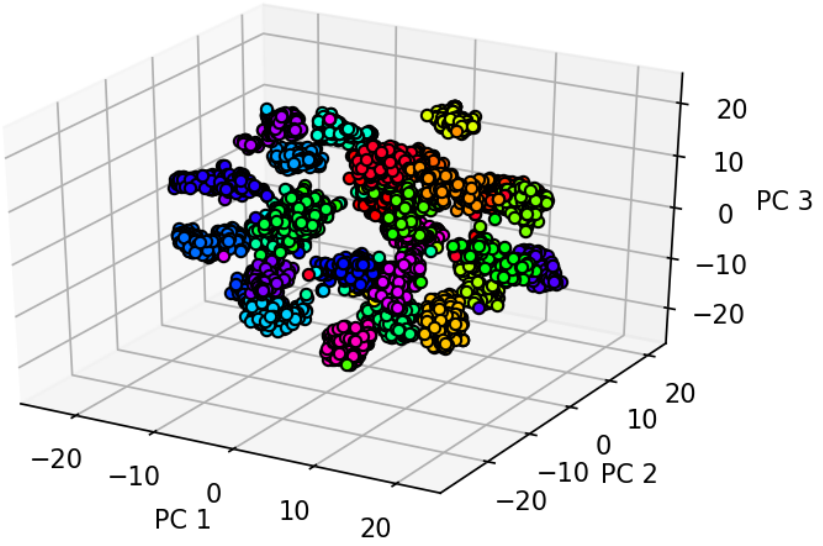
	E-HOL dataset				
	Enrollment time [seconds]				
	5 s	**50 s**	**150 s**	**250 s**	**500 s**
LDA	-	37.33%	25.65%	19.22%	15.94%
DT	-	27.91%	16.64%	10.99%	7.89%
NB	-	6.30%	4.1%	4.09%	3.64%
DeepECG-like	4.71%	4.64%	4.26%	4.15%	4.11%
VAE	14.21%	12.85%	11.05%	-	-

Table 5.4: Results on WESAD dataset for all the enrollment sets. All results are expressed in term of Equal Error Rate [%].

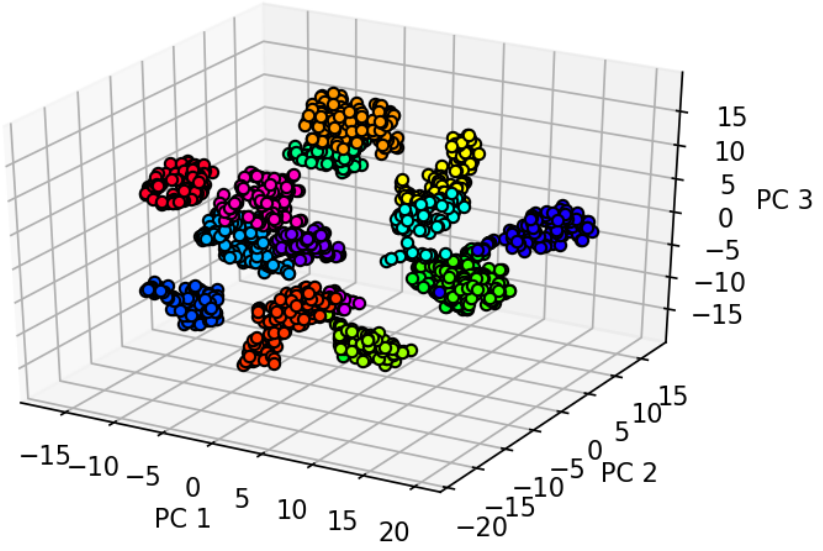
	WESAD dataset				
	Enrollment time [seconds]				
	5 s	**50 s**	**150 s**	**250 s**	**500 s**
LDA	-	32.03%	16.10%	11.68%	9.97%
DT	-	31.88%	9.61%	8.37%	1.60%
NB	-	4.57%	3.67%	3.67%	3.02%
DeepECG-like	7.07%	6.65%	6.59%	6.43%	7.17%
VAE	17.37%	15.17%	13.9%	14.29%	14.98%

Table 5.5: Results on ECG-ID dataset. Being the smallest dataset with very short recordings, only two very short enrollment sizes could be evaluated. Results are expresses in Equal Error Rate [%].

	ECG-ID dataset	
	Enrollment time [seconds]	
	5 s	**7 s**
DeepECG-like	5.7%	5.74%
VAE	14.51%	15.44%



(A) E-HOl dataset.



(B) WESAD dataset.

FIGURE 5.11: 3D projections of embeddings from recorded data after applying T-SNE. 30 subjects from E-Hol dataset (A) and 15 from WESAD dataset (B).Different colors represent different subjects.

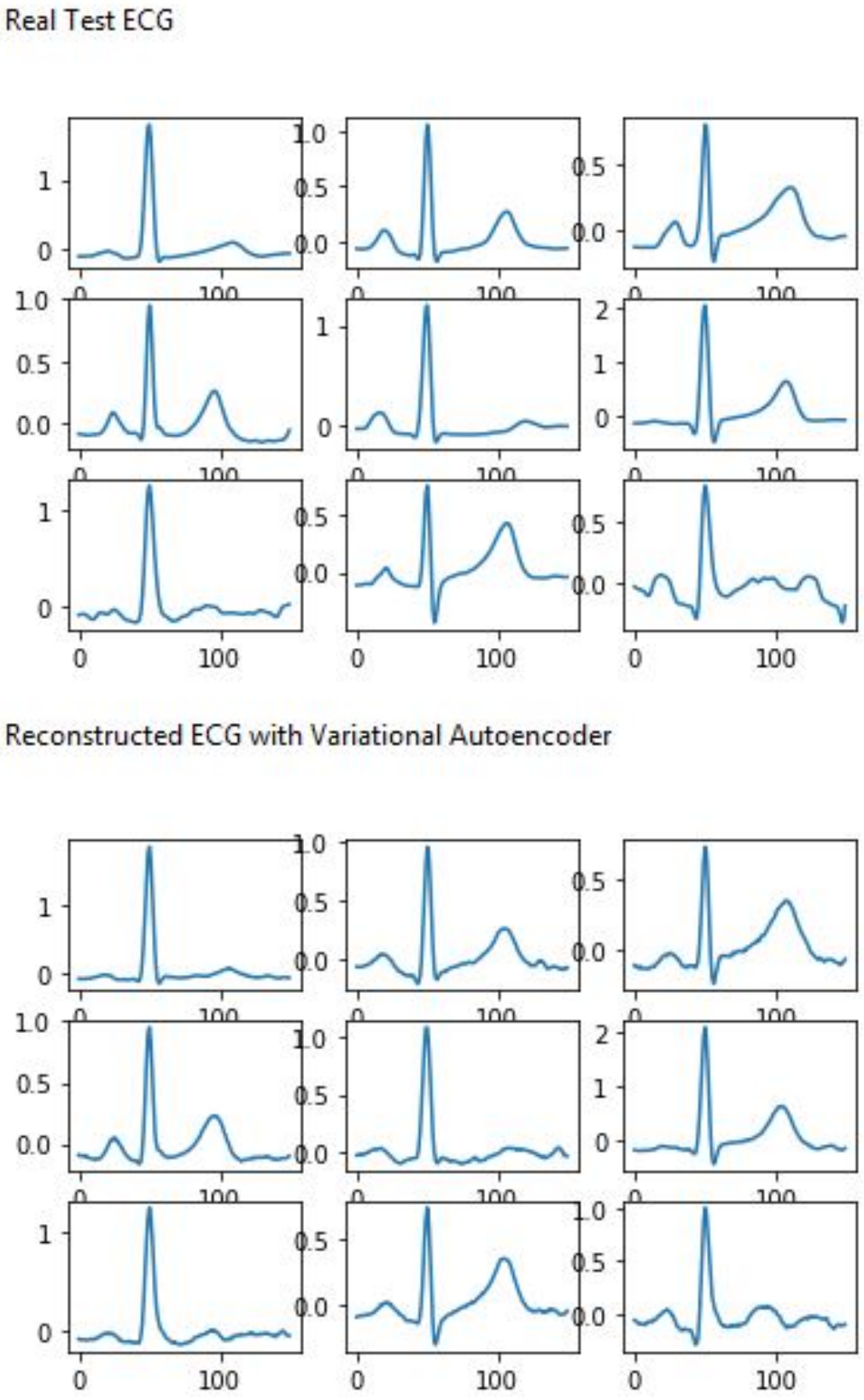


FIGURE 5.12: Original and reconstructed samples using the Variational autoencoder.

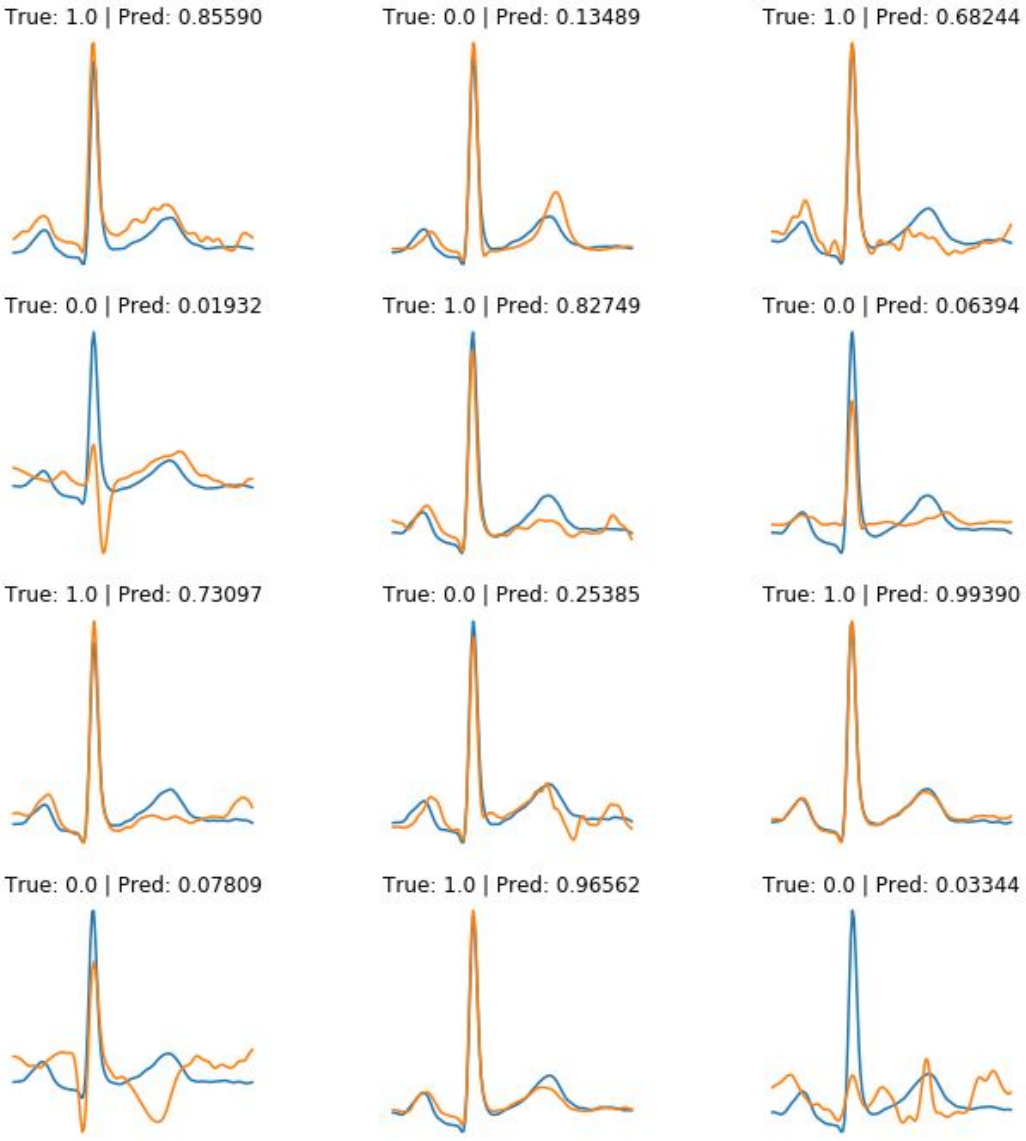


FIGURE 5.13: Predictions for paired data. Blue line corresponds to the reference sample, orange to the matching sample. True: 1.0 means that the two samples come from the same subject. Viceversa for True: 0.0 .

5.6 Discussion

As the last modality of our framework, we explored the electrocardiogram signal (from a physiological perspective and a biometric perspective), we identified an optimal wearable device for recording the data which can be used in a real life scenario, and we assessed a few verification models on a multitude of public datasets.

The best performing model, which also maintained stability in performance over the three datasets (which means that it can generalise despite the device used or the recording condition), is the DeepECG-like architecture.

We evaluate the performances over different enrolling times, maintaining the verification on a single sample (corresponding to a single heartbeat and approximately 2 3 seconds of real time recording), finding that providing additional information for the user templates provides improvement for classical machine learning algorithms only, while deep learning algorithms have very small variations in performance.

The effects of age differences on model's performance were not explored, since only one dataset (ECG-ID) provided the demographic information and it was not enough to draw conclusions. Nevertheless it's reasonable to hypothesize that age difference could affect the performance, due to the fact that the aging process directly affects the heart's physiology and activity.

According to our results, we can conclude that ECG biometric is a very promising modality that can be implemented even for short-time single-sample verification tasks, without the subject even noticing it (the ECG monitor can connect with the mobile device through Bluetooth and the recording process can be prompted in the background).

Unfortunately due to COVID regulation we could not perform a data collection and therefore it was impossible to generate paired data with Swipe and PIN to

furtherly assess the multimodal fusion across all the three modalities, but we still suggest a possible methodology on this matter in Chapter 6.

Chapter 6

Conclusion and future work

6.1 Summary

Biometric verification procedures on mobile and wearable devices is a topic studied by a wide variety of perspectives. The collection time, verification time, and performance of the models are some of the key factors that ensure acceptance by the user population. Few studies evaluated the authentication performance of behavioural biometrics based on the number of samples required by the model for enrollment and verification, but the vast majority could not provide a single sample solution or were focused on continuous authentication.

This research explored the possibility of light computational expense models, enrolled with few samples to perform a ceremony based verification task, in a realistic framework that combines two behavioural modality and ECG. The study assessed the performance of different models, proving not only the feasibility of the single-sample verification, but also exploring how the quality of a subject or a sample affects the outcome (in the case of Swipe verification).

In the following sections, we review our results and findings in relation to the research question. We also provide a further look at the open challenges and follow-up studies that this research brought up. We additionally give an insight of

how the three factors fusion and the data collection would have been conducted, if we were not impaired by covid restrictions.

6.2 Research findings

With this study, we addressed real life issues related to mobile biometric authentication with respect to behavioural data. The research was not just focused on performance evaluation, but also on authentication model stability, recording time, and feasibility of the proposed framework for daily usage. We provide a discussion for each research question (from Chapter 2) concerning the aforementioned challenges.

- ***Is it possible to obtain a steady performance on mobile biometric authentication with behavioural (PIN and Swipe) and electrocardiogram data?***

  To answer this question, it's necessary to remark the whole experimental protocol for each modality with particular attention on the enrollment data, the datasets used and the tasks involved. It's not possible to assess stability in performance based on the authentication error alone. In the case of Swipe authentication, our proposed deep learning models performed consistently on unseen subject, but the overall performance was worse when using public dataset data. This was due to the fact that Callsign data were semi-constrained to a very specific task and the subject were more incentivated to perform it at the best of their ability, unlike the publicly collected dataset. This is not a critique to data collection procedures, but an observation on how a system stability and generalisation power is strongly affected by task constraints and subject consistency, especially considering behavioural data. Other factors hardware-related may have constituted a bias (e. g. latency in capture from the touch sensor), but we contained them during preprocessing of the data, removing outliers.

A further proof of biometric performance stability being highly affected by a user's own behaviours towards the verification task is given by the follow up study on Swipe quality. Our experiments show that not only quality can be estimated for behavioural data on both user and sample perspective, but it's a good indicator for performance. High quality subjects have smaller authentication error and fluctuations, as per high quality samples are more easily predicted correctly by the verification system. This directly implies that authentication performance on behavioural biometrics can be predicted, consistent, and feasible for real life applications.

In the case of PIN authentication, we can see a parallelism with Swipe dynamics, even thought we could provide less comparative results. More interesting conclusions can be drawn regards ECG authentication. As described in Chapter 5, an ECG signal is not behavioural (but emotions or intensity of activity can still alter the signal), implying that data are inherently more consistent for the same subject (if we exclude subjects with heart conditions). We run our experiments evaluating different enrollment sizes and testing our proposed verification model on multiple datasets to assess its consistency over different devices and recording conditions, obtaining very small variation in performance. Additionally, overall performance in terms of EER for ECG biometrics was better compared to Swipe and PIN modalities, confirming its stability and feasibility.

For the assessment of all the modalities, the enrollment samples were chosen amongst the data recorded before the samples used for verification. This ensured that our results were not biased by providing data "from the future". We can conclude that the evaluated modalities with our assessed models provide a steady verification performance.

- ***Is it possible to perform authentication on a ceremony based task with just one sample request?***

Directly connected to the previous research question, we investigated the possibility of authenticating a subject providing one single sample (either a

single Swipe gesture, a single 4-digits PIN entry, or a single recorded heartbeat). The evaluation was conducted following the same premises mentioned in the first research question: outliers removed, unseen subjects in the test set, enrollment data for the subject template temporally older than the testing samples.

The single sample verification task was a very challenging task, but it reflects real world scenarios and expectations, in which when using an online service that requires login verification, the user needs to be authenticated in the least possible time. Especially in the case of PIN biometrics, it would not be realistic to request a PIN multiple times (if not entered it correctly), so the background biometric processing and verification needs to use the few available data provided by that single entry.

With our evaluation we found that our model can generalise and provide valid templates and authentication with few enrollment samples and just one sample for testing. The performance of the models remain stable across the subjects, but are very affected by the required task and behavioural instability of the subjects. Nevertheless, considering the complexity of the challenge and the , we can conclude that 13.72% EER for PIN authentication and 12.9% EER for Swipe authentication are acceptable values.

ECG data, as stated before, are not directly related to user behaviours and in general a single heartbeat contains more information, in terms of recorded data and extractable features, compared to a Swipe gestures or a PIN entry. Due to these reasons, it was expected that ECG biometric verification would perform better than the behavioural biometric modalities. We found not only great values in performance with one single heartbeat, but also that these values can achieved with just 5 seconds of recorded data for enrollment and that the model is able to generalise over three different datasets, making it optimal for mobile authentication in a real life scenario.

It's reasonable to believe that adding more complexity to the models would improve the performances, but the trade-off would be more computational expensive (considering computational cost as the number of parameters in

the model and therefore the memory usage) and result in a prolonged training phase with uncertainty on convergence. Another improvement can be attained by combining the modalities, but this will be discussed in the following research question.

In conclusion, even if not providing the highest accuracy and the lowest error compared to other modalities (e.g. face recognition), the results suggest that single sample authentication with behavioural biometrics and ECG biometrics are feasible and can be implemented with the current mobile devices available on the market, with expectations of further improvement in performance with the growth of new technologies.

- ***Can the performance be improved by combining the aforementioned modalities?*** Our study addressed this issue developing a fusion model designed for this specific framework, considering the optimal Level where to perform the fusion (given the structure of the input data and the contextual information on the sequential pipeline of the framework). We used only paired samples from the two modalities without introducing synthetic data, in order to make our results more consistent and unbiased.

  We assessed our fusion model and compared its performance with the results from the non-fused models and with fusion through another classical model. Both fusion systems decreased the prediction error, but our proposed model performed consistently better than the classical one, due to the contextual information inherited in the algorithm formulation. In addition, our model has only one trainable parameter, it's easily implementable either on a local device or on a server, and does not require the raw input data from the modalities sensors, just the output scores from each modality's model (which decreases the risk of data leakage).

  Results suggests not only that multi-modal biometric fusion can greatly improve authentication performance (EER reduced from 15.32% to 7.84%), but also that it can be performed at a low computing cost and that considering the structure of the framework can lead to further improvements.

Unfortunately we could only assess two factors fusion in this study with Swipe and PIN data, due to the impossibility to conduct a data collection and record paired samples for the three modalities. Nevertheless, considering the presented results, it's reasonable to expect a further improvement in performance with a multimodal system that combines ECG, Swipe and PIN biometrics.

Considering all the research findings, we can conclude that this study introduces a novel framework for mobile biometric verification on ceremony based task, which proves how authentication can be performed rapidly in a real life scenario with minimal number of samples requested for enrollment and verification. We presented a valid wearable solution that allows to include ECG biometric as a background task on query, minimising the amount of data recorded and stored. We proposed a novel fusion method which outperforms classical models.

In the last section, remaining open challenges are discussed, alongside a theoretical solution for three factors fusion.

6.3 Open challenges

The work conducted in this study addressed the research questions, but also put the basis for further evaluation and challenges that could not be addressed for many reasons (e.g. time, COVID-19 restrictions and availability of participants and devices). The first one is the fusion between all the considered modalities, that unfortunately could not be done without a data collection and with paired data from the various sensors. We will still provide a theoretical framework and propose fusion methods based on our previous findings and forecasting an optimal methodology considering the setup. The remaining two challenges are related to the performance over long periods of time or sudden changes.

6.3.1 Three factors authentication

Considering the initial setup, described in Chapter 2, there are various possible steps in which the fusion between the three modalities (Swipe, PIN and ECG) might happen, due to the recording of the ECG signal happening in background. The options, considering just the different authentication phases, can be the followings:

1. *Swipe-ECG fusion followed by PIN*: Only Swipe and ECG modalities are blend together. If the subject is not authenticated, PIN verification is prompted (see Figure 6.1).

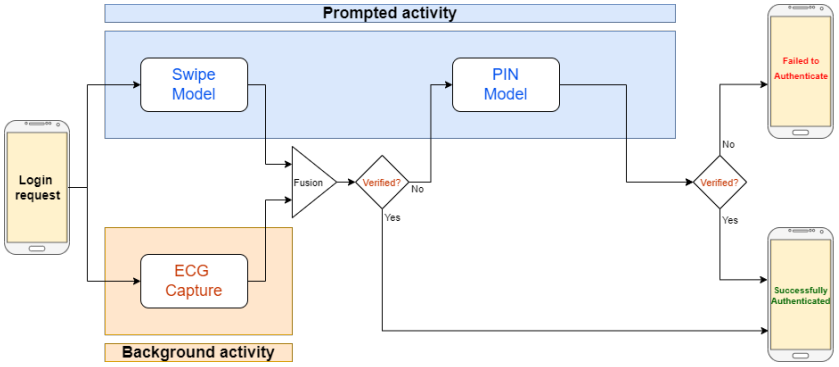


FIGURE 6.1: Pipeline for the first multimodal option.

2. *Swipe followed by PIN-ECG fusion*: PIN and ECG fusion is prompted only after a Swipe failure. This option reduces the amount of data collected from the same subject dramatically, since the ECG data would not be recorded until the first authentication task is failed (see Figure 6.2).

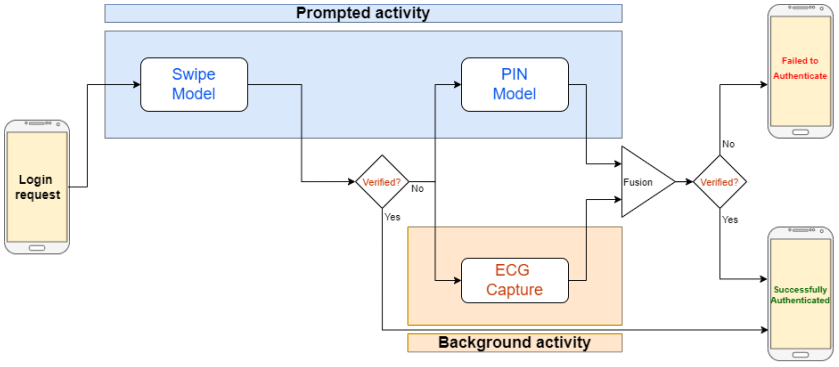


FIGURE 6.2: Pipeline for the second multimodal option.

3. *Swipe followed by Swipe-PIN-ECG fusion*: as for option 2), ECG and PIN data are recorded only after a Swipe failure, but in this case the fusion method combines all the three modalities (see Figure 6.3).

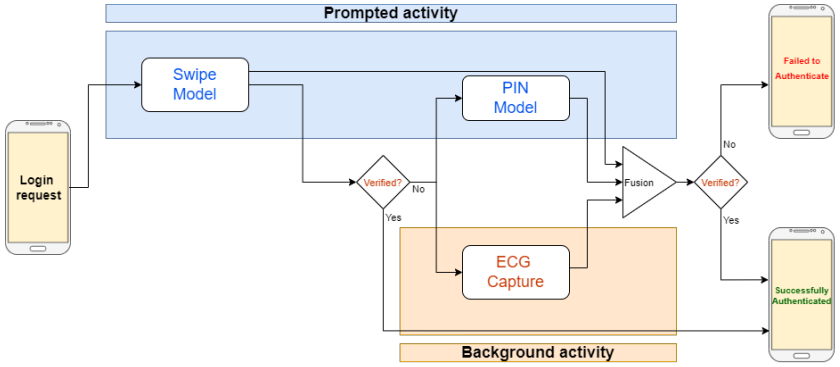


FIGURE 6.3: Pipeline for the third multimodal option.

4. *Swipe-ECG fusion followed by PIN-ECG fusion*: two fusion algorithms with two different ECG recordings (see Figure 6.4). The second multimodal verification is prompted after a failure from the first.

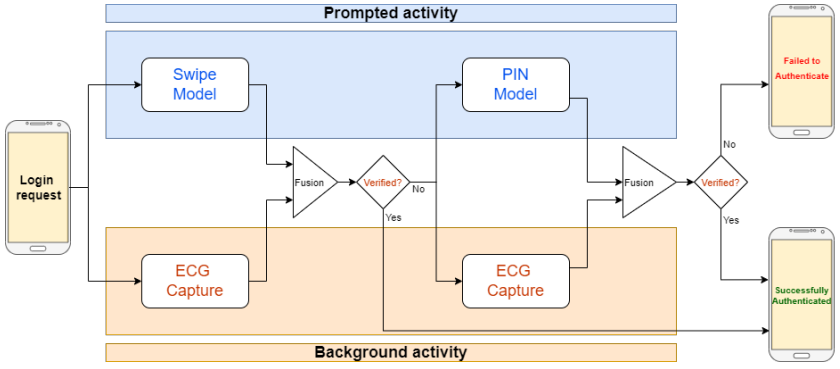


FIGURE 6.4: Pipeline for the fourth multimodal option.

5. *Swipe-ECG fusion followed by Swipe-PIN-ECG fusion*: Fusion is applied at every stage with all the possible available data. This rigorously follows the framework in Figure 2.8 from Chapter 2.

For each option the fusion algorithm can be implemented in many way and at different Levels, but there are recurrent steps as can be seen in the figures. We will propose few theoretical fusion methods for two (PIN-ECG or Swipe ECG) and three (PIN-Swipe-ECG) modalities blend together.

6.3.1.1 Two factors fusion with ECG

If we consider the case of ECG and PIN or Swipe fusion, with the sensors and models evaluated in the previous Chapters and based on the former assumptions on multimodal systems, it would be unlikely to have the fusion at feature or decision level.

In the first case, there are several issues: data structures are not the same (not just in terms of sampling but also in terms of dimensionality) and this would imply some further pre-processing or vectorisation with the risk of losing information. An LSTM network would not be possible to implement, unless considering the whole ECG track as a single feature, but this would imply large padding for Swipe or PIN data, or downsampling for the ECG signal. It could be argued that these premises might not necessarily negatively affect the performance, nevertheless the input data alongside the new model would be more computational expensive and a new training would be needed from scratch to learn all the new weights.

A fusion at decision Level would not be recommended either, since majority vote could result in a tie and "weighting" more the decision of one model in respect to the other model would make the latter superfluous.

A good choice would be to implement the fusion at model level, combining the two original neural network architectures similarly to a Siamese network, but freezing the weights before the merging layer and maintaining an embedding output with triplet loss as cost function. The frozen weights would not be furtherly updated during the training process of the fusion model, as a common procedure during transfer learning. In this configuration, both modalities would be given equal consideration, it would be the training process itself to adjust importance. In Figure 6.5 the general idea is visualised.

The merging operation could be more complex and involve more than one layer, as well as the following part of the architecture may have variable complexity. The training process would be smoother, considering the frozen weights trained

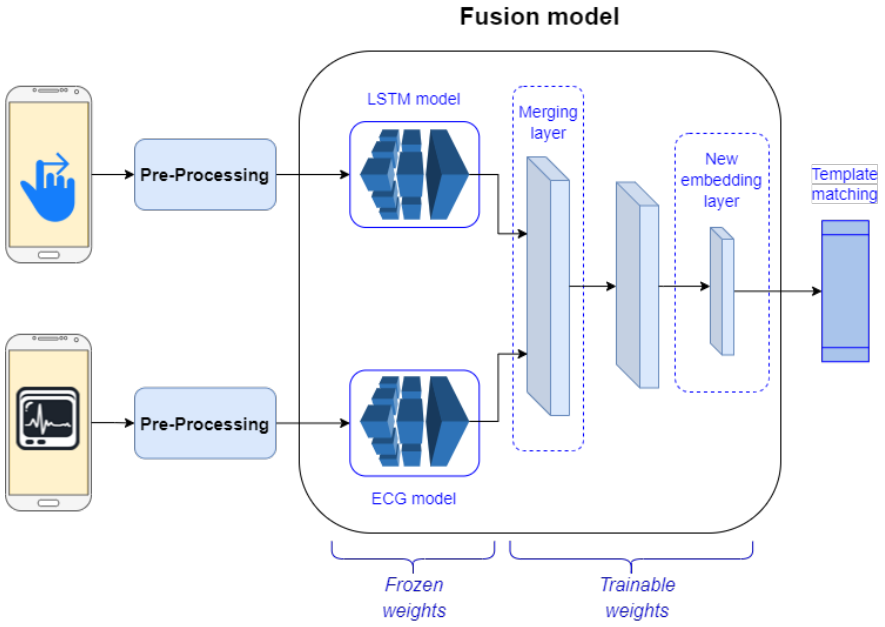


FIGURE 6.5: Two modalities fusion at model Level. Original models with frozen weights are combined with a merging layer. The remaining layers of the new architecture are the only one with trainable weights. The training is conducted with batches of paired data from the same class.

already. This fusion can be implemented for options 1, 2, and 5; in the last case, penalty fusion can be applied to the output scores of the two fusion models.

For ECG fusion, fusion at score level could not be performed with the penalty fusion algorithm, since for both ECG-Swipe and ECG-PIN the main hypothesis of cascade prompting would be missing. A combination of the two scores (linear or non-linear) would still be possible, but with different criteria for the weights. An easy way would be weighting each score based on the overall performance of the corresponding model (the better performing would have a higher scaling parameter for its score in the fusion algorithm). An example is given in Equation 6.1:

$$fusion\_score = \frac{\alpha \cdot s_1 + (1-\alpha) \cdot s_2}{M} \qquad 0 < \alpha < 1 \tag{6.1}$$

With α as a parameter based on the performance of the first model respect to the second model, s_1 as the score from first model, s_2 as the score from the second model, and M as the maximum possible value between s_1 and s_2.

A similar implementation could also take into consideration the quality score of the subject, instead of the relative performances of the model. Assuming $f_{(d_1,d_2)}$ being the fusion algorithm for the two modality scores expressed as distances from the template, d_1 can be corrected based on the quality score Q_i of the i-th subject and on the entity of the error as follows:

$$d_1^{corrected} = d_1 \cdot e^{(Q_i - k)\cdot\sigma_{(8d_1-4)}} \qquad 0 < d_1 < 1 \tag{6.2}$$

With k as trainable parameter and σ as the sigmoid function. Equation 6.2 implies that for low dissimilarity scores (hence when the subject is most likely verified) the correction is close to 0, while for high uncertainty (e.g. when the difference between sample and template is close to 1) the user quality applies a penalty for high quality users (that are expected to be easily authenticated) and is more permissive for low quality (usually inconsistent with themselves).

6.3.1.2 Three factors fusion

Fusing the three modalities together, as suggested by the Framework and seen in Chapter 4, is only possible after a Swipe verification failure; this reduces the number of possible fusion methods that can be applied. In this context, even if there is an odd number of modalities, majority vote would become just a simpler condition, defined as "*if PIN and ECG authentication are successful, then the subject is authenticated*". Feature Level fusion is discouraged for the aforementioned reasons and Model Level might not be possible, if the recorded data from Swipe are not saved for further authentication.

An option would be to combine the embeddings generated separately by Swipe, PIN and ECG models and create a new template. Similarly, the three similarity scores can be fused together. For both solutions there are few criteria that could be followed. Considering f_S, f_P, f_E as the functions that transform the input data from Swipe, PIN, and ECG respectively into embedding or similarity scores.

These functions represent the three pre-trained models. Assuming $h(x)$ and $g(x)$ as parametric functions of class C^1, we can define possible fusion algorithms as follows:

$$\begin{aligned} fusion\_score &= g_{(f_S, f_P, f_E)} \\ fusion\_score &= f_P \cdot g_{(f_S, f_E)} \\ fusion\_score &= h_{(f_P, g_{(f_S, f_E)})} \end{aligned} \tag{6.3}$$

Depending on the chosen functions, the fusion algorithms can be a linear combination or strongly non-linear distributions. Being both g and h of class C^1 ensures the differentiability of the fusion method, but not the convexity. This could affect the training procedure, so the functions must be chosen wisely.

Equation 6.3 does not include other possible combinations, in order to avoid redundancy; it's also taken into consideration the cascade structure of the framework. An example could be the following:

$$\begin{aligned} fusion_{(f_S, f_P, f_E)} &= f_P \cdot e^{g_{(f_S, \theta_S, f_E, \theta_E)}} \\ g_{(f_S, \theta_S, f_E, \theta_E)} &= \theta_S \cdot f_S + \theta_E \cdot f_E \end{aligned} \tag{6.4}$$

6.3.2 Stability over time

Another open challenge for behavioural data is the stability of the system over long periods of time, which directly reflect on the consistency of the user and their change in behaviours. This issue can be directly addressed through repeated user and sample quality estimations, frequency of changes and sensibility of the authentication model on such changes. For an extensive study, it would be optimal to evaluate quality-over-time considering different windows of time between authentication attempts (weeks, months, and years).

In this study we assessed the performance of our models taking in consideration their stability at the best of our possibilities, testing on samples recorded temporally after the enrollment samples or, in case of ECG, also using datasets with samples collected over several months of time. Nevertheless, to explore this issue, it's necessary to conduct a data collection over prolonged periods of times, with

recordings from the same subjects years apart. As mentioned in Chapter 2 with the related studies and in the descriptions of commonly used behavioural datasets (Frank [40], Serwadda [41], Antal [45], and ECG-ID [137]) for biometric authentication, this issue has been identified before but never fully addressed. The existing database for behavioural and ECG data have recordings collected at most with few months apart, with a small number of samples and subjects (compared, for example, with few face databases that have a larger range of collection time).

In conclusion, it would be advised for further studies to take into consideration a prolonged data collection, in accordance with current COVID and GDPR regulations.

6.3.3 Inter-device consistency

This issue is mostly related to behavioural biometrics, for two main reasons: *a)* as behavioural data are affected by subject behaviours and interactions, differences in devices may have effect of such interactions, which may result in a drop of authentication performance. *b)* Our results have shown that ECG biometrics are consistent over different devices, which can be seen by the similar performance obtained over different datasets (each one using different devices and sampling rate for signal capture).

In a real life scenario, with the growth of mobile technologies and the possibility to synchronice multiple devices, it's reasonable to expect a single user to have multiple devices with different sizes and sensors. The authentication model should be able to successfully verify the user regardless of these changes. Further studies could explore those possibilities, assessing the correlation between model performance and device, and trying to solve the issue with further data pre-processing or model weights scaling based on device used.

6.4 Final considerations

In conclusion, the work conducted in this thesis achieved many goals. The major contributions are listed below:

- Novel deep learning models for Pin, Swipe and ECG biometric authentication. Each model performs the verification using only a single sample from the user (and in case of ECG, a single recording of 3 seconds).
- The introduction and evaluation of a novel quality metric for Swipe data, considering both cases of user quality and samples quality.
- A new pre-processing algorithm for data removal, the shifted zero crossing (S0X, see Equation 3.3). it's purpose is to detect swipe data that do not comply with the required task (horizontal swipe) on a threshold basis.
- A multi-modal score-Level fusion algorithm (see equation 4.1), with only one trainable parameter, that considers the contextual information of the modalities and bases the decision on the magnitude of the error from the first modality.
- A theoretical framework for multi-modal mobile biometric authentication, comprised of three different modalities and fusion algorithms at different stages.

As technology progresses, it's reasonable to expect further improvements in mobile authentication systems. This thesis provided promising results on biometric user authentication from a multitude of modalities, addressing the challenge of the single sample authentication.

Nevertheless, it was not expected, nor it was the initial intention, to reach an end-point with this study. Instead, this thesis should be a launching pad for future mobile authentication systems and future studies in the field of behavioural biometrics.

Appendix A

Callsign agreement

8. MATERIALS

8.1. During the term of this Agreement it may be necessary for the Company to provide the University with various proprietary Materials for which the following terms will apply:

8.2. Materials will be provided solely for use in the Project, in the University's laboratories only. The University undertakes that any Materials provided will be used only by the Academic Supervisor, the Student and such persons under the direct supervision of the Academic Supervisor as are required to perform the Project. The Materials will not be provided to any other scientist or Institution (public or private) without prior written permission from the Company.

8.3. Materials are experimental in nature and will be provided without warranties of any kind expressed or implied. The Company, its employees and its Affiliates accept no liability for damages which might arise in connection with their use, storage or disposal by the University. Furthermore, the Company makes no representation that the use of the Materials provided by it will not infringe any patent, copyright, trademark or other proprietary rights.

8.4. On termination of the Agreement the University will discontinue use of the Materials and at the direction of the Company any remaining Materials will be returned to the Company or destroyed, and destruction certified by the University.

8.5. All experimental work within the Project and any destruction of Materials pursuant to Clause 8.3 above will be carried out in accordance with all applicable local, national and international legislation relating to the safe handling, use and disposal of potentially hazardous materials.

FIGURE A.1: Agreement on use of Materials (including production data) from Callsign.

Bibliography

[1] Marco Santopietro, Ruben Vera-Rodriguez, Richard Guest, Aythami Morales, and Alejandro Acien. Assessing the quality of swipe interactions for mobile biometric systems. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–8, 2020. doi: 10.1109/IJCB48548.2020.9304858.

[2] Hazal Su Bıçakcı, Marco Santopietro, Matthew Boakes, and Richard Guest. Evaluation of electrocardiogram biometric verification models based on short enrollment time on medical and wearable recorders. In *2021 International Carnahan Conference on Security Technology (ICCST)*, pages 1–6, 2021. doi: 10.1109/ICCST49569.2021.9717372.

[3] Unipolar and bipolar connection. https://www.wikilectures.eu/w/Unipolar_and_bipolar_connection, .

[4] Sougata Sen and Kartik Muralidharan. Putting 'pressure'on mobile authentication. In *2014 seventh International Conference on mobile computing and ubiquitous networking (ICMU)*, pages 56–61. IEEE, 2014.

[5] Cheng-Jung Tasia, Ting-Yi Chang, Pei-Cheng Cheng, and Jyun-Hao Lin. Two novel biometric features in keystroke dynamics authentication systems for touch screen devices. *Security and Communication Networks*, 7(4):750–758, 2014.

[6] Emanuele Maiorana, Himanka Kalita, and Patrizio Campisi. Deepkey: Keystroke dynamics and cnn for biometric recognition on mobile devices. In *2019 8th European Workshop on Visual Information Processing (EUVIP)*, pages 181–186, 2019. doi: 10.1109/EUVIP47703.2019.8946206.

[7] ISO/IEC JTC1 SC37 Biometrics. Iso/iec 2382-37: 2017. information technology–vocabulary–part 37: biometrics, 2017.

[8] L Goasduff. Gartner says worldwide smartphone sales will grow 3% in 2020, 2020.

[9] Abhijit Das, Chiara Galdi, Hu Han, Raghavendra Ramachandra, Jean-Luc Dugelay, and Antitza Dantcheva. Recent advances in biometric technology for mobile devices. In *2018 IEEE 9th international conference on biometrics theory, applications and systems (BTAS)*, pages 1–11. IEEE, 2018.

[10] Muzhi Zhou and Man-Yee Kan. The varying impacts of covid-19 and its related measures in the uk: A year in review. *PloS one*, 16(9):e0257286, 2021.

[11] Alexander L. Fradkov. Early history of machine learning. *IFAC-PapersOnLine*, 53(2):1385–1390, 2020. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2020.12.1888. URL https://www.sciencedirect.com/science/article/pii/S2405896320325027. 21st IFAC World Congress.

[12] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[13] Marvin Minsky and Seymour Papert. Perceptrons. 1969.

[14] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985. ISSN 0364-0213. doi: https://doi.org/10.1016/S0364-0213(85)80012-4. URL https://www.sciencedirect.com/science/article/pii/S0364021385800124.

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[16] Md. Zahangir Alom, Tarek Taha, Christopher Yakopcic, Stefan Westberg, Mahmudul Hasan, Brian Esesn, Abdul Awwal, and Vijayan Asari. The history began from alexnet: A comprehensive survey on deep learning approaches. 03 2018.

[17] Marcus Liwicki, Alex Graves, Santiago Fernàndez, Horst Bunke, and Jürgen Schmidhuber. A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In *Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007*, 2007.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[19] R. Saravanan and Pothula Sujatha. A state of art techniques on machine learning algorithms: A perspective of supervised learning approaches in data classification. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 945–949, 2018. doi: 10.1109/ICCONS.2018.8663155.

[20] Xiaojin Zhu and Andrew B. Goldberg. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–130, 2009. doi: 10.2200/S00196ED1V01Y200906AIM006.

[21] Aristotelis Lazaridis, Anestis Fachantidis, and Ioannis Vlahavas. Deep reinforcement learning: A state-of-the-art walkthrough. *Journal of Artificial Intelligence Research*, 69:1421–1471, 2020.

[22] Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pages 679–684, 1957.

[23] Philip H. Swain and Hans Hauska. The decision tree classifier: Design and potential. *IEEE Transactions on Geoscience Electronics*, 15(3):142–147, 1977. doi: 10.1109/TGE.1977.6498972.

[24] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[25] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[26] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008. doi: 10.1109/ICDM.2008.17.

[27] Carl Rasmussen. The infinite gaussian mixture model. *Advances in neural information processing systems*, 12, 1999.

[28] Halid Z Yerebakan, Bartek Rajwa, and Murat Dundar. The infinite mixture of infinite gaussian mixtures. *Advances in neural information processing systems*, 27, 2014.

[29] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[30] Glenn W Milligan and Martha C Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2): 159–179, 1985.

[31] Greg Hamerly and Charles Elkan. Learning the k in k-means. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003. URL https://proceedings.neurips.cc/paper/2003/file/234833147b97bb6aed53a8f4f1c7a7d8-Paper.pdf.

[32] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer, 2003.

[33] Ganesh R Naik and Dinesh K Kumar. An overview of independent component analysis and its applications. *Informatica*, 35(1), 2011.

[34] Takio Kurita. Principal component analysis (pca). *Computer Vision: A Reference Guide*, pages 1–4, 2019.

[35] James V Stone. Independent component analysis: a tutorial introduction. 2004.

[36] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

[37] Vincent Spruyt May 3, Vincent Spruyt, and Vincent Spruyt. Loc2vec: Learning location embeddings with triplet-loss networks, Dec 2021. URL https://sentiance.com/2018/05/03/loc2vec-learning-location-embeddings-w-triplet-loss-networks/.

[38] Elakkiya Ellavarason. *Touch-screen Behavioural Biometrics on Mobile Devices*. PhD thesis, University of Kent,, 2021.

[39] AK JainA. Rosss. prabhakar,". *An introduction to biometric recognition*, pages 4–20.

[40] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication. *IEEE Transactions on Information Forensics and Security*, 8(1):136–148, 2013. doi: 10.1109/tifs.2012.2225048.

[41] Abdul Serwadda, Vir V. Phoha, and Zibo Wang. Which Verifiers Work?: A Benchmark Evaluation of Touch-based Authentication Algorithms. In *Proc. IEEE Intl. Conf. on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–8, 2013. doi: 10.1109/btas.2013.6712758.

[42] Lingjun Li, Xinxin Zhao, and Guoliang Xue. Unobservable re-authentication for smartphones. In *NDSS*, volume 56, pages 57–59, 2013.

[43] Hui Xu, Yangfan Zhou, and Michael R. Lyu. Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones. In *10th Symposium On Usable Privacy and Security (SOUPS 2014)*, pages

187–198, Menlo Park, CA, July 2014. USENIX Association. ISBN 978-1-931971-13-3. URL https://www.usenix.org/conference/soups2014/proceedings/presentation/xu.

[44] Tao Feng, Jun Yang, Zhixian Yan, Emmanuel Munguia Tapia, and Weidong Shi. Tips: Context-aware implicit user identification using touch screen in uncontrolled environments. In *Proceedings of the 15th workshop on mobile computing systems and applications*, pages 1–6, 2014.

[45] Margit Antal, Zsolt Bokor, and László Zsolt Szabó. Information Revealed from Scrolling Interactions on Mobile Devices. *Pattern Recognition Letters*, 56:7–13, 2015. doi: 10.1016/j.patrec.2015.01.011.

[46] Oscar Miguel-Hurtado, Sarah V Stevenage, Chris Bevan, and Richard Guest. Predicting sex as a soft-biometrics from device interaction swipe gestures. *Pattern Recognition Letters*, 79:44–51, 2016.

[47] Julian Fierrez, Ada Pozo, Marcos Martinez-Diaz, Javier Galbally, and Aythami Morales. Benchmarking touchscreen biometrics for mobile authentication. *IEEE Transactions on Information Forensics and Security*, 13(11): 2720–2733, 2018.

[48] Parker Lamb, Alexander Millar, and Ramon Fuentes. Swipe dynamics as a means of authentication: Results from a bayesian unsupervised approach. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9, 2020. doi: 10.1109/IJCB48548.2020.9304876.

[49] A Rezaei and S Mirzakuchaki. A recognition approach using multilayer perceptron and keyboard dynamics patterns. In *2013 First Iranian Conference on Pattern Recognition and Image Analysis (PRIA)*, pages 1–5. IEEE, 2013.

[50] Kevin S Killourhy. *A scientific understanding of keystroke dynamics*. PhD thesis, Carnegie Mellon University, 2012.

[51] Romain Giot, Alexandre Ninassi, Mohamad El-Abed, and Christophe Rosenberger. Analysis of the acquisition process for keystroke dynamics. In *2012*

BIOSIG-Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG), pages 1–6. IEEE, 2012.

[52] Shanthi Bhatt and T Santhanam. Keystroke dynamics for biometric authentication—a survey. In *2013 international conference on pattern recognition, informatics and mobile engineering*, pages 17–23. IEEE, 2013.

[53] Nathan L Clarke, SM Furnell, BM Lines, and Paul L Reynolds. Keystroke dynamics on a mobile handset: a feasibility study. *Information Management & Computer Security*, 2003.

[54] Ignacio de Mendizabal-Vazquez, Daniel de Santos-Sierra, Javier Guerra-Casanova, and Carmen Sánchez-Ávila. Supervised classification methods applied to keystroke dynamics through mobile devices. In *2014 International Carnahan conference on security technology (ICCST)*, pages 1–6. IEEE, 2014.

[55] Jiyun Wu and Zhide Chen. An implicit identity authentication system considering changes of gesture based on keystroke behaviors. *International Journal of Distributed Sensor Networks*, 11(6):470274, 2015.

[56] Pin Shen Teh, Ning Zhang, Andrew Beng Jin Teoh, and Ke Chen. Recognizing your touch: Towards strengthening mobile device authentication via touch dynamics integration. In *Proceedings of the 13th International Conference on Advances in Mobile Computing and Multimedia*, pages 108–116, 2015.

[57] Paul Kligfield, Leonard S Gettes, James J Bailey, Rory Childers, Barbara J Deal, E William Hancock, Gerard van Herpen, Jan A Kors, Peter Macfarlane, David M Mirvis, Olle Pahlm, Pentti Rautaharju, Galen S Wagner, Mark Josephson, Jay W Mason, Peter Okin, Borys Surawicz, and Hein Wellens. Recommendations for the standardization and interpretation of the electrocardiogram: part i: the electrocardiogram and its technology a scientific statement from the american heart association electrocardiography and arrhythmias committee, council on clinical cardiology; the american

college of cardiology foundation; and the heart rhythm society endorsed by the international society for computerized electrocardiology. *Journal of the American College of Cardiology*, 49(10):1109–27, Mar 2007.

[58] BW Johansson. [a history of the electrocardiogram]. *Dansk medicinhistorisk arbog*, page 163—176, 2001. ISSN 0084-9588. URL http://europepmc.org/abstract/MED/11848076.

[59] Xiang An and George K. Stylios. A hybrid textile electrode for electrocardiogram (ecg) measurement and motion tracking. *Materials*, 11(10), 2018. ISSN 1996-1944. doi: 10.3390/ma11101887. URL https://www.mdpi.com/1996-1944/11/10/1887.

[60] John G Webster. *Medical instrumentation: application and design*. John Wiley & Sons, 2009.

[61] João Loures Salinet and Olavo Luppi Silva. Chapter 2 - ecg signal acquisition systems. In João Paulo do Vale Madeiro, Paulo César Cortez, José Maria da Silva Monteiro Filho, and Angelo Roncalli Alencar Brayner, editors, *Developments and Applications for ECG Signal Processing*, pages 29–51. Academic Press, 2019. ISBN 978-0-12-814035-2. doi: https://doi.org/10.1016/B978-0-12-814035-2.00008-6. URL https://www.sciencedirect.com/science/article/pii/B9780128140352000086.

[62] Keyun Chen, Lei Ren, Zhipeng Chen, Chengfeng Pan, Wei Zhou, and Lelun Jiang. Fabrication of micro-needle electrodes for bio-signal recording by a magnetization-induced self-assembly method. *Sensors*, 16(9), 2016. ISSN 1424-8220. doi: 10.3390/s16091533. URL https://www.mdpi.com/1424-8220/16/9/1533.

[63] Amer Abdulmahdi Chlaihawi, Binu Baby Narakathu, Sepehr Emamian, Bradley J. Bazuin, and Massood Z. Atashbar. Development of printed and flexible dry ecg electrodes. *Sensing and Bio-Sensing Research*, 20: 9–15, 2018. ISSN 2214-1804. doi: https://doi.org/10.1016/j.sbsr.2018.

05.001. URL https://www.sciencedirect.com/science/article/pii/S2214180418300254.

[64] Katya Arquilla, Andrea K. Webb, and Allison P. Anderson. Textile electrocardiogram (ecg) electrodes for wearable health monitoring. *Sensors*, 20(4), 2020. ISSN 1424-8220. doi: 10.3390/s20041013. URL https://www.mdpi.com/1424-8220/20/4/1013.

[65] Thomas Degen and Heinz Jaeckel. Enhancing interference rejection of preamplified electrodes by automated gain adaption. *IEEE transactions on bio-medical engineering*, 51:2031–9, 11 2004. doi: 10.1109/TBME.2004.834296.

[66] Tuna B Tarim and Mohammed Ismail. Enhanced analog" yields" cost-effective systems-on-chip. *IEEE circuits and devices magazine*, 15(2):12–22, 1999.

[67] Peter L Levin and Reinhold Ludwig. Crossroads for mixed-signal chips. *IEEE Spectrum*, 39(3):38–43, 2002.

[68] Takashi Handa, Shuichi Shoji, Shinichi Ike, Sunao Takeda, and Tetsushi Sekiguchi. A very low-power consumption wireless ecg monitoring system using body as a signal transmission medium. In *Proceedings of International Solid State Sensors and Actuators Conference (Transducers' 97)*, volume 2, pages 1003–1006. IEEE, 1997.

[69] Shuenn-Yuh Lee and Chih-Jen Cheng. Systematic design and modeling of a ota-c filter for portable ecg detection. *IEEE Transactions on Biomedical Circuits and Systems*, 3(1):53–64, 2009.

[70] Ngoc Thang Bui, Tan Hung Vo, Byung-Gak Kim, and Junghwan Oh. Design of a solar-powered portable ecg device with optimal power consumption and high accuracy measurement. *Applied Sciences*, 9(10):2129, 2019.

[71] Seema Nayak, MK Soni, Dipali Bansal, et al. Filtering techniques for ecg signal processing. *International Journal of Research in Engineering & Applied Sciences*, 2(2):671–679, 2012.

[72] Ying-Wen Bai, Wen-Yang Chu, Chien-Yu Chen, Yi-Ting Lee, Yi-Ching Tsai, and Cheng-Hung Tsai. Adjustable 60hz noise reduction by a notch filter for ecg signals. In *Proceedings of the 21st IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No. 04CH37510)*, volume 3, pages 1706–1711. IEEE, 2004.

[73] SW Leung and YT Zhang. Digitization of electrocardiogram (ecg) signals using delta-sigma modulation. In *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol. 20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No. 98CH36286)*, volume 4, pages 1964–1966. IEEE, 1998.

[74] Ashish Kumar, Harshit Tomar, Virender Kumar Mehla, Rama Komaragiri, and Manjeet Kumar. Stationary wavelet transform based ecg signal denoising method. *ISA transactions*, 114:251–262, 2021.

[75] Haroon Yousuf Mir and Omkar Singh. Ecg denoising and feature extraction techniques–a review. *Journal of medical engineering & technology*, 45(8): 672–684, 2021.

[76] LV Rajani Kumari, Y Padma Sai, and N Balaji. R-peak identification in ecg signals using pattern-adapted wavelet technique. *IETE Journal of Research*, pages 1–10, 2021.

[77] N. A. Abdul-Kadir, N.S. Sahar, W. H. Chan, and F. K. C. Harun. A portable wifi ecg. In *2018 IEEE 38th International Electronics Manufacturing Technology Conference (IEMT)*, pages 1–4, 2018. doi: 10.1109/IEMT.2018.8511698.

[78] Ngoc Thang Bui, Duc Tri Phan, Thanh Phuoc Nguyen, Giang Hoang, Jaeyeop Choi, Quoc Cuong Bui, and Junghwan Oh. Real-time filtering

and ecg signal processing based on dual-core digital signal controller system. *IEEE Sensors Journal*, 20(12):6492–6503, 2020. doi: 10.1109/JSEN.2020.2975006.

[79] Gloria Cosoli, Susanna Spinsante, Francesco Scardulla, Leonardo D'Acquisto, and Lorenzo Scalise. Wireless ecg and cardiac monitoring systems: State of the art, available commercial devices and useful electronic components. *Measurement*, 177:109243, 2021. ISSN 0263-2241. doi: https://doi.org/10.1016/j.measurement.2021.109243. URL https://www.sciencedirect.com/science/article/pii/S0263224121002542.

[80] Quasar website. http://www.quasarusa.com/, .

[81] Qardio website. https://www.getqardio.com/it/qardiocore-wearable-ecg-ekg-monitor-iphone/, .

[82] Adam Page, Amey Kulkarni, and Tinoosh Mohsenin. Utilizing deep neural nets for an embedded ecg-based biometric authentication system. In *2015 IEEE biomedical circuits and systems conference (BioCAS)*, pages 1–4. IEEE, 2015.

[83] Lena Biel, Ola Pettersson, Lennart Philipson, and Peter Wide. Ecg analysis: a new approach in human identification. *IEEE Transactions on Instrumentation and Measurement*, 50(3):808–812, 2001.

[84] Steven A Israel, John M Irvine, Andrew Cheng, Mark D Wiederhold, and Brenda K Wiederhold. Ecg to identify individuals. *Pattern recognition*, 38 (1):133–142, 2005.

[85] Yongjin Wang, Foteini Agrafioti, Dimitrios Hatzinakos, and Konstantinos N Plataniotis. Analysis of human electrocardiogram for biometric recognition. *EURASIP journal on Advances in Signal Processing*, 2008:1–11, 2007.

[86] Chuang-Chien Chiu, Chou-Min Chuang, and Chih-Yu Hsu. A novel personal identity verification approach using a discrete wavelet transform of the

ecg signal. In *2008 International Conference on Multimedia and Ubiquitous Engineering (mue 2008)*, pages 201–206. IEEE, 2008.

[87] Adrian DC Chan, Mohyeldin M Hamdy, Armin Badre, and Vesal Badee. Wavelet distance measure for person identification using electrocardiograms. *IEEE transactions on instrumentation and measurement*, 57(2):248–253, 2008.

[88] Janani C Sriram, Minho Shin, Tanzeem Choudhury, and David Kotz. Activity-aware ecg-based patient authentication for remote health monitoring. In *Proceedings of the 2009 international conference on Multimodal interfaces*, pages 297–304, 2009.

[89] Tsu-Wang Shen, WJ Tompkins, and YH Hu. One-lead ecg for identity verification. In *Proceedings of the Second Joint 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society][Engineering in Medicine and Biology*, volume 1, pages 62–63. IEEE, 2002.

[90] Lukasz Wieclaw, Yuriy Khoma, Pawel Fałat, Dmytro Sabodashko, and Veronika Herasymenko. Biometrie identification from raw ecg signal using deep learning techniques. In *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, volume 1, pages 129–133. IEEE, 2017.

[91] Qingxue Zhang, Dian Zhou, and Xuan Zeng. Pulseprint: Single-arm-ecg biometric human identification using deep learning. In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pages 452–456. IEEE, 2017.

[92] Dhananjai Bajpai and Lili He. Evaluating knn performance on wesad dataset. In *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 60–62. IEEE, 2020.

[93] Pramod Bobade and M Vani. Stress detection with machine learning and deep learning using multimodal physiological data. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 51–57. IEEE, 2020.

[94] Arun Ross and Anil Jain. Information fusion in biometrics. *Pattern recognition letters*, 24(13):2115–2125, 2003.

[95] Neeru Bala, Rashmi Gupta, and Anil Kumar. Multimodal biometric system based on fusion techniques: a review. *Information Security Journal: A Global Perspective*, pages 1–49, 2021.

[96] Arun A Ross and Rohin Govindarajan. Feature level fusion of hand and face biometrics. In *Biometric technology for human identification II*, volume 5779, pages 196–204. SPIE, 2005.

[97] Keshav Gupta, Gurjit Singh Walia, and Kapil Sharma. Novel approach for multimodal feature fusion to generate cancelable biometric. *The Visual Computer*, 37(6):1401–1413, 2021.

[98] Ka-Wing Tse and Kevin Hung. User behavioral biometrics identification on mobile platform using multimodal fusion of keystroke and swipe dynamics and recurrent neural network. In *2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pages 262–267. IEEE, 2020.

[99] Anil Jain, Karthik Nandakumar, and Arun Ross. Score normalization in multimodal biometric systems. *Pattern recognition*, 38(12):2270–2285, 2005.

[100] Mingxing He, Shi-Jinn Horng, Pingzhi Fan, Ray-Shine Run, Rong-Jian Chen, Jui-Lin Lai, Muhammad Khurram Khan, and Kevin Octavius Sentosa. Performance evaluation of score level fusion in multimodal biometric systems. *Pattern Recognition*, 43(5):1789–1800, 2010.

[101] Mustafa Berkay Yılmaz and Berrin Yanıkoğlu. Score level fusion of classifiers in off-line signature verification. *Information Fusion*, 32:109–119, 2016.

[102] Karthik Nandakumar, Yi Chen, Sarat C Dass, and Anil Jain. Likelihood ratio-based biometric score fusion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):342–347, 2007.

[103] Mayank Vatsa, Richa Singh, and Afzel Noore. Integrating image quality in 2ν-svm biometric match score fusion. *International Journal of Neural Systems*, 17(05):343–351, 2007.

[104] Salil Prabhakar and Anil K Jain. Decision-level fusion in fingerprint verification. *Pattern Recognition*, 35(4):861–874, 2002.

[105] Priti S Sanjekar and JB Patil. Multimodal biometrics with serial parallel and hierarchical mode at decision level fusion. *Indonesian Journal of Electrical Engineering and Computer Science*, 16(3):1303–1310, 2019.

[106] Sahar A El_Rahman. Multimodal biometric systems based on different fusion levels of ecg and fingerprint using different classifiers. *Soft Computing*, 24 (16):12599–12632, 2020.

[107] Elakkiya Ellavarason, Richard Guest, and Farzin Deravi. A framework for assessing factors influencing user interaction for touch-based biometrics. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 553–557. IEEE, 2018.

[108] Soumik Mondal and Patrick Bours. Swipe gesture based continuous authentication for mobile devices. In *2015 International Conference on Biometrics (ICB)*, pages 458–465. IEEE, 2015.

[109] Ada Pozo, Julian Fierrez, Marcos Martinez-Diaz, Javier Galbally, and Aythami Morales. Exploring a statistical method for touchscreen swipe biometrics. In *2017 International Carnahan Conference on Security Technology (ICCST)*, pages 1–4. IEEE, 2017.

[110] Elakkiya Ellavarason, Richard Guest, and Farzin Deravi. Evaluation of stability of swipe gesture authentication across usage scenarios of mobile device. *EURASIP Journal on Information Security*, 2020(1):1–14, 2020.

[111] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[112] Jeff Heaton. Ian goodfellow, yoshua bengio, and aaron courville: Deep learning, 2018.

[113] Saad Hikmat Haji and Adnan Mohsin Abdulazeez. Comparison of optimization techniques based on gradient descent algorithm: A review. *PalArch's Journal of Archaeology of Egypt/Egyptology*, 18(4):2715–2743, 2021.

[114] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. Introduction to multilayer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39(1):43–62, 1997.

[115] Christopher Olah. Understanding lstm networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015.

[116] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

[117] Sebastian Ruder. Deep learning for nlp best practices, Jun 2020. URL https://ruder.io/deep-learning-nlp-best-practices/.

[118] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

[119] Zhedong Zheng, Liang Zheng, and Yi Yang. A discriminatively learned cnn embedding for person reidentification. *ACM transactions on multimedia computing, communications, and applications (TOMM)*, 14(1):1–20, 2017.

[120] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[121] Elham Tabassi and Patrick Grother. Biometric Quality. The Last 1%-Biometric Quality Assessment for Error Suppression. Technical report, 2009.

[122] Samarth Bharadwaj, Mayank Vatsa, and Richa Singh. Biometric Quality: A Review of Fingerprint, Iris, and Face. *EURASIP Journal on Image and Video Processing*, 2014(1), Feb 2014. doi: 10.1186/1687-5281-2014-34.

[123] Elham Tabassi. NIST Fingerprint Image Quality (NFIQ) Compliance Test. 2005. doi: 10.6028/nist.ir.7300.

[124] Zhigang Yao, Jean-Marie Le Bars, Christophe Charrier, and Christophe Rosenberger. Literature Review of Fingerprint Quality Assessment and its Evaluation. *IET Biometrics*, 5(3):243–251, Jan 2016. doi: 10.1049/iet-bmt.2015.0027.

[125] Barbara Corsetti, Raul Sanchez-Reillo, Richard M Guest, and Marco Santopietro. Face Image Analysis in Mobile Biometric Accessibility Evaluations. In *Proc. 2019 International Carnahan Conference on Security Technology (ICCST)*, pages 1–5, 2019.

[126] Jiansheng Chen, Yu Deng, Gaocheng Bai, and Guangda Su. Face Image Quality Assessment Based on Learning to Rank. *IEEE Signal Processing Letters*, 22(1):90–94, 2015. doi: 10.1109/lsp.2014.2347419.

[127] Javier Hernandez-Ortega, Javier Galbally, Julian Fierrez, Rudolf Haraksim, and Laurent Beslay. FaceQnet: Quality Assessment for Face Recognition based on Deep Learning. In *Proc. 2019 International Conference on Biometrics (ICB)*, pages 1–9, 2019.

[128] Patrick Grother, Austin Hom, Mei Ngan, and Kayee Hanaoka. Ongoing Face RecognitionVendor Test (FRVT). Part 5: Face Image Quality Asssessment. Technical report, NIST, 2020.

[129] Sascha Müller and Olaf Henniger. Evaluating the Biometric Sample Quality of Handwritten Signatures. *Advances in Biometrics, Lecture Notes in Computer Science*, page 407–414, 2007. doi: 10.1007/978-3-540-74549-5_43.

[130] Javier Galbally, Julian Fierrez, Marcos Martinez-Diaz, and Réjean Plamondon. Quality Analysis of Dynamic Signature based on the Sigma-Lognormal Model. In *Proc. ICDAR*, pages 633–637, 2011.

[131] Napa Sae-Bae and Nasir Memon. Quality of Online Signature Templates. In *Proc. IEEE Intl. Conf. on Identity, Security and Behavior Analysis (ISBA 2015)*, pages 1–8, 2015.

[132] George Doddington, Walter Liggett, Alvin Martin, Mark Przybocki, and Douglas Reynolds. Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the nist 1998 speaker recognition evaluation. Technical report, National Inst of Standards and Technology Gaithersburg Md, 1998.

[133] Kristina P Sinaga and Miin-Shen Yang. Unsupervised k-means clustering algorithm. *IEEE access*, 8:80716–80727, 2020.

[134] Structure of the heart. https://studymind.co.uk/notes/structure-of-the-heart/, .

[135] Philip Schmidt, Attila Reiss, Robert Duerichen, Claus Marberger, and Kristof Van Laerhoven. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In *Proceedings of the 20th ACM international conference on multimodal interaction*, pages 400–408, 2018.

[136] Jean-Philippe Couderc, Xia Xiaojuan, Wojciech Zareba, and Arthur J Moss. Assessment of the stability of the individual-based correction of qt interval for heart rate. *Annals of Noninvasive Electrocardiology*, 10(1):25–34, 2005.

[137] Tatiana S Lugovaya. Biometric human identification based on ecg. *PhysioNet*, 2005.

[138] Jiapu Pan and Willis J Tompkins. A real-time qrs detection algorithm. *IEEE transactions on biomedical engineering*, (3):230–236, 1985.

[139] Kiran Kumar Patro and P Rajesh Kumar. Effective feature extraction of ecg for biometric application. *Procedia computer science*, 115:296–306, 2017.

[140] Dominique Makowski, Tam Pham, Zen J Lau, Jan C Brammer, François Lespinasse, Hung Pham, Christopher Schölzel, and SH Annabel Chen. Neurokit2: A python toolbox for neurophysiological signal processing. *Behavior Research Methods*, pages 1–8, 2021.

[141] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[142] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[143] Ruggero Donida Labati, Enrique Muñoz, Vincenzo Piuri, Roberto Sassi, and Fabio Scotti. Deep-ecg: Convolutional neural networks for ecg biometric recognition. *Pattern Recognition Letters*, 126:78–85, 2019.

[144] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[145] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. National Institute of Science of India, 1936.