UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA Y DE LAS COMUNICACIONES

Modelling Human-Computer Interaction: New Applications based on Biometric Signal Processing

–TESIS DOCTORAL–

Modelado de la Interacción Hombre-Máquina: Nuevas Aplicaciones basadas en el Procesado de Señales Biométricas

Author: Alejandro Acién Ayala
(Ingeniero de Telecomunicación)

A Thesis submitted for the degree of:

Doctor of Philosophy

Madrid, July 2021

Colophon

This book was typeset by the author using LaTeX2e. The main body of the text was set using a 11-points Computer Modern Roman font. All graphics and images were included formatted as Encapsulated Postscript (™ Adobe Systems Incorporated). The final postscript output was converted to Portable Document Format (PDF) and printed.

This Thesis was printed with the financial support from EPS-UAM and the Biometrics and Data Pattern Analytics Laboratory - BiDA Lab.
contact: alejandro.acien@uam.es

Department:	Tecnología Electrónica y de las Comunicaciones Escuela Politécnica Superior Universidad Autónoma de Madrid (UAM), SPAIN
PhD Thesis:	Modelling Human-Computer Interaction: New Applications based on Biometrics Signal Processing
Author:	**Alejandro Acién Ayala** Ingeniero de Telecomunicación (Universidad Autónoma de Madrid, SPAIN)
Advisors:	**Aythami Morales Moreno** Doctor Ingeniero de Telecomunicación (Las Palmas de Gran Canaria University, SPAIN) Universidad Autónoma de Madrid, SPAIN **Rubén Vera Rodríguez** Doctor Ingeniero de Telecomunicación (Swansea University, UK) Universidad Autónoma de Madrid, SPAIN
Year:	2021
Committee:	President: **Javier Ortega-García** Universidad Autónoma de Madrid, SPAIN Secretary: **Óscar Déniz Alberto Suarez** Universidad de Castilla la Mancha, SPAIN Vocal 1: **Patrick Bours** Norwegian University of Science and Technology, NORWAY Substitute 1: **Pedro Gómez Vilda** Universidad Politécnica de Madrid, SPAIN Substitute 2: **Javier Galbally Herrero** European Commission - Joint Research Centre, ITALY

BiDA Lab

The research described in this Thesis was carried out within the Biometrics and Data Pattern Analytics Laboratory - BiDA Lab at the Dept. of Tecnología Electrónica y de las Comunicaciones, Escuela Politécnica Superior, Universidad Autónoma de Madrid (from 2017 to 2021). The project was funded by a FPI fellowship from the Spanish MINECO.

Abstract

The research interest in behavioral biometrics has been constantly growing, motivated by the fastest digital revolution that the mankind are experiencing in the last years. This revolution is associated with a massive deployment of digital devices including multiple sensors (e.g., camera, gyroscope, GPS, touch screens, etc.), full connectivity (e.g., bluetooth, Wi-Fi, 4G, etc.) and high computation power (e.g., multiple core CPUs, more advanced GPUs, etc.). As a consequence, services are rapidly migrating from the physical to the digital domain in the information society. Examples can be found in e-government, banking, education, health or commerce. The capacity of these devices to acquire, process, and storage a wide range of heterogeneous data during these human device interactions offers many possibilities and new research lines (e.g., security, health, biology, sociology, etc.).

The main purpose of this Thesis is the analysis and development of new applications based on machine learning algorithms for continuous user modelling, applied to data acquired through the interaction of the user with digital devices. Most of these algorithms can be applied in a transparent way for the users, exploiting the large number of signals generated during this interaction. The Thesis addresses the problem from a holistic perspective, simultaneously analyzing large number of sources of biometric and metadata information that can be acquired not only on mobile devices, but also on desktop computers. The acquisition, preprocessing, analysis, pattern classification and combination of such a huge volume of information is a challenge for the research community due to the heterogeneous nature of the data and the necessity of a continuous monitoring of the users. Besides, many of these new algorithms that model user's behaviour during digital interactions can be applied beyond user authentication, such as health monitoring, behavior analysis, or bot detection; demonstrating the potential of these devices to set a new era of biometric research lines to model human behaviours.

This Dissertation comprises four different parts. *Part I* first introduces the biometric traits that will be used to model Human-Computer Interactions (HCI), as well as a detailed description of the main materials and methods employed throughout the entire Thesis. The first experimental part (*Part II* of this Dissertation) is focused in user mobile authentication algorithms and their multiple implementations like unimodal versus multimodal systems or one time versus active authentication setups. In the second experimental part (*Part III* of this Dissertation) we propose new ways to exploit these mobile biometric traits for different HCI behaviour applications beyond user authentication, such as Parkinson disease characterization through handwriting skills and age detection with touchscreen gestures. In the last experimental part (*Part IV* of this Dissertation), with the knowledge acquired in the previous experimental parts we will combine both security and behavior applications to develop a new generation of bot detection algorithms based on user's behavioural analysis. Finally, *Part IV* presents the main conclusions drawn of this Dissertation and the future work.

Resumen

El interés en la investigación de la biometría del comportamiento ha ido crecido constantemente en los últimos años, motivado en parte por la rápida revolución digital que la humanidad está experimentando. Dicha revolución está asociada a un despliegue masivo de dispositivos digitales que incluyen múltiples sensores (p. ej., cámara, giroscopio, GPS, pantallas táctiles, etc.), una conectividad total (p. ej., bluetooth, Wi-Fi, 4G, etc.) y con una gran potencia de cálculo (p. ej., CPUs con múltiples núcleos, GPUs más avanzadas, etc.). Como consecuencia, los servicios están migrando rápidamente del ámbito físico al digital en esta sociedad de la información que se esta desarrollando. Podemos encontrar ejemplos en la administración nacional electrónica, la banca, la educación online, la sanidad o el comercio. La capacidad de estos dispositivos para adquirir, procesar y almacenar una amplia gama de datos heterogéneos durante las interacciones entre dichos dispositivos y los humanos ofrece muchas posibilidades y líneas de investigación nuevas en hámbitos muy diferentes (p. ej., seguridad, salud, biología, sociología, etc.).

El objetivo principal de esta Tesis es el análisis y desarrollo de nuevas aplicaciones basadas en algoritmos de inteligencia artificial para el modelado y la caracterización contínua del usuario, aplicados a los datos adquiridos a través de la interacción del usuario con los dispositivos digitales. La mayoría de estos algoritmos pueden funcionar de forma totalmente transparente para el usuario, explotando todas estas señales biométricas generadas durante dicha interacción. La Tesis aborda el problema desde una perspectiva holística, analizando simultáneamente un gran número de fuentes de información biométrica y de metadatos que pueden ser adquiridos no sólo en dispositivos móviles, sino también en ordenadores de sobremesa. La adquisición, el preprocesamiento, el análisis, la clasificación de patrones y la combinación de un volumen tan grande de información es un reto para la comunidad investigadora debido a la naturaleza heterogénea de los datos y a la necesidad de un monitoreo continuo de los mismos. Además, muchos de estos nuevos algoritmos que modelan el comportamiento del usuario durante las interacciones hombre-máquina pueden aplicarse más allá de la autenticación de usuarios, como puede ser la monitorización y diagnóstico de enfermedades neurodegenerativas, análisis del comportamiento humano o la detección de softwares maliciosos (bots); lo que demuestra el potencial de estos dispositivos digitales para asentar las bases de un gran número de líneas de investigación biométrica basadas en el modelado y caracterización del comportamiento humano.

Esta Tesis consta de cuatro partes. La *Parte I* presenta los diferentes rasgos biométricos que se utilizarán para modelar las interacciones hombre-máquina (HCI, por sus siglas en inglés), así como una descripción detallada de las principales bases de datos y métodos empleados a lo largo de toda la Tesis. La primera parte experimental (*Parte II*) se centra en algoritmos de autenticación móvil del usuario y sus múltiples implementaciones, como los sistemas unimodales frente a los multimodales o los escenarios de autenticación única frente a escenarios de auten-

ticación continua. En la segunda parte experimental (*Parte III*) proponemos nuevas formas de explotar dichos rasgos biométricos para diferentes aplicaciones más allá de la autenticación del usuario, como la caracterización de la enfermedad de Parkinson a través de la escritura on-line y la detección de edad a partir de gestos realizados en pantallas táctiles. En la última parte experimental (*Parte IV*), combinamos los conocimientos adquiridos en las partes experimentales anteriores para desarrollar una nueva generación de algoritmos de detección de bots (CAPTCHAS, por sus siglas en inglés) basados en el análisis del comportamiento del usuario con diferentes dispositivos digitales. Finalmente, en la *Parte IV* se presentan las principales conclusiones extraídas de esta Tesis y los trabajos futuros que se desprenden de ella.

Nuestras convicciones más arraigadas,
más indubitables, son las más sospechosas.
Ellas constituyen nuestro límite,
nuestros confines, nuestra prisión.

José Ortega y Gasset

Acknowledgements

This Thesis, as a long journey that started 4 years ago has brought me many good times and life experiences that I will never able to thank enough to those wonderful people that made it possible. Although many people come to my mind, I feel that I would never have enough space in these few lines to express my gratitude to all people a would like to, so let me apologize in advance if they are not here, though they never gone from my heart.

Foremost, none of this would have been possible without the support and infinite patience of my advisors: Dr. Aythami Morales, Prof. Julian Fierrez and Dr. Ruben Vera-Rodriguez. They were always there, guiding me when I got lost in this wild sea of papers, conferences, publications and research projects. Of course, special thanks to all my colleagues from the ATVS (now BiDA and AUDIAS): Dr. Ruben Tolosana, Dr. Ruben Zazo, Dr. Alicia Lozano, Ignacio de la Serna, Alvaro Escudero, Javier Hernandez-Ortega, and many more that probably I am forgetting (sorry again!). To all of you I want to say thanks, thanks a lot. Thank you not only for sharing with me the good moments of happiness and success, but also for your unconditional support during my worst moments of frustration and failures. Again, thanks you a lot.

I would like to reserve a few lines to acknowledge those people that I met in my research stays during the Thesis, people who opened my mind beyond my comfort zone and make me to discover beautiful places, cultures, delicious meals, and of course, people who offer me the opportunity to collaborate with them in awesome research projects and share their knowledge with me. These lines are dedicated to Reinel Castrillon and Dr. Juan Rafael Orozco-Arroyave from the University of Antioquia in Medellin. Thanks you for those two months in Medellin, it was the first time in my life I spent so much time abroad, and I have to say it was wonderful. I enjoyed every day, every lunch and every place you took me to visit. I would also like to thank Prof. John Vinnie Monaco for give me the opportunity to spend my second research stay with you in California, where I learnt from one of the best researchers in keystroke dynamics. I owe you a great part of my success in this Thesis.

Time to Spanish!

En primer lugar, me gustaría dar las gracias a todos mis grandes amigos de Tres Cantos, mis amigos de toda la vida, junto a los que he crecido desde que era un enano, con los que he vivido toda clases de aventuras, visitado medio mundo y compartido un sinfín de momentos felices que siempre conseguían opacar los momentos más duros de mi Tesis. En especial me gustaría destacar a todos y a cada uno de los 'miaus', pues se lo merecen: Ruben Aznal, Alejandro Carreras, Elena Chinchilla, Cristina Gallego, Guillermo García, Alberto Gullon, Anicris Marcano y Ryan Roncero. Gracias a todos de corazón. También me gustaría agradecer el apoyo de todos mis compañeros de carrera y que con el tiempo se convirtieron en amigos muy especiales para mí: Ana Moran, Elena Gómez y Alicia Sastre. Sin vuestra ayuda no hubiera llegado tan lejos.

Y finalmente y más importante, mi gran familia. Gracias por vuestro apoyo incondicional que me habéis dado siempre, pero que estos años he necesitado como nunca. En especial me gustaría dar las gracias a mi madre, Francisca Ayala, por su amor incondicional, todos los días esperándome para comer aunque llegara a las mil horas de la universidad, mi padre, Jose Acién, con el que he podido contar siempre para cualquier cosa que he necesitado y finalmente, mi hermana, Alicia Acién, un espejo en el que mirarme y poder verme reflejado algún día. Mi más sincero agradecimiento a todos vosotros. Todo lo que diga aquí es poco comparado con todo lo que os merecéis.

A todos vosotros, gracias.

Alejandro Acién Ayala
Madrid, July 2021

Glossary

- **AA**: Active Authentication.
- **ADD**: Average Detection Delay.
- **AUC**: Area Under the Curve.
- **CAPTCHA**: Completely Automated Public Turing test to tell Computers and Humans Apart.
- **EER**: Equal Error Rate.
- **EHC**: Elder Healthy Control.
- **FAR**: False Acceptance Rate.
- **FMR**: False Match Rate.
- **FNMR**: False Non-Match Rate.
- **FRR**: False Rejection Rate.
- **GAN**: Generative Adversarial Network.
- **GMM**: Gaussian Mixture Model.
- **GRU**: Gated Recurrent Unit.
- **HCI**: Human-Computer Interaction.
- **HMM**: Hidden Markov Model.
- **KL**: Kullback-Leibler.
- k**NN**: k-Nearest Neighbours.
- **LSTM**: Long-Short Term Memory.
- **M-HMM**: Mixture Hidden Markov Model.
- **MLP**: Multilayer Perceptron.
- **NB**: Naive Bayes.
- **OTA**: One-Time Authentication.
- **PD**: Parkinson's Disease.
- **PFD**: Probability of False Detection.

- **PND**: Probability of Non Detection.
- **POHMM**: Partially Observable Hidden Markov Model.
- **QCD**: Quickest Change Detection.
- **RBF**: Radial Basis Function.
- **RF**: Random Forest.
- **RNN**: Recurrent Neural Network.
- **ROC**: Receiver Operating Characteristic.
- **RSSI**: Radio Signal Strength Indicator.
- **SFFS**: Sequential Forward Floating Search.
- **SVM**: Support Vector Machine.
- **UBM**: Gaussian Mixture Model.
- **YHC**: Young Healthy Control.

Contents

List of Figures

List of Tables

Part I

Problem Statement and Contributions

Chapter 1

Introduction

The Interactions between Humans and Machines are undergoing a fast evolution during the last decade, to the point that several aspects of our lives are conditioned by the way we interact with them. Common day-to-day routines such as chatting with acquaintances, go shopping, work or even human relationships have drastically changed due to the proliferation of mobile devices (e.g., smartphones, laptops, tablets) and automatic processes, originally intended to make these day-to-day routines easier. On the other hand, the capacity of these technologies to acquire and store amounts of sensitive data captured during the user-device interaction open a wide range of new possibilities to study human aspects thorough different multidisciplinary fields (e.g., psychology, sociology, biology, behaviour).

At the same time, the evolution of the Human-Computer Interaction (HCI) field is in parallel with a growing interest in the biometrics research community towards more transparent and robust authentication methods that make use of these interaction signals originated when using mobile devices [Crouse *et al.*, 2015; Patel *et al.*, 2016]. Mobile devices possess sensors (e.g., gyroscope, magnetometer, accelerometer, GPS, touchscreen) along with metadata associated to our use of the technology (e.g., internet point access, browsing history, app usage) which could assist in user authentication by analyzing gait [Costilla-Reyes *et al.*, 2020; Muaaz and Mayrhofer, 2017], typing and scrolling touch signals [Fierrez *et al.*, 2018], or certain soft biometric information [Gonzalez-Sosa *et al.*, 2018; Tome *et al.*, 2014]. Those biometric signals are originated naturally during the normal usage of the device, and it has been demonstrated that they may have enough discriminative power for person identification under certain conditions. These kind of biometric signals have been studied under different perspectives in the last years, e.g.: as Behavioral Biometrics [Salah *et al.*, 2011] or Cognitive Biometrics [Al Galib and Safavi-Naini, 2015]. Exploiting these biometric signals in wearables and smartphones, new mobile experiences have the potential to continuously monitor the users and change the way they live and interact with each other. Some examples enable users to: quantify their sleep and exercise patterns, monitor personal commute behaviors, track their emotional state, or even measure how long they spend queuing in retail stores. In another example, by regularly conducting unobtrusive identity checks of the mobile user, continuous authentication applications verifies if the device

is still in an authenticated state [Traore, 2011]. With this active system, if the mobile device is stolen, it should quickly recognize the presence of an unauthorized user. All of these applications are achievable thanks to the combination of powerful algorithms to infer behaviors and contexts from sensor data collected by mobile devices.

In this introductory chapter we first present in Sec. 1.1 a general outlook of the biometric technologies that take advantage of modeling HCI signals to progress in different research areas, in order to establish the main pillars of this Dissertation. Then, we explain the motivations of the Thesis in Sec. 1.2 as well as the milestones and main contributions to the state-of-the-art in biometrics achieved in Sec. 1.3. In Sec. 1.4 we describe the structure followed of this document for a better comprehension. Finally, we dedicate Sec. 1.5 to enumerate the research contributions originated during the development of this Dissertation.

1.1. Biometrics for Modeling Human-Computer Interaction: General Outlook

Biometric technologies improve in several ways traditional recognition algorithms based on passwords or ID cards [Jain *et al.*, 2016]. The advantages of biometric systems are many in terms of security and convenience of use, which has led these technologies to take on a leading role in the last years thanks to the massive deployment of smartphones, tablets and other devices. As an example, the most popular biometric technologies (such as fingerprint, face or iris) have been linked in general to access control applications or forensic science, but nowadays we can also see those traditional biometrics incorporated to high-end mobile devices (e.g., Apple's Touch and Face ID). This massive deployment of mobile devices has encouraged researchers in recent years to study the discriminative ability of biometrics patterns associated with the interaction with this technology [Frank *et al.*, 2013; Tolosana *et al.*, 2020a].

The different biometric modalities are usually divided into two main groups, according to the nature of the human trait that we are analyzing: physiological and behavioral biometrics. Physiological biometrics refers to those biometrics focused in the physical measurements of the human body. These biometrics identify subjects according to their physical aspect, such as face, fingerprint, hand geometry, retina, or iris among others. On the other hand, behavioral biometrics are aimed to measure behavioral patterns during human activities or HCI, such as gait, handwriting, keystroke dynamics or signature among others. Behavioural biometrics not only identify subjects according to their innate human behaviours when interacting with devices, but also these biometrics can be applied to identify other aspect of human beings, like the detection and characterization of different human diseases (e.g., Parkinson, Alzheimer), age detection, or even for bot detection.

This Thesis is mainly focused on four behavioural biometric research areas: smartphone biometrics, keystroke dynamics, mouse dynamics, and on-line handwriting.

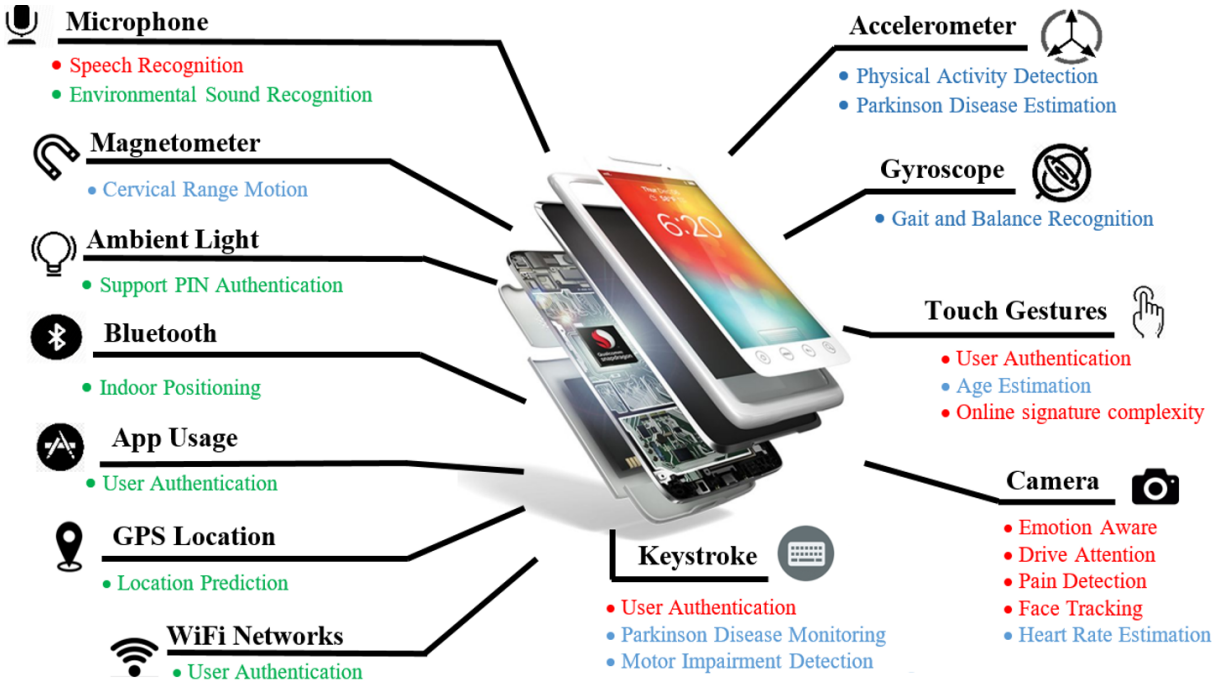

Figure 1.1: *A summary of the different sensors/signals of smartphone and example applications. In blue, applications that reveal neuromotor skills, in red, cognitive functions, and in green, applications revealing behaviors/routines.*

1.1.1. Smartphone Biometrics

Smartphones contain many sensors such as accelerometer, gyroscope, gravity sensor, touch-screen, light sensor, Wi-Fi, Bluetooth, camera, or microphone, among others, which can acquire information as the user is interacting with it or just carrying it. These sources of information can be used to model human-machine interaction and describe human features. Fig. 1.1 presents some examples of different research fields that exploit signals obtained or derived from mobile sensors.

- *Touch Gestures:* this biometric trait involve all kinds of finger movements that we perform over the smartphone screen (e.g., swipe, tap, zoom, etc) and has already been used for user authentication [Fierrez *et al.*, 2018]. More recently, the research community is focusing on the neuromotor patterns that can be extracted from touch gestures. As an example, in [Vera-Rodriguez *et al.*, 2019] the authors model the complexity of online signatures over smartphone touchscreens using the neuromotor patterns associated to touch gestures.

- *Accelerometer and gyroscope:* these sensors are both useful to measure the movements that the smartphone is exposed to. The accelerometer measures the magnitude and direction of acceleration forces applied over the mobile device meanwhile the gyroscope measures orientation. Although these sensors have been studied for mobile user authentication with good results [Deb *et al.*, 2019]. In the last years these qualities make both sensors traditionally useful for gait and balance recognition. For example, in [Barra *et al.*, 2018]

the authors employ these mobile sensors for user recognition trough simple gestures like answering a call in four different user states: standing, sitting, walking, and running. In another example, in [Gafurov *et al.*, 2006] the authors extracted gait patterns from a mobile device attached to the lower part of the leg in three directions: vertical, forward-backward, and sideways motion. They achieved error rates between 5% and 9% for gait authentication combining all three acceleration measures. Accelerometer has been also studied to measure the daily physical activities with the main goal of changing people's sedentary lifestyle [Sun *et al.*, 2010].

- *Wi-Fi, GPS and App Usage:* these mobile signals belong to behavioral-based profiling schemes due to their capacity to provide information about when and where we go and what we do. They record the events (e.g., Wi-Fi networks, Bluetooth signals, GPS locations, or application's name) and the timestamps of their occurrence. This discriminative information is considered as behavioral biometrics due to their capacity to detect variations in our daily routines [Patel *et al.*, 2016]. As an example, in [Mahbub and Chellappa, 2016] the authors developed a modified Hidden Markov Model (HMM) to characterize mobile GPS location histories. They suggest that human mobility can be described as a Markovian Motion and they make predictions of the next user location taking into account the sparseness of the data and previous user locations. In a similar way, in [Mahbub *et al.*, 2019] a variation of HMMs was studied to develop a user authentication mobile system by exploiting application usage data. The authors state that unforeseen events and unknown applications provide more discriminatory information in the authentication process than the most common apps used. In [Li and Bours, 2018c] the authors perform a template-based matching algorithm for user authentication using the Wi-Fi signals stored by the smartphone during the day. The fusion at score level with the accelerometer system achieve authentication error rates under 10%, showing the feasibility of Wi-Fi signals to assist authentication on mobile devices.

- *Bluetooth:* a mobile signal similar to Wi-Fi, which detects other Bluetooth beacons and the timestamps of occurrence. However, thanks to their low power consumption and the fact that works in a short range radio frequency, it is being studied for indoor positioning based on Radio Signal Strength Indicator (RSSI) Probability Distributions. For example, in [Pei *et al.*, 2010] the authors applied the Weibull function to approximate the Bluetooth signal strength distribution in the data training phase.

- *Others Mobile Sensors:* there are less obvious but also useful for modeling human computer interactions such as the light sensor, which measures the ambient-light level that the smartphone is exposed to. In [Spreitzer, 2014] the authors demonstrate that minor tilts and turns in the smartphone cause variations of the ambient-light sensor information. These variations leak enough information to authenticate personal identification numbers. Another sensor, the magnetometer, has been also studied to measure the cervical range

of motion on the horizontal plane using a smartphone placed on the head of the patient during a clinical trial [Tousignant-Laflamme *et al.*, 2013].

- *Camera and Microphone:* these sensors are two of the most important sensors of the mobile device. They take photos, selfies, record voice and sounds. These signals are being used for a wide range of research lines: user recognition [Shi *et al.*, 2011], emotion aware [Chen *et al.*, 2015], driver attention [Dua *et al.*, 2019], pain detection [Tavakolian *et al.*, 2019], face tracking [Lin *et al.*, 2019], heart rate estimation [Hernandez-Ortega *et al.*, 2020b], environmental sound recognition [Demir *et al.*, 2018], and speech recognition [Schuster, 2010] among others. However, their capacity to collect private user information can be perceived as intrusive.

The literature demonstrates the potential of mobile sensors to model inner human features (e.g., cognitive functions, neuromotor skills, and human behaviors/routines). These devices become data hubs that can be used in many different applications related to HCI.

1.1.2. Keystroke Biometrics

Keystroke dynamics is a behavioral biometric trait aimed at recognizing individuals based on their typing habits. The velocity of pressing and releasing different keys [Banerjee and Woodard, 2012], the hand postures during typing [Buschek *et al.*, 2015], and the pressure exerted when pressing a key [Acien *et al.*, 2019a] are some of the features taken into account by keystroke biometric algorithms aimed to discriminate among subjects. Although keystroke biometrics suffer high intra-class variability for person recognition, especially in free-text scenarios (i.e., the input text typed is not fixed between enrollment and testing), the ubiquity of keyboards as a method of text entry makes keystroke dynamics a near universal modality to authenticate subjects on the Internet.

Text entry is prevalent in day-to-day applications: unlocking a smartphone, accessing a bank account, chatting with acquaintances, email composition, posting content on a social network, and e-learning [Hernandez-Ortega *et al.*, 2020a]. As a means of subject authentication, keystroke dynamics is economical because it can be deployed on commodity hardware and remains transparent to the user. These properties have prompted several companies to capture and analyze keystrokes. The global keystroke biometrics market is projected to grow from $129.8 million dollars (2017 estimate) to $754.9 million by 2025, a rate of up to 25% per year[1]. As an example, Google has recently committed $7 million dollars to fund TypingDNA[2], a startup company which authenticates people based on their typing behavior.

At the same time, the security challenges that keystroke biometrics promises to solve are constantly evolving and getting more sophisticated every year: identity fraud, account takeover, sending unauthorized emails, and credit card fraud are some examples[3]. These challenges are

[1] https://www.prnewswire.com/news-releases/keystroke
[2] https://siliconcanals.com/news/
[3] https://150sec.com/fraudulent-fingertips

magnified when dealing with applications that have hundreds of thousands to millions of users. In this context, keystroke biometric algorithms capable of authenticating individuals while interacting with online applications are more necessary than ever. As an example of this, Wikipedia struggles to solve the problem of *'edit wars'* that happens when different groups of editors represent opposing opinions. According to [Yasseri *et al.*, 2012], up to 12% of the discussions in Wikipedia are devoted to revert changes and vandalism, suggesting that the Wikipedia criteria to identify and resolve controversial articles is highly contentious. Large scale keystroke biometrics algorithms could be used to detect these malicious editors among the thousands of editors who write articles in Wikipedia every day. Other applications of keystroke biometric technologies are found in e-learning platforms; student identity fraud and cheating are some challenges that virtual education technologies need to address to become a viable alternative to face-to-face education [Hernandez-Ortega *et al.*, 2020a].

1.1.3. Mouse Dynamics

The mouse is a very common device and its usage is ubiquitous in human-computer interfaces. The way we interact using a mouse with a computer conveys biometric information useful for authentication, especially when combined with other biometric modalities. As an example, in [Ahmed and Traore, 2007; Gamboa *et al.*, 2007] researchers explored characteristics obtained from mouse tasks for user recognition. They analyzed 68 global features (e.g., duration, curvature, mean velocity) from mouse dynamics extracted during login sessions. Their results achieve up to 95% authentication accuracy for passwords with 15 digits. Besides, mouse dynamics can be combined with keystroke biometrics for continuous authentication schemes [Sim *et al.*, 2007]. The fusion of both biometric modalities has been shown to outperform significantly each individual modality achieving up to 98% authentication accuracy [Bailey *et al.*, 2014; Mondal and Bours, 2017].

Mouse dynamics are rich in patterns not only useful for user authentication, but also these mouse interactions generate patterns capable of describing neuromotor capacities of the users (e.g., attitude, emotional state, neuromotor, and cognitive abilities). As an example, in [Martín-Albo *et al.*, 2016] the authors applied the Sigma-Lognormal Model based on the Kinematic Theory [Plamondon, 1995] to compress mouse trajectories. They suggested that mouse movements are the result of complex human motor control behaviors that can be decomposed in a sum of primal movements. In addition, in [Chen *et al.*, 2001] the authors studied the relationship between eye gaze position and mouse cursor position on a computer screen during web browsing and suggested that there are regular patterns of eye/mouse movements associated to the motor cortex system. Modeling the user behavior using mouse dynamics is an ongoing challenge with applications in a variety of fields such as security, e-health, bot detection, or education [Carneiro *et al.*, 2015; Chu *et al.*, 2018; Hernandez-Ortega *et al.*, 2020a].

1.1.4. On-line Handwriting

Signature and handwriting have been studied in depth as a biometric trait during the last 30 years [Fierrez and Ortega-Garcia, 2008], especially in the forensic area, where knowing how to distinguish between genuine and forgery handwriting signatures is a key task, traditionally carried out by forensic experts [Found *et al.*, 1994]. Nowadays, with the massive deployment of mobile devices such biometric trait evolves into on-line handwriting (i.e., handwriting performed over touchscreens). In this scenario, the richness and variety of biometrics patterns that can be extracted from on-line handwriting [Vera-Rodriguez *et al.*, 2015] open a huge range of new lines of work in this field. Current lines of work focus on improving the interoperability between devices [Tolosana *et al.*, 2015], the new scenario of writing using the finger over a touchscreen [Blanco-Gonzalo *et al.*, 2017; Tolosana *et al.*, 2017], the detection of signature complexity to improve the system performance [Caruana *et al.*, 2021; Tolosana *et al.*, 2020c], analysing the effect of aging [Tolosana *et al.*, 2019], the application of deep learning techniques for modeling handwriting and signature signals [Tolosana *et al.*, 2018], and modeling the cognitive and neuromotor processes associated with the generation of the handwriting [Ferrer *et al.*, 2016].

1.2. Motivation of the Thesis

During our day-to-day routines we interact with all kinds of devices (e.g., keyboards, smartphones, tablets, mouse). This interaction is rich in patterns associated with our innate neuromotor features. Modeling this interaction through biometric processing techniques can be useful for different applications ranging from security to health. According to this, the motivations of this Thesis aim to answer the following observations:

1.2.1. Modelling Biometric HCI for Security

The first observation comes from the fact that, according to recent studies, about 34% or more smartphone users did not use any form of authentication mechanism on their devices [Cho *et al.*, 2017]. In similar studies, inconvenience is always shown to be one of the main reasons why users do not use any authentication mechanism. In [Harbach *et al.*, 2014], researchers show that mobile device users spent up to 9% of the time they use their smartphone on unlocking their screens, and the 2018 Meeker Report indicated that the average smartphone user checks his/her device 150 times per day. Those factors lead individuals to make less security conscious decisions like leaving their smartphones unprotected or just protecting them using simple to break authentication mechanisms (e.g., simple Google unlock graphical patterns vulnerable to over-the-shoulder attacks [Martinez-Diaz *et al.*, 2016]).

The second observation is strongly related to the first one. This lack of security conscious of the users with their devices makes mobile web hazards to grow very fast as well. Malicious malware is also adapting to this new mobile era. Mobile bots employ the capacities of smartphones affecting multiples types of online services, such as: social media (e.g., mobile bots accounts

propagate fake twitter messages [Chu *et al.*, 2010]), ticketing/travel, e-commerce, finance, gambling, ATO/Fraud, DDoS attacks, and price scrapping among others. These mobile bots use cellular networks by connecting through cellular gateways[1]. Mobile bots can perform highly advanced attacks while remaining hidden in plain sight. In addition, they are very unlikely to be detected by IP address blocking and more than 5.8% of all mobile devices on cellular networks are used in malicious bot attacks. In other study[2], researchers reveal that mobile fraud reached 150 million global attacks in the first half of 2018 with attack rates rising 24% year-over-year.

The third observation is motivated by the fast proliferation of bot attacks, not only in mobile devices but also in desktop computer interfaces. As an example, bots are expected to be responsible for more than 40% of the web traffic with more than 43% of all login attempts to come from malicious botnets in the next years[3]. Malicious bots cause billionaire loses through web scraping, account takeover, account creation, credit card fraud, denial of service attacks, denial of inventory, and many others. Moreover, bots are used to influence and divide society (e.g., usage of bots to interfere during Brexit voting day [Gorodnichenko *et al.*, 2018], or to spread anxiety and sadness during the COVID-19 outbreak[4,5] through Twitter). Bots are becoming more and more sophisticated, being able to mimic human online behaviors. On the other hand, even though algorithms to distinguish between humans and bots commonly named as CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) are also getting very complex (e.g., Google's ReCAPTCHA), they present limitations. First of all, ensuring a very accurate bot detection makes the tasks difficult to perform even for humans. Second, most of the CAPTCHA systems can be easily solved by the most modern machine learning techniques. For example, the text-based CAPTCHA was defeated by [Bursztein *et al.*, 2011] with 98% accuracy using a ML-based system to segment and recognize the text. In [Bock *et al.*, 2017], the authors designed an AI-based system called unCAPTCHA to break Google's most challenging audio reCAPTCHAs. Third, these algorithms process sensitive information and there are important concerns about how they comply with new regulations such as the European GDPR[6]. Fourth, the CAPTCHA systems become a great barrier to people with visual or other impairments. Finally, the CATPCHA algorithms were originally designed as a task in which machines had to prove they were human, meanwhile in current CAPTCHA systems humans have to prove they are not machines (e.g., *I'm not a robot* from Google's). This means that the responsibility to prove the user's humanity falls over human users instead of bots. At this point, there is still a large room for improvement towards reliable bot detection able to stop malicious software not bothering human users during natural web browsing.

[1] https://resources.distilnetworks.com/reports/mobile-bots-the-next-evolution-of-bad-bots/

[2] https://www.businesswire.com/news

[3] https://resources.distilnetworks.com/white-paper-reports/bad-bot-report-2019

[4] https://www.washingtonpost.com/science/2020/03/17/analysis-millions-coronavirus-tweets-shows-whole-world-is-sad/

[5] https://www.sciencealert.com/bots-are-causing-anxiety-by-spreading-coronavirus-misinformation

[6] https://complianz.io/google-recaptcha-and-the-gdpr-a-possible-conflict/

1.2.2. Modelling Biometric HCI for Health & Behaviour Applications

The fourth observation comes from the fact that, up to date the process to assess the progression of Parkinson's disease (PD) is based on different clinical tests such as the MDS-UPDRS and H&Y scales, which determines a score according to the level of PD. These scales are rather subjective and usually limited to evaluate only upper limb motor skills [Rosenblum *et al.*, 2013; Smits *et al.*, 2014]. Moreover, small variations in the progression are unnoticed thorough these methods with diagnosis errors over 25%, making more difficult the monitoring of the disease. On-line handwriting analysis offers the possibility to diagnosis and monitor the PD progression by analyzing fine motor skills exerted during handwriting tasks, that are not perceptible with traditional scales. The impairment of these fine motor skill induce symptoms such as micrographia (abnormally small letter size), tremor, rigidity, and postural instability [Thomas *et al.*, 2017]. As an example, 5% of the patients manifest micrographia before onset of other symptoms, and up to 30% of those patients report later a worsening in their handwriting skills [McLennan *et al.*, 1972]. Finally, on-line handwriting offers many advantages: they are simple, less intrusive, natural, do not need specialized infrastructure and can be administered remotely.

The last observation comes from the fact that the age is a key attribute in user profiling with direct application on several automatic systems (e.g., parental control, recommender systems, advertising). The most popular way to know the age of the user is by using online questionnaires in which the user directly reveals his age. However, this solution assumes: *i)* honesty on the response of the users, and *ii)* that users can read. Both assumptions cannot be guaranteed because of many practical reasons. Besides the fact that people lie, nowadays children start to use digital platforms and services before learning to read. In the existing literature, there are many experiments exploring the use of technology by children, seeking how to improve the design of adapted interfaces and applications [McKnight and Cassidy, 2012; Tolosana *et al.*, 2021b]. However, modelling and characterising mathematically how children interact with touch devices and how their conduct differs from the adult's one is a field that has not been studied deeply enough.

1.3. The Thesis and Main Contributions

The research works carried out in this Thesis can be stated as follows:

The latest advances in biometric processing technologies, along with the fast technological revolution and massive deployment of mobile devices allow the development of new applications based on Human-Computer Interaction modeling. From this wide HCI area, the research focus of this Thesis has been in the exploration and proposal of new applications in biometric behavioral modeling applied to: i) multimodal biometric user authentication based on mobile sensors and keystroke patterns; ii) modeling neuromotor skills for age detection and PD characterization; and iii) proposing and studying a new generation of bot detector methods based on both mobile and desktop biometric Human-Computer Interactions.

The **main contributions** of this Thesis are:

- *User authentication applications:* we have performed a complete analysis of how discriminative are behavior-based signals obtained from the smartphone sensors. The main aim is to evaluate these signals for person recognition. The recognition based on these signals increases the security of devices, but also implies privacy concerns. We consider seven different data channels and their combinations. Touch dynamics (touch gestures and keystroke), accelerometer, gyroscope, Wi-Fi, GPS location and app usage are all collected during human-mobile interaction to authenticate the users. We evaluate two approaches: one-time authentication and active authentication. In one-time authentication, we employ the information of all channels available during one session. For active authentication we take advantage of mobile user behavior across multiple sessions by updating a confidence value of the authentication score. Our experiments are conducted on the semi-uncontrolled UMDAA-02 database. This database comprises of smartphone sensor signals acquired during natural human-mobile interaction. Our results show that different traits can be complementary and multimodal systems clearly increase the performance with accuracies ranging from 82.2% to 97.1% depending on the authentication scenario. These results confirm the discriminative power of these signals. Moreover, we have studied the performance of Recurrent Neuronal Networks (RNN) for keystroke biometric authentication at large scale in free-text scenarios. Our approach achieves state-of-the-art keystroke biometric authentication performance with an Equal Error Rate of 2.2% and 9.2% for physical and touchscreen keyboards, respectively, significantly outperforming previous approaches. Our experiments demonstrate a moderate increase in error when scaling up to 100,000 the number of subjects employed to evaluate our approach, demonstrating the potential of TypeNet to operate at an Internet scale. The research line has led to the following publications: [Acien *et al.*, 2017, 2021b, 2019a, 2020b,c, 2019b].

- *Neuromotor modeling for health and behaviour applications:* we have explored the suitability of the Sigma-Lognormal theory of rapid human movements to model neuromotor skills exerted during HCI for two applications: *i)* user classification into children and adults according to their interaction with touchscreen devices, and *ii)* we have evaluated the usefulness of on-line handwriting patterns as potential biomarkers to model PD by combining three feature sets extracted from the handwriting signals: neuromotor, global, and nonlinear dynamic features. Regarding the field of children, we have proposed an active detection approach aimed to continuously monitoring the neuromotor user skills by combining two set of features derived from Sigma-Lognormal model and global ones. These feature sets characterize the undeveloped neuromotor skills in children trough touchscreen interaction, differentiating them from the total maturity of neuromotor skills in adults. The experimentation is conducted on a publicly available database with samples obtained from 89 children between 3 and 6 years old and 30 adults. We have used Support Vector Machine algorithm (SVM) to classify the resulting features into age groups. The experiments include single sensor and multi sensor scenarios and fusion scores with temporal features using data from various smartphones and tablets. The results, with correct classification rates over 96%, show the discriminative ability of the proposed neuromotor-inspired features to classify age groups according to the interaction with touchscreen devices. In active detection, our method is able to identify a child in only 3 gestures in average. Then, for the second application based on PD characterization we have employed one of the largest handwriting database in PD with a total of 935 handwriting tasks collected from 55 PD patients and 94 healthy controls (45 young and 49 old). Different classifiers are used to discriminate between PD and healthy subjects: Support Vector Machines (SVM), K-Nearest Neighbors (kNN), and a Multi-Layer Perceptron (MLP). Our proposed approach have achieved remarkable performance with classification results between 81% and 97% of accuracy. The research line has led to the following publications: [Acien *et al.*, 2018; Castrillon *et al.*, 2019; Hernandez-Ortega *et al.*, 2017].

- *Bot detection application:* With the knowledge acquired during the development of previous security and neuromotor analysis applications, we propose a new security application based on the neuromotor analysis of behavioral biometrics: BeCAPTCHA. Specifically, we have studied the suitability of behavioral biometrics to distinguish between computers and humans, commonly named as bot detection. To do that, we have presented BeCAPTCHA-Mouse, a bot detector based on: *i)* a neuromotor model of mouse dynamics to obtain a novel feature set for the classification of human and bot samples; and *ii)* a learning framework involving real and synthetically generated mouse trajectories. We propose two new mouse trajectory synthesis methods for generating realistic data: *i)* a knowledge-based method based on heuristic functions, and *ii)* a data-driven method based on Generative Adversarial Networks (GANs) in which a Generator synthesizes human-like trajectories from a Gaussian noise input. Experiments are conducted on a new testbed also intro-

duced here and available in GitHub: BeCAPTCHA-Mouse Benchmark; useful for research in bot detection and other mouse-based HCI applications. Our benchmark data consists of 15,000 mouse trajectories including real data from 58 users and bot data with various levels of realism. Our experiments show that BeCAPTCHA-Mouse is able to detect bot trajectories of high realism with 93% of accuracy in average using only one mouse trajectory. When our approach is fused with state-of-the-art mouse dynamic features, the bot detection accuracy increases relatively by more than 36%, proving that mouse-based bot detection is a fast, easy, and reliable tool to complement traditional CAPTCHA systems. Moreover, we have adapted the BeCAPTCHA method for mobile scenarios. The heterogeneous flow of data generated during the interaction with the smartphones can be used also to model human behavior when interacting with mobile devices and improve mobile bot detection algorithms. For this, we have proposed BeCAPTCHA-Mobile, a CAPTCHA method based on the analysis of the touchscreen information obtained during a single drag and drop task in combination with the accelerometer data. The goal of BeCAPTCHA-Mobile is to determine whether the drag and drop task was realized by a human or a bot. We evaluate the method by generating fake samples synthesized with Generative Adversarial Neural Networks and handcrafted methods. Our results suggest the potential of mobile sensors to characterize the human behavior and develop a new generation of CAPTCHAs. The experiments are evaluated with HuMIdb (Human Mobile Interaction database), a novel multimodal mobile database captured in this Thesis that comprises 14 mobile sensors acquired from 600 users. HuMIdb is freely available to the research community. The research line has led to the following publications: [Acien *et al.*, 2020a, 2021a].

1.4. Outline of the Dissertation

This Thesis is divided into five main parts (see Fig. 1.2 for details). *Part I* is composed by two chapters; Chapter 1 presents the problem statement and main contributions and Chapter 2 introduces the main materials and methods employed along the Dissertation. There are three experimental parts: *Part II*, *Part III*, and *Part IV*. *Part II* is focused on behavioural biometrics for security applications, in particular keystroke recognition in Chapter 3 and mobile user authentication in Chapter 4. On the other hand, *Part III* presents the most relevant HCI applications develop during the Thesis that take advantage of the neuromotor analysis of these behavioural biometrics signals, such as age detection in Chapter 5 and Parkinson characterization in Chapter 6. In *Part IV* we combine the research work carried out in the previous parts to develop a new security application in the bot detection field based on the neuromotor analysis (Chapter 7). Lastly, *Part IV* concludes the Dissertation.

- *Part I: Problem Statement and Contributions*
  - Chapter 1 first makes a general outlook of behavioural biometrics for modeling

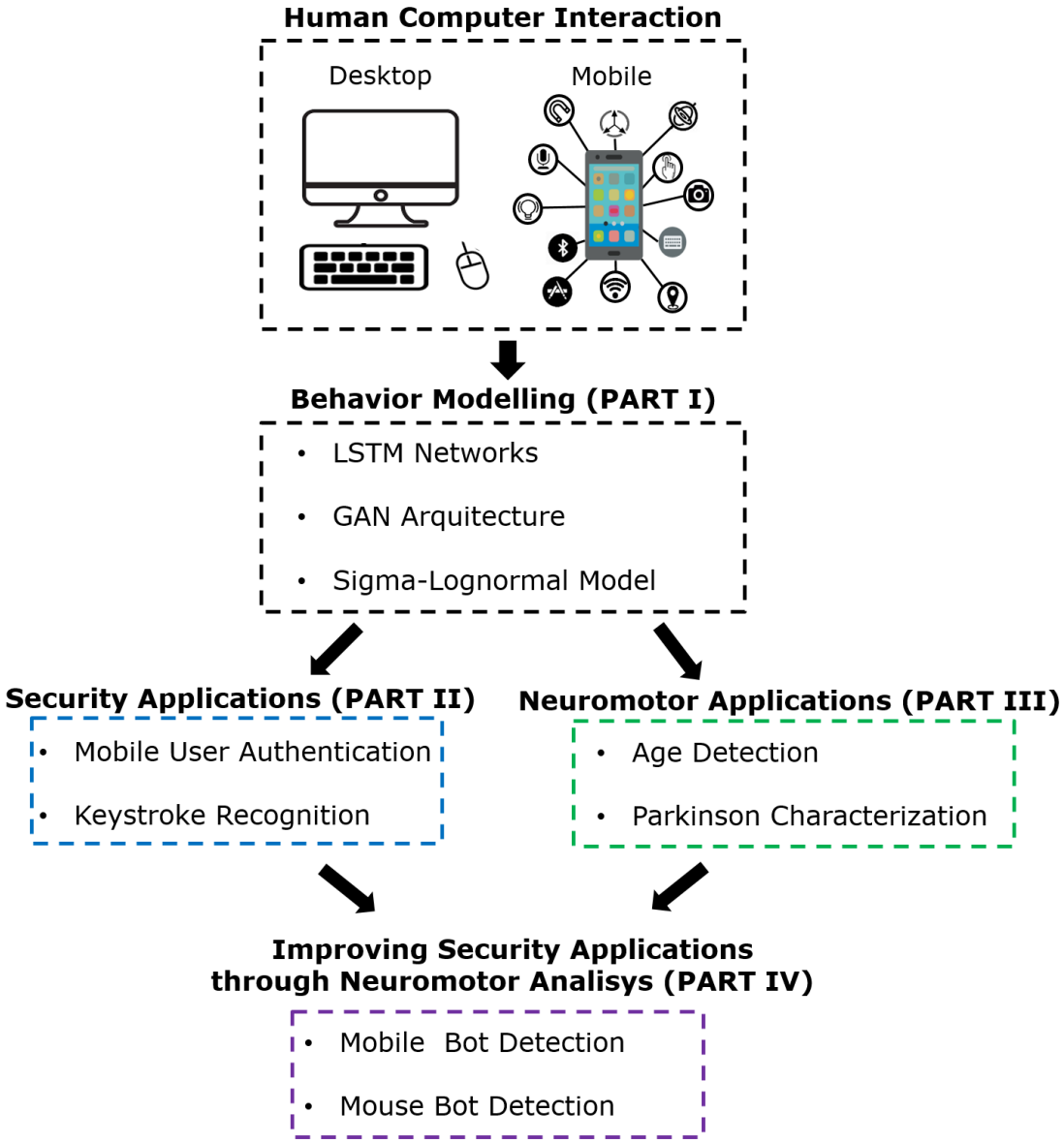

Figure 1.2: *Blocks diagram of the Thesis.*

Human-Computer Interaction, taking into special consideration those biometrics traits employed in this Thesis: smartphone biometrics, mouse dynamics, keystroke biometrics, and on-line handwriting. We finished the chapter by stating the Thesis, giving an outline of the Dissertation, and summarising the research contributions originated from this work.

- Chapter 2 introduces the databases, methods and deep architectures employed in the experimental works of this Dissertation.

- *Part II: Modelling Biometric Device Interaction for Security Applications*

  - Chapter 3 explores keystroke biometrics authentication for two scenarios: *fixed-text*

where we study factors affecting the performance of keystroke authentication systems in which the users employ a proprietary password to authenticate, and *free-text* where we present TypeNet, a Recurrent Neural Network (RNN) for user authentication at large scale trained with a moderate number of keystrokes per identity and evaluated with different learning approaches depending on the loss function, number of gallery samples, length of the keystroke sequences, and device type.

- Chapter 4 provides a taxonomy of applications that can exploit the biometric signals originated by mobile sensors in three different dimensions, depending on the main information content embedded in the signal or signals exploited in the application: neuromotor skills, cognitive functions, and behaviors/routines. We also develop two biometric authentication systems: one based on simple linear touch gestures using a Siamese Recurrent Neural Network architecture, and a second based on the combination of seven different data channels: touch dynamics (touch gestures and keystroke), accelerometer, gyroscope, WiFi, GPS location and app usage that are all collected during HCI.

- *Part III: Modelling Biometrics Device Interaction for Health & Behaviour Applications through Neuromotor Analysis*

  - Chapter 5 studies user classification into children and adults during their interactions with touchscreen devices. We propose two approaches: *i)* one time detection approach in which the classification is performed by employing only one touch gesture; and *ii)* active detection approach aimed to continuously monitor the neuromotor user skills in order to detect a change in the user's profile as soon as possible, employing the minimum number of touch gestures possible.
  - Chapter 6 explores a new set of handwriting features as potential biomarkers to model Parkinson Disease (PD). For this, we employ a novel database with data acquired from PD patients and healthy control (HC) subjects during on-line handwriting tasks distributed in a 3 years time span. The experiments carried out involve up to three different feature sets specifically designed for this task and three different classifiers.

- *Part IV: Improving Security Applications through Neuromotor Analysis*

  - Chapter 7 studies the suitability of a new generation of CAPTCHA algorithms based on human-computer interactions named BeCAPTCHA. We propose two different methods: *i)* BeCAPTCHA-Mobile that exploits mobile sensor signals to develop a mobile bot detector; and *ii)* BeCAPTCHA-Mouse designed for mouse trajectories in desktop computers.

- *Part V: Conclusions*

  - Chapter 8 concludes the Thesis summarising the main results obtained and outlining future research lines.

1.5. Detailed Research Contributions

The research contributions achieved in this Thesis are depicted as follows (journal publications are in bold):

- SECURITY APPLICATIONS.

  - **A. Acien, A. Morales, J. V. Monaco, R. Vera-Rodriguez and J. Fierrez, "TypeNet: Deep Learning Keystroke Biometrics", *IEEE Transactions on Biometrics, Behavior, and Identity Science (TBIOM)* (minor revisions).**
  - **A. Morales, J. Fierrez, A. Acien and R. Tolosana, "SetMargin Loss applied to Deep Keystroke Biometrics with Circle Packing Interpretation", *Pattern Recognition* (major revisions).**
  - A. Acien, J. V. Monaco, A. Morales, R. Vera-Rodriguez and J. Fierrez, "TypeNet: Scaling up Keystroke Biometrics", *in Proc. of the IEEE/IAPR Intl. Joint Conf. on Biometrics (IJCB)*, 2020.
  - A. Acien, A. Morales, R. Vera-Rodriguez and J. Fierrez, "Mobile Active Authentication based on Multiple Biometric and Behavioral Patterns", T. Bourlai and P. Karampelas and V.M. Patel (Eds.), *Securing Social Identity in Mobile Platforms, Springer*, pp. 161-177, 2020.
  - A. Acien, A. Morales, R. Vera-Rodriguez and J. Fierrez, "Smartphone Sensors For Modeling Human-Computer Interaction: General Outlook And Research Datasets For User Authentication", *in Proc. of the EEE Intl. Workshop on Consumer Devices and Systems (CDS)*, July 2020.
  - A. Morales, A. Acien, J. Fierrez, J. V. Monaco, R. Tolosana, R. Vera-Rodriguez and J. Ortega-Garcia, "Keystroke Biometrics in Response to Fake News Propagation in a Global Pandemic", *in Proc. of the IEEE Intl. Workshop on Secure Digital Identity Management (SDIM)*, July 2020.
  - M. Santopietro, R. Vera-Rodriguez, R. Guest, A. Morales and A. Acien, "Assessing the Quality of Swipe Interactions for Mobile Biometric Systems", *in Proc. of the IEEE/IAPR Intl. Joint Conf. on Biometrics (IJCB)*, 2020.
  - A. Acien, A. Morales, R. Vera-Rodriguez, J. Fierrez and R. Tolosana, "MultiLock: Mobile Active Authentication based on Multiple Biometric and Behavioral Patterns", *in Proc. of the ACM Intl. Conf. on Multimedia, Workshop on Multimodal Understanding and Learning for Embodied Applications (MULEA)*, pp. 53-59, Nice, France, October 2019.
  - A. Acien, A. Morales, R. Vera-Rodriguez and J. Fierrez, "Keystroke Mobile Authentication: Performance of Long-Term Approaches and Fusion with Behavioral Profiling", *in Proc. Iberian Conf. on Pattern Recognition and Image Analysis (IBPRIA)*, Vol. 11868, pp. 12-24, Madrid, Spain, July 2019.
  - A. Acien, J. Hernandez-Ortega, A. Morales, J. Fierrez, R. Vera-Rodriguez and J. Ortega-Garcia, "On the Analysis of Keystroke Recognition Performance based on Proprietary Passwords", *in Proc. of the 8th International Conference on Pattern Recognition Systems (ICPRS-17)*, pp. 1-6, Madrid, Spain, July 2017.

- NEUROMOTOR ANALYSIS APPLICATIONS.

  - **A. Acien, A. Morales, J. Fierrez, R. Vera-Rodriguez and J. Hernandez-Ortega, "Active Detection of Age Groups Based on Touch Interaction", *IET Biometrics*, Vol. 8, n. 1, pp. 101-108, January 2019.**
  - **A. Acien, A. Morales, J. Fierrez, R. Vera-Rodriguez and O. Delgado-Mohatar, "BeCAPTCHA: Behavioral Bot Detection using Touchscreen and Mobile Sensors benchmarked on HuMIdb", *Engineering Applications of Artificial Intelligence*, Elsevier, 2021.**
  - **A. Acien, A. Morales, J. Fierrez and R. Vera-Rodriguez, "BeCAPTCHA-Mouse: Synthetic Mouse Trajectories and Improved Bot Detection", *Pattern Reconition* (under review).**
  - A. Acien, A. Morales, J. Fierrez, R. Vera-Rodriguez and I. Bartolome, "BeCAPTCHA: Detecting Human Behavior in Smartphone Interaction using Multiple Inbuilt Sensors", *in AAAI Workshop on Artificial for Cyber Security (AICS)*, New York, USA, February 2020.
  - R. Castrillon, A. Acien, J. Orozco-Arroyave, A. Morales, J. Vargas, R. Vera-Rodriguez, J. Fierrez, J. Ortega-Garcia and A. Villegas, "Characterization of the Handwriting Skills as a Biomarker for Parkinson Disease", *in Proc. of the IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019) - Human Health Monitoring Based on Computer Vision*, Lille, France, April 2019.
  - **J. Hernandez-Ortega, A. Morales, J. Fierrez and A. Acien, "Detecting age groups using touch interaction based on neuromotor characteristics", *IET Electronics Letters*, pp. 1-2, September 2017.**

- SPANISH PATENT APPLICATION.

  - BeCAPTCHA (es, P202030066): Método para generar datos de entrenamiento de un módulo detector de bots, módulo detector de bots entrenado a partir de los datos de entrenamiento generados mediante el método y sistema de detección de bots.

Other contributions so far related to the problem developed in this Thesis but not presented in this Dissertation include:

- FORENSIC TOOLS.

  - R. Vera-Rodriguez, J. Fierrez, J. Ortega-Garcia, A. Acien and R. Tolosana, "e-BioSign Tool: Towards Scientific Assessment of Dynamic Signatures under Forensic Conditions", *in Proc. IEEE 7th International Conference on Biometrics: Theory, Applications and Systems, BTAS*, Arlington, Virginia, USA, 2015.

- FACE RECOGNITION.

  - A. Acien, A. Morales, R. Vera-Rodriguez, I. Bartolome and J. Fierrez, "Measuring the Gender and Ethnicity Bias in Deep Models for Face Recognition", *in Proc. of IAPR Iberoamerican Congress on Pattern Recognition (CIARP)*, Springer, pp. 584-593, Madrid, Spain, November 2018.

Chapter 2

Materials and Methods

This chapter describes the main methods, databases and system architectures employed in this Dissertation and is organized as follows: we first present in Sec. 2.1 the databases employed on the different biometric modalities, exposing their weakness and strengths as well as a detailed description of the new database captured for this Dissertation: the Human-Mobile Interaction database (HuMIdb). Then, we introduce in Sec. 2.2 the Sigma-Lognormal model and the Active Authentication (AA) algorithm. Lastly, Sec. 2.3 describes the most important learning architectures used in the experimental framework of this Thesis.

2.1. Databases

In Table 2.1 we summarize all databases employed in this Thesis. For each database, we include information related to the biometrics modalities captured, the devices employed for the acquisition task, the number of participants as well as the best performance achieved in the state-of-the-art works with each database. The databases are divided into two main groups, according to the acquisition scenario: *i)* Desktop scenario that includes sensors and interfaces commonly employed in desktop applications (e.g., physical keyboard, mouse); and *ii)* Mobile scenario that includes sensors and interfaces developed for mobile applications (e.g., smartphones, tablets).

2.1.1. Mouse Database

The human mouse trajectories employed in this Thesis for bot detection in desktop computers (Chapter 7) were extracted from [Shen *et al.*, 2014] database, which is comprised of more than 200K mouse trajectories acquired from 58 participants with 300 sessions per user, each task was repeated twice in each session. The Acquisition of the data from each subject took between 30 days and 90 days. In each repetition, the task consisted of clicking 8 buttons that appeared in the screen sequentially. The buttons were placed in a particular order to generate mouse trajectories with different directions (rightwards, upwards, downwards, and oblique) and different lengths.

We define a mouse trajectory as the mouse displacement that occurs between two click

Scenario	Study	Modality	# Subjects	# Samples/subject	Sample Size	Supervised	Best Acc.
Desktop	Shen *et al.* [2014]	Mouse	58	∼ 3448K	∼ 2 seconds	Yes	92.19%
	Morales *et al.* [2016]	Keystroke	300	20	∼ 15 keys	Yes	95.0%
	Dhakal *et al.* [2018]	Keystroke	160K	15	∼ 70 keys	No	98.8%
Mobile	Vatavu *et al.* [2015b]	Touchscreen	119	∼ 34	90 seconds	Yes	96.5%
	Mahbub *et al.* [2016]	HCI (Tou, Acc, Blu, Cam, Gyr, GPS, Key, Lig, Mag, Press, Prox, Temp, Wi-Fi)	48	∼ 248	∼ 17 seconds	No	97.1%
	Palin *et al.* [2019]	Keystroke	60K	1 ∼ 20	∼ 70 keys	No	94.7%
	Castrillon *et al.* [2019]	On-line Handwriting	149	∼ 6	∼ 5 seconds	Yes	97.0%
	Acien *et al.* [2021a]	HCI (Tou, Acc, Blu, Gra, GPS, Gyr, Key, LAc, Lig, Mag, Mic, Ori, Prox, Wi-Fi)	600	∼ 500	∼ 10 seconds	No	90.0%

Table 2.1: *Summary of all biometric databases employed in this Dissertation. Modalities: Touchscreen (Tou), Accelerometer (Acc), Bluetooth (Blu), Front camera (Cam), Gravity (Gra), Gyroscope (Gyr), GPS, Keystroke (Key), Light sensor (Lig), Linear Accelerometer (LAc), Magnetometer (Mag), Microphone (Mic), Orientation (Ori), Power consumption (Pow), Pressure (Press), Proximity (Prox), Temperature (Temp), Wi-Fi.*

buttons. Therefore, the mouse movement task is composed of 8 mouse trajectories. The raw data recorded during the acquisition process was: the mouse position over the screen ($\{\mathbf{x}, \mathbf{y}\}$ axis position in pixels), the event (movement or click), and timestamp of the event. The experiments presented in this Dissertation are performed using a subset of the database including 35 samples (randomly chosen) from each of the 58 participants available (more than 5K trajectories in total).

2.1.2. Keystroke KBOC Database

For the analysis of fixed-text keystroke recognition algorithms (Chapter 3), we employ the Keystroke Biometrics Ongoing Competition (KBOC) database [Morales *et al.*, 2016] is composed of keystroke sequences from 300 subjects acquired in 4 different sessions distributed in a 4 months time span. Thus, three different levels of temporal variability were taken into account: *i)* within the same session (the samples are not acquired consecutively), *ii)* within weeks (between two consecutive sessions), and *iii)* within months (between non-consecutive sessions). Each session comprises 4 case-insensitive repetitions of the subject's name and surname (2 in the middle of the session and two at the end) typed in a natural and continuous manner. Note that passwords based on name and surname are very familiar sequences that are typed almost on a daily basis. This allows them to reduce the intra-class variability and to increase the inter-class variability. The database was captured in a university environment, being the vast majority of acquired subjects proficient in the use of computers and keyboards. No mistakes were permitted (i.e., pressing the backspace), if the subject gets it wrong, he/she was asked to

start the sequence again. The names of three other subjects in the database were also captured as forgeries, again with no mistakes permitted when typing the sequence. However, during the acquisition around 4% of samples (equally distributed among genuine and impostors) presented inconsistencies that produced different lengths in the sequences. The use of shift keys can vary the number of keys pressed even if the final result does not change. For example, the sequences $Shift + Shift + a = A$ and the sequences $Shift + a = A$ have different lengths but same text as output. They considered these samples as matching and therefore they are part of the database employed for the competition. The time (in milliseconds) elapsed between key events (press and release) was provided as the keystroke dynamics sequence. Imitations were carried out in a cyclical way (i.e., all the subjects imitate the previous subjects, and the first one imitates the last subjects).

2.1.3. Aalto Keystroke Databases

For free-text keystroke recognition at large scale (Chapter 3) we employ two different keystroke datasets from the Aalto University: *i*) [Dhakal *et al.*, 2018] which comprises more than 5GB of keystroke data collected on desktop keyboards from 168,000 participants; and *ii*) [Palin *et al.*, 2019] dataset which comprises almost 4GB of keystroke data collected on mobile devices from 260,000 participants. The same data collection procedure was followed for both datasets. The acquisition task required subjects to memorize English sentences and then type them as quickly and accurate as they could. The English sentences were selected randomly from a set of 1,525 examples taken from the Enron mobile email and Gigaword Newswire corpus. The example sentences contained a minimum of 3 words and a maximum of 70 characters. Note that the sentences typed by the participants could contain more than 70 characters because each participant could forget or add new characters when typing. All participants in the Dhakal database completed 15 sessions (i.e., one sentence for each session) on either a desktop or a laptop physical keyboard. However, in the Palin dataset the participants who finished at least 15 sessions are only 23% (60,000 participants) out of 260,000 participants that started the typing test.

For the data acquisition, the authors launched an online application that records the keystroke data from participants who visit their webpage and agree to complete the acquisition task (i.e., the data was collected in an uncontrolled environment). Press (keydown) and release (keyup) event timings were recorded in the browser with millisecond resolution using the JavaScript function `Date.now`. The authors also reported demographic statistics for both datasets: 72% of the participants from the Dhakal database took a typing course, 218 countries were involved, and 85% of the them have English as native language, meanwhile only 31% of the participants from the Palin database took a typing course, 163 countries were involved, and 68% of the them were English native speakers.

2.1.4. Touchscreen Database

The touchscreen database used [Vatavu *et al.*, 2015b] for age detection (Chapter 5) is a database with touchscreen activity of both children and adults performing predesigned tasks in an ad-hoc app. The database comprises samples from different guided activities such as tap, double tap and drag-and-drop (swipe) tasks. Swipe activities consist in picking one object on the device screen and moving it to a target area, meanwhile tap activities consist in touching the screen over a target area for a moment. This kind of tasks has been selected because they are simple and common neuromotor tasks as they consist in moving the finger on a surface and also they are widespread gestures in touchscreen device interaction. Multidevice information is available as long as the participants have completed the tasks in both a smartphone and a tablet. The dataset is composed by 89 children between 3 to 6 years old and 30 young adults under 25 years old. The mean age of the children is 4.6 years. The total number of samples is 2,912 for children and 1,157 for adults. To the best of our knowledge, this is the largest database in the field of interaction with touchscreen technology with children under 6 years old. The main issue when acquiring data from children activity is to maintain the kids' attention during a long time period. The authors of the database have adapted the activities interfaces to make the tasks more interesting to children. Thank to this, they have managed to obtain a completion rate near to 100% in tap tasks and above 90% in all types of tasks.

2.1.5. UMDAA-02 Multimodal Database

For mobile user authentication (Chapter 4) we employ UMDAA-02 [Mahbub *et al.*, 2016], a multimodal mobile database that comprises 141 GB of smartphone sensor signals collected from 48 Maryland University students over a period of 2 months. The participants used a smartphone provided by the researchers as their primary device during their daily life (unsupervised scenario). The sensors captured are touchscreen (i.e., touch gestures and keystroke), gyroscope, accelerometer, magnetometer, light sensor, GPS, and Wi-Fi, among others. Information related to mobile user's behavior such as lock and unlock time events, start and end time stamps of calls and app usage are also stored. During a session, the data collection application stored the information provided by the sensors in use. UMDAA-02 contains 10 days of data collection and 248 sessions per participant in average. In each session, the participants spent up to 224 seconds using their smartphones with an average of 5 data sensors captured.

2.1.6. On-line Handwriting Database

The on-line handwriting database used [Castrillon *et al.*, 2019] for Parkinson characterization (Chapter 6) contains a total of 935 handwriting tasks collected from 55 PD patients of 60 years old in average and two groups of healthy participants: one group composed by 49 elderly participants (with ages over 50 years) that we will name EHC (Elder Healthy Control), and a second group of 45 young healthy controls (with ages between 17 and 42 years) namely YHC (Young Healthy Control). They consider this division into young and elder healthy control participants

Sensors	Sampling Rate	Features	Power Consumption
Accelerometer	200 Hz	x, y, z	Low
L.Accelerometer	200 Hz	x, y, z	Low
Gyroscope	200 Hz	x, y, z	Low
Magnetometer	200 Hz	x, y, z	Low
Orientation	NA	L or P	Low
Proximity	NA	cm	Low
Gravity	NA	m/s^2	Low
Light	NA	lux	Low
TouchScreen	E	x, y, p	Medium
Keystroke	E	key, p	Medium
GPS	NA	Lat., Lon., Alt., Bearing, Accuracy	Medium
WiFi	NA	SSID, Level, Info, Channel, Frenquency	High
Bluetooth	NA	SSID, MAC	Medium
Microphone	8 KHz	Audio	High

Table 2.2: *Description of all sensor signals captured in HuMIdb. E = Event-based acquisition, L = Landscape, P = Portrait. The timestamp parameter is captured for all sensors.*

to differentiate between patterns associated to the PD disease and patterns associated to aging. Additionally, the PD participants were asked by their medication, the level of Parkinson (in UPDRS scale) as well as whether they were under their effects at the moment of the acquisition.

In each session the participants were asked to complete 17 different handwriting tasks following a template (e.g., writing words, digits, phrases, drawing figures, and performing a sign). During the acquisition, the handwriting signals were recorded using a commercial tablet Wacom Cintiq (13HD Touch, 180 Hz of sampling frequency), which captures different signals including $\{\mathbf{x}, \mathbf{y}\}$ axis position, pressure, in-air movement, and timestamps.

2.1.7. HuMIdB Database

For this Thesis we captured a novel multimodal mobile database called HuMIdb (the Human Mobile Interaction database), that comprises more than 5 GB from a wide range of mobile sensors acquired under unsupervised scenario. The database includes 14 sensors (described in Table 2.2) during natural human-mobile interaction performed by 600 participants. For the acquisition, we implemented an Android application that collects the sensor signals while the participants complete 8 simple tasks with their own smartphones and without any supervision whatsoever (i.e., the participants could be standing, sitting, walking, indoors, outdoors, at daytime or night, etc.). The acquisition app was launched on Google Play Store and advertised in our research web site and various research mailing lists. After that, the participants were self-selected around the globe producing more varied participants than previous state-of-the-art mobile databases. All data captured in this database have been stored in private servers and anonymized with previous participant consent according to the GDPR (General Data Protection Regulation).

The different tasks are designed to reflect the most common interaction with mobile devices:

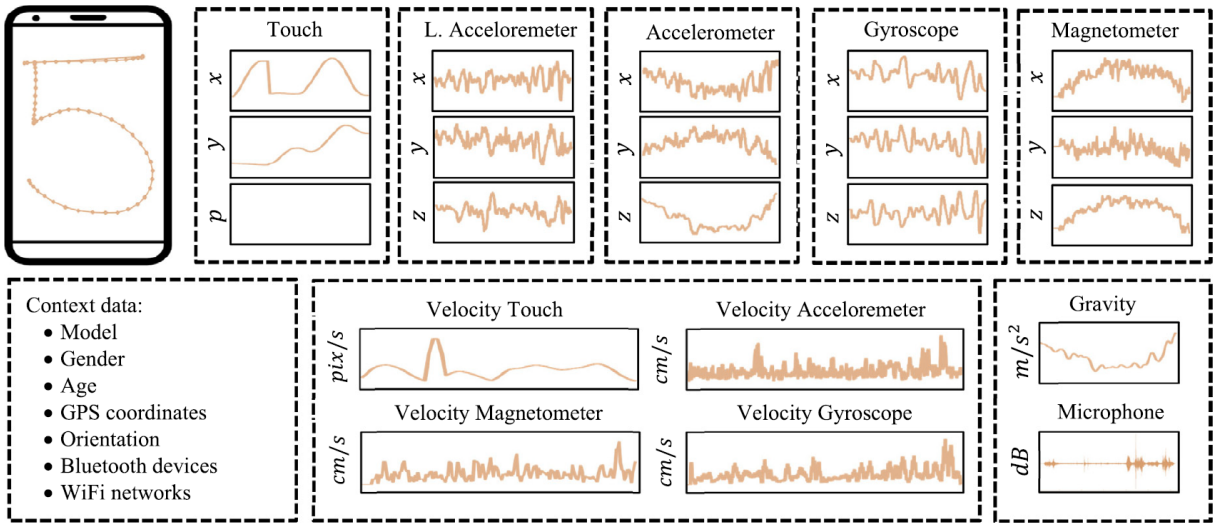

Figure 2.1: *Full set of data generated during one of the HuMIdb task.*

keystroke (name, surname, and a pre-defined sentence), tap (press a sequence of buttons), swipe (up and down directions), air movements (circle and cross gestures in the air), handwriting (digits from 0 to 9), and voice (record the sentence "*I am not a robot*"). Additionally, there is a drag and drop button between tasks.

The acquisition protocol comprises 5 sessions with at least 1-day gap among them. It is important to highlight that in all sessions, the 1-day gap refers to the minimum time between one subject finishes a session and the next time the app allows to have the next session. At the beginning of each task, the app shows a brief pop-up message explaining the procedure to complete each task. The application also captures the orientation (landscape/portrait) of the smartphone, the screen size, resolution, the model of the device, and the date when the session was captured. Regarding the age distribution, 25.6% of the participants were younger than 20 years old, 49.4% are between 20 and 30 years old, 19.2% between 30 and 50 years old, and the remaining 5.8% are older than 50 years old. Regarding the gender, 66.5% of the participants were males, 32.8% females, and 0.7% others. Participants performed the tasks from 14 different countries (52.2%/47.0%/0.8% are European, American, and Asian respectively) using 600 different devices.

Fig. 2.1 shows an example of the handwriting task (for digit "5") and the information collected during the task. Note how a simple task can generate a heterogeneous flow of information related with the user behavior: the way the user holds the device, the power and velocity of the gesture, the place, etc. The richness in number of sensors acquired and population diversity of HuMIdb offer many other research possibilities. In this Thesis we will employ HuMIdb for the development of a new bot detection algorithm for mobile devices in Chapter 7. However, some of the possible research lines to explore beyond bot detection with this dataset include:

- *Demographic modeling*: HuMIdb comprises users from the 4 continents and 14 different countries. The database is diverse in gender and age of the participants.

Parameter	Description
D_i	Input pulse: covered distance
t_{0i}	Initialization time: displacement in the time axis
μ_i	Log-temporal delay
σ_i	Impulse response time of the neuromotor system
θ_{si}	Starting angle of the stroke
θ_{ei}	Ending angle of the stroke

Table 2.3: *Sigma-Lognormal parameters description.*

- *Cross-sensor interoperability*: HuMIdb includes signals from 600 (one per user) different devices, with a total of 230 different smartphone models. Analyzing the impact of different device characteristics on human behavior is a challenging research line.

- *User recognition*: HuMIdb comprises behavioral patterns from 600 users. Continuous authentication based on biometric behavioral patterns is a popular research line with applications in the security market.

2.2. Methods

2.2.1. The Sigma-Lognormal Model

The Sigma-Lognormal model [Fischer and Plamondon, 2017] from the kinematic theory of rapid human movements [Plamondon, 1995] allows to describe and characterize neuromotor-fine hand skills exerted during Human-Computer interactions. This model has been applied successfully in the past to handwriting tasks like handwritten signature [Djioua and Plamondon, 2008; Ferrer *et al.*, 2014]. In this Thesis we will apply the Sigma-Lognormal model to extract neuromotor features from swipe gestures (Chapter 5), on-line handwriting (Chapter 6), and mouse movements (Chapter 7).

The model states that the velocity profile of the human hand movements can be decomposed into primitive strokes with a Lognormal shape that describes well the nature of the hand movements ruled by the motor cortex. The velocity profile of these strokes is modeled as:

$$|\vec{v_i}(t)| = \frac{D_i}{\sqrt{2\pi}\sigma_i (t - t_{0i})} \exp\left(\frac{(\ln(t - t_{0i}) - \mu_i)^2}{-2\sigma_i^2}\right) \tag{2.1}$$

where the parameters are described in Table 2.3. The velocity profile of the entire hand movement is calculated as the sum of all these individual strokes:

$$\vec{v_r}(t) = \sum_{i=1}^{N} \vec{v_i}(t) \tag{2.2}$$

where N is the number of velocity strokes considered in the model. A complex hand movement like swipe gesture or mouse trajectory, is a summation of these lognormals, each one character-

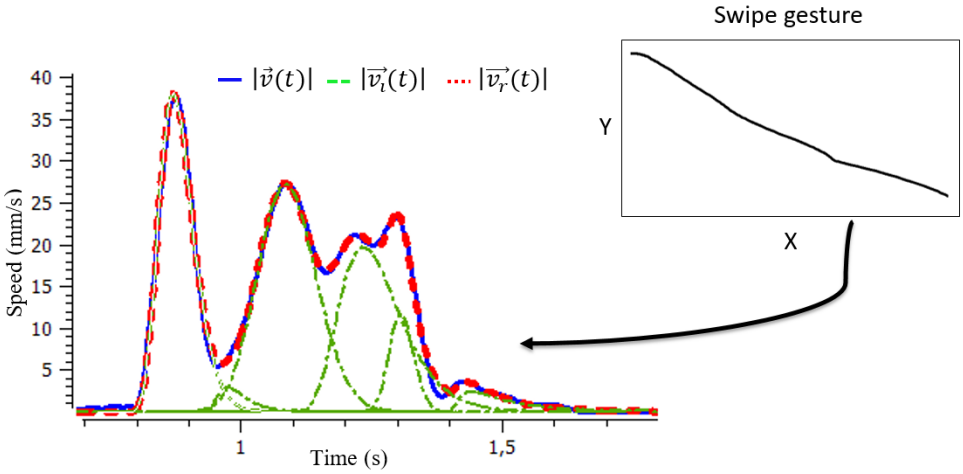

Figure 2.2: *An example of the Lognormal decomposition of a swipe gesture. The blue line is the velocity profile of the swipe gesture provided as input to the Sigma-Lognormal model, which generates as output the lognormal signals (the green dashed lines) extracted from the velocity profile. The red dashed line is the reconstruction of the original velocity profile from the lognormal signals.*

ized by the six parameters in Table 2.3. An example of this is shown in Fig. 2.2 with a swipe gesture, where the blue line is the velocity profile $|\vec{v}\,(t)|$ of the swipe gesture, which is used as the input of the Sigma-Lognormal model. The green dashed lines correspond to the individual lognormal signals $|\vec{v_i}\,(t)|$ generated as in [Fischer and Plamondon, 2017], which describes a method to automatically estimate both N and the parameters in Table 2.3 from an input trajectory $|\vec{v}\,(t)|$. Finally, the red dotted line $|\vec{v_r}\,(t)|$ is the reconstruction of the original velocity profile by summing all these generated individual lognormal signals. We can observe that the reconstructed signal matches almost perfectly with the original velocity profile of the swipe gesture, suggesting the potential of the Sigma-Lognormal model to describe hand movements. Lognormals with a high amplitude are typically observed during the first part of the movement (agonist and antagonist activations), while smaller lognormals occur during the fine correction. The differences in lognormal sizes provide us information about the length of the trajectory (long trajectories have usually larger velocities).

The neuromotor feature set proposed is computed from the six lognormal parameters described in Table 2.3. The swipe gesture N lognormal signals and each lognormal generates those 6 parameters from Table 2.3. For each parameter, we calculate 6 features: maximum, minimum, and mean for both halves of the trajectory. This is done because in natural swipe gestures the lognormal parameters are usually very different between both halves of a given trajectory. Additionally, we added the number of lognormals N that each swipe trajectory generates as an additional feature. This additional feature measures the complexity of the trajectory [Vera-Rodriguez *et al.*, 2019], having many lognormals means that the swipe trajectory has many changes in the velocity profile while few of them usually indicates more basic and soft trajectories (as we will see in Chapter 5). The neuromotor feature set extracted for swipe gestures can be extrapolated with little modifications, as we will see later, to other hand movements such as

mouse trajectories or on-line handwriting tasks.

2.2.2. Active Authentication Algorithm

Active Authentication (AA) refers to those experiments where we take a sequence of events during a period of time to detect a change in the user profile (e.g., the device has been stolen and the impostor user starts to operate with it). In this Thesis we will apply the AA algorithm for continuous mobile authentication (Chapter 4), and continuous monitoring for children detection (Chapter 5). For AA experiments we consider the QCD algorithm (Quickest Change Detection) as explained in [Perera and Patel, 2017a]. The QCD-based algorithm updates a confidence score based on previous events (e.g., smartphone session, swipe gesture, keystroke sequence) by performing a cumulative sum of scores. This cumulative sum will be almost zero if the scores belong to the genuine user, and will grow if an impostor takes the control, until it reaches a certain threshold that would detect the intruder. The cumulative sum is calculated as follow:

$$score_j^{AA} = \max(score_{j-1}^{AA} + L_j, 0) \tag{2.3}$$

where j means the actual event and $score_{j-1}^{AA}$ is the previous cumulative score. L_j is the contribution of the actual event calculated as the log-likelihood ratio between score distributions:

$$L_j = \log(\frac{f_I(score_j)}{f_G(score_j)}) \tag{2.4}$$

where f_G and f_I are the probability distributions of the genuine and impostor scores respectively calculated previously in the OTA (One-Time Authentication) scenario (see Fig. 2.3 left), and $score_j$ is the OTA score of the actual event. According to Eq. 2.4, the log-likelihood ratio L_j will be negative if $score_j$ belongs to a genuine user and positive in the opposite case, and therefore, multiple consecutive events of an impostor will increase the cumulative sum $score_j^{AA}$. Fig. 2.3 depicts an example of $score_j^{AA}$ evolution. At the time the mobile starts to be operated by an intruder (event number sixteen in Fig. 2.3 right) the $score_j^{AA}$ ($j > 16$) will tend to increase until reaching the threshold. The selection of the threshold to calculate when the user is detected as an impostor is crucial in performance terms: a high threshold could decrease the number of false detections (genuine user detected as an impostor), but also it could increase the time delay (time between the impostor user starts to operate the device till he is detected).

In order to choose the best threshold, we will employ ADD (Average Detection Delay), PFD (Probability of False detection) and PND (Probability of Non Detection) curves, previously used in [Perera and Patel, 2016]. ADD curves show the number of samples necessary to detect an impostor as a function of the threshold. On the other hand, PFD curves depict the percentage of false detection. It means that $score_j^{AA}$ reaches the intruder detection threshold during a genuine session sequence, PFD is similar to FMR (False Match Rate) in OTA. Finally, the PND (Probability of Non Detection) curve depicts the percentage of impostor who are not detected by the system. It means that $score_j^{AA}$ does not reach the intruder detection threshold during

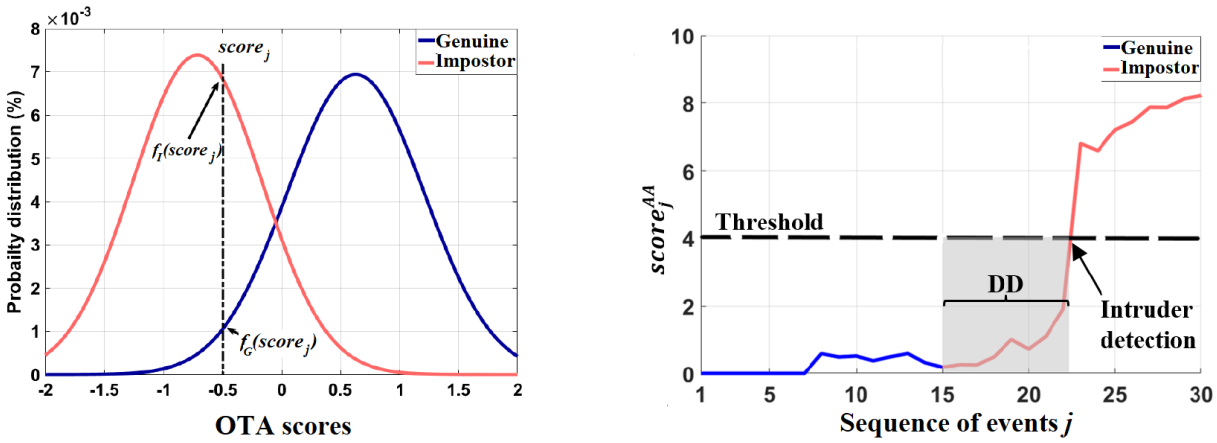

Figure 2.3: *Left: Probability distribution of genuine and impostors scores for OTA scenario. The score* $score_j$ *shows that* $f_I(score_j)$ *is higher than* $f_G(score_j)$ *so the log likelihood ratio* L_j *will be positive. Right: an example of QCD-based curve with a sequence of* 30 *events (*15 *genuine and* 15 *impostors). The dashed line is the intruder detection threshold and the grey box shows the Detection Delay (DD).*

the intruder sessions sequence, PND is similar to FNMR (False Non-Match Rate) in OTA.

2.3. Deep Architectures

This section describes the most important Deep Neuronal Network architectures employed in this Dissertation, as well as their learning frameworks. Owing to the nature of the majority of biometric signals we will employ have a temporal evolution, the Recurrent Neuronal Networks (RNNs) plays an important role thorough the entire Thesis and will be employed several times: in mobile authentication with swipe gestures (Chapter 4), for keystroke recognition (Chapter 3), and bot detection (Chapter 7). Although some implementation details varies from work to work, the main architectures are depicted as follows:

2.3.1. The Recurrent Architecture

RNNs have demonstrated to be one of the best algorithms to deal with temporal data and are well suited for keystroke sequences [Deb *et al.*, 2019; Lu *et al.*, 2019], touchscreen signals [Acien *et al.*, 2021a] or handwriting signatures [Tolosana *et al.*, 2020b, 2021c] among others.

Our standard RNN architecture employed in this Thesis is depicted in Fig. 2.4. It is composed of two Long Short-Term Memory (LSTM) layers of 128 units (*tanh* activation function). Between the LSTM layers, we perform batch normalization and dropout rate of 0.5 to avoid overfitting. Additionally, each LSTM layer has a recurrent dropout rate of 0.2.

One constraint when training a RNN using standard backpropagation through time applied to a batch of sequences is that the number of elements in the time dimension (e.g., number of keystrokes in the keystroke sequence or number of samples in the touchscreen gesture) must be the same for all sequences. We set the size of the time dimension to M. In order to train the model with sequences of different lengths N within a single batch, we truncate the end of

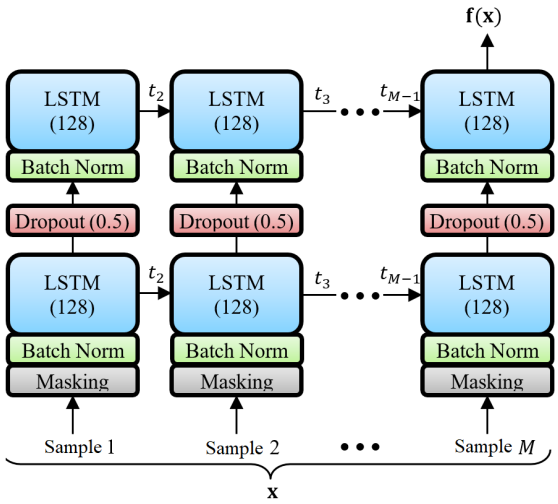

Figure 2.4: *The architecture of the RNNs for temporal sequences. The input* $\mathbf{x}$ *is a time series with shape* $M \times F$ *(# samples* $\times$ *# features) and the output* $\mathbf{f}(\mathbf{x})$ *is an embedding vector with shape* 1×128.

the input sequence when $N > M$ and zero pad at the end when $N < M$, in both cases to the fixed size M. Error gradients are not computed for those zeros and do not contribute to the loss function at the output layer as a result of the masking layer shown in Fig. 2.4. Finally, the output of the model $\mathbf{f}(\mathbf{x})$ is an array of size 1×128 that we will employ later as an embedding feature vector to recognize subjects.

The RNN models can be trained following three different approaches, according to the loss functions employed to train them: *Softmax loss*, which is widely used in classification tasks; *Contrastive loss*, a loss for distance metric learning based on two samples [Hadsell *et al.*, 2006]; and *Triplet loss*, a loss for metric learning based on three samples [Weinberger and Saul, 2009]. These are defined as follows:

- *Softmax loss:* let $\mathbf{x}_i$ be a temporal sequence of individual I_i, and let us introduce a dense layer after the embeddings aimed at classifying the individuals used for learning (see Fig. 2.5.a). The Softmax loss is applied as

$$\mathcal{L}_S = -\log\left(\frac{e^{f_{I_i}^C(\mathbf{x}_i)}}{\sum_{c=1}^{C} e^{f_c^C(\mathbf{x}_i)}}\right) \tag{2.5}$$

  where C is the number of classes used for learning (i.e., identities), $\mathbf{f}^C = [f_1^C, \ldots, f_C^C]$, and after learning all elements of $\mathbf{f}^C$ will tend to 0 except $f_{I_i}^C(\mathbf{x}_i)$ that will tend to 1. Softmax is widely used in classification tasks because it provides good performance on closed-set problems. Nonetheless, Softmax does not optimize the margin between classes. Thus, the performance of this loss function usually decays for problems with high intra-class variance. In order to train the architecture proposed in Fig. 2.4, we have added an output

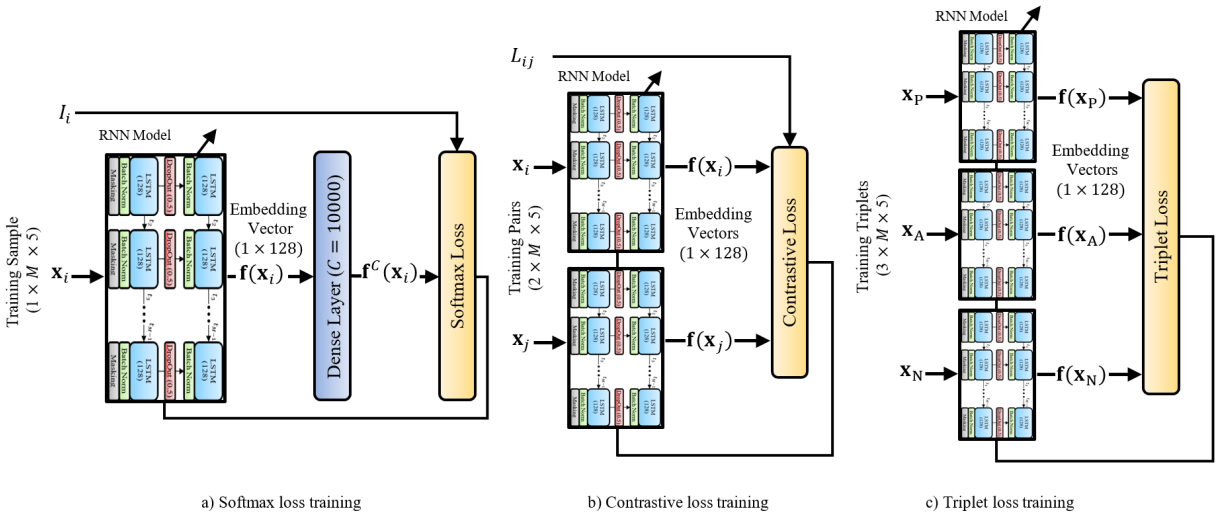

Figure 2.5: *Learning architecture for the different loss functions a) Softmax loss, b) Contrastive loss, and c) Triplet loss. The goal is to find the most discriminant embedding space* $\mathbf{f}(\mathbf{x})$.

classification layer with C units (see Fig. 2.5.a). During the training phase, the model will learn discriminative information from the input sequences and transform this information into an embedding space where the embedding vectors $\mathbf{f}(\mathbf{x})$ (the outputs of the model) will be close in case both inputs sequences belong to the same subject (genuine pairs), and far in the opposite case (impostor pairs).

- *Contrastive loss:* let $\mathbf{x}_i$ and $\mathbf{x}_j$ each be a temporal sequence that together form a pair which is provided as input to the model. The Contrastive loss calculates the Euclidean distance between the model outputs,

$$d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{f}(\mathbf{x}_i) - \mathbf{f}(\mathbf{x}_j)\| \tag{2.6}$$

where $\mathbf{f}(\mathbf{x}_i)$ and $\mathbf{f}(\mathbf{x}_j)$ are the model outputs (embedding vectors) for the inputs $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. The model will learn to make this distance small (close to 0) when the input pair is genuine and large (close to α) for impostor pairs by computing the loss function $\mathcal{L}_{CL}$ defined as follows:

$$\mathcal{L}_{CL} = (1 - L_{ij})\frac{d^2(\mathbf{x}_i, \mathbf{x}_j)}{2} + L_{ij}\frac{\max^2\{0, \alpha - d(\mathbf{x}_i, \mathbf{x}_j)\}}{2} \tag{2.7}$$

where L_{ij} is the label associated with each pair that is set to 0 for genuine pairs and 1 for impostor ones, and $\alpha \geq 0$ is the margin (the maximum margin between genuine and impostor distances). The Contrastive loss is trained using a Siamese architecture (see Fig. 2.5.b) that minimizes the distance between embeddings vectors from the same class ($d(\mathbf{x}_i, \mathbf{x}_j)$ with $L_{ij} = 0$), and maximizes it for embeddings from different class ($d(\mathbf{x}_i, \mathbf{x}_j)$ with $L_{ij} = 1$).

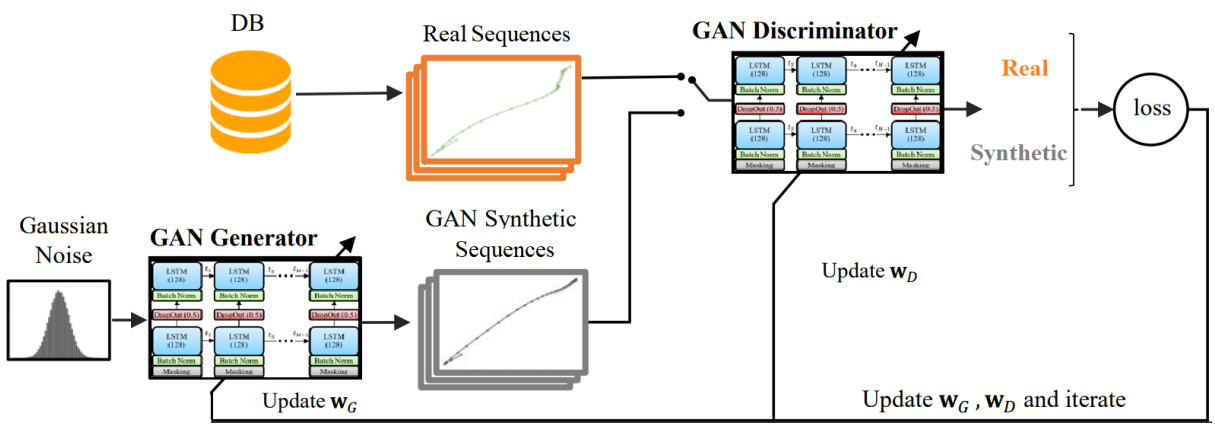

Figure 2.6: *The proposed architecture to train a GAN Generator of synthetic sequences. The Generator learns the features of the real sequences from the database and generate real-like ones from Gaussian Noise. Note that the weights of the Discriminator* $\mathbf{w}_D$ *are trained after the update of the weights of the Generator* $\mathbf{w}_G$*.*

- *Triplet loss:* the *Triplet loss* function enables learning from positive and negative comparisons at the same time (note that the label L_{ij} eliminates one of the distances for each pair in the Contrastive loss). A triplet is composed by three different samples from two different classes: Anchor (A) and Positive (P) are different sequences from the same subject, and Negative (N) is a sequence from a different subject. The Triplet loss function is defined as follows:

$$\mathcal{L}_{TL} = \max\left\{0, d^2(\mathbf{x}_{\mathrm{A}}^i, \mathbf{x}_{\mathrm{P}}^i) - d^2(\mathbf{x}_{\mathrm{A}}^i, \mathbf{x}_{\mathrm{N}}^j) + \alpha\right\} \tag{2.8}$$

  where α is a margin between positive and negative pairs and d is the Euclidean distance calculated with Eq. 2.6. In comparison with Contrastive loss, Triplet loss is capable of learning intra- and inter-class structures in a unique operation (removing the label L_{ij}). The Triplet loss is trained using an extension of a Siamese architecture (see Fig. 2.5.c) for three samples. This learning process minimizes the distance between embedding vectors from the same class ($d(\mathbf{x}_{\mathrm{A}}, \mathbf{x}_{\mathrm{P}})$), and maximizes it for embeddings from different classes ($d(\mathbf{x}_{\mathrm{A}}, \mathbf{x}_{\mathrm{N}})$).

2.3.2. The GAN Architecture

The GAN (Generative Adversarial Network) architecture is composed by two neuronal networks, commonly named Generator (defined by its parameters $\mathbf{w}_G$) and Discriminator (defined by its parameters $\mathbf{w}_D$), that are trained one against the other. The architecture of the GAN is depicted in Fig. 2.6 and will be employed in Chapter 7 to generate synthetic mouse trajectories and synthetic swipes gestures. The aim of the Generator is to fool the Discriminator by generating fake sequences very similar to the real ones. The sampling rate is determined by the real sequence used in the learning framework. Therefore, the synthesized sequences are generated

with the same sampling rate. Other frequencies can be obtained subsampling the generated ones or re-training the GAN for a different sampling rate. The input of the Generator consist of a seed vector of R random numbers. The output of the Generator are two coordinate vectors $\{\hat{\mathbf{x}}, \hat{\mathbf{y}}\}$ (i.e., the coordinates of the synthetic swipe gesture or mouse trajectory generated) with length equal to M (M can be fixed to generate different sequence lengths). The input of the Discriminator consists of a batch including two types of sequence: 1) *Synthetic*: sequences generated by the Generator ($\{\hat{\mathbf{x}}, \hat{\mathbf{y}}\}$); 2) *Real*: sequences chosen randomly from the database $\{\mathbf{x}, \mathbf{y}\}$. The aim of the Discriminator is to predict whether the sequences comes from the real set or is a fake created by the Generator. During the training phase, the GAN architecture will improve the ability of the Generator to fool the Discriminator. This architecture turns latent space points defined by the random seed into a classification decision: '*Synthetic*' (from the Generator) or '*Human*'. This learning process is guided by the real sequences.

The topology employed in the Discriminator consist of two LSTM (Long Short-Term Memory) layers (with 128 and 64 units respectively, with '*LeakyReLU*' activation) followed by a dense layer (with 1 unit and '*Sigmoid*' activation), very similar to the RNN model depicted previously in Sec. 2.3.1. The dense layer of the Discriminator is used as a classification layer to distinguish between fake and real sequences ('*Binary Cross-Entropy*' loss function). For the Generator, we employ two LSTM layers (with 128 and 64 units respectively, with '*ReLU*' activation) followed by a dense layer with '*TimeDistributed*' activation.

Part II

Modelling Biometric Device Interaction for Security Applications

Chapter 3

User Authentication based on Keystroke Biometrics

We dedicate this chapter to go deep on the analysis of keystroke biometrics for the two scenarios: *fixed-text*, where the keystroke sequence typed by the subject is prefixed, such as a username or password, and *free-text*, where the keystroke sequence is arbitrary, such as writing an email or transcribing a sentence with typing error.

For fixed-text scenario, we study factors affecting the keystroke recognition performance with proprietary passwords. Despite the great efforts made during the last decades, the performance of keystroke recognition systems is far from the performance achieved by traditional hard biometrics. This is very pronounced for some users, who generate many recognition errors even with the most sophisticate recognition algorithms. Our purpose here is to study factors affecting the performance of users for approaches in which each user employ a proprietary password based on familiar information to authenticate.

For free-text scenarios we present TypeNet, a new keystroke biometric authentication algorithm based on Long Short-Term Memory (LSTM) networks to authenticate users at large scale. The literature on free-text keystroke biometrics is extensive, but to the best of our knowledge, previous systems have only been evaluated with up to several hundred subjects and cannot deal with the recent challenges that massive usage applications are facing. TypeNet outperforms previous state-of-the-art keystroke biometric authentication approaches when scaling the number of users to authenticate, demonstrating the potential of TypeNet to operate at large scale.

The chapter is organized as follows: we first summarize in Sec. 3.1 related works in keystroke dynamics. Then, we present in Sec. 3.2 the results for the fixed-text study and the conclusions achieved. Finally, we introduce TypeNet in Sec. 3.3 and evaluates its performance for different devices and set-ups.

Study	Scenario	#Subjects	#Seq.	Sequence Size	#Keys	Best Performance
Monrose and Rubin [1997]	Desktop	31	N/A	N/A	N/A	Acc. = 23%
Gunetti and Picardi [2005]	Desktop	205	$1 \sim 15$	$700 \sim 900$ keys	688K	EER = 7.33%
Gascon *et al.* [2014]	Mobile	315	$1 \sim 10$	~ 160 keys	67K	EER= 10.0%
Ceker and Upadhyaya [2016]	Desktop	34	2	$\sim$ 7K keys	442K	EER = 2.94%
Morales *et al.* [2016]	Desktop	300	20	$\sim$ 15K keys	90K	EER = 4.68%
Murphy *et al.* [2017]	Desktop	103	N/A	1,000 keys	12.9M	EER = 10.36%
Monaco and Tappert [2018]	Both	55	6	500 keys	165K	EER = 0.6%
Lu *et al.* [2019]	Desktop	75	3	$\sim$ 5,700 keys	1,2M	EER = 3.04%
Deb *et al.* [2019]	Mobile	37	180K	3 seconds	6.7M	81.61% TAR at 0.1% FAR
Kim and Kang [2020]	Mobile	50	20	$\sim$ 200 keys	200K	EER = 0.05%
Ours (2020)	Both	**228**K	**15**	**$\sim$ 70 keys**	**199M**	**EER= 2.2**%

Table 3.1: *Comparison among different keystroke datasets employed in relevant related works. N/A = Not Available. Acc = Accuracy, EER = Equal Error Rate, TAR = True Acceptance Rate, FAR = False Acceptance Rate.*

3.1. State-of-the-art on Keystroke Authentication

The measurement of keystroke dynamics depends on the acquisition of key press and release events. This can occur on almost any commodity device that supports text entry, including desktop and laptop computers, mobile and touchscreen devices that implement soft (virtual) keyboards, and PIN entry devices such as those used to process credit card transactions. Generally, each keystroke (the action of pressing and releasing a single key) results in a keydown event followed by keyup event, and the sequence of these timings is used to characterize an individual's keystroke dynamics. Within a web browser, the acquisition of keydown and keyup event timings requires no special permissions, enabling the deployment of keystroke biometric systems across the Internet in a transparent manner.

Biometric authentication algorithms based on keystroke dynamics for desktop and laptop keyboards have been predominantly studied in fixed-text scenarios where accuracies higher than 95% are common [Morales *et al.*, 2016]. Approaches based on sample alignment (e.g., Dynamic Time Warping) [Morales *et al.*, 2016], Manhattan distances [Monaco, 2016], digraphs [Bergadano *et al.*, 2002], and statistical models (e.g., HMMs) [Ali *et al.*, 2016] have shown to achieve the best results in fixed-text.

Nevertheless, the performances of free-text algorithms are generally far from those reached in the fixed-text scenario, where the complexity and variability of the text entry contribute to intra-subject variations in behavior, challenging the ability to recognize subjects [Sim and Janakiraman, 2007]. In [Monrose and Rubin, 1997], authors proposed a free-text keystroke algorithm based on subject profiling by using the mean latency and standard deviation of digraphs and computing the Euclidean distance between each test sample and the reference profile. Their results worsened from 90% to 23% of correct classification rates when they changed both subject

profiles and test samples from fixed-text to free-text. In [Gunetti and Picardi, 2005], the authors extended the previous algorithm to n-graphs. They calculated the duration of n-graphs common between training and testing and defined a distance function based on the duration and order of such n-graphs. Their results of 7.33% classification error outperformed the previous state of the art. Nevertheless, their algorithm needs long keystroke sequences (between 700 and 900 keystrokes) and many keystroke sequences (up to 14) to build the subject profile, which limits the usability of that approach. More recently, in [Murphy *et al.*, 2017] the authors collected a very large free-text keystroke dataset ($\sim$ 2.9M keystrokes) and applied the Gunetti and Picardi algorithm achieving 10.36% classification error using sequences of 1,000 keystrokes and 10 genuine sequences to authenticate subjects. The effect of the data size on the performance of free-text keystroke algorithms has been studied by [Huang *et al.*, 2015]. Their results suggested that a sample size of 10,000 keystrokes for the reference profile and 1,000 keystrokes for the test sample are needed to achieve good authentication performance for those algorithms based on n-graph features. The main drawback when using large keystroke sequences was that the subject needed on average six minutes of typing to generate a valid sample. Finally, in [Ayotte *et al.*, 2020] the authors implemented a new metric based on Random Forest classifier to select the best features for keystroke recognition when using digraph algorithms. Their results on the Clarkson II ([Murphy *et al.*, 2017]) dataset achieved a 7.8% EER with 200 digraphs, demonstrating the potential of such algorithms with an appropriate selection of the keystroke features.

More recently than the pioneering works of Monrose and Gunetti, some algorithms based on statistical models have shown to work very well with free-text, like the Partially Observable Hidden Markov Model (POHMM) [Monaco and Tappert, 2018]. This algorithm is an extension of the traditional HMM, but with the difference that each hidden state is conditioned on an independent Markov chain. This algorithm is motivated by the idea that keystroke timings depend both on past events and the particular key that was pressed. Performance achieved using this approach in free-text is close to fixed-text, but it again requires several hundred keystrokes and has only been evaluated with a database containing less than 100 subjects.

Unlike physical keyboards, touchscreen keyboards support a variety of input methods, such as swipe which enables text entry by sliding the finger along a path that visits each letter and lifting the finger only between words. The ability to enter text in ways other than physical key pressing has led to a greater variety of text entry strategies employed by typists [Palin *et al.*, 2019]. In addition to this, mobile devices are readily equipped with additional sensors which offer more insight to a users keystroke dynamics. This includes the touchscreen itself, which is able to sense the location and pressure, as well as accelerometer, gyroscope, and orientation sensors.

Like desktop keystroke biometrics, many mobile keystroke biometric studies have focused on fixed-text sequences [Teh *et al.*, 2016]. Some recent works have considered free-text sequences on mobile devices. In [Gascon *et al.*, 2014], the authors collected freely typed samples from over 300 participants and developed a system that achieved a True Acceptance Rate (TAR) of 92% at 1% False Acceptance Rate (FAR) (an EER of about 10%). Their system utilized

accelerometer, gyroscope, time, and orientation features. Each user typed an English pangram (sentence containing every letter of the alphabet) approximately 160 characters in length, and classification was performed by Support Vector Machine (SVM). In [Kim and Kang, 2020], the authors utilized microbehavioral features to obtain an EER below 0.05% for 50 subjects with a single reference sample of approximately 200 keystrokes for both English and Korean input. The microbehavioral features consist of angular velocities along three axes when each key is pressed and released, as well as timing features and the coordinate of the touch event within each key.

Because mobile devices are not stationary, mobile keystroke biometrics depend more heavily on environmental conditions, such as the user's location or posture, than physical keyboards which typically remain stationary. This challenge of mobile keystroke biometrics was examined in [Crawford and Ahmadzadeh, 2017]. They found that authenticating a user in different positions (sitting, standing, or walking) performed only slightly better than guessing, but detecting the user's position before authentication can significantly improve performance.

Nowadays, with the proliferation of machine learning algorithms capable of analysing and learning human behaviors from large scale datasets, the performance of keystroke dynamics in the free-text scenario has been boosted. As an example, [Ceker and Upadhyaya, 2016] the authors propose a combination of the existing digraphs method for feature extraction plus an SVM classifier to authenticate subjects. This approach achieves almost 0% error rate using samples containing 500 keystrokes. These results are very promising, even though it was evaluated using a small dataset with only 34 subjects. In [Deb *et al.*, 2019], the authors employ an RNN within a Siamese architecture to authenticate subjects based on 8 biometric modalities on smartphone devices. They achieved results in a free-text scenario of 81.61% TAR at 0.1% FAR using just 3 second test windows with a dataset of 37 subjects.In other work [Giot and Rocha, 2019], the authors tested the Siamese architecture for verification in a fixed-text scenario by employing the CMU dataset ([Killourhy and Maxion, 2009]), achieving poor results of 31% ERR with 200 enrollment samples per subject over a population of 51 subjects and exposing the limitations of RNN architectures in fixed-text keystroke authentication.

In [Çeker and Upadhyaya, 2017], the authors employed CNN (Convolutional Neural Network) with Gaussian data augmentation technique for fixed-text keystroke authentication over a population of 267 subjects. Their results of 2.02% EER in the best scenario suggest the combined benefit of CNN architectures and data augmentation for keystroke biometric systems. Finally in [Lu *et al.*, 2019], the authors combined a CNN with a RNN architecture. They argued that adding a 1D convolutional layer at the top of the RNN architecture makes the model able to extract higher-level keystroke features that are processed by the following RNN layers. Their results tested with the SUNY Buffalo ([Sun *et al.*, 2016]) dataset showed a relative error reduction of 35% (from 5.03% to 2.67% EER) when employing the 1D convolutional layer with a population of 75 users and keystrokes sequences of 30 keys. The main drawback of this method is that they need to train an independent model for each subject in order to extract enough high-level keystroke features from the subject.

Previous works in free-text keystroke dynamics have achieved promising results with up to

several hundred subjects (see Table 3.1), but they have yet to scale beyond this limit and leverage emerging machine learning techniques that benefit from vast amounts of data.

3.2. On the Analysis of Keystroke Recognition Performance based on Proprietary Passwords

The performance of fixed-text keystroke authentication systems are difficult to predict for some users [Morales *et al.*, 2014]. There is a large margin between performance of different users and it is possible to observe users with performances 10 time worse than others independently of the fixed-text keystroke authentication systems employed. The reasons of this variable performance have attracted the interest of researchers [Mondal *et al.*, 2013; Montalvão *et al.*, 2015; Morales *et al.*, 2014; Syed *et al.*, 2011].

In this section we extend the previous studies by: *i)* analyzing different factors that affect the keystroke recognition performance for scenario in which each user type a proprietary password (300 passwords); *ii)* we employ one of the largest databases available with 300 users acquired in 4 different sessions and four state-of-the art algorithms recently evaluated during the Keystroke Biometrics Ongoing Competition (see Sec. 2.1.2 for database details); *iii)* we provide new insights on keystroke recognition performances including results that contradict what has been known to date about the length of the passwords and its performances.

3.2.1. Experimental Protocol

The experimental protocol used in this section is the same proposed during the KBOC Competition [Morales *et al.*, 2016]. It is based on the following steps, for each user: *i)* Participants have 4 training samples (genuine samples from the 1st session) as enrolment data. *ii)* 20 test samples (genuine and impostor samples randomly selected from remaining samples not used for training) are used to evaluate the performance of the systems. The number of genuine and impostor samples per user varies between 8 and 12 (but the sum is equal to 20 for all of them). This variable number of genuine and impostor samples helps to avoid algorithms that exploit cohort information. *iii)* Each test sample is labelled with its corresponding user model and performance is evaluated according to the verification task (one to one comparisons). The performance is evaluated in the form of user-dependent EER. The EER has been calculated independently for each of the 300 subjects (300 different decision thresholds). The final EER value is the average of the individual EER from all subjects.

We will analyze the performance of 4 state-of-the-art keystroke recognition systems evaluated during the KBOC Competitions [Loy *et al.*, 2005; Morales *et al.*, 2016]. The systems were submitted by 4 different participants. We have chosen the best system from each participant among the 31 systems submitted during the competition (see [Morales *et al.*, 2016] for details). Table 3.2 shows the performance of all 4 systems according to the experimental protocol proposed. This performance will be used as baseline for the rest of experiments of this section. The

System	Baseline	Good Users	Bad Users
P1	11.3	6.1	25.3
P2	8.9	4.6	25.8
P3	14.7	5.5	24.8
P4	4.6	3.1	23.9

Table 3.2: *Baseline equal error rates (%) per user for all systems and averaged for good and bad users. The threshold calculated to discriminate between both groups was* 10% *EER for all systems. P = Participant #.*

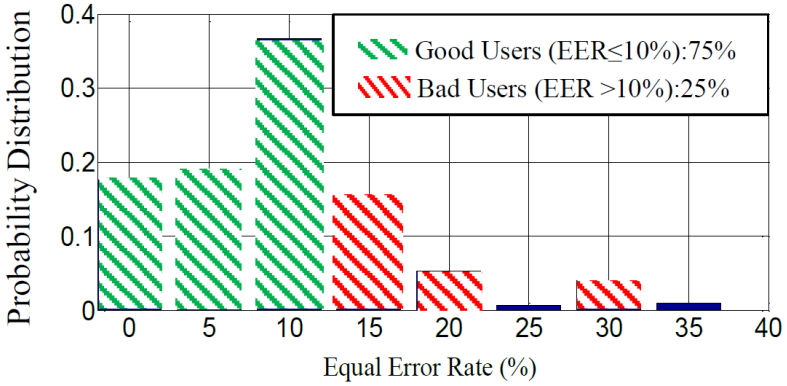

Figure 3.1: *Probability distribution of Equal Error Rate (averaged from all* 4 *systems) among the database population.*

results show a large difference between the performance of the Participant 4 (P4) and the rest of participants. The largest differences between participants lie in the pre-processing (sequence alignment and feature normalization) and post-processing techniques (score normalization) applied. The score normalization applied by P4 allows reducing the EER up to 4.62%. In the next sections we will analyze different factors affecting the performance of keystroke recognition systems at three levels: Classification level (by analyzing the scores obtained by the systems), Feature level (by analyzing the features used as input for the systems) and Score level (by analyzing techniques used for score normalization).

3.2.2. Results: Performance Analysis at Classification Level

- *Baseline:* the performance of keystroke dynamics is strongly user-dependent. As an example, Fig. 3.1 shows the probability distribution of the EER (averaging the performance of all 4 systems) obtained independently for each of the 300 users. The results show a large margin between performances of different users (from 0% to 35% of EER). In addition, it is remarkable the large number of users with 0% of EER for all 4 systems (around 20% of users). The final aim of this section is to find the main factors affecting this large difference between performances obtained among users.

- *Good vs. Bad Users:* in order to analyze the performance of users, the database was

	P1	P2	P3	P4
P1	100	48.5	42.5	42.0
P2	72.1	100	57.7	54.9
P3	36.0	32.5	100	30.0
P4	94.2	82.2	79.5	100

	P1	P2	P3	P4
P1	100	86.0	75.8	95.3
P2	68.7	100	61.7	78.5
P3	80.4	81.9	100	86.0
P4	47.7	49.1	40.5	100

Table 3.3: *Confusion matrix for good users (left) and bad users (right). System P4 (row 4) has the largest number of good users in comparison with the others systems.*

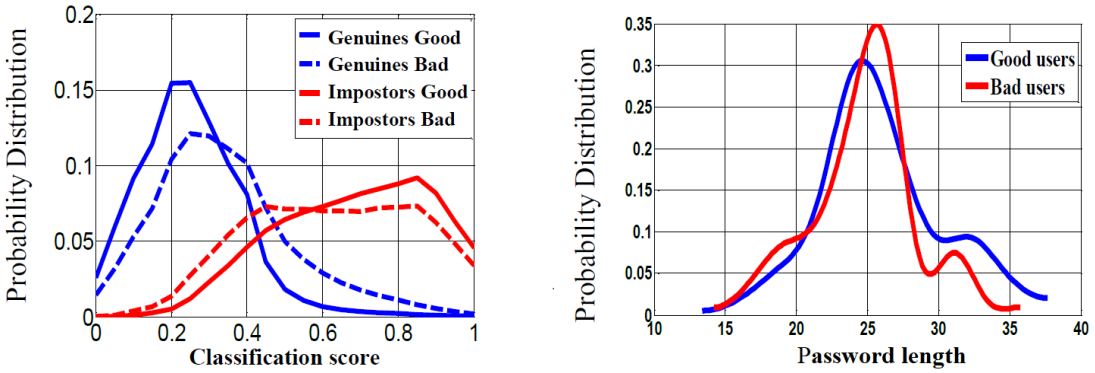

Figure 3.2: *Probability distributions of classifications scores (left) and length of passwords (right) for good and bad users (curves averaged from all four systems).*

divided into two groups (independently for each system) attending to the EERs of the users. Users with lower EER ($\leq 10\%$) were named as good users while users with higher EER ($> 10\%$) where named as bad users. The average of the EER for each group are summarized in Table 3.2. While good users show mean EER ranging between 3% and 6%, the bad users show up to 25% mean EER. The good users represent around 75% of the database while bad users the remaining 25%. The probability distribution of classification scores from test samples (normalized between $0-1$ for all 4 systems) can be seen in Fig. 3.2. The distributions shown that overlap between both genuine and impostor scores is greater for bad users as is expected. However, the degradation of the genuine scores is higher, suggesting that intra-class variability (difference between samples of the same user along different sessions) is more important than the inter-class variability (ability of the impostor) in this scenario. Table 3.3 shows confusions matrices for both groups and all 4 systems. The average percentage of coincidence between good users is 55% and 70% for bad ones. The superior percentage of bad users suggests that worst users are difficult to identify for all 4 systems. On the other hand, there are 30% and 45% of bad and good users respectively that were classified into different quality groups depending of the system. These results suggest a large complementarity between systems (i.e., users with bad performances for system A can show good performances for system B).

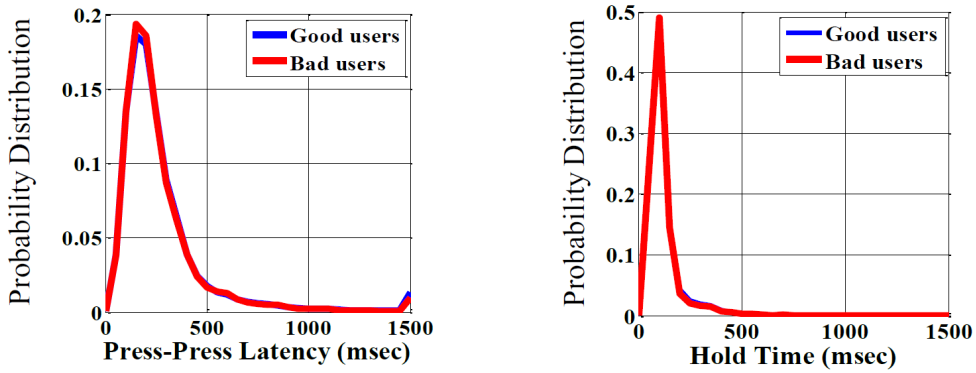

Figure 3.3: *Probability distribution of features for good and bad users (curves averaged from all four systems).*

3.2.3. Results: Performance Analysis at Feature Level

- *Length of the Password:* the first experiment is based on the idea that the length of the passwords can affect the performance of the systems [Mondal *et al.*, 2013; Montalvão *et al.*, 2015]. Long passwords can be better to discriminate between impostors and genuine users as they carry more biometric user information. However, the results showed in Fig. 3.2 (right) suggests there is not dependence between length of the passwords and system performance. These results contradict previous works [Mondal *et al.*, 2013; Montalvão *et al.*, 2015] which states clear differences between performances obtained by long and short passwords. There are two main reasons to explain these results: *i)* passwords used in this KBOC database are composed by familiar words (name and surname) instead of alphanumeric sequences of symbols (e.g., '*tie5Roanl*' and '*try4-mbs*'). The users of KBOC database show very stable features as they type very familiar sequences; *ii)* the length of the passwords in KBOC database ranges between 12 and 38 symbols while previous studies were based in passwords with a maximum length of 16 symbols. Based on our experiments and scenarios, he length of the password is not a key factor which determine the keystroke performance.

- *Timing:* regarding two of the most popular characteristics on keystroke dynamics, we calculated the values of Hold Time and Press-Latency for each user. Fig. 3.3 shows both features for good and bad users and any difference between them have been appreciated. Good and bad users show exactly the same distributions of time. This result suggests that there are no differences in terms of time features (i.e., time between individual key events).

- *Misalignment:* around 4% of the samples in the database have different number of keys pressed (mainly because of the use of the shift keys). These sequences may produce misalignment during the comparison of training and test samples and performance degradation up to 300% (see [Morales *et al.*, 2016] for details). The number of misaligned samples in bad users is two times greater than good users. These results suggest that the correct alignment of sequences is critical for keystroke recognition performance.

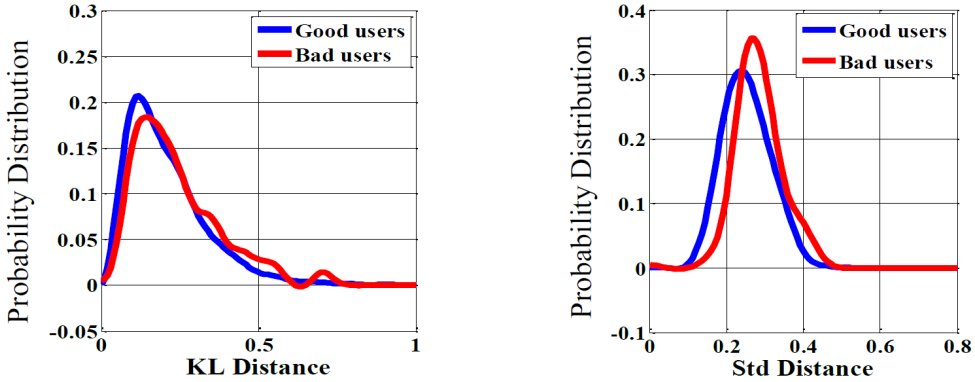

Figure 3.4: *Probability distribution of enrolment set variability (measured in the form of Kullback-Leibler divergence and standard deviation) for good and bad users (curves averaged from all four systems).*

System	EER_G	EER'_G
P1	15.7	12.0 ($\downarrow$ 23%)
P2	11.8	9.1 ($\downarrow$ 23%)
P3	18.0	14.5 ($\downarrow$ 19%)
P4	20.1	5.3 ($\downarrow$ 73%)

Table 3.4: *EER for all systems with (EER'_G) and without (EER_G) score normalization. In brackets we show the improvement.*

- *Stability of the Features:* for this experiment we measured the distance between training samples and genuine test samples for each user. In order to measure the distance, we propose two methods: standard deviations (std) and Kullback-Leibler divergence (KL). For each test sample, both distances are calculated as the distance between the test features and the enrolment feature vector (calculated averaging the 4 training feature vectors). Fig. 3.4 shows distances for good and bad users. KL distance seems to be very similar for both groups but small differences in std distance were observed. This difference in std distance suggests that good users tend to have less keystrokes variations.

3.2.4. Results: Performance Analysis at Score Level

EERs showed in Table 3.2 were calculated independently for each of the 300 subjects (300 different decision thresholds). These EERs are calculated as the average of the individual EER from all subjects [Giot *et al.*, 2009; Killourhy and Maxion, 2009; Shanmugapriya and Padmavathi, 2009]. To analyse the impact of the score normalization in the performance, the average EERs from the whole database (using only one decision threshold for all users) are summarized in Table 3.4 denoted as EER_G. Three different techniques of score normalization are proposed for this experiment with similar results: min-max, mu-sigma and *tangh* (see [Snelick *et al.*, 2005] for details). The best performance was achieved with a relative min-max normalization

Factors	Performance	Usability	Computational Cost
Length	↑	↑↑↑	↑
Timing	↑	↑	↑↑
Misalignment	↑↑↑	↑↑	↑↑↑
Stability	↑↑	↑	↑↑
Normalization	↑↑↑	↑	↑↑↑

Table 3.5: *Summary of the impact (↑ low, ↑↑ medium and ↑↑↑ high) for each factor based on our experimentation in keystroke dynamics for KBOC database.*

technique proposed in [Monaco, 2016] and described below:

$$score' = \frac{score - min_i}{max_i - min_i} \tag{3.1}$$

where:

$$min_i = \mu_i - 2 \times \sigma_i \tag{3.2}$$

$$max_i = \mu_i + 2 \times \sigma_i \tag{3.3}$$

these μ_i and σ_i are the mean and standard deviation of the user i obtained from the 20 test scores available for each user (optimist a posteriori normalization). Table 3.4 shows a significant improvement for all systems when score normalization is applied. The experiment show that score normalization can be used to improve performance by 20%. System P4 had the largest improvement ranging from 20.1% of EER_G to 5.3% of EER'_G. These results suggest a strong impact in the performance when employing normalization techniques. Note that best results are obtained using normalization parameters (mean and std of EER'_G) optimized according to the scores of each user. In some applications the scores available to model each user are limited and other strategies should be explored.

Finally, in Table 3.5 we summarize the impact for each factor based on our experiments. The results suggest that: *i)* the length of the password does not affect the performance of keystroke authentication for long passwords ($>$ 12 symbols) and familiar sequences; *ii)* intra-class variability has higher influence than inter-class variability; *iii)* misaligned samples have a strong impact on the performance; *iv)* the timing features from good and bad users are similar; *v)* score normalization techniques offers a huge improvement for algorithms with good intra-class adaptation but does not represent a realistic scenarios where a few training samples are available for these techniques.

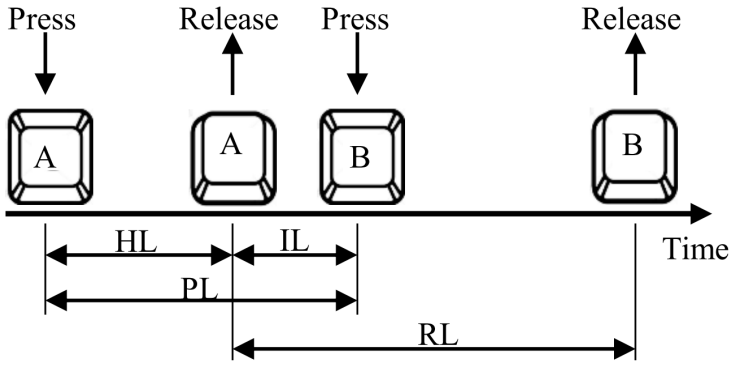

Figure 3.5: *Example of the 4 temporal features extracted between two consecutive keys: Hold Latency (HL), Inter-key Latency (IL), Press Latency (PL), and Release Latency (RL).*

3.3. TypeNet: Deep Learning Keystroke Biometric in Free-text

In keystroke dynamics, it is thought that idiosyncratic behaviors that enable authentication are characterized by the relationship between consecutive key press and release events (e.g., temporal patterns, typing rhythms, pauses, typing errors). Unlike fixed-text scenario, in free-text scenario the keystroke sequences between enrollment and testing may differ in both length and content. This reason motivates us to choose a Recurrent Neural Network as our keystroke authentication algorithm. RNNs have demonstrated to be one of the best algorithms to deal with temporal data [Tolosana *et al.*, 2020b, 2021c] and are well suited for free-text keystroke sequences [Deb *et al.*, 2019; Lu *et al.*, 2019]).

TypeNet is a RNN architecture composed of two Long Short-Term Memory (LSTM) layers of 128 units, following the same architecture as presented in Sec. 2.3.1. Our goal is to build a keystroke biometric system capable of generalizing to new subjects not seen during model training, and therefore, having a competitive performance when it deploys to applications with thousands of users. TypeNet is trained only once on an independent set of subjects. This model then acts as a feature extractor that provides input to a distance-based recognition scheme. After training TypeNet once, we evaluate in the experimental section the recognition performance for a varying number of subjects and enrollment samples per subject. For this, we train up to 6 TypeNet versions, one for each loss function (i.e., *Softmax loss*, *Contrastive loss* and *Triplet loss*) for both devices: desktop and mobile, using the Dhakal [Dhakal *et al.*, 2018] and Palin [Palin *et al.*, 2019] databases respectively (see Sec. 2.1.3 for database details).

The raw data captured for each session in both databases includes a time series with three dimensions: the keycodes, press times, and release times of the keystroke sequence. Timestamps are in UTC format with millisecond resolution, and the keycodes are integers between 0 and 255 according to the ASCII code.

We extract 4 temporal features for each sequence (see Fig. 3.5 for details): *i)* Hold Latency (HL), the elapsed time between key press and release events; *ii)* Inter-key Latency (IL), the elapsed time between releasing a key and pressing the next key; *iii)* Press Latency (PL), the

elapsed time between two consecutive press events; and *iv)* Release Latency (RL), the elapsed time between two consecutive release events. These 4 features are commonly used in both fixed-text and free-text keystroke systems [Alsultan and Warwick, 2013]. Finally, we include the keycodes as an additional feature.

The 5 features are calculated for each keystroke in the sequence. Let N be the length of the keystroke sequence, such that each sequence provided as input to the model $\mathbf{x}$ is a time series with shape $N \times 5$ (N keystrokes by 5 features). Following the same nomenclature, the output of the model $\mathbf{f}(\mathbf{x})$ is an array of size 1×128 that we will employ later as an embedding feature vector to recognize subjects.

All feature values are normalized before being provided as input to the model. Normalization is important so that the activation values of neurons in the input layer of the network do not saturate (i.e., all close to 1). The keycodes are normalized to between 0 and 1 by dividing each keycode by 255, and the 4 timing features are converted to seconds. This scales most timing features to between 0 and 1 as the average typing rate over the entire dataset is 5.1 ± 2.1 keys per second. Only latency features that occur either during very slow typing or long pauses exceed a value of 1.

3.3.1. Experimental Protocol

In the desktop scenario, we train the models using only the first 68,000 subjects from the Dhakal dataset. For the models trained with the *Softmax loss* function we employ $C = 10{,}000$ subjects for classification, due to the *Softmax loss* requires a very wide final layer with many classes. In this case, we used $15 \times 10{,}000 = 150{,}000$ keystroke sequences for training and the remaining 58,000 subjects were discarded (due to hardware limitations). For the *Contrastive loss* we generate genuine and impostor pairs using all the 15 keystroke sequences available for each subject. This provides us with $15 \times 67{,}999 \times 15 = 15.3$ million impostor pair combinations and $15 \times 14/2 = 105$ genuine pair combinations for each subject. The pairs were chosen randomly in each training batch ensuring that the number of genuine and impostor pairs remains balanced (512 pairs in total in each batch including impostor and genuine pairs). Similarly, we randomly chose triplets for the *Triplet loss* training.

The remaining 100,000 subjects were employed only for model evaluation, so there is no data overlap between the two groups of subjects. This reflects an open-set authentication paradigm. The same protocol was employed for the mobile scenario but adjusting the amount of subjects employed to train and test. In order to have balanced subsets close to the desktop scenario, we divided by half the Palin database such that 30,000 subjects were used to train the models, generating $15 \times 29{,}999 \times 15 = 6.75$ million impostor pair combinations and $15 \times 14/2 = 105$ genuine pair combinations for each subject. The other 30,000 subjects were used to test the mobile TypeNet models. Once again $C = 10{,}000$ subjects were used to train the mobile TypeNet model with *Softmax loss*.

Regarding the hyper-parameters employed during training, the best results for all models were achieved with a learning rate of 0.05, Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and

$\epsilon = 10^{-8}$, and the margin set to $\alpha = 1.5$ for the *Contrastive loss* function. The models were trained for 200 epochs with 150 batches per epoch and 512 sequences in each batch. The models were built in `Keras-Tensorflow`.

The results are divided into two groups (i.e., identification and authentication results) following two different experimental protocols:

- *Authentication Protocol:* we authenticate subjects by comparing gallery samples $\mathbf{x}_{i,g}$ belonging to the subject i in the test set to a query (i.e., a sample from unknown user) sample $\mathbf{x}_{j,q}$ from either the same subject (genuine match $i = j$) or another subject (impostor match $i \neq j$). The test score is computed by averaging the Euclidean distances between each gallery embedding vector $\mathbf{f}(\mathbf{x}_{i,g})$ and the query embedding vector $\mathbf{f}(\mathbf{x}_{j,q})$ as follows:

$$s_{i,j}^{q} = \frac{1}{G} \sum_{g=1}^{G} ||\mathbf{f}(\mathbf{x}_{i,g}) - \mathbf{f}(\mathbf{x}_{j,q})|| \tag{3.4}$$

  where G is the number of sequences in the gallery (i.e., the number of enrollment samples) and q is the query sample of subject j. Taking into account that each subject has a total of 15 sequences, we retain 5 sequences per subject as the test set (i.e., each subject has 5 genuine test scores) and let G vary between $1 \leq G \leq 10$ in order to evaluate the performance as a function of the number of enrollment sequences.

  To generate impostor scores, for each enrolled subject we choose one test sample from each remaining subject. We define k as the number of enrolled subjects. In our experiments, we vary k in the range $100 \leq k \leq K$, where $K = 100{,}000$ for the desktop TypeNet models and $K = 30{,}000$ for the mobile TypeNet. Therefore each subject has 5 genuine scores and $k - 1$ impostor scores. Note that we have more impostor scores than genuine ones, a common issue in keystroke dynamics authentication. The results reported in the next section are computed in terms of Equal Error Rate (EER), which is the value where False Acceptance Rate (FAR, proportion of impostors classified as genuine) and False Rejection Rate (FRR, proportion of genuine subjects classified as impostors) are equal. The error rates are calculated for each subject and then averaged over all k subjects [Morales *et al.*, 2014].

- *Identification Protocol:* identification tasks are common in forensics applications, where the final decision is based on a bag of evidences and the biometric recognition technology can be used to provide a list of candidates, referred to as background set $\mathfrak{B}$ in this work. The Rank-1 identification rate reveals the performance to unequivocally identifying the target subject among all the subjects in the background set. Rank-n represents the accuracy if we consider a ranked list of n profiles from which the result is then manually or automatically determined based on additional evidence [Fierrez *et al.*, 2018b].

  The 15 sequences from the k test subjects in the database were divided into two groups: Gallery (10 sequences) and Query (5 sequences). We evaluate the identification rate by

		#Enrollment Sequences per Subject G				
		1	**2**	**5**	**7**	**10**
#Keys per Sequence M	**30**	17.2/10.7/8.6	14.1/9.0/6.4	13.3/7.3/4.6	12.7/6.8/4.1	11.5/3.3/3.7
	50	16.8/8.2/5.4	13.1/6.7/3.6	10.8/5.4/2.2	9.2/4.8/1.8	8.8/4.3/1.6
	70	14.1/7.7/4.5	10.4/6.2/2.8	7.5/4.8/1.7	6.7/4.3/1.4	6.0/3.9/1.2
	100	13.8/7.7/4.2	10.1/6.0/2.7	7.4/4.7/1.6	6.4/4.3/1.4	5.7/3.9/1.2
	150	13.8/7.7/4.1	10.1/6.0/2.7	7.4/4.7/1.6	6.5/4.3/1.4	5.8/3.8/1.2

Table 3.6: *Equal Error Rates (%) achieved in* ***desktop*** *scenario using Softmax/Contrastive/Triplet loss for different values of the parameters M (sequence length) and G (number of enrollment sequences per subject).*

comparing the Query set of samples $\mathbf{x}^{\mathrm{Q}}_{j,q}$, with $q = 1, ..., 5$ belonging to the test subject j against the Background Gallery set $\mathbf{x}^{\mathrm{G}}_{i,g}$, with $g = 1, ..., 10$ belonging to all background subjects. The distance was computed by averaging the Euclidean distances $||\cdot||$ between each gallery embedding vector $\mathbf{f}(\mathbf{x}^{\mathrm{G}}_{i,g})$ and each query embedding vector $\mathbf{f}(\mathbf{x}^{\mathrm{Q}}_{j,q})$ as follows:

$$s^{Q}_{i,j} = \frac{1}{10 \times 5} \sum_{g=1}^{10} \sum_{q=1}^{5} ||\mathbf{f}(\mathbf{x}^{\mathrm{G}}_{i,g}) - \mathbf{f}(\mathbf{x}^{\mathrm{Q}}_{j,q})|| \tag{3.5}$$

We then identify a query set (i.e., subject $j = J$ is the same gallery person $i = I$) as follows:

$$I = \arg\min_{i} s^{Q}_{i,J} \tag{3.6}$$

The results reported in the next section are computed in terms of Rank-n accuracy. A Rank-1 means that $d_{i,J} < d_{I,J}$ for any $i \neq I$, while a Rank-n means that instead of selecting a single gallery profile, we select n profiles starting with $i = I$ by increasing distance $d_{i,J}$. In forensic applications, it is traditional to use Rank-20, Rank-50, or Rank-100 in order to generate a short list of potential candidates that are finally identified by considering other evidence.

#Keys per Sequence M	#Enrollment Sequences per Subject G = 1	2	5	7	10
30	17.7/15.7/14.2	16.0/14.1/12.5	15.2/13.0/11.3	14.9/12.6/10.9	14.5/12.1/10.5
50	17.2/14.6/12.6	15.4/13.1/10.7	13.8/12.1/9.2	13.4/11.5/8.5	12.7/11.0/8.0
70	17.8/13.8/11.3	15.5/12.4/9.5	13.5/11.2/7.8	13.0/10.7/7.2	12.1/10.4/6.8
100	18.4/13.6/10.7	15.8/12.3/8.9	13.6/10.9/7.3	13.0/10.4/6.6	12.3/10.0/6.3
150	18.4/13.7/10.7	15.9/12.3/8.8	13.7/10.8/7.3	13.0/10.4/6.6	12.3/10.0/6.3

Table 3.7: *Equal Error Rates (%) achieved in* ***mobile*** *scenario using Softmax/Contrastive/Triplet loss for different values of the parameters M (sequence length) and G (number of enrollment sequences per subject).*

3.3.2. Results and Discussion

3.3.2.1. Authentication: Varying Amount of Enrollment Data

As commented in the related works section (Sec. 3.1), one key factor when analyzing the performance of a free-text keystroke authentication algorithm is the amount of keystroke data per subject employed for enrollment. In this section, we study this factor with two variables: the keystroke sequence length M and the number of gallery sequences used for enrollment G.

Our first experiment reveals to what extent M and G affect the authentication performance of our TypeNet models. Note that the input to our models has a fixed size of M after the masking process shown in Fig. 2.4 (Sec. 2.3.1). For this experiment, we set $k = 1{,}000$ (where k is the number of enrolled subjects). Tables 3.6 and 3.7 summarize the error rates in both desktop and mobile scenarios respectively, achieved by the TypeNet models for the different values of sequence length M and enrollment sequences per subject G.

In the desktop scenario (Table 3.6) we observe that for sequences longer than $M = 70$ there is no significant improvement in performance. Adding three times more key events (from $M = 50$ to $M = 150$) lowers the EER by only 0.7% in average for all values of G. However, adding more sequences to the gallery shows greater improvements with about 50% relative error reduction when going from 1 to 10 sequences independent of M. Comparing among the different loss functions, the best results are always achieved by the model trained with *Triplet loss* for $M = 70$ and $G = 10$ with an error rate of 1.2%(with a standard deviation of $\sigma \leq 4.1\%$), followed by the *Contrastive loss* function with an error rate of 3.9%; the worst results are achieved with the *Softmax loss* function (6.0%). For one-shot authentication ($G = 1$), our approach has an error rate of 4.5% using sequences of 70 keystrokes.

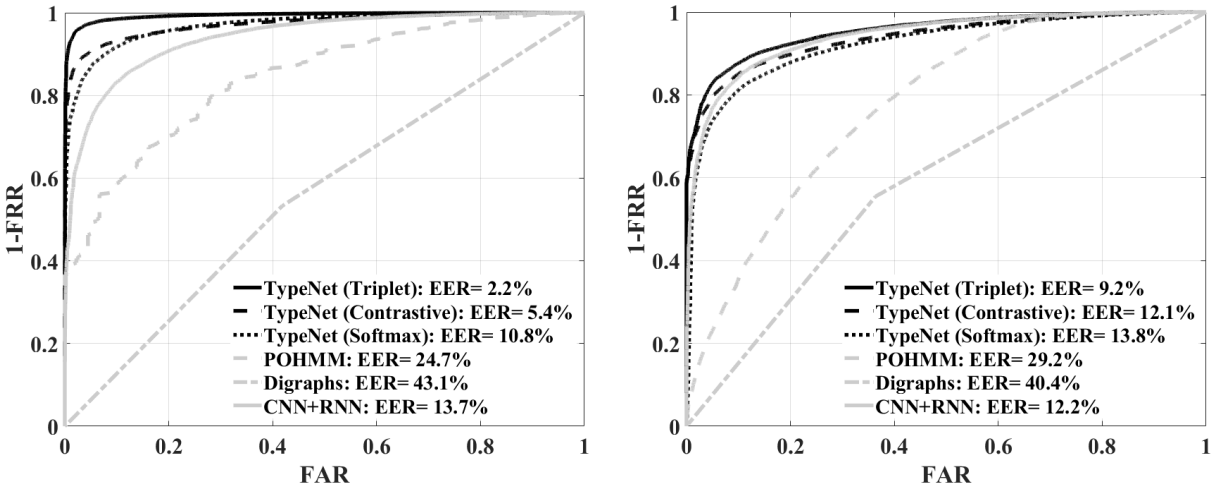

Figure 3.6: *ROC comparisons in free-text biometric authentication for desktop (left) and mobile (right) scenarios between the three proposed TypeNet models and three state-of-the-art approaches: POHMM (Partially Observable Hidden Markov Model) from [Monaco and Tappert, 2018], digraphs/SVM from [Ceker and Upadhyaya, 2016], and CNN+RNN (Convolutional Neuronal Network + Recurrent Neuronal Network) model from [Lu* et al.*, 2019]. $M = 50$ keystrokes per sequence, $G = 5$ enrollment sequences per subject, and $k = 1{,}000$ test subjects.*

Similar trends are observed in the mobile scenario (Table 3.7) compared to the desktop scenario (Table 3.6). First, increasing sequence length beyond $M = 70$ keystrokes does not significantly improve performance, but there is a significant improvement when increasing the number of sequences per subject. The best results are achieved for $M = 100$ and $G = 10$ with an error rate of 6.3% by the model trained with *Triplet loss* (with a standard deviation of $\sigma \leq 9.2\%$), followed again by the *Contrastive loss* (10.0%), and *Softmax loss* (12.3%). For one-shot authentication ($G = 1$), the performance of the triplet model decays up to 10.7% EER using sequences of $M = 100$ keystrokes.

Comparing the performance achieved by the three TypeNet models between mobile and desktop scenarios, we observe that in all cases the results achieved in the desktop scenario are significantly better to those achieved in the mobile scenario. These results are consistent with prior work that has obtained lower performance on mobile devices when only timing features are utilized [Banovic *et al.*, 2017; Buschek *et al.*, 2015; Teh *et al.*, 2016].

Next, we compare TypeNet with our implementation of two state-of-the-art algorithms for free-text keystroke authentication: a statistical sequence model, the POHMM (Partially Observable Hidden Markov Model) from [Monaco and Tappert, 2018], an algorithm based on digraphs and SVM from [Ceker and Upadhyaya, 2016], and a deep model based on the combination of CNN and RNN architectures introduced in [Lu *et al.*, 2019]. To allow fair comparisons, all approaches are trained and tested with the same data and experimental protocol: $G = 5$ enrollment sequences per subject, $M = 50$ keystrokes per sequence, $k = 1{,}000$ test subjects. The CNN+RNN architecture proposed in [Lu *et al.*, 2019] was trained following the same protocol employed with the TypeNet model.

In Fig. 3.6 we plot the error rates of the four approaches (i.e., Digraphs, POHMM, CNN+RNN,

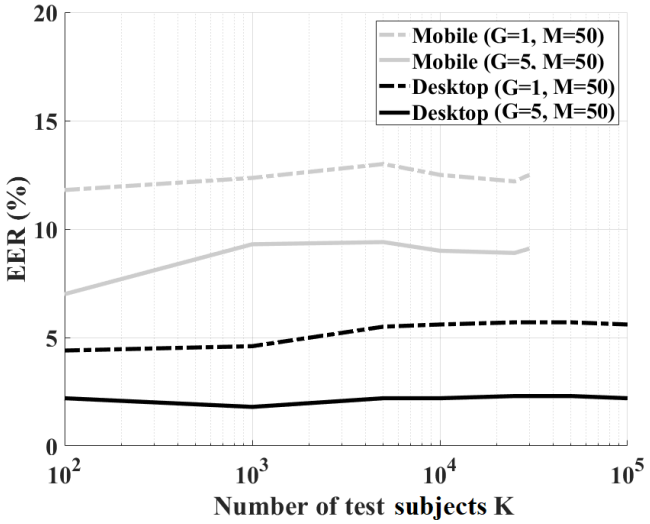

Figure 3.7: *EER (%) of our proposed TypeNet models when scaling up the number of test subjects k in one-shot ($G = 1$ enrollment sequences per subject) and 5-shot ($G = 5$) authentication cases. $M = 50$ keystrokes per sequence.*

and TypeNet) trained and tested on both desktop (left) and mobile (right) datasets. The TypeNet models outperform previous state-of-the-art free-text algorithms in both mobile and desktop scenarios with this experimental protocol, where the amount of enrollment data is reduced ($5 \times M = 250$ training keystrokes in comparison to more than 10,000 in related works, see Sec. 3.1). This can largely be attributed to the rich embedding feature vector produced by TypeNet, which minimizes the amount of data needed for enrollment. The SVM generally requires a large number of training sequences per subject (~ 100), whereas in this experiment we have only 5 training sequences per subject. We hypothesize that the lack of training samples contributes to the poor performance (near chance accuracy) of the Digraphs system based on SVMs. Finally, the results achieved by the model based on CNN+RNN are the closest to those achieved by the TypeNet models. The deep learning architectures clearly outperform traditional approaches. However, the performance of TypeNet is significantly better than the performance achieved by the architecture proposed in [Lu *et al.*, 2019], especially for the desktop scenario.

3.3.2.2. Authentication: Varying Number of Subjects

In this experiment, we evaluate to what extent our best TypeNet models (those trained with *Triplet loss*) are able to generalize without performance decay. For this, we scale the number of enrolled subjects k from 100 to K (with $K = 100{,}000$ for desktop and $K = 30{,}000$ for mobile). For each subject we have 5 genuine test scores and $k-1$ impostor scores, one against each other test subject. The models used for this experiment are the same trained in previous the section (68,000 independent subjects included in the training phase for desktop and 30,000 for mobile).

Fig. 3.7 shows the authentication results for one-shot enrollment ($G = 1$ enrollment sequences, $M = 50$ keystrokes per sequence) and the case ($G = 5$, $M = 50$) for different values of

		TypeNet Model		
		Desktop	**Mobile**	**Mixture**
Test Dataset	**Aalto Desktop**	2.2	21.4	17.9
	Aalto Mobile	13.7	9.2	12.6
	Buffalo (Free)	7.6	33.2	22.1
	Buffalo (Transc)	9.5	32.8	23.1
	Clarkson II	26.8	36.6	35.8
	Clarkson II*	17.2	33.0	30.4

Table 3.8: *Equal Error Rates (%) achieved in the cross-database scenario for the three TypeNet models (Desktop, Mobile, and Mixture) when testing on Aalto Desktop ([Dhakal* et al.*, 2018]), Aalto Mobile([Palin* et al.*, 2019]), Clarkson II ([Ayotte* et al.*, 2020]), and Buffalo ([Sun* et al.*, 2016]) dataset. Buffalo (Free) = free text, Buffalo (Transc) = transcripted text. *Experiment using all the data available per subject.*

k. For the desktop devices, we can observe that in both cases there is a slight performance decay when we scale from 1,000 to 10,000 test subjects, which is more pronounced in the one-shot case. However, for a large number of subjects ($k \geq$ 10,000), the error rates do not appear to demonstrate continued growth. For the mobile scenario, the results when scaling from 100 to 1,000 test subjects show a similar tendency compared to the desktop scenario with a slightly greater performance decay. However, we can observe an error rate reduction when we continue scaling the number of test subjects up to 30,000. In all cases the variation of the performance across the number of test subjects is less than 2.5% EER. These results demonstrate the potential of the RNN architecture in TypeNet to authenticate subjects at large scale in free-text keystroke dynamics. We note that in the mobile scenario, we have utilized only timing features; prior work has found that greater performance may be achieved by incorporating additional sensor features [Kim and Kang, 2020].

3.3.2.3. Authentication: Cross-database Interoperability

In this experiment we measure the cross-device interoperability between the best TypeNet models trained with the triplet loss. We also study the capacity of both desktop and mobile TypeNet models to generalize to other input devices and state-of-the-art databases. For this, we test both models with a different keystroke dataset than the one employed in their training. Additionally, for this experiment we train a third TypeNet model called Mixture-TypeNet with triplet loss using keystroke sequences from both datasets (half of the training batch for each dataset) but keeping the same train/test subject division as the other TypeNet models to allow fair comparisons. To be consistent with the other experiments we keep the same experimental protocol: $G = 5$ enrollment sequences per subject, $M = 50$ keystrokes per sequence, $k =$ 1,000 test subjects.

Table 3.8 shows the error rates achieved for the three TypeNet models when we test with desktop (Dhakal) and mobile (Palin) datasets. We can observe that error rates increase significantly in the cross-device scenario for both desktop and mobile TypeNet models. This performance

decay is alleviated by the Mixture-TypeNet model, which still performs much worse than the other two models trained and tested in the same-sensor scenario. These results suggest that multiple device-specific models may be superior to a single model when dealing with input from different device types. This would require device type detection in order to pass the enrollment and test samples to the correct model [Alonso-Fernandez *et al.*, 2010].

Secondly, we test the generalization capacity of the three TypeNet models with two public free-text keystroke databases: the Clarkson II dataset collected in [Murphy *et al.*, 2017] and the Buffalo dataset collected in [Sun *et al.*, 2016]. Table 3.8 presents the performance of the proposed approaches over the Clarkson II database and Buffalo database in transcribed (Transc) and free-text (Free) scenarios. Note that the models were trained and evaluated with different databases. This experiment is aimed to explore the generalization capacity between various data collection environments. Due to the number of subjects in both Clarkson II and Buffalo databases, which is much fewer than those present in the Aalto datasets, we modified the experimental protocol. For Clarkson II we employed $k = 91$ (the number of subjects for which we could extract at least 15 samples of 150 keys), $G = 5$ enrollment sequences per subject, $M = 50$ keystrokes per sequence. For the Buffalo database we employed $k = 147$, $G = 2$ enrollment sequences per subject (as we only have three sessions per subject, we employ two for gallery and one for query), and $M = 50$ keystrokes per sequence.

The last three rows of Table 3.8 show the results achieved when testing with both Clarkson II and Buffalo databases. The performance of the Desktop version of TypeNet remained competitive for the Bufallo dataset even when we only employed $G = 2$ gallery samples per subject. Nonetheless, there is a large increase of the error rates for Clarkson II database. This drop of performance might be caused by the uncontrolled acquisition of the Clarkson II database over a long time period (i.e., two years) and the fully free-text typing behavior. However, when we employ all keystroke data available in the database per subject for testing (i.e., $G = 10$ and $M = 150$) the error rate drops up to 17.2%. Note that the benchmark published in [Murphy *et al.*, 2017] achieved EERs around 10% training and testing with the same database. The results obtained by the owner of the database demonstrate the uncontrolled conditions of this database. We want to highlight that the TypeNet models were not retrained with any kind of keystroke data from Clarkson II or Buffalo databases, these databases were employed only for testing. These results suggest that re-training is necessary to improve the performance of the proposed models, especially for the Clackson II database. On the other hand, the performance achieved by the Mobile and Mixed versions of TypeNet was very poor with EERs greater than 20%. Both databases were acquired with desktop keyboards and these results indicates the importance of the device in the generalization capacity of the models.

3.3.2.4. Identification based on Keystroke Dynamics

Table 3.9 presents the identification accuracy for a background of $\mathfrak{B} = 1{,}000$ subjects, $k = 10{,}000$ test subjects, $G = 10$ gallery sequences per subject, and $M = 50$ keystrokes per sequence. The accuracy obtained for an identification scenario is much lower than the accuracy

Method	Scenario	Rank		
		1	50	100
Digraph [Ceker and Upadhyaya, 2016]	D	0.1	9.5	15.2
Digraph [Ceker and Upadhyaya, 2016]	M	0.0	8.5	14.4
POHMM [Monaco and Tappert, 2018]	D	6.1	48.4	63.4
POHMM [Monaco and Tappert, 2018]	M	6.5	41.8	53.7
CNN+RNN [Lu *et al.*, 2019]	D	44.2	95.5	98.2
CNN+RNN [Lu *et al.*, 2019]	M	24.5	86.3	90.5
TypeNet (*softmax*)	D	47.5	96.3	98.7
TypeNet (*softmax*)	M	23.5	82.6	91.4
TypeNet (*contrastive*)	D	29.4	97.2	99.3
TypeNet (*contrastive*)	M	19.0	80.4	89.8
TypeNet (*triplet*)	D	67.4	99.8	99.9
TypeNet (*triplet*)	M	25.5	87.5	94.2

Table 3.9: *Identification accuracy (Rank-n in %) for a background size* $\mathfrak{B} = 1{,}000$. *Scenario: D = Desktop, M = Mobile.*

reported for authentication. In general, the results suggest that keystroke identification enables a 90% size reduction of the candidate list while maintaining almost 100% accuracy (i.e., 100% rank-100 accuracy with 1,000 subjects). However, the results show the superior performance of the *Triplet loss* function and significantly better performance compared to traditional keystroke approaches [Ceker and Upadhyaya, 2016; Monaco and Tappert, 2018]. While traditional approaches are not suitable for large-scale free text keystroke applications, the results obtained by TypeNet demonstrate its usefulness in many applications.

The number of background profiles can be further reduced if auxiliary data is available to realize a pre-screening of the initial list of gallery profiles (e.g., country, language). The Aalto University dataset contains auxiliary data including age, country, gender, keyboard type (desktop vs. laptop), among others. Table 3.10 shows also subject identification accuracy over the 1,000 subjects with a pre-screening by country (i.e., contents generated in a country different to the country of the target subject are removed from the background set). The results show that pre-screening based on a unique attribute is enough to largely improve the identification rate: Rank-1 identification with pre-screening ranges between 5.5% to 84.0%, while the Rank-100 ranges between 42.2% to 100%. These results demonstrate the potential of keystroke dynamics for large-scale identification when auxiliary information is available.

3.3.2.5. Input Text Dependency in TypeNet Models

For the last experiment, we examine the effect of the text typed (i.e., the keycodes employed as input feature in the TypeNet models) on the distances between embedding vectors and how this may affect the model performance. The main drawback when using the keycode as an input feature to free-text keystroke algorithms is that the model could potentially learn text-based

Method	Scenario	Rank		
		1	50	100
Digraph [Ceker and Upadhyaya, 2016]	D	5.5	37.6	42.2
POHMM [Monaco and Tappert, 2018]	D	21.8	78.3	89.7
CNN+RNN [Lu *et al.*, 2019]	D	65.1	99.1	99.7
TypeNet (*softmax*)	D	68.3	99.39	99.9
TypeNet (*contrastive*)	D	56.3	99.7	99.9
TypeNet (*triplet*)	D	84.0	99.9	100

Table 3.10: *Identification accuracy (Rank-n in %) for a background size $\mathfrak{B} = 1{,}000$ and pre-screening based on the location of the typist. Scenario: D = Desktop. There is not metadata related to the mobile scenario.*

features (e.g., orthography, linguistic expressions, typing styles) rather than keystroke dynamics (e.g., typing speed and style) features. To analyze this phenomenon, we first introduce the Levenshtein distance (commonly referred as *Edit distance*) proposed in [Hyyro, 2005]. The Levenshtein distance d_L measures the distance between two words as the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into another. As an example, the Levenshtein distance between *"kitten"* and *"sitting"* is $d_L = 3$, because we need to substitute *"s"* for *"k"*, substitute *"i"* for *"e"*, and insert *"g"* at the end (three editions in total). With the Levenshtein distance metric we can measure the similarity of two keystroke sequences in terms of keys pressed and analyze whether TypeNet models could be learning linguistic expressions to recognize subjects. This would be revealed by a high correlation between Levenshtein distance d_L and the Euclidean distance of test scores d_E.

In Fig. 3.8 we plot the test scores (Euclidean distances) employed in one-shot scenario ($G = 1$ enrollment sequence per subject, $M = 50$ keystrokes per sequence, $k = 1{,}000$ test subjects) versus the Levenshtein distance between the gallery and the query sample that produced the test score (i.e., $d_E(\mathbf{f}(\mathbf{x}_g), \mathbf{f}(\mathbf{x}_q))$ vs. $d_L(\mathbf{x}_g, \mathbf{x}_q)$). To provide a quantitative comparison, we also calculate the Pearson coefficient p and the Linear Regression response as a measure of correlation between both distances (smaller slope indicates a weaker relationship). In mobile scenarios (Fig. 3.8 right) we can observe a significant correlation (i.e., higher slope in the Linear Regression response and high p value) between the Levenshtein distances and the test scores: genuine distance scores show lower Levenshtein distances (i.e., more similar typed text) than the impostor ones, and therefore, this metric provides us some clues about the possibility that TypeNet models in the mobile scenario could be using the similarity of linguistic expressions or keys pressed between the gallery and the query samples to recognize subjects. These results suggest us that the TypeNet models trained in the mobile scenario may be performing worse than in the desktop scenario, among other factors, because mobile TypeNet embeddings show a significant dependency to the entry text. On the other hand, in desktop scenarios (Fig. 3.8 left) this correlation is not present (i.e., the small slope in the Linear Regression response and $p \sim 0$) between test scores and Levenshtein distances, suggesting that the embedding vector produced

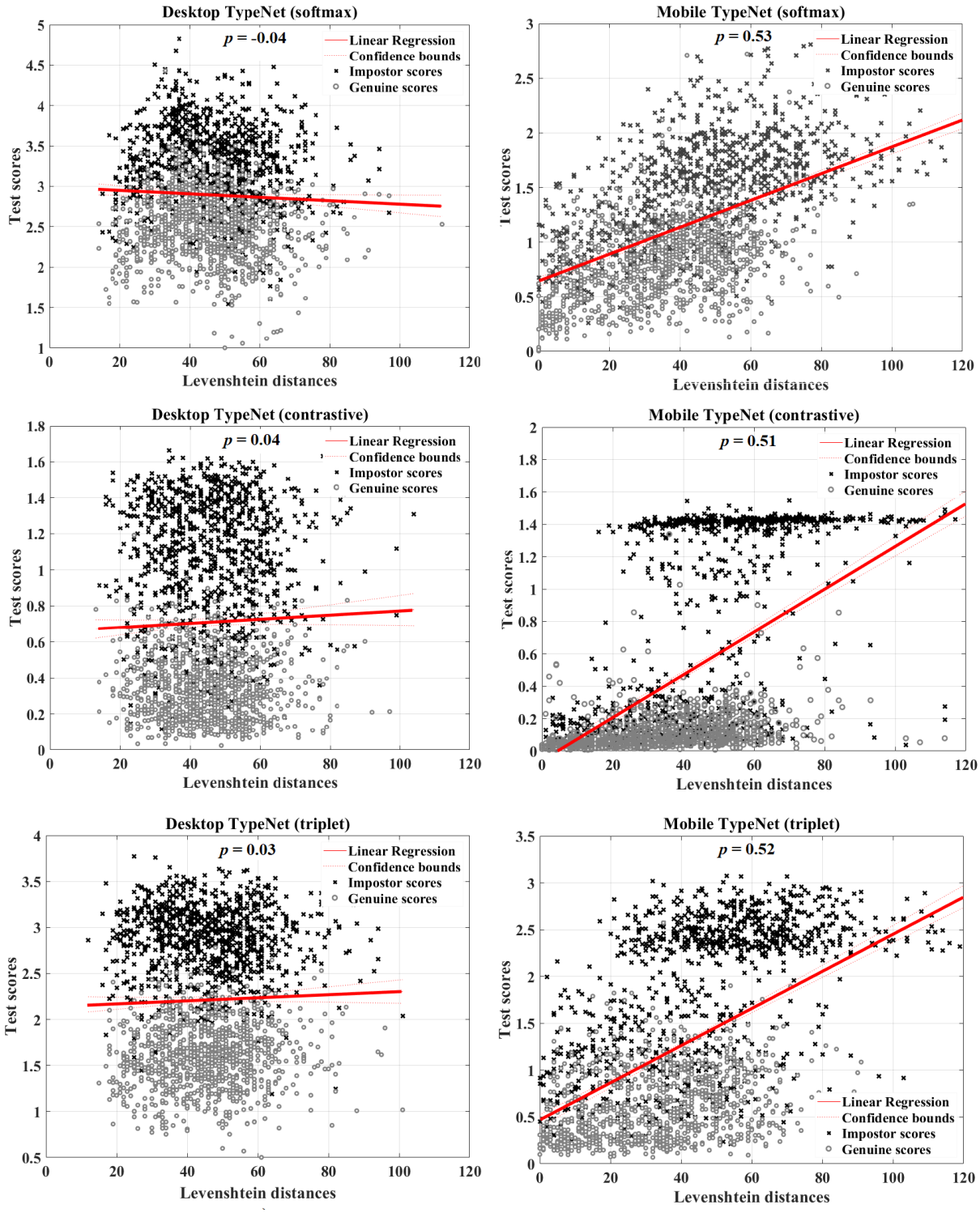

Figure 3.8: *Levenshtein distances vs. test scores in desktop (left) and mobile (right) scenarios for the three TypeNet models. For qualitative comparison we plot the linear regression results (red line), and the Pearson correlation coefficient p. Note: we only plot one genuine and one impostor score (randomly chosen) for each of the* 1,000 *subjects to improve the visualization of the results.*

by TypeNet models trained with the desktop dataset are largely independent of the input text.

3.4. Chapter Summary and Conclusions

In all authentication scenarios, the TypeNet models trained with triplet loss have shown a superior performance, especially when there are many subjects but few enrollment samples per subject. The results achieved in this work outperform previous state-of-the-art algorithms. Our results range from 17.2% to 1.2% EER in desktop and from 17.7% to 6.3% EER in mobile scenarios depending on the amount of subject data enrolled. A good balance between performance and the amount of enrollment data per subject is achieved with 5 enrollment sequences and 50 keystrokes per sequence, which yields an EER of 2.2/9.2% (desktop/mobile) for 1,000 test subjects. These results suggest that our approach achieves error rates close to those achieved by the state-of-the-art fixed-text algorithms [Morales *et al.*, 2016], within $\sim 5\%$ of error rate even when the enrollment data is scarce.

Scaling up the number of test subjects does not significantly affect the performance: the EER in the desktop scenario increases only 5% in relative terms with respect to the previous 2.2% when scaling up from 1,000 to 100,000 test subjects, while in the mobile scenario decays up to 15% the EER in relative terms. Evidence of the EER stabilizing around 10,000 subjects demonstrates the potential of this architecture to perform well at large scale. However, the error rates of both models increase in the cross-device interoperability scenario. Evaluating the TypeNet model trained in the desktop scenario with the mobile dataset the EER increases from 2.2% to 13.7%, and from 9.2% to 21.4% for the TypeNet model trained with the mobile dataset when testing with the desktop dataset. A solution based on a mixture model trained with samples from both datasets outperforms the previous TypeNet models in the cross-device scenario but with significantly worse results compared to single-device development and testing. When testing the generalization capacity of the proposed models with the Buffalo and Clarkson II keystroke datasets, TypeNet is able to maintain a competitive performance (between 7.6% and 17.2% of EER for the best scenario) without any kind of transfer learning or retraining, demonstrating the potential of TypeNet models to generalize well in other databases acquired under similar conditions. However, the performance decreased quickly when testing with databases acquired with different conditions or devices (e.g., touchscreen sensors).

In addition to authentication results, identification experiments have been also conducted. In this case, TypeNet models trained with triplet loss have shown again a superior performance in all ranks evaluated. For Rank-1, TypeNet models trained with triplet loss have an accuracy of 67.4/25.5% (desktop/mobile) with a background size of $\mathfrak{B} = 1{,}000$ identities, meanwhile previous related works barely achieve 6.5% accuracy. For Rank-50, the TypeNet model trained with triplet loss achieves almost 100% accuracy in the desktop scenario and up to 87.5% in the mobile one. The results are improved when using auxiliary-data to realize a pre-screening of the initial list of gallery profiles (e.g., country, language), showing the potential of TypeNet models to perform great not only in authentication, but also in identification tasks. Finally we have

demonstrated that the text-entry dependencies in TypeNet models are irrelevant in desktop scenarios, although in mobile scenarios the TypeNet models have some correlation between the input text typed and the performance achieved.

For fixed-text scenario, We have analyzed the performance of four state-of-the-art keystroke recognition systems. The experiments suggest that: *i)* the length of the password does not affect the performance of keystroke authentication for long passwords (> 12 symbols) and familiar sequences; *ii)* intra-class variability has higher influence than inter-class variability; *iii)* misaligned samples have a strong impact on the performance; *iv)* the timing features from good and bad users are similar, *v)* score normalization techniques offers a huge improvement for algorithms with good intra-class adaptation but does not represent a realistic scenarios where a few training samples are available for these techniques.

Chapter 4

User Mobile Authentication based on in-built Sensors

This chapter provides a general outlook of the different ways the sensors commonly available in modern smartphones can be used for modeling the interaction between human and smartphones. The main aim is to evaluate the signals generated by these sensors for person recognition.

For this, we first summarize in Sec. 4.1 a representative selection of existing databases on user mobile interaction, with special focus on applications related to user authentication, including key features and a selection of the main research results obtained on them so far. Then, we present the two proposed approaches for mobile authentication: one approach based on simple linear touch gestures using a Recurrent Neural Network architecture in Sec. 4.2 and a multi-modal approach based on smartphone usage under realistic conditions by including up to four different biometric traits (touch gestures, keystroke, gyroscope, and accelerometer) and three behavioral-based profiling techniques (GPS, Wi-Fi, and app usage) in Sec. 4.3.

4.1. State-of-the-art on Mobile Authentication

Mobile authentication based on signals acquired from the interaction of subjects with mobile devices has been extensively studied in the last years [Buschek *et al.*, 2015; Fierrez *et al.*, 2018; Li and Bours, 2018c]. In Table 4.1 we summarize some of the most relevant state-of-the-art works in this field. Swipe dynamics (touch gesture that consist on sliding the finger over the touchscreen) is one of the most popular traits analyzed [Fierrez *et al.*, 2018]. However, it has been shown not to have enough discriminative power to replace traditional technologies until now.

Accelerometer and gyroscope sensors have been studied traditionally for gait recognition, and some works have demonstrated also their utility for user authentication with acceptable performance [Li and Bours, 2018b].

Geo-location based verification approaches are scarce in the literature. In [Mahbub and

Study	Modality	Classifier	Database	Best Acc.
Shi *et al.* [2011]	4 (Mic, GPS, Tou, Gait)	Naive Bayes	Prop. DB	90%
Buschek *et al.* [2015]	Keystroke	KNN, SVM, NB	Prop. DB (28)	$67 \sim 74\%$
Fridman *et al.* [2016]	4 (Sty, App, Web, GPS)	Binary Classifiers, SVM	Prop. DB (200)	95%
Mahbub and Chellappa [2016]	GPS	M-HMM	UMDAA-02 (48), GeoLife (182)	$69 \sim 79\%$
Fierrez *et al.* [2018]	Touch Gestures	SVM, GMM	Serwadda (190), Antal (71), Frank (41), UMDAA-02 (48)	$80 \sim 90\%$
Li and Bours [2018c]	Acc, Wi-Fi	Templates, RF	Prop. DB (321)	90%
Li and Bours [2018b]	Acc, Gyr	Random Forest	Prop. DB (304)	77%
Monaco and Tappert [2018]	Keystroke	POHMM, HMM, SVM	Mutiple Databases (247)	90%
Liu *et al.* [2018]	5 (Tou, Pow, Acc, Gyr, Mag)	Suport Vector Machine	Prop. DB (10)	95%
Li and Bours [2018a]	4 (Wi-Fi, Blu, Acc, Gyr)	Random Forest	Prop. DB (321)	90%
Deb *et al.* [2019]	8 (Key, GPS, Acc, Gyr, Mag, Grav, LAc, Rot)	Siamese LSTM	Prop. DB (37)	97%
Mahbub *et al.* [2019]	App Usage	M-HMM	UMDAA-02 (48)	$70 \sim 84\%$
Acien *et al.* [2019b]	**7 (Tou, keys, Acc, Gyr, Wi-Fi, GPS, App)**	**Templates, SVM**	**UMDAA-02 (48)**	**98%**
Acien *et al.* [2020b]	**Touch Gestures**	**Siamese LSTM**	**HuMI (600)**	**87%**

Table 4.1: *Summary of the state-of-the-art in biometric mobile authentication. The number of users for each database is in brackets. Modalities: Touchscreen (Tou), Accelerometer (Acc), Linear Accelerometer (LAc), Stylometry (Sty), Bluetooth (Blu), App Usage (App), Web Browsing (Web), GPS, Keystroke (Key), Magnetometer (Mag), Microphone (Mic), Power consumption (Pow), Gravity (Grav), Rotation (Rot), Wi-Fi.*

Chellappa, 2016], the authors developed a mobile authentication system using trace histories by generating a confidence score of the new user location taking into account the sparseness of the geo-location data and past locations. For this purpose, they employed modified Hidden Markov Models (HMMs) considering the human mobility as a Markovian motion. In a similar way, in [Mahbub *et al.*, 2019] a variation of HMMs was used to develop a user authentication mobile system by exploiting application usage data. They suggest that unforeseen events and unknown applications have more impact in the authentication performance than the most common apps used by the user. The potential of Wi-Fi history data was analyzed in [Li and Bours, 2018c] for mobile authentication. They explored: *i)* the Wi-Fi networks detected by the smartphone, *ii)* when the detection occurs, and *iii)* how frequently those networks are detected during a period of time. Regarding keystroke traits, in [Buschek *et al.*, 2015] a fixed-text keystroke system for mobile user authentication was studied using not only time and space based features (e.g., hold and flight times, jump angle or drag distance) but also studying the hands postures during typing as discriminative information. In [Monaco and Tappert, 2018], a novel fixed-text

authentication system for laptops and mobile devices based on Partially Observable HMMs was studied. This model is an extension of HMMs, in which the hidden state is conditioned on an independent Markov chain. The algorithm is motivated by the idea that typing events depend both on past events and also on a separate process. Finally, building a multimodal system that integrates all these heterogeneous intra class variation or spoofing attacks [Marcel *et al.*, 2019] are some inevitable problems in unimodal systems that can be overcome by multimodal architectures [Fierrez *et al.*, 2018b; Patel *et al.*, 2016]. In [Shi *et al.*, 2011], a multimodal user authentication system was based on the fusion at decision level of voice, location, multi-touch, and accelerometer data. Their preliminary results suggest that these four modalities are suitable for continuous authentication. In [Fridman *et al.*, 2016], a fusion was performed also at decision level of behavioral-based profiling signals such as web browsing, application usage, and GPS location with keystroke data achieving 95% of user authentication accuracy using information from one-minute window. More recently, in [Liu *et al.*, 2018] a fusion also at decision level of touch dynamics, power consumption, and physical movements modalities achieved 94.5% of accuracy with a dataset that was captured under supervised conditions. In [Li and Bours, 2018a], an unobtrusive mobile authentication application is designed for single and multimodal approaches. They collected data from Wi-Fi, Bluetooth, accelerometer, and gyroscope sources in unsupervised conditions and fused them at score level achieving up to 90% of accuracy in the best scenario. In [Deb *et al.*, 2019], they propose a Siamese Long Short Term Memory network architecture to merge up to 8 modalities (keystroke dynamics, GPS location, accelerometer, gyroscope, magnetometer, linear accelerometer, gravity, and rotation sensors) for mobile authentication, achieving 97.15% of accuracy using data from a 3 seconds window for each of the modalities considered individually.

4.2. Mobile Authentication Based on Swipe Gestures

For this approach we explore a new authentication system based on the touch gestures acquired in HuMIdb. In particular, we employ to authenticate the right-swipe gestures captured when the users scroll the drag and drop button to proceed between tasks. This is a common gesture used in many touch interfaces (e.g., unlock, next step confirmation).

4.2.1. Experimental Protocol

Lets define the interaction of the user with the touchscreen as a time sequence $\{\mathbf{x}, \mathbf{y}, \mathbf{p}, \mathbf{t}\}$ with length N, composed by the coordinates $\{\mathbf{x}, \mathbf{y}\}$, the pressure $\mathbf{p}$, and the timestamp $\mathbf{t}$. The coordinates $\{\mathbf{x}, \mathbf{y}\}$ are normalized by the size of the screen. Then, we extract eleven temporal features adapted from [Tolosana *et al.*, 2018, 2021c] for on-line signatures: velocity, acceleration, jerk, and the Fourier transform for both axis $\{\mathbf{x}, \mathbf{y}\}$ plus the raw data $\{\mathbf{x}, \mathbf{y}, \mathbf{p}\}$. Note that we discard the timestamp $\mathbf{t}$ because it depends on the device and the network could be learning to discriminate among devices instead of users.

The architecture proposed to model swipe gestures is a RNN already depicted in Sec. 2.3.1. Note that for this approach we reduce the number of neurons in both LSTM layers to 64 due to swipe gesture are commonly shorter than other temporal signals, and therefore, to much neurons could overfit the model during the training phase. We train the RNN model in a Siamese setup (i.e., employing the *Contrastive Loss* function) in which the model has two inputs (the two swipe samples to compare) and two embedding vectors as outputs (see Fig. 2.5.b for details). During the training phase, the RNN model learns to project embedding vectors from same user close to each other and embedding vectors from different users far from each other. The input of the RNN model $\mathbf{s}$ (we changed the nomenclature of the RNN input from the previous Chapter to $\mathbf{s}$ in order to avoid misunderstandings with the $\mathbf{x}$ coordinate) is a feature set of size $11 \times N$ extracted for each swipe. The output of the model $\mathbf{f}(\mathbf{s})$ is an embedding vector of size 1×64 that contains discriminative information extracted from each swipe gesture to authenticate users. To test our model we compute the Euclidean distance between pairs of embedding vectors: one from the genuine user that claims to authenticate in our system, called gallery sample $\mathbf{f}(\mathbf{s}_g)$, and the unknown sample $\mathbf{f}(\mathbf{s}_u)$ that we want to verify.

Regarding the training details, the best results were achieved with a learning rate of 0.05, Adam optimizer with $\beta_1 = 0.9, \beta_1 = 0.999, \epsilon = 10^{-8}$ and the margin set (see Eq. 2.7) to $\alpha = 1.5$ without learning decay. The model was trained after 30 epochs with 100 batches per epoch. Each batch has a size of 512 pairs. The pairs were chosen randomly but keeping the number of genuine and impostor pairs balanced in each batch. The model was built in `Keras-Tensorflow`. The RNN network is trained with 70% of HuMIdb users and tested with the remaining ones (open-set authentication paradigm). We want to highlight that there are a total of 30K swipe gestures in our experimental dataset. The size of the input features vector is set to $N = 50$, filling with zeros when the vectors are smaller and truncating in the opposite case.

Additionally, we compare our proposed RNN model with our implementation of one of the best state-of-the-art systems traditionally employed in mobile touch authentication: global features extraction plus binary Support Vector Machine (SVM) classifier with Gaussian kernel. In particular, we compute the global features presented in [Martinez-Diaz *et al.*, 2014] (commonly used for online handwriting sequence modeling) and adapted for swipe biometrics in [Fierrez *et al.*, 2018]. Mean velocity, max acceleration, distance between adjacent points, or angles are some examples of this subset of 29 features extracted. We employ the same experimental protocol for both systems. This means that we compute a binary classifier to authenticate each user by using the gallery samples as genuine training samples, and then we test with the remaining ones. This method allows us to compare the amount of user data each architecture needs.

4.2.2. Results and Discussion

The results are depicted in Fig. 4.1 in terms of Equal Error Rate (EER), where EER refers to the value where False Acceptance Rate (FAR, percentage of impostors classified as genuine) and False Rejection Rate (FRR, percentage of genuine users classified as impostors) are equal. The curve shows the variation in the performance according to the number of gallery samples

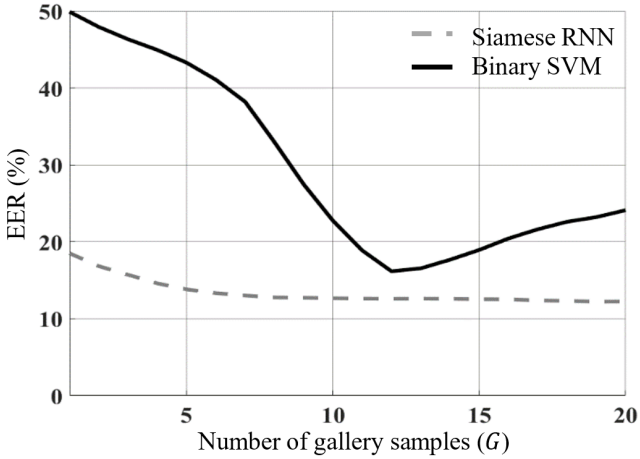

Figure 4.1: *Authentication based on touchscreen signals (single swipe): Error Rates (%) for increasing number of gallery samples (G) employed to model each user.*

G used to compute the score for each user, as the average of the Euclidean distances between the gallery samples $\mathbf{f}(\mathbf{s}_g)$ and the unknown sample $\mathbf{f}(\mathbf{s}_u)$. For one-time authentication ($G = 1$) our proposed system achieves an EER of 19%, and the performance improves when scaling up the number of gallery samples. In fact, with 6 gallery samples the EER is reduced to 13%, with no significant improvements for larger G. Comparing with the SVM architecture, we can observe that the Siamese RNN architecture obtains much better results. These results prove the richness of touch gestures to model the interaction between humans and smartphones, in particular for user authentication. With a simple gesture (drag and drop) we have built an authentication system with good performance. Note that this performance is achieved under uncontrolled conditions including almost 600 different devices and non-supervised acquisition. Although these error rates can be considered high for some applications (e.g., in comparison with fingerprint or face authentication), the authentication based on touch gestures can be useful in continuous authentication scenarios where identity management is based on multiple evidences evaluated in a transparent setup.

4.3. Multimodal Authentication Approach

Previous works fusing different modalities ([Deb *et al.*, 2019; Fridman *et al.*, 2016; Li and Bours, 2018a]) have focused their approach on obtaining time windows from the different modalities and then carry out the fusion. However, this does not represent a realistic scenario due to not all modalities fused can always be captured in a specific time windows. In this approach we go a step forward by merging the modalities at session level (time during an unlock and the next lock of the device), and therefore fusing only the modalities available at each session.

We analyze two architectures for user authentication (see Fig. 4.2) according the two scenarios proposed: the first scenario (continuous line in Fig. 4.2), referred to as One-Time Authentication (OTA) is based on unimodal systems trained with the information extracted from

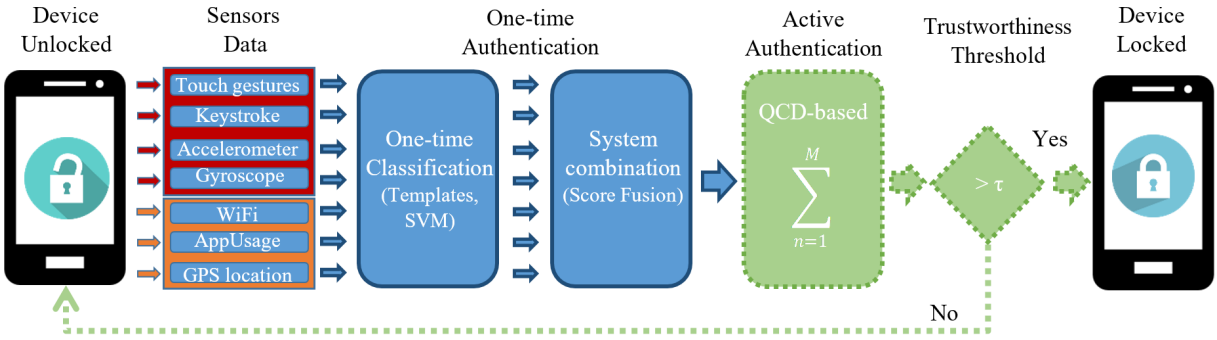

Figure 4.2: *The pipeline of the multimodal approach proposed for mobile authentication. Continuous line corresponds to one-time authentication, and dotted line indicates add-on modules for active authentication.*

the mobile sensors during a user session. A session is defined as the elapsed period between the device unlock and the next lock. Therefore, sessions have a variable duration and information obtained from sensors varies depending on the usage of the device during the session. The information provided by the sensors is employed to model the user according to seven systems: keystroke, touch gestures, accelerometer, gyroscope, Wi-Fi, app usage, and GPS location. Each system provides a single authentication score and these scores are combined to generate a unique score for each session. The second scenario, called Active Authentication (dotted line in Fig. 4.2), is based on updating a confidence value generated from the One-Time Authentication during consecutive sessions.

The 7 systems are categorized into two main groups according to the nature of the information employed to model the user: biometric and behavior-based profiling systems. In this approach, biometric systems refer to the top 4 channels in the Sensors Data module of Fig. 4.2 (red box). The way we realize touch gestures, typing, or handle the device is determined by behavioral aspects (e.g., emotional state, attention) and neuromotor characteristics of users (e.g., ergonomic, muscles activation/deactivation timing, motor abilities). Behavioral-based profiling refers to those systems that model the owners of the device according to the services they use during their daily habits (orange box in Fig. 4.2, bottom 3 channels in the Sensors Data module).

4.3.1. Experimental Protocol

Wi-Fi, app usage, and GPS location authentication systems are based on a similar template-based matching algorithm. A user template is defined as a table containing the time stamps and the frequency of the events [Li and Bours, 2018c]. For this, we divided the time (24 hours of the day) into N equal time slots (e.g., if we choose $N = 48$ we will have 48 time slots of 30 minutes), giving to each time slot a number ID. Then we store in the template the event's name, the number ID of the time slot and the occurrence frequency of that event (number of times this event occurs during this particular time slot on a window of consecutive days). Table 4.2 shows an example of the app-usage template for a given user generated according the data obtained during 6 days; in this case '*WhatsApp*' application is detected in the fourth slot for five days out of the 6 days considered meanwhile the same app is detected only one day in the fifth slot.

Event	Time Slot	Frequency
WhatsApp	4	5
Navigator	4	3
YouTube	5	1
WhatsApp	5	1
Facebook	7	2

Table 4.2: *Example of an app-usage user template generated according the data captured during six days.*

Note that multiple detections of the same event in the same time slot and day are ignored but they are stored if they belong to different time slots or days. Depending on the system, the event could be the name of the Wi-Fi network, latitude and longitude of a location (with two decimals of accuracy), or the name of a mobile app for Wi-Fi, GPS location, and app usage systems, respectively. Finally, we test the systems by calculating a behavior-based confidence score [Li and Bours, 2018c] for each test session as:

$$score = \sum_{i=1}^{S} f_i^2 \tag{4.1}$$

where f_i^2 is the frequency of the event stored in the template that match with the test event i in the same time slot and S is the total number of events detected in that test session. For example, if the test session includes the usage of '*WhatsApp*' and '*Navigator*' apps during the fourth slot, the score confidence will be $5^2 + 3^2 = 34$ (according to the template showed in Table 4.2). Based on this, a higher score in the test session implies higher confidence for authentication.

For touch gestures, keystroke, accelerometer and gyroscope systems, the feature extraction and classification algorithms are adapted to model the user information. The features employed in the system based on touch gestures is a reduced set of the global features presented in [Martinez-Diaz *et al.*, 2014] (commonly used for online handwriting sequence modeling) and adapted for swipe biometrics in [Fierrez *et al.*, 2018]. Mean velocity, max acceleration, distance between adjacent points, or total duration are some examples of this subset of 28 features extracted from the $\{\mathbf{x}, \mathbf{y}\}$ touch coordinates (see [Martinez-Diaz *et al.*, 2014] for details).

For accelerometer and gyroscope, the data captured are comprised of the $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ coordinates of the inclination vector of the device (gyroscope) and the acceleration vector (accelerometer) in each time stamp. For these two sensors we use the feature set proposed in [Li and Bours, 2018b]: mean, median, maximum, minimum, distance between maximum and minimum, and the standard deviation for each array of coordinates. Moreover, we propose the 1 and 99 percentiles and the distance between them as additional features. Regarding keystroke dynamics, the keys pressed were encrypted in order to ensure users' privacy. Thus, systems based on graphs were discarded and we adopted traditional timing features: hold time, press-press latency, and press-release latency as in [Morales *et al.*, 2016; O'Neal *et al.*, 2016]. Finally, we

System	Single	+Wi-Fi	+GPS	+AppUsage	All	AA
Touch gestures	72.0	78.2	78.3	75.4	83.1	95.0(6)
keystroke	62.5	72.6	70.9	67.8	79.1	92.9(7)
Accelerometer	61.3	70.8	77.3	64.7	78.7	93.7(7)
Gyroscope	59.5	69.7	72.6	63.4	78.4	92.3(6)
Combined	73.2	77.3	78.9	75.3	82.2	97.1(5)

Table 4.3: *Results achieved for both OTA and AA scenarios in terms of accuracy (%) according to different number of biometric systems and their fusion with behavior-based profiling systems. In brackets, average number of sessions employed (ADD).*

propose a feature set based on 6 statics (mean, median, standard deviation, 1 percentile, 99 percentile, and $99-1$ percentile). For classification we train different SVM with a Radial Basis Function (RBF) kernel, one for each biometric system and user with an optimization of both hyperparameters (C, σ).

As commented before, the experiments are divided into two different scenarios: One-Time Authentication (OTA) and Active Authentication (AA). In OTA experiments, all 7 systems are trained separately for each user and the scores are calculated at session level, generating 7 scores for each test session as maximum (note that the number of systems available during a session varies). The 4 biometric systems considered can produce more than one score per session (e.g. multiple gestures or multiple keystroke sequences during a text chat). In those cases, the scores available during the session are averaged to obtain one score for each biometric system and session. Finally, we normalize with *tanh* normalization and fuse the scores (mean rule) to calculate a single score [Fierrez *et al.*, 2018b] according to the different fusion set-ups proposed. The scores from the best fusion set-up will be used in the AA scenario. For AA scenario, we employ multiple consecutive sessions in order to improve the confidence in the authentication by updating a confidence score based on the scores of previous sessions (see Sec. 2.2.2 for AA algorithm details).

All experiments were conducted with the UMDAA-02 database (described in Sec. 2.1.5) and dividing the database into 60% days for training (first sessions) and the remaining 40% days for testing. This means that we employ 6 days in average to model the user and 4 days in average to test such a model. The performance for both scenarios is presented in terms of accuracy computed as $100 - EER$ (Equal Error Rate).

4.3.2. Results and Discussion

Table 4.3 summarizes the results for OTA scenario by ranking from the best individual biometric system performance to the worst one. The first column shows the performance obtained for each single biometric system. From the second to the fourth column, we show the performance for the fusion of each biometric system with each behavior-based profiling system, and the fifth column shows the fusion with all of them. Firstly, the poor performance achieved by

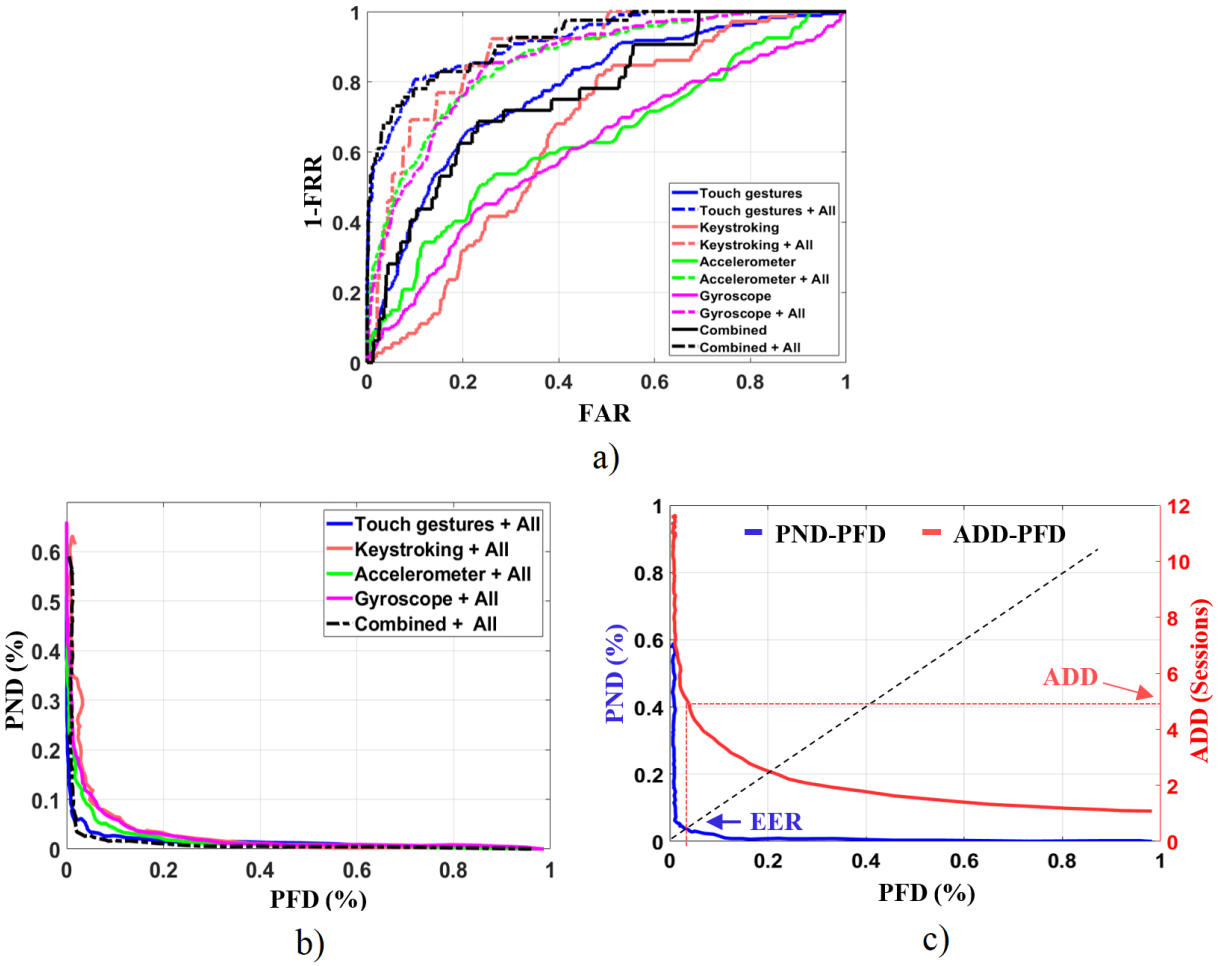

Figure 4.3: *ROC curves (a) in OTA scenario for individual biometrics and the best fusion set-up incorporating the three considered behavior profiling sources (All = Wi-Fi + GPS + App usage). PND vs PFD curves of active authentication for the best fusion schemes (b), PND vs PFD and ADD vs PFD curves for the best fusion set-up (c). The dark dashed line shows the EER and the red one shows the Average Detection Delay (ADD) for that EER in the lower plot.*

some biometric systems can be caused by the uncontrolled acquisition conditions of the database and the limited number of samples per session (e.g., free text keystroke usually requires large sequences) but the combination of all of them (last row in Table 4.3) shows acceptable performance for unsupervised scenarios. Secondly, we can observe that behavior-based profiling systems always improve biometric system performances in all fusion schemes. In fact, the combination of all behavior-based profiling approaches with each biometric system achieves the most competitive performance, improving them in more than 18% of accuracy in the best of cases. If we analyze each single behavior-based profiling fusion, we can observe that the GPS system achieves the best improvements, boosting biometric systems performances in more than 13% of accuracy.

Finally, in Fig. 4.3.(a) we plot the Receiver Operating Characteristic (ROC) curves for each single biometric system and the best fusion set-up (i.e., the fusion of all behavior-based profiling

systems with each biometric system, column 5 in Table 4.3). The results in OTA scenario suggest that behavior-based profiling systems always improve the biometric ones and the best performance is achieved by fusing with all of them, and therefore, the scores obtained from this fusion scheme will be use in AA scenario.

To calculate the correct classification rate in AA we plot in Fig. 4.3.(b) and Fig. 4.3.(c) the PND vs. PFD and the ADD vs. PFD curves respectively. The PND-PFD curves are similar to FMR-FNMR curve in one-time authentication with the main difference that those results are obtained from a sequence of stacked scores instead of only one. The EER will be the value where PND and PFD are equal and the accuracy will be computed as $100 - EER$. The ADD-PFD curve shows the number of sessions needed to detect an intruder according to the PFD. This curve allows us to know how many sessions are needed to achieve the EER reported. For instance, the PND-PFD curves in Fig. 4.3.(c) show that the EER in Active Authentication is 2.9% for an ADD equal to 5 sessions. These results suggest that we can improve OTA results at the cost of having more sessions to detect an intruder. All curves were calculated for each user and averaged.

Finally, all AA results are summarized in the last column of Table 4.3. Remember that scores employed in the QCD-based algorithm come from the fusion scores of the best OTA scenario (fusing with all behavior-based profiling systems) so both performances are correlated. Each performance in Table 4.3 for AA is followed by the average detection delay in brackets needed to achieve it. As we expected, in all different fusion set-ups the AA algorithm improves the accuracy at the cost of needing more sessions to detect the intruder. In fact, for the best fusion set-up the performance improves from 82.2% to 97.1% by using 5 consecutive intruder sessions to detect the impostor. Comparing all scenarios, the greatest improvement occurs with all biometric systems combined (14.9% of improvement in the last row of Table 4.3) with an average 5 sessions.

4.4. Chapter Summary and Conclusions

We have explored the potential of mobile devices to model human-machine interaction. We presented a taxonomy of applications that can exploit the signals originated in those sensors in three different dimensions, depending on the main information content embedded in the signal or signals exploited in the application: neuromotor skills, cognitive functions, and behaviors/routines. We have overviewed the databases employed in the literature. These databases have been used traditionally for user authentication, but they provide signals useful for other applications as well beyond security and related to human behavior analysis. As example application, we experimented with HuMIdb, which to the best of our knowledge is the largest database of mobile sensor signals acquired during human mobile interaction to date, with 14 sensor signals collected from 600 users across 5 sessions and more than 600 devices involved. For that experiments we introduced a new method for user authentication based only on one touch gesture (drag and drop) and RNNs resulting in an error rate of 13%.

In another example application, we have developed a multimodal mobile authentication system that include up to four different biometric traits (touch gestures, keystroke, gyroscope, and accelerometer) and three behavioral-based profiling techniques (GPS, Wi-Fi, and app usage). The experiments were conducted on the UMDAA-02 mobile database, a challenging dataset acquired under uncontrolled conditions. Our results over 97.1% of accuracy when combining all data channels in an active authentication scenario show the potential of multimodal mobile authentication approaches based on biometric signal processing.

Part III

Modelling Biometrics Device Interaction for Health & Behaviour Applications through Neuromotor Analysis

Chapter 5

Modelling Human Interactions for Children Detection

In this chapter we analyse a way to classify subjects with touchscreen gestures according to two age groups: children and adults. We will also apply AA algorithms in order to take advantage of physiological and behavioural mannerism of the subject while interacting with a touchscreen device during a short period of time to detect a change in the subject's profile. Such mannerisms are often distinctive among different subjects, they are stable over a period of time and difficult to mimic [Perera and Patel, 2017b]. Therefore, AA systems are well protected against spoofing or hacking and recent works have shown that they report better results than OTA systems [Patel *et al.*, 2016]. The main objective of the method proposed in this chapter is to detect a child with the minimum possible delay from the moment he or she starts using a touch based device. Thanks to the usage of AA systems, in this context it would be possible for example to adequate the content shown on the screen for the new subject profile instantly, avoiding locking the session and asking for a password or traditional parental control systems.

The chapter is organized as follows: we first summarize in Sec. 5.1 related works in age detection. Then, we present the system proposed in Sec. 5.2 and evaluates its performance for OTA and AA scenarios in Sec. 5.3.

5.1. State-of-the-art on Age Detection

In the existing literature, there are many experiments exploring the use of technology by children, seeking how to improve the design of adapted interfaces and applications [McKnight and Cassidy, 2012]. However, modelling and characterizing mathematically how children interact with touch devices and how their conduct differs from the adult's one is a field that has not been studied deeply enough. A work related to this topic is [Aziz *et al.*, 2013] where the authors analysed different types of touch tasks like tap, rotate or drag and drop, and they found that children have different success rates when trying to perform different tasks. Simple

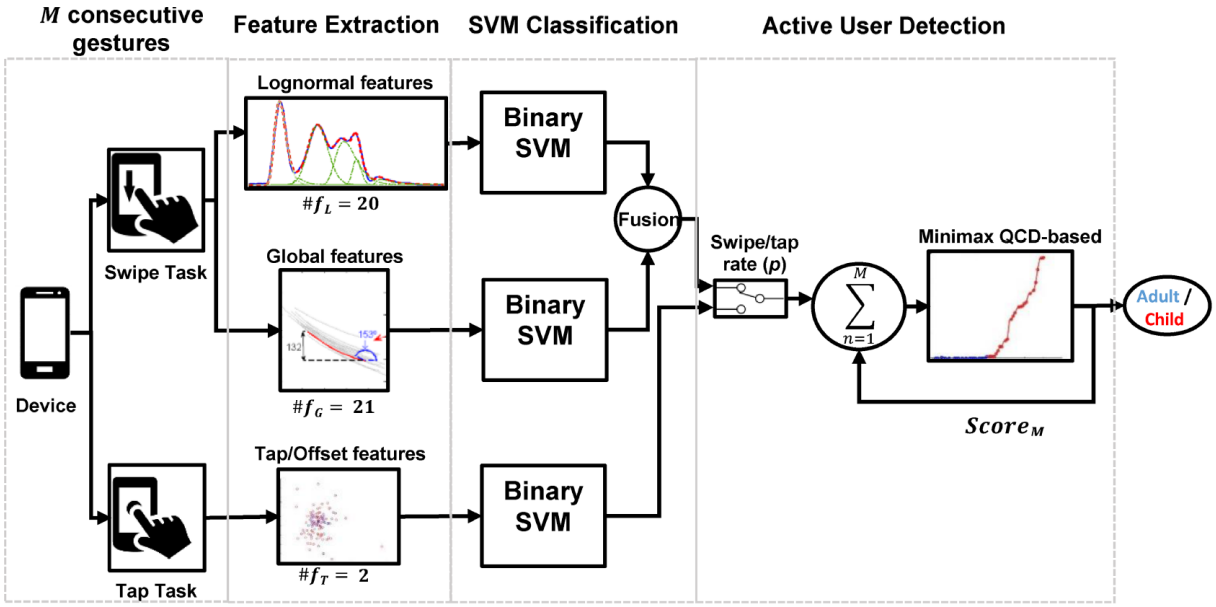

Figure 5.1: *For each sequence of M input consecutive touch gestures, three feature sets are generated: Lognormal (f_L), Global (f_G), and Tap/Offset (f_T)*

tasks (e.g., swiping, tapping) can be done by all children without any problem, but the more complex ones are very difficult to complete for short age children. In [Anthony *et al.*, 2012], the authors measured the touch gestures of children and compared it to gestures from adults. They discovered that children have a larger miss rate compared to adults when trying to hit small targets. The difference between adults and children is mainly caused by the different grade of maturity of their anatomy and neuromotor system. These features are less mature in children, so they have worse manual dexterity causing rougher movements [Inhelder and Piaget, 1969; O'Reilly and Plamondon, 2009]. In a complementary case of our study [Suleyman *et al.*, 2015], the authors show high classification rates between young adults (20-50 years) and older adults (70+ years) based on touch-gestures, demonstrating differences in neuromotor skills during human ageing whilst our work studies differences between undeveloped neuromotor skills in children and total maturity in young adults. In [Bevan and Fraser, 2016], the authors show that people with long thumb complete swipe gesture over a smartphone faster than those with shorter thumb. This could be a key to identify children due to their shorter thumbs and therefore longer time task.

5.2. Experimental Protocol

We will analyse two different types of touchscreen gestures: swipe and tap. In swipe tasks, subjects slide their finger over the screen, while tap tasks consist on tapping the touchscreen for a short period of time. We choose these gestures because they are the most common ones in touchscreen interaction and they are easy to be performed by children. To do this, we use information of swipe and tap patterns from a publicly and available database [Vatavu *et al.*,

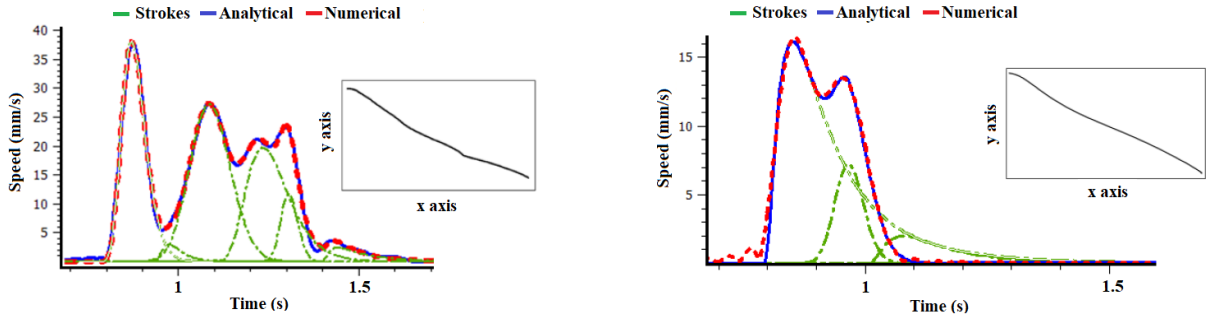

Figure 5.2: *Child (left) and adult (right) speed profiles from a touchscreen pattern (swipe). Numerical: is the captured velocity signal $|\vec{v}(t)|$ the touch activity (input of the model). Analytical: is the reconstructed Sigma Lognormal velocity $|\vec{v_r}(t)|$ profile (output of the model). Strokes: is the decomposition in individual strokes of the model $|\vec{v_i}(t)|$.*

2015b] comprised of 119 subjects (89 children and 30 adults) using two different types of devices: a smartphone and a tablet (see Sec. 2.1.4 for more database details) .

The architecture proposed is divided into three consecutive stages depicted in Fig. 5.1: feature extraction to compute the most suitable features for each touch-based task, SVM classification to classify between child or adult for OTA scenario, and AA scenario where a sequence of M touch gestures performed during the interaction with the device is taken into account to decide whether it has been produced by an adult or a child.

5.2.1. Feature Extraction

The feature extraction approach followed depends on the task performed: tap or swipe. For swipe tasks we use two feature approaches, one based on the Sigma-Lognormal model (already presented in Sec. 2.2.1) and a different one based on global features. It is worth noting that Sigma-Lognormal features extract information related to neuromotor skills involved in the action performed, meanwhile global features extract holistic information from the trajectory of the stroke performed. For tap tasks we follow a feature approach based on Tap/Offset features, proposed in [Suleyman *et al.*, 2015] due to they are more suitable for tap gestures (few samples, lack of fine movement).

- *Sigma-Lognormal features:* studies like [Duval *et al.*, 2015; Meulenbroek and Van Galen, 1988] have proved that the Sigma-Lognormal model can be used to characterize children handwriting. They conclude there are two main groups of children that are separable by looking at their learning stage. Children's neuromotor skills become more similar to the adults' skills when they grow up, namely, when they finish their preoperational stage. At age 10 children know how to activate each little muscle properly to produce determinate fine movements [Vatavu *et al.*, 2015b]. As they are based on the same neuromotor skills, the principles applied to the handwriting models can be used to model touchscreen patterns.

  In Fig. 5.2, the speed profiles of an example of touchscreen pattern is shown. The numerical signal $|\vec{v}(t)|$ is the velocity profile of the touchscreen pattern acquired by the device, which

Space-based Features	Time-based Features	General Features
$f_1 = D_i$	$f_8 = \Delta t_0 = t_{0_i} - t_{0_{i-1}}$	f_{19} = Task Time
$f_2 = \mu_i$	$f_9 = v_2 = \lvert\vec{v_i}\,(t_{2_i})\rvert$	f_{20} = # of lognormals
$f_3 = \sigma_i$	$f_{10} = v_3 = \lvert\vec{v_i}\,(t_{3_i})\rvert$	
$f_4 = \sin\theta_{s_i}$	$f_{11} = v_4 = \lvert\vec{v_i}\,(t_{4_i})\rvert$	
$f_5 = \cos\theta_{s_i}$	$f_{12} = \delta t_{05} = t_{5_i} - t_{0_i}$	
$f_6 = \sin\theta_{e_i}$	$f_{13} = \delta t_{15} = t_{5_i} - t_{1_i}$	
$f_7 = \cos\theta_{e_i}$	$f_{14} = \delta t_{13} = t_{3_i} - t_{1_i}$	
	$f_{15} = \delta t_{35} = t_{5_i} - t_{3_i}$	
	$f_{16} = \delta t_{24} = t_{4_i} - t_{2_i}$	
	$f_{17} = \Delta t_1 = t_{1_i} - t_{1_{i-1}}$	
	$f_{18} = \Delta t_3 = t_{3_i} - t_{3_{i-1}}$	

Table 5.1: *Sigma-Lognormal model extracted features. These features are calculated for each lognormal of the decomposition of the numerical signal* $|\vec{v_i}\,(t)|$.

is employed as input of the Sigma Lognormal model. The model decomposes this velocity profile into individual strokes $|\vec{v_i}\,(t)|$, each stroke represents a Lognormal signal with their own parameters. The analytical signal $|\vec{v_r}\,(t)|$ is calculated as the summation of these individuals strokes extracted from the numerical signal (see Sec. 2.2.1 for more details). Finally, the parameters of the Sigma-lognormal model (see Table 2.3) have been adapted to calculate 18 different features that can be used to depict the neuromotor properties of the subjects [Fischer and Plamondon, 2015]. Table 5.1 summarize these features and classify them into three different groups according to their temporal or spatial nature. Additionally, the task time and the number of lognormals in each task have been added as features number 19 and 20. Note that every swipe is composed of at least one lognormal with its own parameters, and therefore, their own features so a combination of features is needed to obtain a single value for each feature in those swipe with multiple lognormals (i.e., multiple $|\vec{v_i}\,(t)|$). In this work, the values of the features have been combined by computing the arithmetic mean of the features obtained from each $|\vec{v_i}\,(t)|$.

Quite often it is possible to differentiate between children and adults by simply looking at the velocity profile of a touchscreen task. A visual comparison between children and adults speed profiles (Fig. 5.2 left and right respectively) shows that children speed signals are composed by a higher number of strokes than the adult's signals. The larger maturity on the neuromotor skills of adults produces soft velocity profiles that reveals a fine control of the movements.

- *Global features:* the global features set refers to those features calculated from the entire touch task pattern, such as the mean velocity, max acceleration or total duration. For this purpose, many global features set have been proposed in the literature [Jain *et al.*, 2005; Martinez-Diaz *et al.*, 2014; Serwadda *et al.*, 2013] for signature verification. We use the 28-dimensional features set applied in [Fierrez *et al.*, 2018] due to good results obtained

Parameter	Description
T	Task Time
v', v_σ, $v_{1_{st}}$, $v_{2_{st}}$, $v_{3_{st}}$	Velocity: mean, standard deviation, first quartile, second quartile and third quartile.
a', a_σ, $a_{1_{st}}$, $a_{2_{st}}$, $a_{3_{st}}$	Acceleration: mean, standard deviation, first quartile, second quartile and third quartile.
d_{N-1}	Distance between end points.
θ	Angle between line and horizontal axis.
$\sum_{i=0}^{N} d_i$	Summation of distance between adjacent points.
d_x, d_y	Distances between mean and min point.
σ_{ax}, σ_{ay}	Standard deviation of x and y axis acceleration.
H, V	Horizontal and vertical span ratio.
A	Swipe area.

Table 5.2: *Global features set.*

in swipe patterns, but with some limitations (pressure measurements were not acquired in this database). After removing the features related to pressure, a 21-dimensional feature vector was computed, as shown in Table 5.2.

- *Tap/Offset features:* tap tasks are characterized by two features: *i)* the distance between the target point and the point touched (Offset-distance); and *ii)* the time that the subject touches the screen during the tap task (Tap-time).

Finally, we use feature selection based on Sequential Forward Floating Search (SFFS) to calculate the best subset of features for each feature set in swipe task. Up to 5 features were extracted to achieve the best result for lognormals features: f_1, f_2,f_9, f_{19}, f_{20} and other 5 features where selected from the global feature set: v', $v_{2_{st}}$, $v_{3_{st}}$, $a_{2_{st}}$, $a_{3_{st}}$.

5.2.2. Classification

As a classifier we use SVM with a RBF (with $C = 30$ and $\sigma = 10$) kernel because of its good general performance in binary classification tasks. As showed in Fig. 5.1, three binary SVM classifiers were implemented, one for each feature set. Four different classification algorithms were tested: Decision Tree, k-Nearest Neighbours (kNN), SVM-RBF and Logistic Regression. We choose SVM-RBF due to it achieves the best and stables results in all scenarios. Each SVM is trained using samples from children and adults over the training data. Owing to we have two different feature sets for swipe task, we compute the final score by merging both feature sets

following two fusion approaches: fusion at feature level and fusion at score level. The two fusion approaches followed for swipe task achieved similar performance independently (as later we will see in Sec. 5.3). For fusion at feature level, both sets of features were concatenated to form the feature vector. For fusion at the score level, the final score is obtained as the average of the previous two.

As commented before, the experiments were divided into two well differentiated scenarios: one-time authentication (OTA) and active authentication (AA). OTA refers to the features extraction and SVM classifiers experiments, where only one sample is used to discriminate among children and adults. For this, the subjects have been separated randomly in training (60%) and test (40%). It is guaranteed that subjects (children and adults) employed for training are different from those employed for test (open-set classification paradigm). The OTA experiments were repeated 50 times and the final performance is presented in terms of average correct classification rate computed as $100 - EER$. Owing to the higher number of children tasks in the database compared with the adults, selecting a percentage of the total subjects makes the two scenarios to be unbalanced. Experiments balancing the number of both classes in training and testing processes have been made. Nevertheless, the results show small variations around 1% of accuracy (variation that can be related to the statistical variation due to the data set).

On the other hand, in AA experiments we simulate a sequence of events (i.e., sequences of swipe and tap gestures) during a period of time to detect a change in the subject profile (from adult to child or vice versa). To simulate this change in the subject profile, we build sample sequences with a first half of adult samples and the second half of child samples, very similar to the sample sequence showed previously in Fig. 2.3 (left). Moreover, the rate between taps and swipe in each sequence ρ can vary depending on the application used (remember that each sample of the sequence could be a tap or swipe); for instance, in reading applications swipe gestures are more common than taps gestures meanwhile, in videogames applications it would be the opposite. The different combinations have a significant impact in results due to tap gestures have a worse performance in one-time subject detection and it could yield a drop of performance for active detection in those sequences where tap gestures are more common. To analyse the effects of this rate we separate the experiments in three different set-ups taking into account the percentage between tap and swipe gestures in the sample sequences made. Remark that active detection samples are the scores from the one-time detection system (see Fig. 5.1), so the performance of the first one is crucial for both scenarios.

5.3. Results and Discussion

5.3.1. One-time Detection

In Table 5.3 we summarize all experiments performed in OTA scenario. For swipe tasks, the best result was achieved with fusion at the score level between the two approaches based on Global features and Sigma-Lognormal features. The mean value of correct classification

Device	Our implementation of Vatavu *et al.* [2015a]	Swipe				Tap
		Sigma-Lognormal	Global	Feature Fusion	Score Fusion	Tap
Mobile	86.5	93.6	92.1	92.7	94.1	85.4
Tablet	90.5	96.3	94.5	94.9	96.5	80.0

Table 5.3: *Results achieved for each OTA system in terms of correct classification rate (%).*

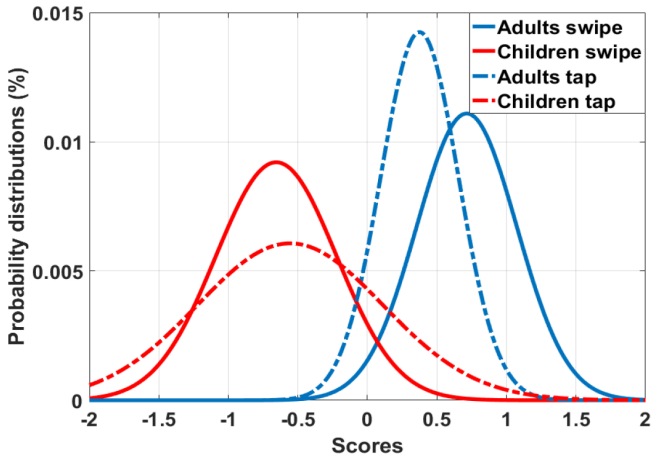

Figure 5.3: *Probability distribution of adults and children for tap and swipe tasks with phone device.*

rate having into account all evaluated scenarios and both devices is 94.4%. The best results are obtained with tablets as sensors, while when using smartphone's data slightly worse results are achieved. Note that swipe gestures have longer trajectories in tablet screens compared to smartphone screens. Larger movements imply more information available to classify subjects. However, tap gestures do not take advantage of the screen's size due to the target point has the same size in both devices (remember that the distance between the target point and the point touched is one of the two features in Tap/Offset feature set). Fig. 5.3 shows the probability distribution functions of the scores calculated in the classification process for both swipe and tap task in phone device. For swipe task, scores from children and adults are visibly separated into two different zones, making possible to get high accuracy rates (over 93%). There are also other zones where the score distributions overlap. These regions are the source of incorrect classifications. Combining scores from several samples (as we will see later in AA scenario) of the same subject could reduce the overlapping areas, increasing even more the accuracy rate. Regarding tap task, both probability distributions show greater overlap causing a worse performance.

Table 5.3 also shows the classification accuracies obtained from the method proposed in [Vatavu *et al.*, 2015a]. The improvement in accuracy rate can be associated to the better discrimination due to a combination of Lognormal and global features that describes better touched based gestures, while in [Vatavu *et al.*, 2015a] their features are basically related to the precision of the gestures. Finally, the age of children is a key factor to take into account in classification

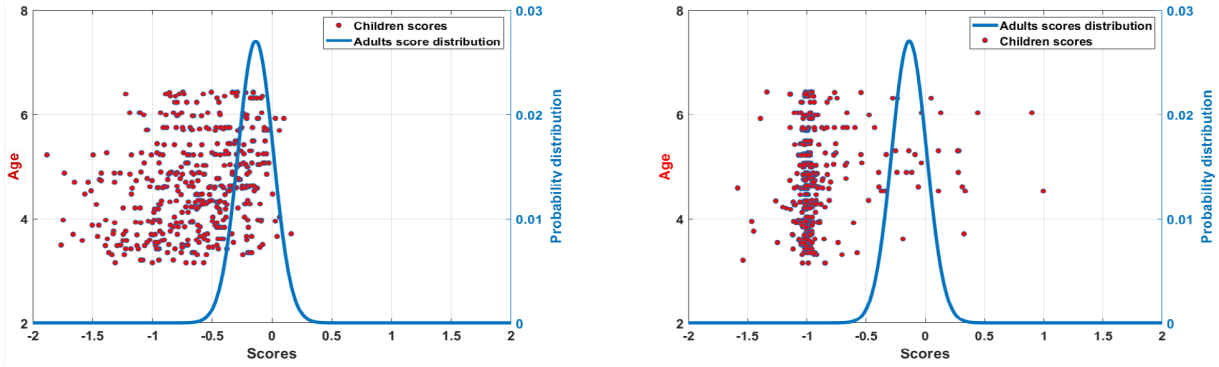

Figure 5.4: *Probability distribution of adults (right y-axis) and children scores sorted by age (left y-axis) for swipe task (right) and tap task (left) with tablet device.*

performance. Neuromotor skills in children become more similar to the adult ones as they grow up. At the age of ten, children have their neuromotor skills completely developed making the classification task more complicated [Duval *et al.*, 2015], in order to analyse the impact of the growing we compare children scores by age obtained in the classification task with the distributions of adults scores, expecting to observe a similarity between adults and children as children are older. Fig. 5.4 shows the scores obtained by the SVM of the children in tap and swipe task sorted by age (left y-axis). Besides, the distribution of adults scores is plotted to easy comparison. We can observe that children scores get close to the adult ones as children grow up due to maturity of their neuromotor skills, especially in tap task. In accuracy terms, we divide children into three age groups: under 4 years old, between $4-5$ and older than 5; the accuracies in tablet device were 85.0%, 80.8% and 76.3% respectively in tap task and 98.2%, 96.5% and 93.2% respectively in swipe task.

5.3.2. Active Detection

As mentioned before, the swipe/task rate (ρ) could be crucial for AA results due to worse performance in tap tasks. Real subject interaction with touch screens involves swipe and tap gestures. Thus, we decided to distinguish among three set-ups taking into account different percentage of swipe and tap events:

- *First set-up* ($\rho \leq 0.25$)*:* sample sequences with a 25% or less of swipe gestures.
- *Second set-up* ($0.25 < \rho < 0.75$)*:* sample sequences with swipes and tap gestures balanced.
- *Third set-up* ($\rho \geq 0.75$)*:* sample sequences with a majority of swipe gestures.

The swipe and tap gestures from each subject are randomly chosen for all three set-ups to build each sample sequence. The experiments are repeated up to 100 times so we have 100 different samples sequences for each subject. In order to analyse the AA system performance, we present ADD, PFD and PND curves shown in Fig. 5.5. All curves were calculated individually for each subject, and finally averaged.

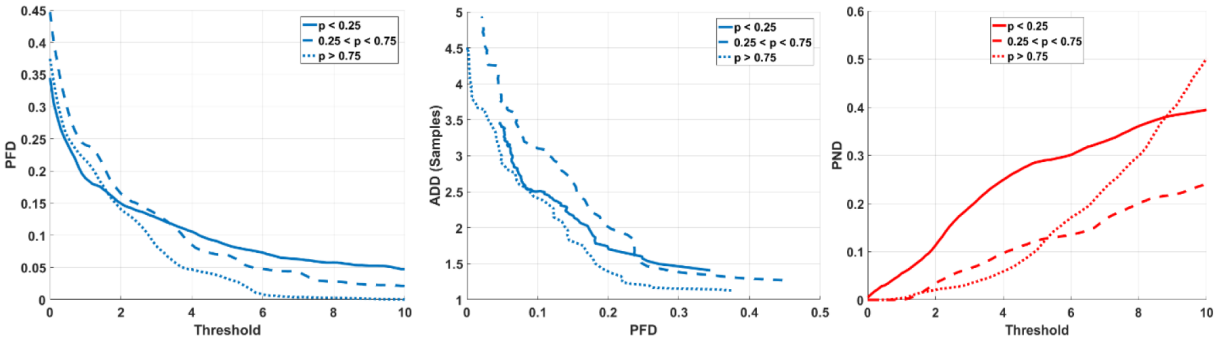

Figure 5.5: *PFD (left), PFD ADD (middle) and PND (right) curves for smartphone.*

PFD curves show how many adults in percentage are identified as children (false detections). All set-ups show similar PFD performance; as it is expected, false detections decrease when thresholds increase. Furthermore, the third set-up ($\rho > 0.75$) where swipes gestures are more common decrease faster due to having a better performance in OTA scenarios for swipe, so false detection are relatively uncommon among swipe gestures. In addition, ADD-PFD curve denotes how many samples are necessary to identify a child on average depending on false adult detections. This quantity is a significant factor to take into account when an AA system is designed. The system tries to identify a child with the minimum amount of samples as possible in order to reduce the time delay (time between the child starts to operate the device till he is detected) but avoiding false detections as well. It can be seen that the number of samples necessary to identify a child increases when we decrease the false detection, so there is always a trade-off between both curves. Moreover, ADD curve for the third set-up ($\rho > 0.75$) has better performance again.

PND curves depicts the percentage of children which are not detected by the system. In this case, the first set-up ($\rho < 0.25$) increases faster with the threshold. Regarding the third set-up we expected to achieve the best results but it tends to obtain the highest PND rate with high thresholds. The main reason of this effect is that the lack of swipe samples in some children suggest that they would never be detect by the system for high thresholds. Note that the third set-up has the best results for low thresholds where few samples are enough to reach it. It can be seen that there is always a trade-off between false child detection (PFD) and non-child detection (PND), we can decrease the false adult detection at the cost of having more children who are not detected by the system and vice versa. Therefore, performance will vary depending on the system design and application.

The PFD-PND curves showed in Fig. 5.6 are useful performance metrics to analyse AA systems. Note that PFD/PND in active detection are similar to FMR/FNMR in OTA results. The main difference is PFD and PND curves are obtained from a sequence of stacked scores meanwhile FMR/FNMR come from OTA scenarios. In order to differentiate among both cases, we decided to keep the same nomenclature as in [Duval *et al.*, 2015]. In these curves we can appreciate the trade-off effects: reducing the false child detection rates (PFD) makes the system more prone to non-child detection (PND) as a consequence. In this figure we can analyse better

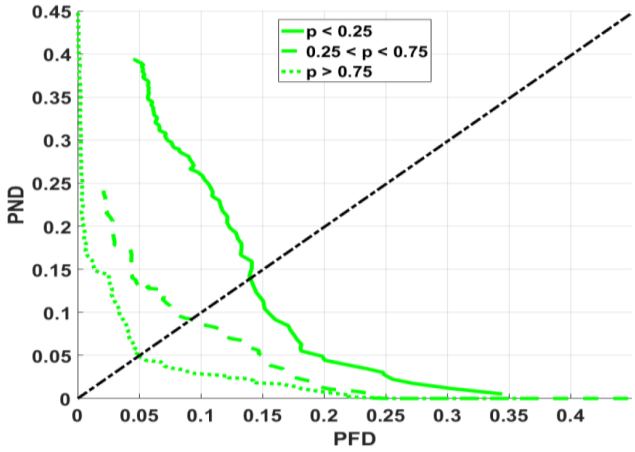

Figure 5.6: *Probability of non-detection (PND) vs probability of false detection (PFD) with smartphone device. Points where curves cross the black line are the EER values.*

	Active Detection					One-time Detection	
Device	$\rho = 0$	$\rho < 0.25$	$0.25 \leq \rho \leq 0.75$	$\rho > 0.75$	$\rho = 1$	**Swipe**	**Tap**
Mobile	86.2	86.2	91.1	94.5	95.6	94.1	85.4
Tablet	82.3	82.7	89.5	95.0	97.0	96.5	80.0

Table 5.4: *Results achieved in correct classification rate terms (%) for both one-time detection and active subject detection algorithms.*

the ρ rates effects over the sample sequence. The case having more swipe than tap gestures yielded better performance as expected.

Finally, Table 5.4 summarises the correct classification rates, computed as the opposite of the EER (the points where curves in Fig. 5.6 cross the black line). Each correct classification rate has been calculated independently for each device and ρ rates. This table shows that AA results are slightly better in smartphone devices due to it achieves better results in tap gestures. In fact, the difference between smartphone and tablet devices tends to decrease when tap gestures are less common. Besides, the correct classification rates for OTA scenarios were added to the table to compare among algorithms. It can be seen that AA results are always between OTA results: results in AA scenario where most of samples are swipe are close to swipe results in OTA scenario meanwhile AA algorithm improves OTA results when tap gestures are more common. In fact, if we only consider swipe gestures ($\rho = 1$) or tap gestures ($\rho = 0$) in AA, the results improve OTA marks so AA algorithms could improve OTA systems at the cost of having more time to detect a child.

5.4. Chapter Summary and Conclusions

In this chapter we have studied an age classification algorithm based on swipe and tap gestures. For this, we have employed three feature sets: *i)* one based on the parameters of the

sigma-lognormal model that characterize better the neuromotor skills of the subjects, making possible to discern between children and adults, *ii)* the global features set that extract features from the whole gesture, and *iii)* two features extracted for the tap gestures. An evaluation of performance has been made, using a public database with touchscreen activity of both children and adults. Depending on the type of gesture, our methods achieve accuracies ranging between 80% and 96.5%. Secondly, we developed an active detection system aimed to detect a child with the minimum delay as possible from the time he starts to interact with the device. We simulate this situation generating sequences of tap and swipe gestures during a period of time. Depending on the type of sequences, our methods achieve accuracies ranging between 82.3% and 97%. These accuracies can be obtained using features extracted from only four simple touch gestures made by adults or children. Although our error rates are very low, the results should be interpreted with care. A major limitation of this work comes from the database used, as it does not contain data from subjects with ages between 6 and 25 years old. To the best of our knowledge, there is no such database with touch screen interaction data available to the research community.

Chapter 6

Characterization of the Handwriting Skills for Parkinson Detection

In this chapter we analyse a new set of handwriting features as potential biomarkers to model PD. For this, we employ a novel database with data acquired from PD patients and healthy control (HC) subjects during on-line handwriting tasks distributed in a 3 years time span (see Sec. 2.1.6 for more database details). To the best of our knowledge, this is the largest handwriting database for PD collected until now with 150 subjects (between PD and controls subjects) and a total of 935 handwriting tasks. Additionally, we propose a new benchmark for the evaluation of the different handwriting tasks (individually and merged) to classify between PD and HC considering three feature sets: kinematics, nonlinear dynamics, and neuromotor feature sets; and three different classifiers: kNN, SVM, and Multi-Layer Perceptron (MLP).

The chapter is organized as follows: we first make a brief discussion in Sec. 6.1 of the related works for the evaluation of PD in handwriting biometrics. Then, we present the proposed benchmark for the detection of PD in Sec. 6.2 and evaluates its performance for the different tasks proposed in Sec. 6.3.

6.1. State-of-the-art on Handwriting Parkinson Detection

The literature of the automatic evaluation of handwriting skills in PD patients is extensive, involving a huge range of tasks and machine learning algorithms. In [Drotár *et al.*, 2016] the authors perform an analysis of the kinematic and pressure features to classify between PD patients and healthy subjects with a database of 37/38 PD/HC subjects. Their results of up to 82% of accuracy considering different tasks such as spirals, sentences and characters demonstrates the potential of handwriting features to asses and monitor the progression of PD. In [Mucha *et al.*, 2018] the authors introduced a new algorithm named '*Fractional Derivative*' aimed to improve classification results in PD detection by considering kinematic handwriting signals extracted from drawing of a spiral. They report results of 72.4% of accuracy with a database of 30/36

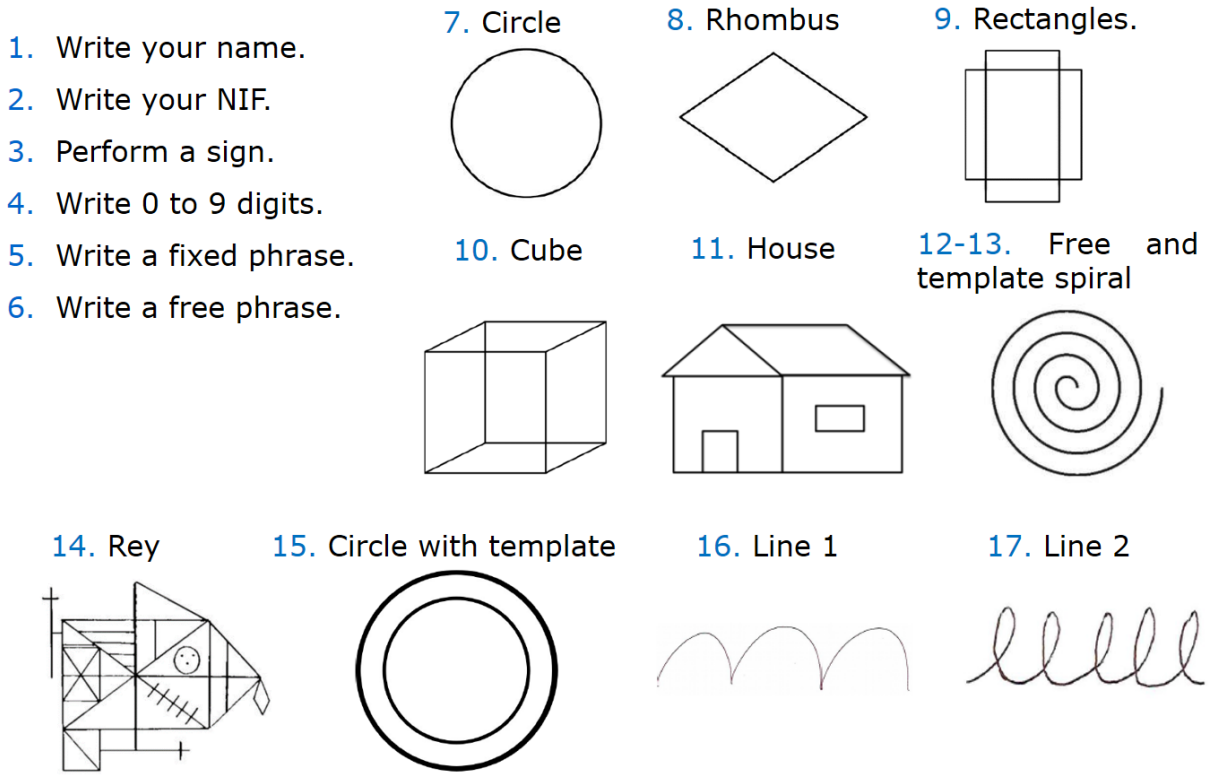

Figure 6.1: *Example of the template for each of the 17 on-line handwriting tasks: the first tasks consisted of writing the letters l and m in a continuous and long trace. Other tasks include the digits (0 to 9), the ID, name and signature of the participant, a free sentence, and the alphabet. The other nine tasks consist of geometrical figures including an Archimedean spiral, a circle with and without a template, a house, two concentric rectangles, a rhombus, a cube, and the Rey-Osterrieth complex figure.*

PD/HC subjects. In [Heremans *et al.*, 2016] the authors found a correlation of $r = -0.4$ between the handwriting kinematic features and the medical scales by using repetitive cursive loops for the evaluation, with a population of 30/15 PD/HC subjects. In other work [Kotsavasiloglou *et al.*, 2017] authors report classification rates of 91% of accuracy by employing a kinematic features and entropy analysis from drawings of horizontal lines. Finally, in [Taleb *et al.*, 2017] the authors report results of 96.87% of accuracy when classifying between PD patients and HC subjects with a SVM classifier and a feature set composed by spatio-temporal, pressure, energy, entropy, and intrinsic features. Seven tasks were considered from a corpus with 16/16 PD/HC subjects. All these efforts and others have been summarized in recent surveys [De Stefano *et al.*, 2019; Impedovo and Pirlo, 2018].

6.2. Experimental Protocol

Kinematic, non-linear dynamics and neuromotor features sets were extracted from the handwriting signals acquired during the execution of the 17 different handwriting tasks (see Fig. 6.1 for details). Owing to handwriting tasks are composed by more than one stroke (defined as the handwriting segment between pen-down and pen-up movements), the three feature sets

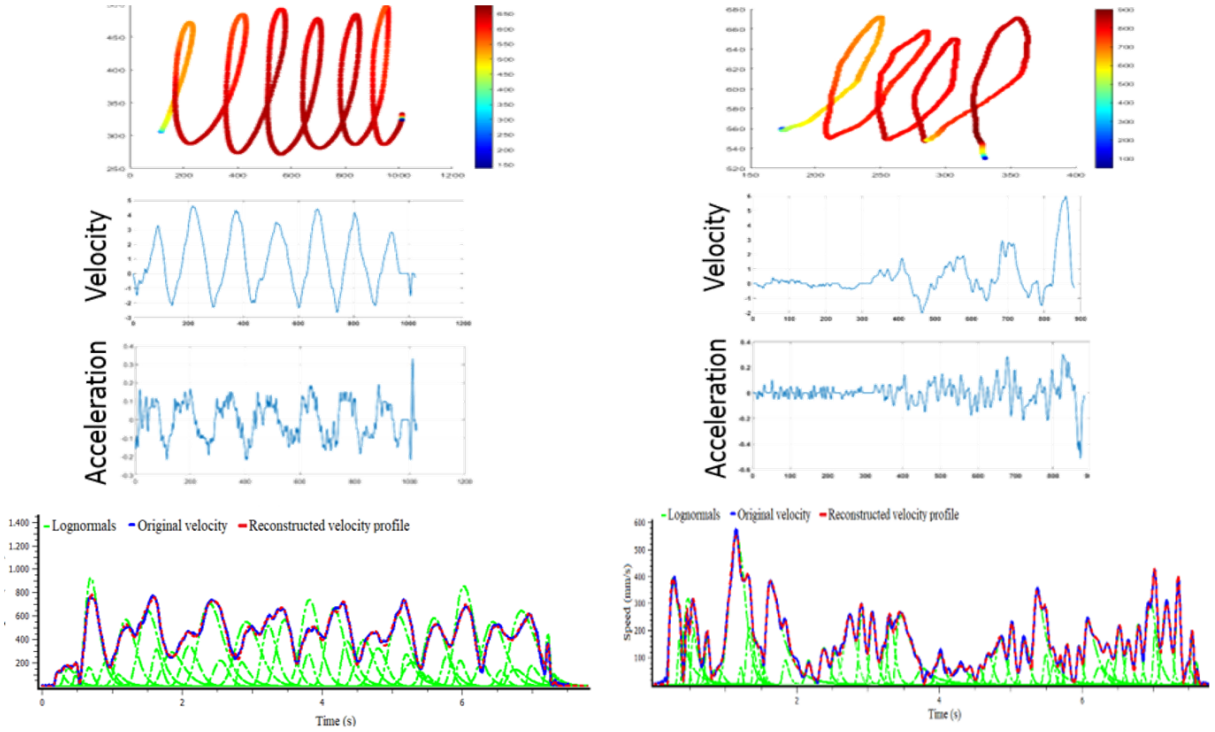

Figure 6.2: *Example of feature extraction from healthy control (left) and PD (right) patients when performing the handwriting task nº17. PD patients show a large number of lognormals with shorter bandwidths as well as more irregular velocity signals due to the Parkinson symptoms.*

were extracted at stroke level to compute a total of eleven statistical functions for each feature (between all strokes): mean value, median, standard deviation, 1st percentile, 99th percentile, difference between the 99th and 1st percentiles, maximum, minimum, kurtosis and skewness. This procedure results in a 921-dimensional feature vector per task containing a total of 452 kinematic features, 354 nonlinear dynamics features and 115 neuromotor features. Although we have not included all features proposed in the literature, we consider that this feature set is representative of the state-of-the-art ([De Stefano *et al.*, 2019]):

- *Kinematic features:* these features are extracted from the whole handwriting pattern (stroke in this chapter). Mean velocity, max acceleration, distance between adjacent points or total duration are examples of kinematic features. In this chapter we employ the global feature set already introduced in Chap. 5 and summarized in Table 5.2 as the kinematic feature set. Although global feature set has been used to characterize handwriting signatures for many years with good performance [Fierrez and Ortega-Garcia, 2008], they have not been used before to characterize PD, so in this chapter we will analyze whether they are suitable for this purpose. In Fig. 6.2 we have an example of the velocity and acceleration signals extracted from both healthy and PD patient. We can see how these signals are more irregular for the PD patient due to the hand tremor symptom of the disease.

- *Nonlinear Dynamics features:* these features model stability, non-stationarity conditions

	Experiments and Feature Sets											
	YHC vs. PD				EHC vs. PD				YHC vs. EHC			
	Kinematic	**Non Linear**	**Neuromotor**	**All**	**Kinematic**	**Non Linear**	**Neuromotor**	**All**	**Kinematic**	**Non Linear**	**Neuromotor**	**All**
Alphabet	91.8	87.6	-	90.7	73.3	71.3	-	71.3	88.3	74.5	-	77.7
Circletemplate	70.3	73.0	59.5	71.6	55.7	65.7	55.7	65.7	68.4	73.7	65.8	75.0
Cube	84.2	72.3	70.3	76.2	68.6	65.7	61.0	68.6	76.6	69.1	62.8	60.6
Freewriting	83.7	80.6	-	81.6	65.7	63.7	-	68.6	79.3	75.0	-	72.8
House	81.2	77.2	72.3	82.2	76.2	72.4	65.7	72.4	67.0	60.6	62.8	72.3
ID	78.8	84.8	69.7	83.8	68.0	66.0	55.3	67.0	86.2	78.7	73.4	75.5
Name	91.0	81.0	69.0	82.7	63.5	63.5	62.5	60.6	84.0	76.6	59.6	80.9
Numbers	83.8	84.8	68.7	72.0	65.0	61.2	61.2	65.0	73.4	73.4	64.9	75.5
Line 1	86.7	72.0	77.3	90.0	62.0	62.0	47.9	69.0	78.9	69.7	73.7	76.3
Line 2	76.0	77.3	54.7	85.9	64.8	56.3	53.5	62.0	80.3	80.3	71.1	80.3
Rectangles	83.2	79.2	68.3	71.3	61.0	65.7	50.5	73.3	62.8	67.0	61.7	63.8
Rey	87.5	77.1	-	90.6	68.7	64.6	-	73.7	80.6	77.4	-	78.5
Rhombus	72.3	78.2	65.3	74.3	54.3	61.0	54.3	59.0	72.3	64.9	64.9	70.2
Signature	87.9	81.8	70.7	93.9	71.8	69.9	65.0	75.7	78.7	62.8	58.5	76.6
Spiral	71.3	73.3	68.3	77.2	61.0	61.9	59.0	63.8	58.5	73.4	55.3	61.7
Spiraltemplate	78.2	77.2	79.2	80.2	60.6	62.5	53.8	72.1	69.9	64.5	63.4	66.7
Circle	78.2	77.2	73.3	73.3	61.9	58.1	55.2	67.6	78.7	66.0	72.3	75.5

Table 6.1: *Classifications results (%) per task for the SVM classifier with RBF kernel. Note: 'Circle' results correspond to the training process for optimizing the meta-parameters.*

in muscular movements that cannot be accurately modeled with other features. The nonlinear features set has been applied to model other biosignals like voice and gait with good results ([Pérez-Toro *et al.*, 2018; Travieso *et al.*, 2017]). Different nonlinear features are extracted such as correlation dimension, Lempel-Ziv complexity, largest Lyapunov exponent, Hurst exponent, empirical mode decomposition, and entropy. Other non linear features typically extracted to model handwriting signals are considered including the Shannon entropy, 2nd and 3rd order Renyi entropy, and the signal-to-noise ratio calculated using the conventional energy definition and the Teager-Kaiser energy [Drotár *et al.*, 2014].

- *Neuromotor features:* the neuromotor features set (already presented in Sec. 2.2.1) extracted from the Sigma-Lognormal model [Fischer and Plamondon, 2017] allows to characterize neuromotor-fine skills from the velocity profile of rapid human hand movements like handwriting. These features can be used as a marker of neurological disorders. Healthy patients tend to show velocity signals with less number of lognormals and stable bandwidths while the PD patients velocity signals show a large number of lognormals (due to poor motor control ability) and variable bandwidths, as showed in Fig. 6.2.

For classification, we employ three binary classifiers (i.e., SVM, kNN, and MLP) with a meta-parameter optimization that consists of a leave-one-out cross-validation strategy in a grid-search over a set with different candidate values. For this, we employ one of the 17 tasks (the '*Circle*' task) to train the classifiers and found the best hyper-parameters. The other 16 tasks are employed for evaluation. The experiments are divided into three different scenarios, according to the subjects employed to train and test the classifier: YHC (Young Healthy Control) vs. PD, EHC (Elderly Healthy Control) vs. PD, and YHC vs. EHC. The division between young (with ages between 17 and 42 years) and elder (older than 50 years) control subjects is aimed to discern between the PD neuromotor impairment and typical neuromotor degradation caused by the age.

6.3. Results and Discussion

Table 6.1 shows the accuracies obtained with the SVM classifier for the three scenarios proposed: YHC vs. PD, EHC vs. PD, and YHC vs. EHC. Similar results were obtained with MLP, but there were not satisfactory with kNN. When comparing among feature sets, we can observe that the best results for all tasks are achieved with the Kinematic feature set followed by the Non linear features. Moreover, the accuracies obtained when combining all feature sets are lower than the results achieved by the Kinematic features by their own. This happens due to the poor performance of the Neuromotor feature set which is outperformed by the other feature sets in most of the tasks. The reason of this is that the computer tool employed to extract the neuromotor features do not work well with long trajectories, due to the Sigma-Lognormal decomposition took to much iterations to converge and the tool stops at some point without finishing the process. In fact, in some tasks like '*Alphabet*' or '*Freewriting*' the tool is not able

Set of Features	Classifier	YHC vs. PD	EHC vs. PD	YHC vs. EHC
Kinematics	SVM	96.9	78.3	94.4
	kNN	93.7	74.9	87.5
	MLP	96.9	73.4	94.4
Non Linear	SVM	95.6	78.3	91.6
	kNN	94.3	61.4	86.2
	MLP	96.9	78.3	94.4
Neuromotor	SVM	88.0	51.6	81.9
	kNN	89.3	65.2	79.1
	MLP	81.8	59.9	81.9
All	SVM	96.9	81.7	97.2
	kNN	96.9	73.4	87.5
	MLP	96.9	78.3	94.4

Table 6.2: *Classifications results (%) for all tasks. Note: the 'Circle' task was not considered, as it was used before for training.*

even to start so we decided to not include the neuromotor results for those tasks with too long trajectories.

Regarding the analysis by task, '*Alphabet*' and '*Signature*' tasks present the best classification accuracy by merging all features sets according to the main rule to obtain the new scores: over 90% for YHC vs. PD and over 70% for EHC vs. PD. On the other hand, '*Circle*' with template and '*Rectangle*' present performances below 60% in some cases.

Table 6.2 summarizes the results when combining all tasks following a late-fusion strategy (i.e., combining at score level instead of at feature level as in Table 6.1). Again, the best performance is achieved by the SVM classifier followed closely by the MLP. The results obtained with the fusion strategy are better than those obtained with individual models for each task. Further research in this topic may help to clarify which are the most important tasks to discriminate between PD and healthy subjects, meanwhile we suggest that the best strategy is to combine all them at score level. When comparing among scenarios, the worst results are achieved in EHC vs. PD scenario as we expected. We suggest that the degradation of the neuromotor skills by aging could affects the classification performance due to their symptoms are very similar to the PD neuromotor impairments, and therefore, the classifiers are more prone to mistakes.

Finally, Fig. 6.3 shows the ROC curves for the classification between PD and HC subjects. The four curves correspond to the results obtained with the SVM classifier considering the three feature sets and their combination following the same late-fusion strategy as in Table 6.2. We can observe that the classification between YHC and PD patients presents the best results in most of the cases. The combination of all of the models presents the best results which confirms that complimentary information can be obtained from each feature set. Finally, the most difficult classification scenario is always between EHC and PD patients, which confirms other works in the literature where the effect of aging in handwriting can be misunderstood as PD symptoms for the classifiers.

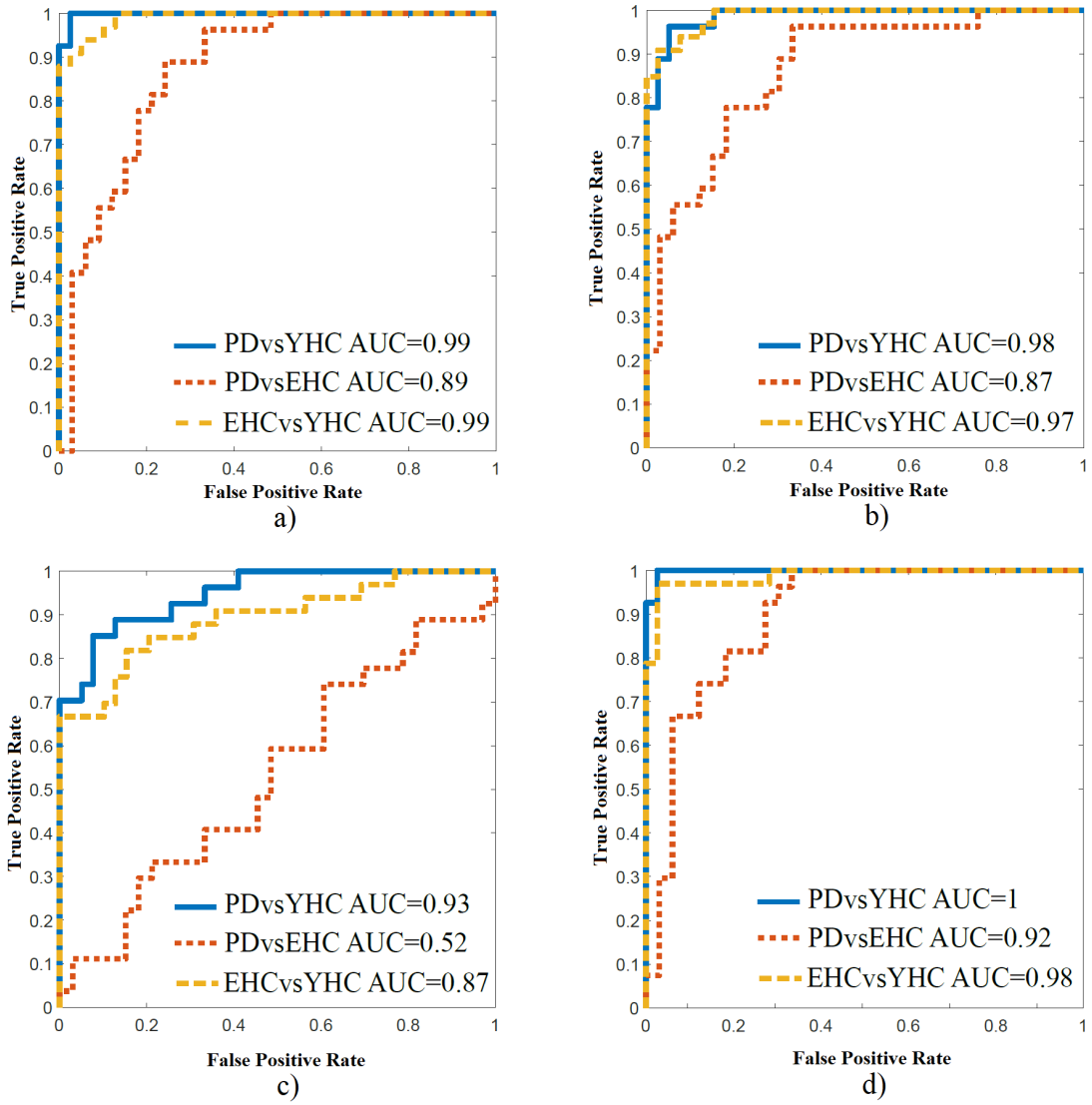

Figure 6.3: *ROC curves YHC vs. PD, EHC vs. PD, and EHC vs. YHC with SVM classifier for Kinematics (a), Non linear (b), Neuromotor (c), and the feature fusion (d). Area Under the Curve AUC* $= 1$ *for perfect classification.*

6.4. Chapter Summary and Conclusions

In this chapter we have analyzed three different feature sets (kinematics, non-linear, and neuromotor features) for the characterization of PD thorough handwriting analysis. For this, we have employed one the largest PD databases in on-line handwriting with 149 patients performing 17 handwriting tasks. The richness of the database is not only in the number of PD patients and healthy control subjects, but also in the quantity and diversity of the tasks performed. Our results over 96% of correct classifications rates in the best scenario (i.e., classifying between YHC versus PD patients) show the potential of these handwriting feature sets to model and characterize PD. When comparing among the different handwriting tasks, some tasks (e.g., alphabet, signature and house) are more discriminant than others (e.g., rhombus, rectangles and

cube) achieving superior performance in most of the classification scenarios proposed. Finally, the worst classification scenario was between EHC versus PD patients, where the effect of aging can be very similar to PD symptoms in handwriting tasks.

Part IV

Improving Security Applications through Neuromotor Analysis

Chapter 7

Modelling the Human Interaction for Bot Detection

This chapter studies the suitability of a new generation of CAPTCHA algorithms based on human-computer interactions named BeCAPTCHA. We present two approaches: *i)* BeCAPTCHA-Mobile, a bot detector based on the analysis of the touchscreen information obtained during a single drag and drop task in combination with the accelerometer data; and *ii)* BeCAPTCHA-Mouse, a bot detector based on the neuromotor analysis of mouse dynamics to obtain a novel feature set for the classification of human and bot. We evaluate both approaches by generating realistic synthetic data with two novel methods: *a)* a function-based method based on heuristic functions, and *b)* a data-driven method based on Generative Adversarial Networks (GANs) in which a Generator synthesizes human-like data from a Gaussian noise input.

The chapter is organized as follows, in Sec. 7.1 we review the state-of-the-art works in the bot detection field and the most advanced CAPTCHA algorithms, exposing their strengths and weakness. Then, we present the two proposed BeCAPTCHA approaches: one approach based on mouse dynamics in Sec. 7.2 and another one based on simple linear touch gestures in combination with the accelerometer data for mobile devices in Sec. 7.3.

7.1. State-of-the-art on Bot Detection

How to distinguish between human users and artificial intelligence during computer interactions is not a trivial task. This challenge was firstly discussed by Alan Turing in 1950. He investigated whether machines could show an intelligent behavior, and also how humans could be aware of these artificial behaviors. For this, he developed the famous Turing Test, commonly named as '*The Imitation Game*', in which a human evaluator would judge natural language conversations between a human and a computer designed to generate human-like responses. The Turing Test was both influential and widely criticized and became an important concept in the artificial intelligence field [Saygin *et al.*, 2000]. However, at the epoch of Alan Turing research,

the problem of machines acting like humans were commonly associated to science-fiction topics.

Nowadays, boosted by the last advances of machine learning technologies and worldwide connections, that 'science-fiction topic' becomes a real hazard. As an example, bots are expected to be responsible for more than 40% of the web traffic with more than 43% of all login attempts to come from malicious botnets in the next years[1]. Malicious bots cause billionaire losses through web scraping, account takeover, account creation, credit card fraud, denial of service attacks, denial of inventory, and many others. Moreover, bots are used to influence and divide society (e.g., usage of bots to interfere during Brexit voting day [Gorodnichenko *et al.*, 2018], or to spread anxiety and sadness during the COVID-19 outbreak[2,3] through Twitter). Bots are becoming more and more sophisticated, being able to mimic human online behaviors. On the other hand, algorithms to distinguish between humans and bots are also getting very complex. We can distinguish two types of bot detection methods in response to those sophisticated bots:

- *Active Detection:* traditionally named as CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart), these algorithms determinate whether or not the user is human by performing online tasks that are difficult for software bots to solve while being easy for legitimate human users to complete. Some of the most popular CAPTCHA systems are based on: characters recognition from distorted images (text-based), class-objects identification in a set of images (image-based), and speech translation from distorted audios (audio-based).
- *Passive Detection:* these detectors are transparent and analyze the users behavior while they interact with the device. The last version of Google reCAPTCHA v3 replaces traditional cognitive tasks by a transparent algorithm capable of detecting bots and humans from their web behavior[4]. In other work [Xie and Yu, 2009], the authors describe browsing behavior of web users for the detection of DDoS Attacks (Distributed Denial of Service) task.

Machine Learning and Pattern Recognition communities have made great advances during the last decades. These advances have boosted several research fields including Computer Vision, Audio Processing, and Natural Language Processing. Nonetheless, the application of these advances to the bot detection field has been rather low. While previous works [Bock *et al.*, 2017; Bursztein *et al.*, 2011] the authors focus their efforts in beating the existing CAPTCHA systems and exposing their vulnerabilities with the latest advances in machine learning techniques, in this chapter we employ them to develop better bot detectors and harden the existing ones.

The main drawback of traditional CAPTCHA methods is that they only measure cognitive human skills (e.g., character recognition from distorted images, class-objects identification in a set of images, or speech translation from distorted audios). Trying to ensure a very accurate bot

[1]https://resources.distilnetworks.com/white-paper-reports/bad-bot-report-2019

[2]https://www.washingtonpost.com/science/2020/03/17/analysis-millions-coronavirus-tweets-shows-whole-world-is-sad/

[3]https://www.sciencealert.com/bots-are-causing-anxiety-by-spreading-coronavirus-misinformation

[4]https://www.google.com/recaptcha/intro/v3.html

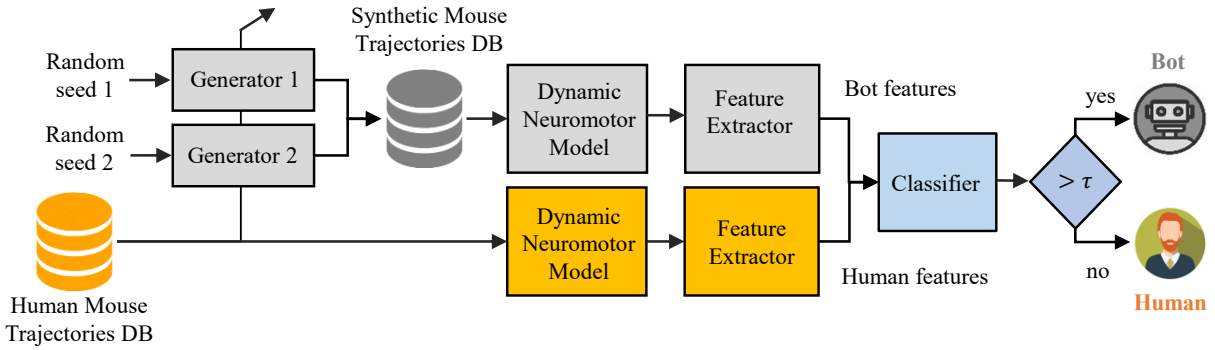

Figure 7.1: *Learning framework of BeCAPTCHA-Mouse: i) we propose two novel methods to generate realistic synthetic mouse trajectories that allow to train and evaluate bot detection systems based on mouse dynamics; ii) we propose a neuromotor model to characterize human and synthetic mouse trajectories; iii) we evaluate the proposed features using multiple classifiers and learning scenarios; and iv) the proposed Generators can be also helpful for other HCI applications.*

detection makes these CAPTCHAs difficult to perform even for humans. The main goal of our proposed approaches is to focus more on human behavioral skills rather than on cognitive ones. To the best of our knowledge, there are only a very limited number of works using behavioural biometrics for bot detection. The most related to our research are [Chu *et al.*, 2018] and [Akrout *et al.*, 2019]. In [Akrout *et al.*, 2019] the authors synthesize mouse trajectories over a grid to hack the Google reCAPTCHA v3 algorithm, and in Chu *et al.* [2018] the authors extract global features (e.g., duration, average speed, displacement) from mouse and keystroke patterns to conduct a case study in the detection of blog bots for online blogging systems.

While previous work in mouse dynamics [Ahmed and Traore, 2007; Chu *et al.*, 2018] the authors focused on basic cues like duration or average speed, in BeCAPTCHA-Mouse we go a step forward by focusing on the analysis and synthesis of entire mouse trajectories. We propose to use the Sigma-Lognormal model to extract neuromotor features that characterizes better human behaviors while performing mouse movements. Additionally, we also propose novel generation methods to synthesize human-like trajectories to improve the training and evaluation of our methods. On the other hand, most of the current CAPTCHAs have been designed to be used in a web interaction based on mouse and keyboard interfaces. In BeCAPTCHA-Mobile we explore the potential of our approach for bot detection in smartphone devices, by employing the data from two of the most common mobile sensor that can be easily acquired: touchscreen and accelerometer.

7.2. BeCAPTCHA-Mouse

The mouse is a very common device and its usage is ubiquitous in human-computer interfaces. Bot detection based on mouse dynamics can be therefore applied either in active or passive detectors.

Our BeCAPTCHA-Mouse bot detector is based on two main pillars: *i)* we use mouse dy-

namics to extract neuromotor features capable to distinguish human behavior from bots (see Fig. 7.1); *ii)* we generate synthetic mouse trajectories to improve the learning framework of bot detectors.

Mouse dynamics are rich in patterns capable of describing neuromotor capacities of the users. Note that we do not claim to replace other approaches (e.g., Google's reCAPTCHA) by mouse-based bot detection, our purpose is to enhance them by exploiting the ancillary information provided by mouse dynamics.

Our proposed method for bot detection consists in characterizing each mouse trajectory (real and synthetic) with a fixed-size feature vector obtained from a neuromotor decomposition of the velocity profile, followed by a standard classifier. Each trajectory characterized in this way can be classified individually using standard classifiers into human or bot based on supervised training using a development groundtruth dataset. When multiple trajectories are available, standard information fusion techniques can be applied [Fierrez *et al.*, 2018a]. The more realistic the synthetic data used as groundtruth for training the classifier the stronger the classifier.

In our experimental work we demonstrate the effectiveness of the neuromotor features and the synthetic samples for different classifiers. The contribution and success of our BeCAPTCHA-Mouse bot detector is not in the particular classifier used, but in two other fronts (see Fig. 7.1): the high realism of the groundtruth data used for training our classifiers, and our proposed trajectory modeling using neuromotor features.

7.2.1. Neuromotor Analysis of Mouse Trajectories

By looking at typical mouse movements (see Fig. 7.2.a), we can observe some aspects typically performed by humans during mouse trajectories execution: an initial acceleration and final deceleration performed by the antagonist (activate the movement) and agonist muscles (opposing joint torque) [Plamondon, 1995], and a fine-correction in the direction at the end of the trajectory when the mouse cursor gets close to the click button (characterized by a low velocity that serves to improve the precision of the movement). These aspects motivated us to use neuromotor analysis to find distinctive features in human mouse movements. Neuromotor-fine skills, that are unique of human beings are difficult to emulate for bots and could provide distinctive features in order to tell humans and bots apart.

For this, we propose to model the trajectories according to the Sigma-Lognormal model already introduced in Sec. 2.2.1. The neuromotor feature set proposed for bot detection is computed from the six lognormal parameters described in Table 2.3. Each mouse trajectory generates N lognormal signals and each lognormal generates those 6 parameters from Table 2.3. For each parameter, we calculate 6 features: maximum, minimum, and mean for both halves of the trajectory. This is done because in natural mouse movements the lognormal parameters are usually very different between both halves of a given trajectory (e.g., Fig. 7.2.b). As a result, the neuromotor feature set has size 37.

Fig. 7.2.c shows the decomposition of a synthetic trajectory with linear shape. We can observe the huge differences between both lognormal decompositions (the human trajectory and

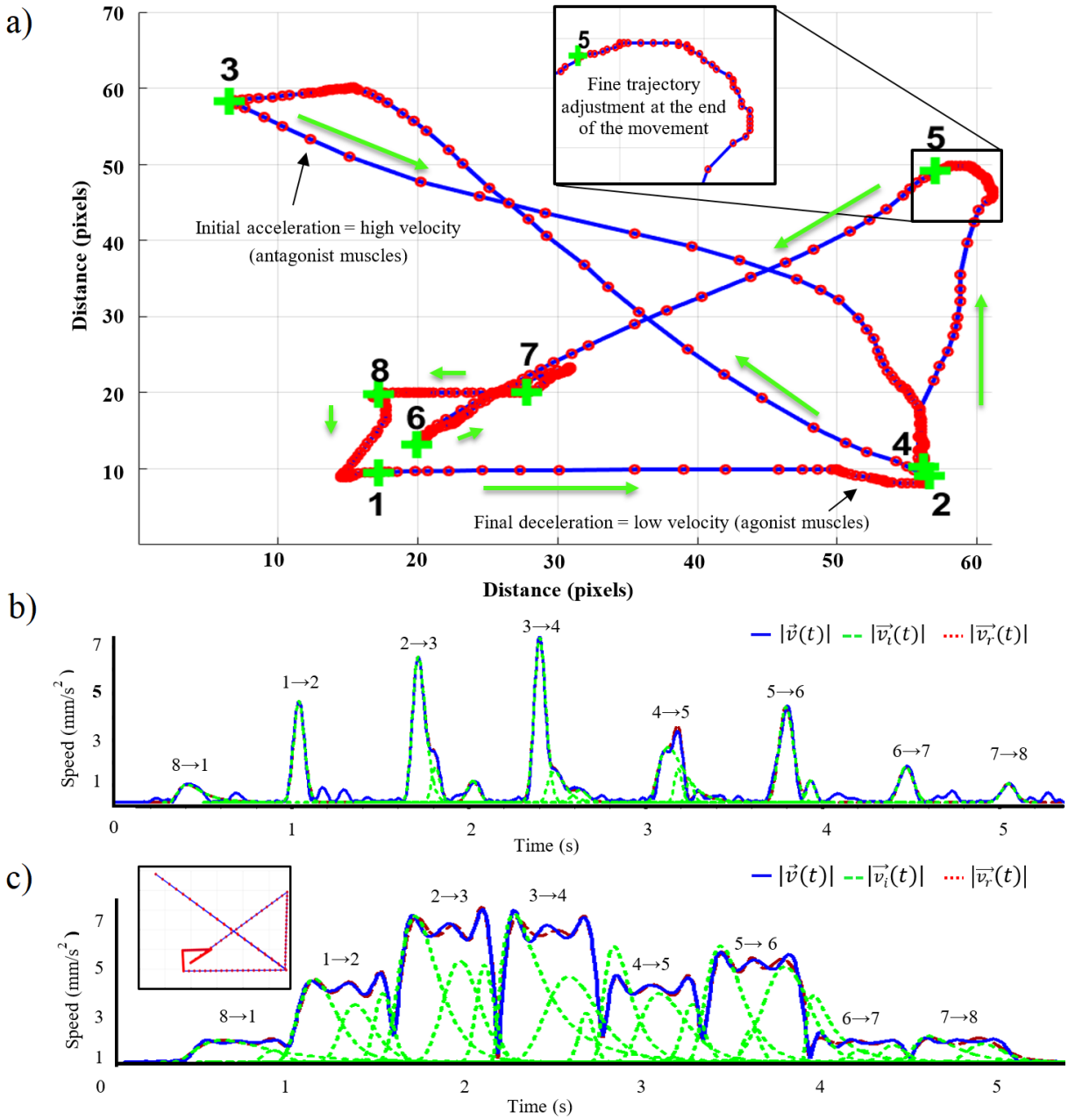

Figure 7.2: *a) Example of the mouse task determined by 8 key-points: the crosses represent the key-points where the user must click, red circles are the* (**x**, **y**) *coordinates obtained from the mouse device, and the black line is the mouse trajectory. b) and c) are examples of the Lognormal decomposition of a human mouse movement and a synthetic linear trajectory respectively.*

the synthetic one) by looking at the shape of the lognormal signals. The synthetic trajectory has wider lognormals and they are more symmetric than the human ones. Note that the Sigma-Lognormal algorithm introduces a low-pass filter to the input signal, that is the reason why the velocity profile of the synthetic trajectory (Fig. 7.2.c) is a bit smoothed, but the difference between both synthetic and human velocity profiles is still patent.

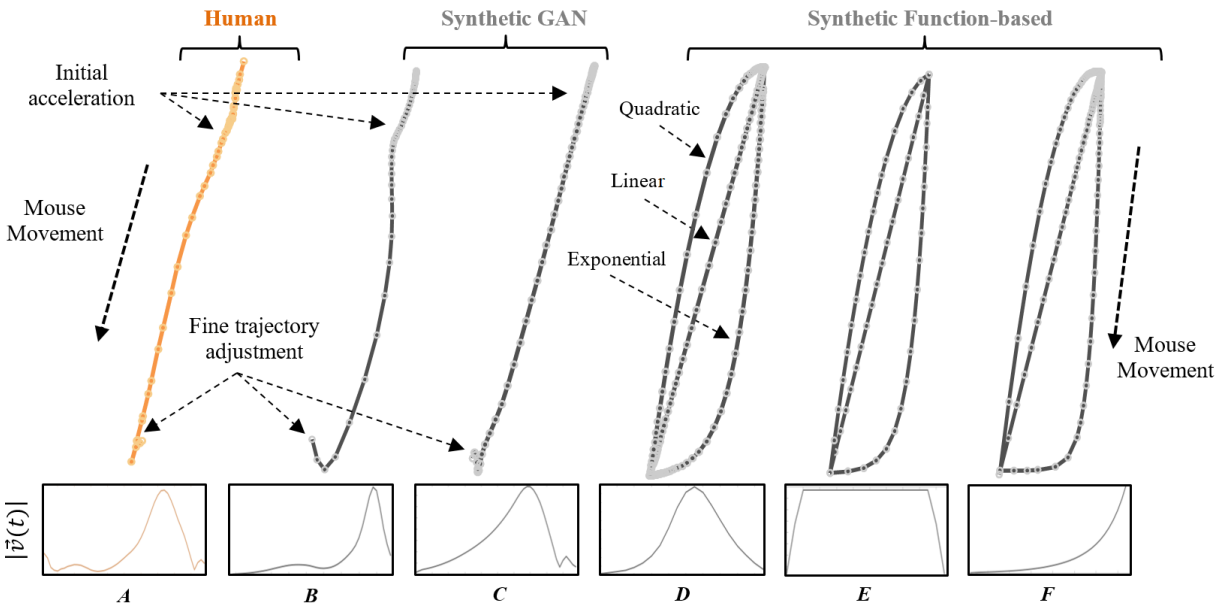

Figure 7.3: *Examples of mouse trajectories and their velocity profiles employed in this work: A is a real one extracted from a task of the database; B and C are synthetic trajectories generated with the GAN network; D, E and F are generated with the Function-based approach. Note that for each velocity profile (D = Gaussian, E = constant, F = logarithmic), we include the three Function-based trajectories (linear, quadratic, and exponential).*

7.2.2. Trajectory Synthesis

We define a mouse movement as the spatial trajectory across time between two consecutive clicks, i.e., a sequence of points $\{\mathbf{x}, \mathbf{y}\}$ and a velocity profile $|\vec{v}(t)|$, where $\mathbf{x} = [x_1, \ldots, x_M]$, $\mathbf{y} = [y_1, \ldots, y_M]$, and M is the number of time samples. A mouse trajectory is defined by two main characteristics: the shape (defined by $\{\mathbf{x}, \mathbf{y}\}$) and the velocity profile (defined by $|\vec{v}(t)|$). In order to generate realistic synthetic samples, both characteristics must be considered in the generation method. We propose two methods for synthetically generating such mouse movement.

- *Function-based trajectories:* we generate mouse trajectories according to three different trajectory shapes (linear, quadratic, and exponential) and three different velocity profiles (constant, logarithmic, and Gaussian).

  We can synthesize many different mouse trajectories that mimic human movements by varying the parameters of each function. To generate a synthetic trajectory $\{\hat{\mathbf{x}}, \hat{\mathbf{y}}\}$ with M points, first we define the initial point $[\hat{x}_1, \hat{y}_1]$ and ending point $[\hat{x}_M, \hat{y}_M]$. Second, we select one of three velocity profiles $\left|\vec{\hat{v}}(t)\right|$: *i)* constant velocity, where the distance between adjacent points is constant; *ii)* logarithmic velocity, where the distances are gradually increasing (acceleration); and *iii)* Gaussian velocity, in which the distances first increase and then decrease when they get close to the end of the trajectory (acceleration and deceleration). Third, we generate a sequence $\hat{\mathbf{x}}$ between $\hat{x}_1$ and $\hat{x}_M$ spaced according to the selected velocity profile. The $\hat{\mathbf{y}}$ sequence is then generated according to the shape

function. For example, for a shape defined by the quadratic function $\hat{y} = a\hat{x}^2 + b\hat{x} + c$, we fit b and c for a fixed value of a by using the initial and ending points. We repeat the process fixing either b or c. The range of the parameters $\{a, b, c\}$ explored is determined by analyzing real mouse movements fitted to quadratic functions. Linear and exponential shapes are generated similarly.

Fig. 7.3 (trajectories D, E, and F) shows some examples of these mouse trajectories synthesized. That figure also shows the 3 different velocity profiles considered: the 3 trajectories in E have constant velocity, F shows acceleration (the distance between adjacent samples increases gradually), and D has initial acceleration and final deceleration. We can generate infinite mouse trajectories with this approach by varying the parameters of each function.An important factor when synthesizing mouse trajectories is the number of points (M) of the trajectory. This usually varies depending not only on the length of the trajectory, but also on the direction, because different muscles are involved when we perform mouse trajectories in different directions. To emulate this phenomenon, we calculate the mean and standard deviation of the number of points for each of the 8 mouse trajectories from the human data used in the experiments. Then, we synthesize trajectories with different number of points following a Gaussian distribution with the calculated mean and standard deviation.

- *GAN-based trajectories:* for this approach we employ a GAN network already introduced in Sec. 2.3.2, in which two neuronal networks, commonly named Generator (defined by its parameters $\mathbf{w}_G$) and Discriminator (defined by its parameters $\mathbf{w}_D$), are trained one against the other. During the GAN training, the weights of the Discriminator ($\mathbf{w}_D$) remain frozen. The iterative training process will update the weights $\mathbf{w}_G$ of the Generator in a way that makes Discriminator more likely to predict '*Human*' when looking at synthetic mouse trajectories generated by the Generator. If the Discriminator is not frozen during this process, it will tend to predict '*Human*' for all trajectories. The Discriminator is trained (weights $\mathbf{w}_D$ updated) after the update of the weights of the Discriminator ($\mathbf{w}_G$). This process is repeated iteratively. Once the Generator is trained this way, then we can use it to synthesize sequences very similar to the human ones.

  The GAN network was trained using 60% of the human mouse trajectories in the mouse database (see Sec. 2.1.1 for more database details). Training details: learning rate $\alpha = 2 \times 10^{-4}$, Adam optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, 50 epochs with a batch size of 128 samples for both Generator and Discriminator.

  Fig. 7.3 shows two examples (trajectories B and C) of synthetic mouse trajectories generated with the GAN network and the comparison with a real one. We can observe high similarity between the two synthetic examples and the real one. Human mouse patterns such as the initial acceleration and the final trajectory fine correction that we discussed before are automatically learned by the GAN network and reproduced in the synthetic trajectories generated.

Trajectories		Bot: Function-based									Bot: GAN
		Linear			Quadratic			Logarithmic			
		VP = 1	VP = 2	VP = 3	VP = 1	VP = 2	VP = 3	VP = 1	VP = 2	VP = 3	
Individual Trajectories	$8 \rightarrow 1$	98.6	96.3	99.0	91.0	91.0	92.3	89.0	88.6	89.3	96.9
	$1 \rightarrow 2$	99.7	98.6	97.2	91.6	98.3	92.2	95.8	92.3	92.5	96.7
	$2 \rightarrow 3$	99.4	99.1	99.7	95.3	96.4	88.0	94.4	98.9	90.5	99.9
	$3 \rightarrow 4$	99.7	97.5	97.0	94.2	96.6	90.5	94.2	95.1	93.0	99.7
	$4 \rightarrow 5$	99.9	98.0	99.4	95.5	94.7	92.5	93.9	95.4	93.9	97.0
	$5 \rightarrow 6$	99.9	98.9	99.1	92.8	97.5	91.4	93.3	95.1	94.4	98.3
	$6 \rightarrow 7$	99.1	98.3	98.6	90.2	89.7	93.6	88.8	92.3	93.6	98.1
	$7 \rightarrow 8$	97.0	96.6	97.5	92.2	93.3	93.0	88.3	88.6	93.1	98.7
All	Neuromotor	99.1	98.7	99.3	96.9	96.3	94.7	96.3	95.2	94.7	98.0
	Global Features	99.7	99.6	99.7	95.3	96.7	96.8	97.2	96.5	97.3	**99.8**
	Both	**99.9**	**99.7**	**99.8**	**98.0**	**99.0**	**98.4**	**98.2**	**98.9**	**98.9**	99.7

Table 7.1: *Accuracy rates (%) in the binary classification between each of the 8 human trajectories and the synthetic ones. VP (Velocity Profile): VP = 1 constant velocity, VP = 2 initial acceleration, VP = 3 initial acceleration and final deceleration.*

7.2.3. Results and Discussion

The BeCAPTCHA-Mouse Benchmark is composed of 5K human trajectories and 10K synthetic trajectories generated according to the two methods proposed (5K Function-Based and 5K GAN trajectories). Both real and synthesized samples are characterized by a variety of lengths, directions, and velocities.

7.2.3.1. Role of the Direction and Length of the Trajectory

We have extracted the proposed neuromotor features from human and synthetic mouse trajectories. For this first experiment, we use a Random Forest (RF) classifier because of its best performance among all classifiers evaluated (as we will see in the next section). The experiments are divided according to the 8 real mouse trajectories present in the whole task. This means that we classify at trajectory level (i.e., the mouse trajectory performed between two consecutive click buttons) instead of classifying the whole task. This is because the task was designed to take into account trajectories with different directions and lengths, and therefore, different muscles configurations are involved in each trajectory. In this way, we can analyze which mouse trajectories are better to discriminate between humans and bots. We train 10 different RFs (one for each type of attack, see columns in Table 7.1) using both human and synthetic trajectories. For each RF, we train the classifier by using 70% of all samples (up to 1,500 samples available for each type of trajectory between both synthetic and real ones) randomly chosen as the training set. The other 30% samples are employed for evaluation. The results are obtained by repeating each experiment 5 times and averaging, with a standard deviation of $\sigma \sim 0.1\%$.

Table 7.1 shows the results for all classification schemes. The first 8 rows present the 8 trajectories derived from the movements between the 8 key-points (plotted in Fig. 7.2.a). The table shows the classification accuracy in % (human vs bot) for the different synthetic trajectories (in columns) generated in this work.

First, comparing among the different trajectories, we can observe that the shorter ones

Features	Training	
	Only Real	**Real+Fake**
Global Features	66.3% (baseline)	96.6% (↓ 90.1%)
Neuromotor	64.4% (↑ 5.6%)	89.8% (↓ 79.7%)
Both	59.9% (↑ 19.0%)	98.2% (↓ 95.4%)

Table 7.2: *Accuracy rates (%) in bot detection of the different feature sets for models trained with and without synthetic samples (fakes) and evaluated using human samples and fake samples. One-Class SVM (first column) and Multiclass SVM (second column). Relative error reduction with respect to the baseline [Chu* et al.*, 2018] in brackets.*

($8 \rightarrow 1$, $6 \rightarrow 7$, and $7 \rightarrow 8$) show higher classification errors compared to the larger ones. Short trajectories generate less neuromotor information: initial acceleration, final deceleration, and trajectory corrections are less pronounced in short trajectories. Second, logarithmic trajectory shapes achieve the worst classification performance, as we expected, because the shape of logarithmic functions fit better the human trajectories shapes. Third, the most significant parameter when synthesizing trajectories is the velocity profile. When VP = 3 (i.e., initial acceleration and final deceleration), the synthetic trajectories are able to fool the classifier up to 17% of the times. This confirms that the velocity profile of human mouse trajectories plays and important role when describing human features in mouse dynamics. Four, the GAN Generator (last column in Table 7.1) results in lower classification errors compared with the Function-based method. This is surprising after visualizing the high similarity between human and GAN-generated trajectories (see Fig. 7.3 A vs B and A vs C). We interpret this result with care: on the one hand it demonstrates that our bot detection approach is powerful against realistic and sophisticate fakes, but on the other hand the GAN Generator can be improved to better fool our detector. Although the synthetic samples generated by the GAN Generator seems very realistic to the human eye, the RF classifiers were capable of detecting synthetic samples with high accuracy. These high classification rates suggest that GAN generators introduce patterns that allow its detection [Neves *et al.*, 2020].

The last three rows in Table 7.1 present the results when features from all 8 trajectories are combined (each RF is trained using features from all 8 trajectories). Additionally, we compare the performance achieved with existing approaches [Chu *et al.*, 2018]. The feature set proposed in [Chu *et al.*, 2018] consists of 6 global features: duration, distance, displacement, average angle, average velocity, and move efficiency (distance over displacement). The results suggest that the feature set proposed in [Chu *et al.*, 2018] outperforms the neuromotor features proposed here only for Linear synthetic trajectories. The best performance is obtained overall with an extended set composed by both sets of features. The extended set has the best results with an average around 99% of accuracy independently of the type of synthetic trajectory.

7.2.3.2. Role of Synthetic Samples

Table 7.2 shows the accuracy when all types of attacks are used to train and test the system. In this case, the classifier is trained using trajectories from all 8 directions and synthetic samples

	Bot														
	Function-based					GAN					Combination				
Classifiers	**Acc**	**AUC**	**Pre**	**Re**	**F1**	**Acc**	**AUC**	**Pre**	**Re**	**F1**	**Acc**	**AUC**	**Pre**	**Re**	**F1**
SVM	98.0	99.4	98.6	96.7	97.7	98.5	99.6	99.2	97.9	98.5	98.2	99.4	97.3	99.0	97.4
***k*NN**	93.4	98.1	93.6	93.2	93.5	94.1	99.4	99.8	88.3	93.6	92.0	97.4	90.7	93.2	92.1
RF	**98.5**	**99.8**	**98.6**	**98.8**	**98.7**	**99.7**	**99.9**	**99.5**	**99.9**	**99.7**	**98.7**	**99.9**	**98.8**	**99.0**	**99.0**
MLP	94.6	94.1	95.0	94.2	94.6	93.4	93.5	95.4	92.3	93.9	92.2	91.5	89.8	95.4	92.5
LSTM	98.2	99.8	97.6	98.8	98.2	99.2	98.0	99.7	98.9	99.5	97.3	99.7	96.7	97.9	97.3
GRU	98.4	99.4	98.5	98.6	98.6	99.3	99.2	99.2	90.2	99.0	99.8	99.8	94.4	99.0	96.9

Table 7.3: *Bot detection performance metrics in % (Acc = Accuracy, AUC = Area Under the Curve, Pre = Precision, Re = Recall, and F1) for the different scenarios: Function-based, GAN, and Combination.*

from all 10 types of attacks. The Table shows the impact of introducing the synthetic samples (i.e., Real+Fake) in the learning process. For this experiment, we decided to use as classifiers a One-Class SVM (trained using only real trajectories) and a Multiclass SVM (trained using real and synthetic trajectories). The aim of the experiment is to evaluate to what extent the inclusion of synthetic samples in the learning framework serves to improve the accuracy of the model. The results show that the synthetic samples and neuromotor feature set proposed in this work allows to reduce the error by 95.4% in comparison with the previous existing method [Chu *et al.*, 2018]. These results demonstrate the potential of synthetically generated trajectories and mouse dynamics features to boost the performance of new bot detection algorithms.

The results obtained show how training methods based on both real and synthetic trajectories clearly outperform training methods based exclusively on real samples. As can be seen, the classifier trained only with real samples was not capable to detect most of the attacks with accuracy rates lower than 70% either for global features and neuromotor features. The importance of synthetic samples is twofold: i) evaluation of bot detection algorithms under challenging attacks generated according to different methods; and ii) training better detectors to model both human and synthetic behaviors. The results in Table 7.2 show the potential of the synthetic samples and its usefulness to train better models capable to deal with all types of attacks.

7.2.3.3. Ablation Study

In this section we perform an ablation study on different classifiers to analyze their performance in bot detection for the 3 multi-class scenarios proposed, according to the synthetic samples employed to train and test them: Function-Based, GAN, and their Combination. It is worth noting that all classifiers are trained using trajectories from all 8 directions and synthetic samples from all 10 types of attacks, as reported in Table 7.2 to allow fair comparisons.

Table 7.3 shows the performance of classification algorithms: Support Vector Machine (SVM) with a Radial Basis Function (RBF), K-Nearest Neighbors (kNN) with $k = 10$, Random Forest (RF), Multi-Layer Perceptron (MLP), and 2 Recurrent Neuronal Networks (RNN), (one composed by Long Short-Term Memory (LSTM) units and the other with Gated Recurrent Units (GRU). The RNNs (i.e., LSTM and GRU) were trained directly with the raw data (i.e., the sequence of points $\{\mathbf{x}, \mathbf{y}\}$ of the mouse trajectories) instead of extracting the global features

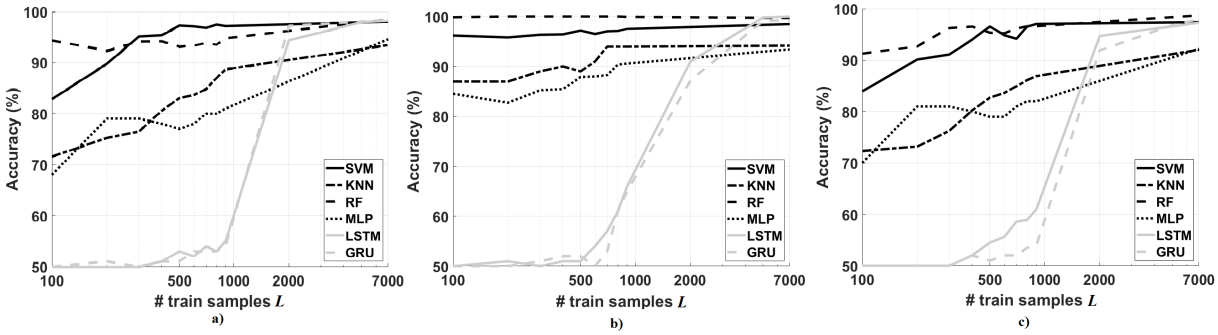

Figure 7.4: *Accuracy curves (%) against the number of train samples (*$100 \leq L \leq 7{,}000$*) to train the different classifiers in Function-based (a), GAN (b), and Combination (c) classification scenarios.*

(i.e., Neuromotor + Baseline [Chu *et al.*, 2018]) as done with the statistical classifiers. The RNNs have the same architecture as the Discriminator of the GAN: two recurrent layers of 128 and 64 units respectively, followed by a dense layer to classify between fake and real mouse trajectories. All classifiers were trained and tested following the same experimental protocol as in Sec. 7.2.3.1, using 70% of all samples (up to 10K samples between both real and synthetic samples when combining all types of trajectories) randomly chosen as the training set (named L in this section, with $L = 7{,}000$). The results are reported in terms of Accuracy, AUC (Area Under the Curve), Precision, Recall, and F1.

First, we can observe that the best results among the statistical classifiers are achieved by the RF classifier followed by the SVM. kNN and MLP perform worst, although all classifiers have accuracy rates over 90%. Secondly, among the different RNNs, the configuration with LSTM units performs sightly better than the one with GRU units, even though both recurrent network setups are outperformed by the RF classifier. These results suggest that the feature set chosen to train and test the statistical classifiers is suitable for the mouse bot detection task, outperforming other approaches based on deep neuronal networks architectures. Nonetheless, the RNNs demonstrate its capacity to extract useful features from the raw data.

In the next experiment we explore whether the number of training samples (L) plays and important role in the classification performance. We want to highlight that the training and the evaluation sets have the same number of human (L_h) and synthetic (L_s) samples, i.e.: $L_h = L_s = L/2$.

For this, in Fig. 7.4 we plot the accuracy curves of the previous classifiers according to the number of samples employed in their training set. As expected, the accuracy improves in all scenarios when we enlarge the number of train samples. However, there are important differences between the statistical and the RNNs approaches. Meanwhile all statistical classifiers achieve their maximum performance with $L = 500$, both LSTM and GRU are not able to reach the same performance with only 500 train samples. In fact, they need at least $L = 2{,}000$ to perform as well as the statistical classifiers. This shows the superior performance of the statistical classifiers in those scenarios where the number of samples to train the classifiers are scarce.

Finally, in the last experiment we replaced the previously introduced RNNs classifiers by

	Bot														
	Function-based					GAN					Combination				
Discriminators	**Acc**	**AUC**	**Pre**	**Re**	**F1**	**Acc**	**AUC**	**Pre**	**Re**	**F1**	**Acc**	**AUC**	**Pre**	**Re**	**F1**
LSTM (128/64)	**89.9**	**93.2**	**88.5**	**90.0**	**89.3**	96.8	99.6	95.0	98.7	96.8	**89.6**	**93.9**	**89.2**	**90.0**	**89.6**
LSTM (64/32)	74.0	72.1	67.0	95.6	78.7	**99.9**	**99.9**	**99.9**	**99.9**	**99.9**	73.0	76.1	65.9	96.0	78.1
LSTM (32/16)	81.4	80.2	77.9	88.0	82.6	99.7	98.9	99.6	99.9	99.8	78.8	76.0	74.4	88.0	80.6
LSTM (16/8)	56.8	58.6	54.2	86.8	66.7	56.2	91.3	53.3	99.9	69.5	64.0	67.0	59.5	87.2	70.7

Table 7.4: *Performance metrics in % (AUC = Area Under the Curve, Acc, Pre, Re, and F1) for the different setups of GAN Discriminator in bot detection. In brackets the number of neurons for the first/second LSTM layer respectively used in the Discriminator.*

the Discriminator model of the GAN architecture. The idea is to analyze in what extent the Discriminator of the GAN Network trained only with the synthetic samples generated by the Generator (and the real ones) during the GAN training could perform better in classification than the previous RNNs trained from scratch. For this, we tuned the number of neurons of the two LSTM layers of the Discriminator and trained a new GAN network for each Discriminator setup proposed.

Table 7.4 shows the performance of 4 GAN Discriminator setups for the 3 classification scenarios proposed: the Function-based, GAN, and their Combination. As we expected, the performance using GAN classification is much better than the performance achieved by the LSTM and GRU networks of the previous experiment, due to the Discriminators were trained specifically to discriminate between the synthetic mouse trajectories generated by the GAN Generator and the human ones. However, the Discriminators also classify quite well in the Function-based scenario, even though no Function-based sample was employed to train them ($L_s = 0$). In fact, as we increase the complexity of the Discriminator with more neurons in both layers, the performance improves up to 90% of accuracy, close to the results achieved by the LSTM and GRU networks trained with $L_s = 7{,}000$ samples. These results show the potential of the GAN architecture, not only to generate synthetic mouse trajectories with similar shape to the human ones with the Generator, but also for classification purposes, as the Discriminator is able to classify between human and bot trajectories even against synthetic trajectories not seen during the training phase.

7.3. BeCAPTCHA-Mobile

We focus here on building a CAPTCHA system based on swipe gestures (i.e., drag and drop tasks). We model this gesture according to features obtained from the touchscreen and accelerometer sensors in order to extract cognitive and neuromotor human features that help us to discriminate between bots and human users just with simple drag and drop gestures. To evaluate BeCAPTCHA-Mobile, we will employ human samples from HuMIdb database (already presented in Sec. 2.1.7) and synthetic ones (bot like samples) generated using two different approaches: a handcrafted synthesis and using GANs (see Fig. 7.5 for details) . The goal is to determine whether a simple swipe gesture has been performed by a human or generated by a bot.

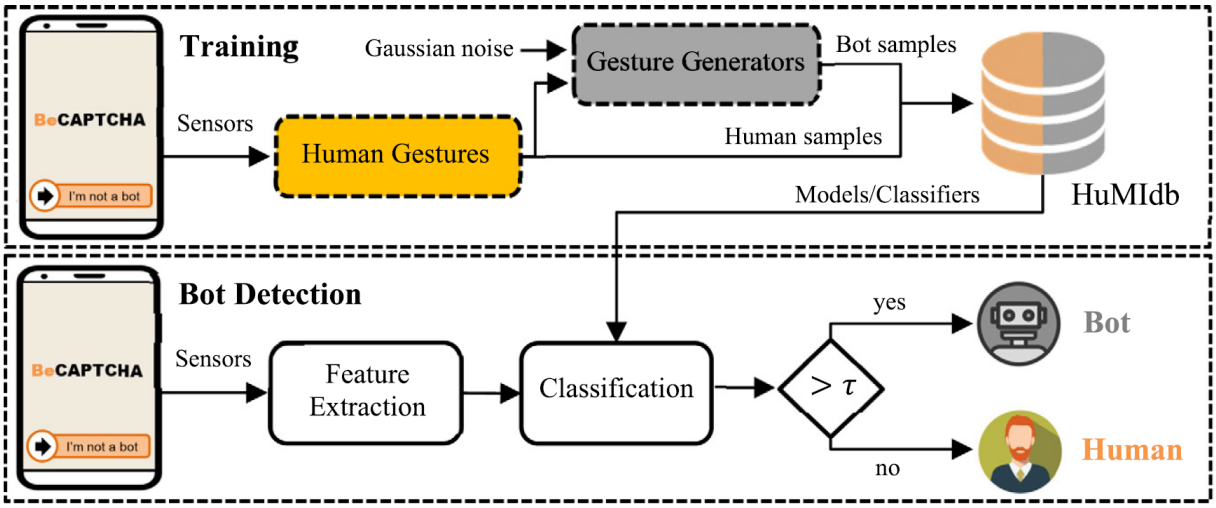

Figure 7.5: *Block diagram of our proposed BeCAPTCHA-Mobile approach. The response of the bot detector is a combination of responses from two different modalities: touch and accelerometer. τ is a decision threshold.*

Parameter	Description
Duration (D)	$t_N - t_0$
Distance (L)	$\|(x_{N-1}, y_{N-1}) - (x_0, y_0)\|$
Displacement (P)	$\sum_{i=0}^{N-1} \|(x_{i+1}, y_{i+1}) - (x_i, y_i)\|$
Angle (α)	$\tan^{-1}(\|(y_{N-1} - y_0)\| \,/\, \|(x_{N-1} - x_0)\|)$
Mean velocity (V)	$(1/N) \sum_{i=0}^{N-1} \|(x_{i+1}, y_{i+1}) - (x_i, y_i)\| \,/(t_{i+1} - t_i)$
Move efficiency (E)	P/L

Table 7.5: *Touch features extracted for the characterization of the gestures.*

7.3.1. Feature Extraction: Characterizing Swipe Gestures

To characterize swipe gestures from the touchscreen and accelerometer signals, we have adapted two feature sets previously employed in [Chu *et al.*, 2018; Li and Bours, 2018a] for bot detection and user authentication respectively.

The interaction of the user with the Touchscreen is defined by a time sequence $s_T = \{\mathbf{x}, \mathbf{y}, \mathbf{p}, \mathbf{t}\}$ with length N, composed by the coordinates $\{\mathbf{x}, \mathbf{y}\}$ the pressures $\mathbf{p}$ (when available), and the timestamps $\mathbf{t}$. First, the coordinates $\{\mathbf{x}, \mathbf{y}\}$ normalized by the size of the screen. Second, the pressure is discarded as it is not available in most of the devices. Third, six global features are generated according to Table 7.5. The Accelerometer signal is defined by a sequence $s_A = \{\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{t}\}$. The feature set chosen for the accelerometer signal was adapted from [Li and Bours, 2018c], in which they calculate the mean, median, rootmean-square, and standard deviation of the three accelerometer axes $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ user authentication.

7.3.2. Generating Human-like Gestures: Bot Samples

A swipe gesture can be defined by a spatial trajectory (sequence of points $\{\mathbf{x}, \mathbf{y}\}$ and a velocity profile determined by the timestamp sequence $\mathbf{t}$. To generate synthetic swipe patterns, we will follow two approaches: handcrafted synthesis and Generative Adversarial Network (GAN) synthesis.

- *Handcrafted synthesis:* we observed that most of the human swipe trajectories obtained from our drag and drop task are linear. The handcrafted approach generates swipe trajectories according to a straight-line shape and a realistic velocity profile. For this, we first estimate the probability distribution of length and angle of human swipe gestures in HuMIdb. Note that the size and coordinates of each human swipe varies depending on the device features so we have normalized each one by the total size of the screen. The synthetic trajectories are defined by the initial point (x_0, y_0), duration $(t_N - t_0)$, angle (α), and the velocity profile $\{\mathbf{v}, \mathbf{t}\}$. We have synthesized the fake trajectories according to distributions of these parameters fitted from human data (except for the velocity profile). With the aim to emulate human behaviors, we spaced the points of the linear trajectory on a log scale (emulating a velocity profile with the initial acceleration observed in human samples). The accelerometer signals are synthesized as random sequences generated from a Gaussian distribution with mean and standard deviation estimated from real accelerometer signals from HuMIdb.
- *GAN synthesis:* For this approach, we employ a GAN (Generative Adversarial Network) architecture firstly proposed in [Goodfellow *et al.*, 2014], in which two neuronal networks, commonly named Generator and Discriminator, are trained in adversarial mode. The Generator tries to fool the Discriminator by generating fake samples (touch trajectories and accelerometer signals in this work) very similar to the real ones, while the Discriminator has to discriminate between the real samples and the fake ones created. Once the Generator is trained, then we can use it to synthesize swipe trajectories very similar to the real ones.

  The topology employed in both Generator and Discriminator had already presented in Sec. 2.3.2 and consist of two LSTM (Long Short-Term Memory) layers followed by a dense layer, very similar to a recurrent auto-encoder. The LSTM layers learn the time relationships of human swipe sequences, while the dense layer is used as a classification layer to distinguish between fake and real swipe trajectories in the Discriminator or to build synthetic swipe trajectories in the Generator. To synthesize accelerometer signals, we follow the same GAN architecture described before, but extending the input of the generator from $\{\mathbf{x}, \mathbf{y}\}$ swipe coordinates to $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ accelerometer axes.

7.3.3. Experimental Protocol

Both GAN networks were trained using more than 10K human samples extracted from the HuMIdb. Training details: learning rate of 10^{-4}, Adam optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.999$,

and $\epsilon = 10^{-8}$. The system was trained for 50 epochs with a batch size of 128 samples for both Generator and Discriminator. The loss function was '*binary crossentropy*' for the Discriminator. The model was trained and tested in *Keras-Tensorflow*.

We generated 12K synthetic samples according to the two methods proposed (up to 18K samples between all groups: 6K human samples, 6K GAN synthetic samples, and 6K handcrafted synthetic samples). Once we have extracted the global features from human and synthetic swipe trajectories and accelerometer data we classify them employing three classification algorithms: an SVM (Support Vector Machine) with an RBF (Radial Basis Function), k-NN with $k = 10$, and RF (Random Forest). The experiments are divided into two different scenarios depending on the synthetic data (i.e., handcrafted or GAN) employed in training: multiclass or agnostic. In multiclass classification, we train and test the classifiers with the same kind of synthetic samples in order to analyze whether the classifier can find discriminative features between both human and bots samples. In the agnostic classification, we train the classifiers using samples of one bot generation method and test with the other one, in order to study whether the classifiers are able to detect bot samples from unknown bot generation methods not seen during the training phase.

In both classification setups, there is no overlap between the data used for training and evaluation. We use 70% of all samples (randomly chosen) as the training set, which is further divided into development (90%) and validation set (10%) in order to choose the best hyperparameters of the classifiers. The remaining 30% of the samples is used for the evaluation of the system. Both development and evaluation sets are balanced with same number of human and bot samples in each set. All experiments were repeated 5 times (with random selection of the data sets) and the results were computed as the average of the 5 iterations with a standard deviation of $\sigma \sim 0.1\%$.

7.3.4. Results and Discussion

7.3.4.1. Performance of bot detection: Multiclass vs agnostic training

Table 7.6 shows the bot detection performance metrics (%) for different synthetic trajectories (columns) generated when comparing with the human ones. For this experiment the number of training samples (for both human and synthetic samples) is set to $M = 1000$. The results are presented in terms of AUC (Area Under the Curve), Accuracy, Precision, Recall, and F1.

First, we observe that the results achieved for the agnostic classification are always significantly worse (lower performance) than those achieved in multiclass classification as expected. The synthetic samples generated by the two methods present their own specific features and the inclusion of both types of samples in training clearly improves the detection accuracy. Secondly, when comparing among classifiers we can observe that the RF classifier performs better in multiclass classification meanwhile in agnostic classification, RF is outperformed by k-NN. Finally, we can observe that classifiers trained with both accelerometer and touch samples perform better than those systems trained only with the touch data, especially in agnostic classification,

		Bot Detection														
		Handcrafted					GAN					HandCrafted + GAN				
	Classifiers	AUC	Acc	Re	Pre	F1	AUC	Acc	Re	Pre	F1	AUC	Acc	Re	Pre	F1
Touch	SVM (M)	99.2	94.2	89.4	98.8	93.9	98.6	95.5	95.0	95.6	95.5	93.6	85.8	82.9	88.1	85.4
	k-NN (M)	88.3	80.6	74.7	84.8	79.4	98.6	94.6	92.0	97.0	94.5	90.0	80.1	78.0	82.3	80.0
	RF (M)	**100**	**99.9**	**99.9**	**100**	**99.9**	**99.3**	**97.3**	**94.7**	**98.2**	**96.4**	**99.7**	**96.5**	**96.8**	**97.7**	**97.3**
	SVM (A)	**61.3**	51.7	**96.6**	**49.1**	**65.2**	70.4	56.6	88.5	43.0	61.4	-	-	-	-	-
	k-NN (A)	57.5	**53.8**	91.9	48.3	63.3	**76.7**	**63.6**	74.5	**57.0**	54.6	-	-	-	-	-
	RF (A)	56.6	52.2	93.9	48.8	64.3	50.8	50.1	**99.9**	50.1	**66.6**	-	-	-	-	-
Touch + Acce	SVM (M)	99.9	99.2	99.2	99.3	99.2	99.1	**99.8**	99.4	99.2	99.5	99.2	99.2	99.6	98.8	99.2
	k-NN (M)	99.8	99.0	8.9	99.2	99.0	98.7	99.7	99.1	99.3	99.4	99.1	98.9	99.2	98.5	98.9
	RF (M)	**100**	**99.9**	**99.8**	**99.9**	**99.9**	**99.7**	99.6	**99.9**	**99.8**	**99.8**	**99.9**	**99.8**	**99.7**	**100**	**99.8**
	SVM (A)	**93.6**	82.4	99.9	74.0	85.0	88.6	**68.8**	98.8	**62.0**	**76.2**	-	-	-	-	-
	k-NN (A)	**87.7**	**86.0**	**99.9**	**78.2**	**87.7**	81.2	60.1	98.6	60.0	64.7	-	-	-	-	-
	RF (A)	92.4	85.8	99.9	78.0	87.6	**99.2**	54.4	**99.8**	53.2	66.9	-	-	-	-	-

Table 7.6: *Bot detection performance metrics in % (AUC = Area Under the Curve, Acc = Accuracy, Re = Recall, Pre = Precision, and F1) for the different scenarios: Multiclass (M), Agnostic (A). Touch = Touchscreen, Acce = Accelerometer*

where the multimodal systems doubled their performance. These results suggest the potential of multimodal approaches, even in this challenging scenario where the synthetic training samples are not generated with the same method employed for the evaluation, in which the systems can maintain bot detection rates over 90%.

To better understand the results, Fig. 7.6 shows the probability functions of the six features proposed for the three types of touch signals (i.e., humans and both synthetic generation methods). Synthetic distributions do not completely fit the human distributions, but they present a behavior like the human samples. First, we can observe that the Move Efficiency of the handcrafted trajectories is equal to 1, this happens because in swipe trajectories with straight line shape the distance and displacement are equal. This is the reason why the multiclass classifiers detect these synthetic trajectories so easily. Note that the Duration (length) of both handcrafted and GAN synthetic swipes were computed as a Gaussian distribution with the same mean and standard deviation as the human ones so both probability distributions are equal. Regarding Distance and Displacement, the GAN trajectories fit worse than the handcrafted ones. We suggest that the main reason for this is that the GAN network generates smoother swipe trajectories than the human ones without abrupt direction changes, causing longer displacements in less distance (like a parabolic function). Finally, the Velocity Profile of both synthetic swipe trajectories are very similar to the human ones, the initial acceleration applied to the Function-based trajectories reproduces human behaviors with great similarity while the GAN network learns very realistic Velocity Profiles of human swipe trajectories as well.

7.3.4.2. Ablation study: Number of training samples

In Fig. 7.7 we first explore to what extent the number of training samples affects the classification performance. For this, we plot accuracy curves for the best classifier (i.e., RF for

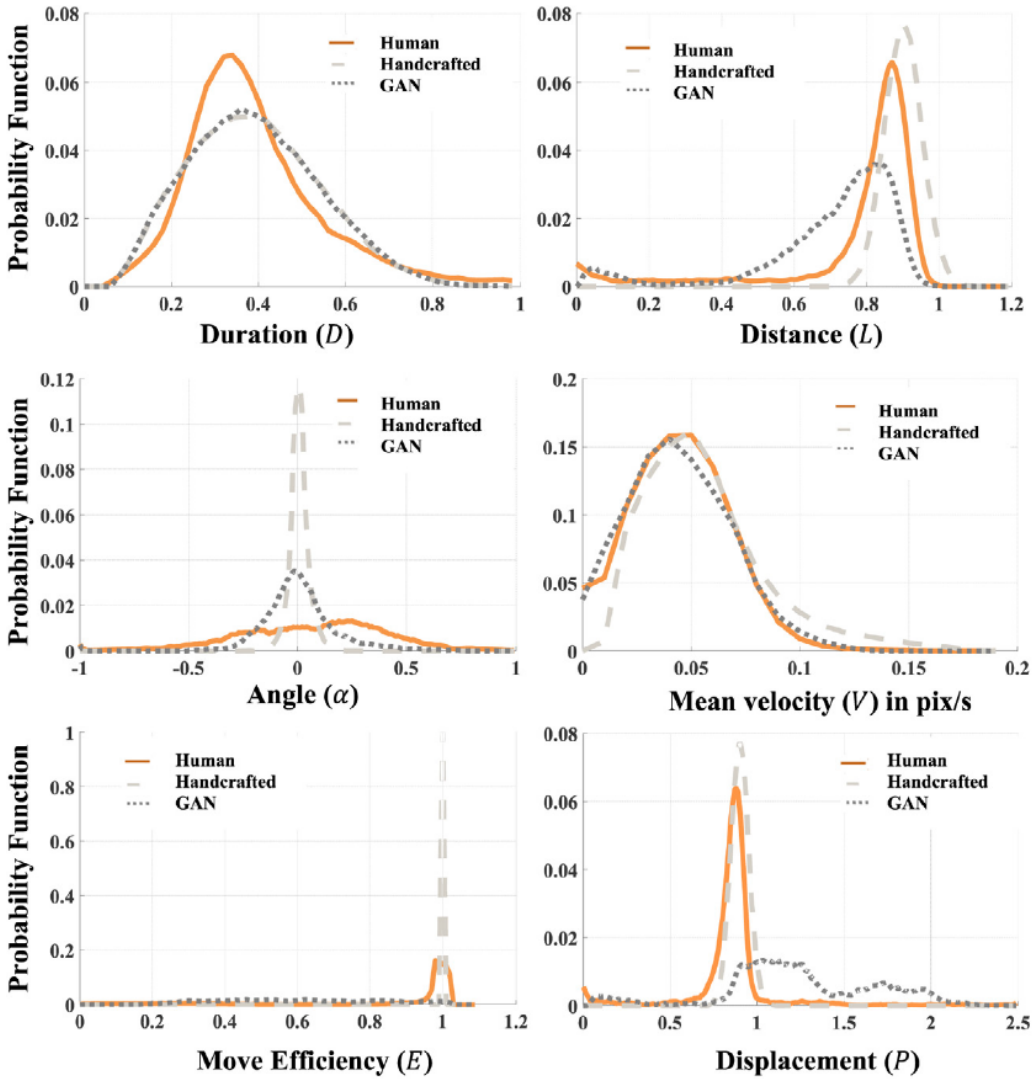

Figure 7.6: *Probability functions of the six global features for Human, Handcrafted, and GAN touch trajectories.*

multiclass and k-NN for agnostic classification) against the number of samples employed to train them (M). Remember that both training and evaluation sets are balanced so the number of human (M_h) and synthetic (M_s) train samples are equal, i.e.: $M_h = M_s = M/2$.

We can observe in Fig. 7.7 (left) that the accuracy improves when scaling up the number of train samples as we expected. The accuracy improves significantly up to $M = 1000$. On the other hand, it is surprising that the opposite tendency is observed in agnostic classification (Fig. 7.7 right), where the accuracy rates decay when scaling up the number of train samples. We suggest that the problem in agnostic classification is that classifiers are better trained to detect a specific synthetic generation method, making more difficult for them to detect synthetic samples generated with other methods as we increase the number of training samples with a specific method (i.e., some kind of overfitting to the specific bot generation method used for training).

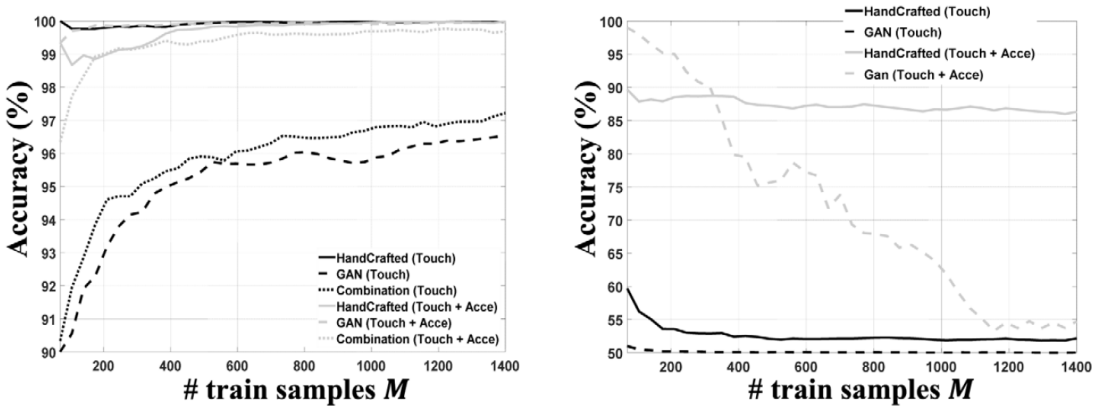

Figure 7.7: *Accuracy curves (%) against the number of train samples ($70 \leq M \leq 1400$) to train the different classifiers in multiclass (left) and agnostic (right) classification scenarios.*

	Bot Detection		
SVM Classifiers	**HandCrafted**	**GAN**	**HandCrafted + GAN**
One-class (Touch)	62.3	54.6	57.1
One-class (Touch + Acce)	89.2	79.4	80.5

Table 7.7: *Accuracy rates (%) in bot detection for the one-class SVM classifiers, where the SVM is trained with only human samples and tested with both synthetic generation methods.*

7.3.4.3. Performance of bot detection: One-class classification

The previous results encourage us to explore one-class classification scenario, where we train the classifier using only the human samples and test with both human and synthetic samples, in order to study whether the classifier is able to detect bots as abnormal human behavior. For this, we employ a SVM classifier that usually works well in one-class classification and set $M = 1000$. Table 7.7 shows the accuracy rates (%) for one-class SVM bot detection where rows represent the modality of the human samples (i.e., touch or touch plus accelerometer) employed to train the classifiers and in columns the bot generation method employed in the test. We can observe that synthetic samples generated with GAN can fool the classifier more times than the handcrafted samples as we expected, showing the potential of GAN networks to reproduce human trajectories with a great similarity, making almost impossible for the classifier to discriminate between synthetic GAN trajectories and human ones. The fusion of touch trajectories with accelerometer data improves the accuracy rates by more than 30%. GAN networks are not able to reproduce human accelerometer signals as well as touch trajectories, due to the complexity of the accelerometer signals, suggesting again the potential of multimodal approaches to deal with bot attacks.

7.3.4.4. Performance of bot detection: GAN discriminator

Besides the comparison among different classifier algorithms, we conduct another experiment in which we employ the GAN Discriminator as the classifier. In this experiment the previous

		Bot Detection									
		Handcrafted					GAN				
	GAN Discriminator	**AUC**	**Acc**	**Re**	**Pre**	**F1**	**AUC**	**Acc**	**Re**	**Pre**	**F1**
Tou	**LSTM (32/16)**	**92.2**	**86.8**	**85.7**	**89.2**	**87.4**	**78.3**	**77.8**	**79.2**	**76.7**	**78.3**
	LSTM (16/8)	70.0	65.2	64.3	67.3	66.3	54.4	52.2	54.3	55.1	54.7
	LSTM (32)	89.1	86.7	87.7	84.7	86.1	66.2	64.3	64.1	64.5	64.4
	LSTM (16)	89.9	87.4	89.9	86.6	87.2	52.5	52.2	53.3	51.9	52.6
Tou+Acce	**LSTM (32/16)**	85.8	77.7	74.2	80.5	77.3	63.8	59.2	60.0	62.1	61.1
	LSTM (16/8)	85.5	84.4	82.1	85.7	84.1	76.2	70.4	71.3	73.4	72.2
	LSTM (32)	61.1	65.3	68.4	64.4	66.3	61.7	57.7	58.8	55.6	56.7
	LSTM (16)	**93.4**	**88.8**	**89.9**	**91.2**	**90.8**	**81.1**	**74.4**	**77.3**	**75.5**	**76.4**

Table 7.8: *Performance metrics in % (AUC = Area Under the Curve, Acc, Pre, Re, and F1) for the different setups of GAN Discriminator in bot detection. In brackets the number of neurons for the first/second LSTM layer respectively used in the Discriminator. Tou = Touchscreen, Acce = Accelerometer*

feature extraction plus statistical classifier is replaced by a LSTM network (the Discriminator of the GAN). The fact that the Discriminator was trained with synthetic samples generated by the Generator during GAN training could perform better in the classification task than a neural network trained from scratch. Remember that the GAN Discriminator consists of two LSTM (Long Short-Term Memory) layers followed by a dense layer with a '*sigmoid*' activation function to discriminate between bots and humans, so in this experiment we tune the number of neurons of these two layers and train a new GAN network for each Discriminator setup.

Table 7.8 shows the bot detection performance metrics (%) for the different synthetic trajectories (columns) generated when comparing with the human ones. In rows, the different GAN Discriminator setups chosen for this experiment: two LSTM layers with 32 and 16 neurons respectively, two layers with 16 and 8 neurons respectively, one layer with 32 neurons, and one layer with 16 neurons.

First, it is surprising to observe that the GAN Discriminator performs better detecting synthetic handcrafted samples, even when the GAN Discriminator was trained only to discriminate between GAN synthetic and human samples. According to these results the GAN Discriminator can perform better than statistical classification algorithms as abnormal human behavior detector (e.g., agnostic and one-class classification scenarios). Regarding GAN Discriminator setups, configurations with larger number of neurons (i.e., 32 neurons in the first layer and 16 in the second one) seem to perform better for touch trajectories, and the opposite for the fusion with the accelerometer signals. We suggest that smooth and complex signals such as touch gestures need larger GAN Discriminator setups to be detected meanwhile more simple and noisy signals such as the accelerometer ones can be detected with smaller GAN Discriminator setups.

7.4. Chapter Summary and Conclusions

We have proposed BeCAPTCHA-Mouse, a bot detection algorithm based on mouse dynamics and the first one public for research in bot detection. Our method is based on neuromotor features extracted from each mouse trajectory and a learning framework including both real and synthetic samples. We have proposed and studied two new methods for generating synthetic mouse trajectories of varying level of realism. These generators are very useful both training stronger bot detectors and evaluating them in comprehensive and worst case scenarios. These generators are also valuable for related research problems beyond bot detection involving mouse dynamics. In our experiments we have observed the main features of human mouse trajectories (e.g., initial acceleration, final deceleration, and fine trajectory correction). Based on that we have developed a neuromotor feature representation using the Sigma-Lognormal model. Using the proposed neuromotor feature representation and training standard classifiers making use of the proposed synthetic mouse trajectories, we have been able to discriminate between humans and bots with up to 98.7% of accuracy, even with bots of high realism, and only one mouse trajectory as input (between two consecutive clicks). This proves the potential of mouse dynamics for Turing tests. Additionally, we also provided an exhaustive ablation study on different classifiers to explore the capacity of these algorithms for the bot detection task. Random Forests (RF) have demonstrated to perform the best in all scenarios evaluated followed by an LSTM network. However, when the number of train samples is reduced ($L \leq 1{,}000$), the LSTM is not able to classify as well as the RF classifier. In fact, the LSTM can be replaced by the Discriminator of the GAN network when the lack of bot samples to train the system makes the deep learning approaches unavailable, showing a superior performance even against bot samples not seen during the training phase. This results suggest that the GAN architecture is a powerful tool not only to generate human-like mouse trajectories, but also to detect bot samples from other synthetic generation methods.

We also proposed BeCAPTCHA-Mobile, a new BeCAPTCHA version that combines swipe touchscreen trajectories and accelerometer signals. We provide results in various experimental configurations and classifiers, considering or not synthetic bot data for training BeCAPTCHA-Mobile (multi-class, agnostic, and one-class). Bot detection results for agnostic classification (i.e., training with one synthetic bot method and testing with the other method) and one-class classification (i.e., training only with the human samples) just using touch gestures are poor with accuracies of around 60%, but the combination with accelerometer data improves the results to the range 80%-90% of accuracy. In addition, the case of multi-class training (i.e., training with both bot data generation methods) achieves very good performance, with results against very realistic synthetic attacks of over 90% of accuracy for bot detection. Regarding classifiers, Random Forest (RF) seems to perform the best in multi-class scenario while K-Nearest Neighbors (kNN) performs better in the agnostic scenario. In addition, the number of samples (human and bot) employed to train the classifiers affect considerably the performance, meanwhile in multiclass scenario, classifiers perform better as we increase the amount of samples to train

them. The opposite tendency is observed in agnostic scenario, where the classifiers reduce their capacity to detect bot samples from other methods as we increase the amount of training data to detect a specific kind of synthetic bot samples. Finally, employing the GAN Discriminator as a classifier reveals the potential of this LSTM network to detect bot samples generated using the handcrafted method, with a performance like using RF in multi-class scenario. Considering that the GAN Discriminator is only trained with human and GAN Generator samples, the potential of the GAN Discriminator for agnostic and one-class classification scenario is patent.

Part V

Conclusions

Chapter 8

Conclusions and Future Work

This final chapter summarizes the most important results and goals achieved in this Dissertation with reference to the research objectives of Chapter 1. The **major contributions** made in this Thesis are:

- Overview of signals and sensors employed in the literature to model human-machine interaction based on mobile devices.
- Performance analysis of user authentication based on 4 biometric data channels (touch gestures, keystroke, accelerometer, and gyroscope) and 3 behavior profiling data sources (WiFi, GPS, and App usage), obtained during natural human-smartphone interaction.
- Study of multimodal approaches for smartphone user authentication based on various combinations of the previous 7 data channels, both for One-Time Authentication and for Active Authentication schemes (i.e., continuously over multiple sessions). Our results demonstrate that signals from the smartphone can be used to improve user authentication under realistic usage conditions.
- Exploring a novel free-text keystroke biometrics approaches based on Deep Recurrent Neural Networks, suitable for authentication and identification at large scale. We have conducted an exhaustive experimentation and evaluated how performance is affected by the following factors: the length of the keystroke sequences, the number of gallery samples, and the device (touchscreen vs physical keyboard). We have presented TypeNet, a Recurrent Neural Network trained with keystroke sequences from more than 100,000 subjects. We have analyzed the performance of three different loss functions (softmax, contrastive, triplet) used to train TypeNet.
- Analyzing different factors that affect fixed-text the keystroke recognition performance for scenario in which each user type a proprietary password (300 passwords). We have provided new insights on fixed-text keystroke recognition performances including results

that contradict what has been known to date about the length of the passwords and its performances.

- Studying user age group classification based on the combination (at the feature level and score level) of neuromotor characteristics with global features obtained from touch interaction (i.e., swipe and tap gestures), following an active detection framework for different use cases.
- Evaluation of the suitability of handwriting patterns as potential biomarkers to model Parkinson's disease (PD). We have computed three feature sets extracted from the handwriting signals (i.e., neuromotor, kinematic, and nonlinear dynamic) and evaluate their performance with three different classifiers (i.e., SVM, k-NN, and MLP) for healthy control and Parkinson subjects classification.
- Proposing two new methods for generating realistic synthetic data: *i)* a Function-based method based on heuristic functions, and *ii)* a data-driven method based on GANs in which a Generator synthesizes data from a Gaussian noise input. We demonstrate the usefulness of these synthetic data to train more accurate bot detectors. These Generators can be helpful in many HCI research areas and applications.
- Development of BeCAPTCHA-Mouse, a new bot detector based on neuromotor modeling of mouse trajectories and supervised classification trained with human and synthetic data. Our proposed mouse detection algorithm can be added in a transparent setup and enhance traditional CAPTCHAs based on cognitive challenges, for example when you select the images in a visual CAPTCHA, or when you navigate through a website.
- Development of BeCAPTCHA-Mobile, a new bot detection approach based on modeling the user behavior in smartphone interaction using multiple inbuilt sensors. We also experiment with a particular implementation of the proposed approach by combining touch dynamics and accelerometer data from HuMIdb, acquired when the users perform swipe gestures. This is a very common gesture used in many touch interfaces (e.g., unlock devices, confirm will to advance to other step).
- Collection of the new public HuMIdb dataset (Human Mobile Interaction database) that characterizes the interaction of 600 users according to 14 sensors during normal human-mobile interactions in an unsupervised scenario with more than 200 different smartphone models.

8.1. Conclusions

Then, we describe the main conclusions drawn from the major contributions:

- In Chapter 3 we have presented new free-text keystroke biometrics systems based on an RNN architecture, trained with different learning strategies and evaluated over 4 public

databases. We present a comprehensive performance analysis including authentication and identification results obtained at very large scale. These experiments comprise more than 136 million keystrokes from 168,000 subjects captured on desktop keyboards and 60,000 subjects captured on mobile devices with more than 63 million keystrokes. Deep neural networks have shown to be effective in face recognition tasks when scaling up to hundreds of thousands of identities [Kemelmacher-Shlizerman *et al.*, 2016]. The same capacity has been shown by TypeNet models in free-text keystroke biometrics, with accuracies over 97% when testing with 100,000 users, showing the potential of our proposed method to operate at Internet scale.

- In Chapter 4 we have studied new algorithms for user mobile authentication based on multiple biometric and behavior-based profiling systems. For this, we studied two scenarios according to the number of mobile sessions used: one session (One-Time Authentication) and multiple sessions (Active Authentication). The results suggest that some biometric systems work better than others, and that the fusion with behavior-based profiling systems always improves the results, achieving accuracies up to 83.1% in the best case for an OTA scenario. Our experiments also suggest that Active Authentication always improve the accuracies with up to 14% of enhancement with respect to One-Time Authentication using between 5 and 7 mobile sessions.

- In Chapter 5 we have studied a new algorithm for age detection between children and adults according to their interaction with touchscreen devices like smartphones and tablets. Furthermore, we present an Active Authentication (AA) algorithm that takes advantage from the previous classifier results to identify children during a continuous interactions with the device. The correct classification rates are over 96% in the best scenario by combining both sigma-lognormal and global features, showing the potential of the proposed method to discriminate between children and young adults.

- In Chapter 6 we have employed one of the largest online handwriting database for Parkinson Disease (PD) research. Our experiments testbed with this database shows results of 96.9% accuracy in the classification of PD patients vs YHC (Young Healthy Controls), 81.7% in the classification of PD patients vs EHC (Elderly Healthy Controls), and 97.2% in the classification of EHC vs YHC when combining the three set of features proposed, showing the potential of online handwriting biometrics to identify specific characteristics of the disease in early stages for the opportune diagnosis and monitoring of PD.

- In Chapter 7 we have explored behavioral biometrics for bot detection during human-computer interaction. Our conclusions in comparison to state-of-the-art works suggest that there is unexploited potential of behavioral biometrics for bot detection tasks. We strongly believe that the combination of these behavioral signals with traditional CAPTCHA methods can harden significantly existing algorithms for bot detection, as we have demonstrated with our patented BeCAPTCHA application (es, P202030066). When combining both hu-

man and synthetic data to train our approach BeCAPTCHA has up to 95% of relative error reduction to discriminate between human and synthetic samples. The expected improvements will be even larger when considering additional biometric traits in extended BeCAPTCHA implementations beyond mouse, touchscreen and accelerometer data.

8.2. Future Work

Finally, a number of research lines arise from the work carried out in this Thesis. We consider of special interest the following ones:

- We will improve the way training pairs/triplets are chosen in Siamese/Triplet training for TypeNet models. Currently, the pairs are chosen randomly; however, recent work has shown that choosing *hard pairs* during the training phase can improve the quality of the embedding feature vectors [Wu *et al.*, 2017]. We will also explore improved learning architectures based on a combination of short- and long-term modeling, which has demonstrated to be very useful for modeling behavioral biometrics [Tolosana *et al.*, 2021a].

  In addition, we plan to test our model with other free-text keystroke databases to analyze the performance in other scenarios [Acien *et al.*, 2020b], and investigate alternate ways to combine the multiple sources of information [Fierrez *et al.*, 2018b] originated in the proposed framework, e.g., the multiple distances. Integration of keystroke data with other information captured at the same time in desktop [Hernandez-Ortega *et al.*, 2020a] and mobile acquisition [Acien *et al.*, 2019b] will be also explored.

  Finally, the proposed TypeNet models will be valuable beyond user authentication and identification, for applications related to human behavior analysis like profiling [Acien *et al.*, 2018], bot detection [Acien *et al.*, 2021a], and e-health [Giancardo *et al.*, 2018].

- We aim at building a reliable and fast classifier of users from all ages based on their touchscreen interaction [Fierrez *et al.*, 2018] by employing a combination of different expert systems [Fierrez *et al.*, 2018a]. The main drawback of other methods like using the browsing history or social network profiles is that they need a high amount of data. Our system allows us to classify users using data from simple and short tasks. This makes our solution suitable for applications that require classification on the fly. As further improvements of our developments, two aspects can be taken into account. First, the patterns used in this work are very simple: swipe and tap gestures. Better classification rates may be achieved if the information comes from more complex tasks or from continuous monitoring. Second, this study includes the analysis of touch patterns from children under 6 years. However, how to recognise users with mature neuromotor skills (from 10 years old onwards) is a challenging task and new models and methodologies should be proposed for that purpose in the future. The classification of older users using the sigma-lognormal model is a possibility since it is demonstrated that the neuromotor abilities decay with the age [Plamondon *et al.*, 2013].

- In bot detection, we aim at improving the neuromotor feature set by calculating secondary features inferred from the main ones. Also, we propose to improve the GAN model in two ways: *i)* combine both synthesis methods by using the function-based trajectories as the input of the GAN model instead of Gaussian noise, and *ii)* experimenting with different amount of layers/units in the GAN Generator to increase the variety of the synthetic mouse trajectories generated. Both techniques could generate more sophisticate and human-like trajectories. Finally, in this paper we only considered mouse trajectories acquired from mouse devices. We also propose to analyze mouse-pad trajectories normally performed when using laptops as another line of research.

  The exploitation of behavioral biometrics for bot detection is an open research line with large opportunities and challenges. We want to highlight that behavioral CAPTCHAs are compatible with previous CAPTCHA technologies and it could be added as a new cue to improve existing bot detection schemes in a multiple classifier combination [Fierrez *et al.*, 2018a].

8.3. Conclusions (Spanish)

A continuación, describimos las principales conclusiones extraídas de las contribuciones de esta Tesis:

- En el Capítulo 3 hemos presentado nuevos sistemas de reconocimiento biométrico basados en la dinámica de tecleo para texto libre, usando en una arquitectura RNN entrenada con diferentes estrategias de aprendizaje y evaluada sobre 4 bases de datos públicas. Presentamos un análisis de rendimiento exhaustivo que incluye resultados de autenticación e identificación obtenidos a gran escala. Dichos experimentos comprenden más de 136 millones de teclas pulsadas provenientes de 168.000 sujetos capturados en teclados físicos y 60.000 sujetos capturados en dispositivos móviles con más de 63 millones de teclas pulsadas. Las redes neuronales profundas han demostrado ser eficaces en tareas de reconocimiento facial a medida que se escalaban a cientos de miles de identidades [Kemelmacher-Shlizerman *et al.*, 2016]. La misma capacidad han mostrado los modelos TypeNet basados en la dinámica de tecleo para texto libre, con precisiones superiores al 97% al realizar pruebas con 100.000 usuarios, lo que demuestra el potencial de nuestro método propuesto para operar a gran escala.

- En el Capítulo 4 hemos estudiado nuevos algoritmos para la autenticación del usuario durante la interacción con el móvil, mediante la combinación de múltiples rasgos biométricos y rasgos basados en el perfil de conducta. Para ello, hemos estudiado dos escenarios según el número de sesiones móviles utilizadas: una sesión (Autenticación Única) y múltiples sesiones (Autenticación Activa). Los resultados sugieren que algunos sistemas biométricos funcionan mejor que otros, y que la fusión con sistemas basados en perfiles de comportamiento siempre mejora los resultados, logrando precisiones de hasta 83.1% en el mejor

caso para el escenario de Autenticación Única. Nuestros experimentos también sugieren que la Autenticación Activa siempre mejoran las precisiones con hasta un 14% con respecto a la Autenticación Única, utilizando entre 5 y 7 sesiones móviles.

- En el Capítulo 5 hemos desarrollado un algoritmo de detección de edad entre niños y adultos jóvenes según su interacción con dispositivos de pantalla táctil como smartphones y tabletas. Además, presentamos un algoritmo de Autentificación Activa (AA) que aprovecha los resultados del algortimo anterior y así poder identificar niños de forma activa durante múltiples interacciones consecutivas con el dispositivo. Los precisiones conseguidas superan el 96% en el mejor escenario, combinando características neuromotoras y globales, lo que demuestra el potencial del método propuesto para discriminar entre niños y adultos jóvenes.

- En el Capítulo 6 hemos empleado una de las mayores bases de datos de escritura on-line para la investigación de la enfermedad de Parkinson (EP). Nuestro resultados preliminares consiguen una precisión del 96,9% en la clasificación de pacientes con EP frente a participantes CJS (Controles Jóvenes Sanos), un 81,7% en la clasificación de pacientes con EP frente a CMS (Controles Mayores Sanos), y un 97,2% en la clasificación de CMS frente a CJS cuando se combinan los tres conjuntos de características propuestos. Lo que demuestra el potencial de la biometría basada en escritura on-line para identificar características específicas de la enfermedad de Parkinson en etapas tempranas y así poder ayudar para un óptimo diagnóstico y monitoreo.

- En el Capítulo 7 hemos estudiado la biometría del comportamiento para la detección de bots durante la interacción Hombre-Máquina. Nuestras conclusiones en comparación con trabajos previos del estado del arte sugieren que existe un potencial aún sin explotar en la biometría de comportamiento para su aplicación en la detección de bots. Creemos firmemente que la combinación de estas señales de comportamiento con otros métodos CAPTCHA tradicionales puede mejorar significativamente la detección de bots, como hemos demostrado con nuestra aplicación patentada BeCAPTCHA (es, P202030066). Cuando se combinan datos humanos y sintéticos para entrenar nuestro alogoritmo, BeCAPTCHA tiene hasta un 95% de reducción del error relativo discriminando entre muestras humanas y sintéticas. Las mejoras esperadas serán aún mayores cuando se consideren rasgos biométricos adicionales en futuras implementaciones de BeCAPTCHA ampliadas más allá de los paptrones del ratón, gestos en pantallas táctiles y las señales extraídas del acelerómetro.

References

A. Acien, J. Hernandez-Ortega, A. Morales, J. Fierrez, R. Vera-Rodriguez, and J. Ortega-Garcia. On the analysis of keystroke recognition performance based on proprietary passwords. In *Proc. of the 8th International Conference on Pattern Recognition Systems (ICPRS-17)*, pages 1–6, July 2017. 12

A. Acien, A. Morales, J. Fierrez, and R. Vera-Rodriguez. Becaptcha-mouse: Synthetic mouse trajectories and improved bot detection. *arXiv preprint arXiv:2005.00890*, 2020a. 14

A. Acien, A. Morales, J. Fierrez, R. Vera-Rodriguez, and O. Delgado-Mohatar. Becaptcha: Behavioral bot detection using touchscreen and mobile sensors benchmarked on humidb. *Engineering Applications of Artificial Intelligence*, 98:104058, 2021a. 14, 20, 28, 122

A. Acien, A. Morales, J. Fierrez, R. Vera-Rodriguez, and J. Hernandez-Ortega. Active detection of age groups based on touch interaction. *IET Biometrics*, 8(1):101–108, 2018. 13, 122

A. Acien, A. Morales, J. V. Monaco, R. Vera-Rodriguez, and J. Fierrez. Typenet: Deep learning keystroke biometrics. *arXiv preprint arXiv:2101.05570*, 2021b. 12

A. Acien, A. Morales, R. Vera-Rodriguez, and J. Fierrez. Keystroke mobile authentication: Performance of long-term approaches and fusion with behavioral profiling. In *Proc. of the Iberian Conference on Pattern Recognition and Image Analysis (IBPRIA)*, pages 12–24, 2019a. 7, 12

A. Acien, A. Morales, R. Vera-Rodriguez, and J. Fierrez. Smartphone sensors for modeling human-computer interaction: General outlook and research datasets for user authentication. In *Proc. of the 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1273–1278, 2020b. 12, 60, 122

A. Acien, A. Morales, R. Vera-Rodriguez, J. Fierrez, and J. V. Monaco. Typenet: Scaling up keystroke biometrics. In *Proc. of the IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–7, 2020c. 12

A. Acien, A. Morales, R. Vera-Rodriguez, J. Fierrez, and R. Tolosana. Multilock: Mobile active authentication based on multiple biometric and behavioral patterns. In *Proc. of the ACM International. Conference on Multimedia, Workshop on Multimodal Understanding and Learning for Embodied Applications (MULEA)*, pages 53–59, 2019b. 12, 60, 122

A. A. E. Ahmed and I. Traore. A new biometric technology based on mouse dynamics. *IEEE Transactions on Dependable and Secure Computing*, 4(3):165–179, 2007. 8, 97

I. Akrout, A. Feriani, and Akrout. Hacking google reCAPTCHA v3 using reinforcement learning. In *Proc. of the Conference on Reinforcement Learning and Decision Making*, 2019. 97

A. Al Galib and R. Safavi-Naini. User authentication using human cognitive abilities. In *Proc. of the Financial Cryptography and Data Security (ICFCDS)*, pages 254–271, 2015. 3

M. L. Ali, K. Thakur, C. C. Tappert, and M. Qiu. Keystroke biometric user verification using Hidden Markov Model. In *Proc. of the IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)*, pages 204–209, 2016. 36

F. Alonso-Fernandez, J. Fierrez, D. Ramos, and J. Gonzalez-Rodriguez. Quality-based conditional processing in multi-biometrics: application to sensor interoperability. *IEEE Trans. on Systems, Man and Cybernetics Part A*, 40(6):1168–1179, 2010. 53

A. Alsultan and K. Warwick. Keystroke dynamics authentication: A survey of free-text. *International Journal of Computer Science Issues (IJCSI)*, 10:1–10, 2013. 46

L. Anthony, Q. Brown, J. Nias, B. Tate, and S. Mohan. Interaction and recognition challenges in interpreting children's touch and gesture input on mobile devices. In *Proc. of the ACM international conference on Interactive tabletops and surfaces*, pages 225–234, 2012. 74

B. Ayotte, M. Banavar, D. Hou, and S. Schuckers. Fast free-text authentication via instance-based keystroke dynamics. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(4):377–387, 2020. xxvi, 37, 52

N. A. A. Aziz, F. Batmaz, R. Stone, and P. W. H. Chung. Selection of touch gestures for children's applications. In *Proc. of the Science and Information Conference*, pages 721–726, 2013. 73

K. O. Bailey, J. S. Okolica, and G. L. Peterson. User identification and authentication using multi-modal behavioral biometrics. *Computers & Security*, 43:77–89, 2014. 8

S. Banerjee and D. Woodard. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research*, 7:116–139, 2012. 7

N. Banovic, V. Rao, A. Saravanan, A. K. Dey, and J. Mankoff. Quantifying aversion to costly typing errors in expert mobile text entry. In *Proc. of the Conference on Human Factors in Computing Systems (CHI)*, pages 4229–4241, 2017. 50

S. Barra, G. Fenu, M. De Marsico, A. Castiglione, and M. Nappi. Have you permission to answer this phone?. In *Proc. of the International Workshop on Biometrics and Forensics (IWBF)*, pages 1–7, 2018. 5

F. Bergadano, D. Gunetti, and C. Picardi. User authentication through keystroke dynamics. *ACM Transactions on Information and System Security*, 5(4):367–397, 2002. 36

C. Bevan and D. S. Fraser. Different strokes for different folks? revealing the physical characteristics of smartphone users from their swipe gestures. *International Journal of Human-Computer Studies*, 88:51–61, 2016. 74

R. Blanco-Gonzalo, R. Sanchez-Reillo, J. Liu-Jimenez, and C. Sanchez-Redondo. How to assess user interaction effects in biometric performance. In *Proc. of the IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, pages 1–6, 2017. 9

K. Bock, D. Patel, G. Hughey, and D. Levin. uncaptcha: a low-resource defeat of recaptcha's audio challenge. In *Proc. of the 11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*, 2017. 10, 96

E. Bursztein, M. Martin, and J. Mitchell. Text-based captcha strengths and weaknesses. In *Proc. of the 18th ACM conference on Computer and Communications Security*, pages 125–138, 2011. 10, 96

D. Buschek, A. De Luca, and F. Alt. Improving accuracy, applicability and usability of keystroke biometrics on mobile touchscreen devices. In *Proc. of the ACM Conference on Human Factors in Computing Systems*, pages 1393–1402, 2015. 7, 50, 59, 60

D. Carneiro, P. Novais, J. M. Pêgo, N. Sousa, and J. Neves. Using mouse dynamics to assess stress during online exams. In *Proc. of the Hybrid Artificial Intelligent Systems*, pages 345–356, 2015. 8

M. Caruana, R. Vera-Rodriguez, and R. Tolosana. Analysing and exploiting complexity information in on-line signature verification. In *Proc. of the International Conference on Pattern Recognition Workshops, ICPRw*, 2021. 9

R. Castrillon, A. Acien, J. Orozco-Arroyave, A. Morales, J. Vargas, R.Vera-Rodriguez, J. Fierrez, J. Ortega-Garcia, and A. Villegas. Characterization of the handwriting skills as a biomarker for parkinson disease. In *Proc. of the IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, April 2019. 13, 20, 22

H. Ceker and S. Upadhyaya. User authentication with keystroke dynamics in long-text data. In *Proc. of the IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2016. xxii, 36, 38, 50, 54, 55

H. Çeker and S. Upadhyaya. Sensitivity analysis in keystroke dynamics using convolutional neural networks. In *Proc. of the IEEE Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2017. 38

M. Chen, Y. Zhang, Y. Li, S. Mao, and V. C. Leung. Emc: Emotion-aware mobile cloud computing in 5g. *IEEE Network*, 29(2):32–38, 2015. 7

M. C. Chen, J. R. Anderson, and M. H. Sohn. What can a mouse cursor tell us more? correlation of eye/mouse movements on web browsing. In *Proc. of the CHI '01 Extended Abstracts on Human Factors in Computing Systems*, pages 281–282, 2001. 8

G. Cho, J. H. Huh, J. Cho, S. Oh, Y. Song, and H. Kim. Syspal: System-guided pattern locks for android. In *Proc. of the 2017 IEEE Symposium on Security and Privacy (SP)*, pages 338–356, 2017. 9

Z. Chu, S. Gianvecchio, and H. Wang. *Bot or Human? A Behavior-Based Online Bot Detection System*, pages 432–449. Springer International Publishing, Cham, 2018. xxvii, 8, 97, 103, 104, 105, 107

Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Who is tweeting on twitter: human, bot, or cyborg?. In *Proc. of the 26th Annual Computer Security Applications Conference*, pages 21–30, 2010. 10

O. Costilla-Reyes, R. Vera-Rodriguez, A. S. Alharthi, S. U. Yunas, and K. B. Ozanyan. Deep learning in gait analysis for security and healthcare. In *Deep Learning: Algorithms and Applications*, volume 865, pages 299–334. Springer, 2020. 3

H. Crawford and E. Ahmadzadeh. Authentication on the go: Assessing the effect of movement on mobile device keystroke dynamics. In *Proc. of the 13th Symposium on Usable Privacy and Security (SOUPS)*, pages 163–173, 2017. 38

D. Crouse, H. Han, D. Chandra, B. Barbello, and A. K. Jain. Continuous authentication of mobile user: Fusion of face image and inertial measurement unit data. In *Proc. of the 2015 International Conference on Biometrics (ICB)*, pages 135–142, 2015. 3

C. De Stefano, F. Fontanella, D. Impedovo, G. Pirlo, and A. S. di Freca. Handwriting analysis to support neurodegenerative diseases diagnosis: A review. *Pattern Recognition Letters*, 121: 37–45, 2019. 86, 87

D. Deb, A. Ross, A. K. Jain, K. Prakah-Asante, and K. V. Prasad. Actions speak louder than (pass)words: Passive authentication of smartphone' users via deep temporal features. In *Proc. of the International Conference on Biometrics (ICB)*, pages 1–8, 2019. 5, 28, 36, 38, 45, 60, 61, 63

F. Demir, A. Sengur, H. Lu, S. Amiriparian, N. Cummins, and B. Schuller. Compact bilinear deep features for environmental sound recognition. In *Proc. of the International Conference on Artificial Intelligence and Data Processing (IDAP)*, pages 1–5, 2018. 7

V. Dhakal, A. Feit, P. O. Kristensson, and A. Oulasvirta. Observations on Typing from 136 Million Keystrokes. In *Proc. of the Conference on Human Factors in Computing Systems (CHI)*, 2018. xxvi, 20, 21, 45, 52

M. Djioua and R. Plamondon. A new algorithm and system for the characterization of handwriting strokes with delta-lognormal parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2060–2072, 2008. 25

P. Drotár, J. Mekyska, I. Rektorová, L. Masarová, Z. Smékal, and M. Faundez-Zanuy. Decision support framework for parkinson's disease based on novel handwriting markers. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 23(3):508–516, 2014. 89

P. Drotár, J. Mekyska, I. Rektorová, L. Masarová, Z. Smékal, and M. Faundez-Zanuy. Evaluation of handwriting kinematics and pressure for differential diagnosis of parkinson's disease. *Artificial intelligence in Medicine*, 67:39–46, 2016. 85

I. Dua, A. U. Nambi, C. V. Jawahar, and V. Padmanabhan. Autorate: How attentive is the driver?. In *Proc. of the 14th IEEE International Conference on Automatic Face Gesture Recognition (FG 2019)*, pages 1–8, 2019. 7

T. Duval, C. Rémi, R. Plamondon, J. Vaillant, and C. O'Reilly. Combining sigma-lognormal modeling and classical features for analyzing graphomotor performances in kindergarten children. *Human Movement Science*, 43:183–200, 2015. 75, 80, 81

M. A. Ferrer, M. Diaz, C. Carmona-Duarte, and A. Morales. A behavioral handwriting model for static and dynamic signature synthesis. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1041–1053, 2016. 9

M. A. Ferrer, M. Diaz-Cabrera, and A. Morales. Static signature synthesis: A neuromotor inspired approach for biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):667–680, 2014. 25

J. Fierrez, A. Morales, R. Vera-Rodriguez, and D. Camacho. Multiple classifiers in biometrics. part 1: Fundamentals and review. *Information Fusion*, 44:57–64, 2018a. 98, 122, 123

J. Fierrez, A. Morales, R. Vera-Rodriguez, and D. Camacho. Multiple classifiers in biometrics. part 2: Trends and challenges. *Information Fusion*, 44:103–112, 2018b. 47, 61, 66, 122

J. Fierrez and J. Ortega-Garcia. On-line signature verification. In *Handbook of biometrics*, pages 189–209. Springer, 2008. 9, 87

J. Fierrez, A. Pozo, M. Martinez-Diaz, J. Galbally, and A. Morales. Benchmarking touchscreen biometrics for mobile authentication. *IEEE Transactions on Information Forensics and Security*, 13(11):2720–2733, 2018. 3, 5, 59, 60, 62, 65, 76, 122

A. Fischer and R. Plamondon. A dissimilarity measure for on-line signature verification based on the sigma-lognormal model. In *Proc. of the 17th Biennial Conference of the International Graphonomics Society*, 2015. 76

A. Fischer and R. Plamondon. Signature verification based on the kinematic theory of rapid human movements. *IEEE Transactions on Human-Machine Systems*, 47(2):169–180, 2017. 25, 26, 89

B. Found, D. Dick, and D. Rogers. The structure of forensic handwriting and signature comparisons. *International Journal of Speech Language and the Law*, 1(2):183–196, 1994. 9

M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Transactions on Information Forensics and Security*, 8(1):136–148, 2013. 4

L. Fridman, S. Weber, R. Greenstadt, and M. Kam. Active authentication on mobile devices via stylometry, application usage, web browsing, and gps location. *IEEE Systems Journal*, 11 (2):513–521, 2016. 60, 61, 63

D. Gafurov, K. Helkala, and T. Søndrol. Biometric gait authentication using accelerometer sensor. *Journal Of Computers*, 1(7):51–59, 2006. 6

H. Gamboa, A. L. N. Fred, and A. K. Jain. Webbiometrics: User verification via web interaction. In *Proc. of the Biometrics Symposium*, pages 1–6, 2007. 8

H. Gascon, S. Uellenbeck, C. Wolf, and K. Rieck. Continuous authentication on mobile devices by analysis of typing motion behavior. *Sicherheit 2014–Sicherheit, Schutz und Zuverlässigkeit*, 2014. 36, 37

L. Giancardo, A. Sánchez-Ferro, T. Arroyo-Gallego, I. Butterworth, C. S. Mendoza, P. Montero, M. Matarazzo, J. A. Obeso, M. L. Gray, and R. S. J. Estépar. Computer keyboard interaction as an indicator of early parkinson's disease. *Scientific Reports*, 6, October 2018. 122

R. Giot, M. El-Abed, and C. Rosenberger. Greyc keystroke: a benchmark for keystroke dynamics biometric systems. In *Proc. of the 3rd International Conference on Biometrics: Theory, Applications, and Systems*, pages 1–6, 2009. 43

R. Giot and A. Rocha. Siamese networks for static keystroke dynamics authentication. In *Proc. of the IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2019. 38

E. Gonzalez-Sosa, J. Fierrez, R. Vera-Rodriguez, and F. Alonso-Fernandez. Facial soft biometrics for recognition in the wild: Recent works, annotation, and cots evaluation. *IEEE Transactions on Information Forensics and Security*, 13(8):2001–2014, 2018. 3

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proc. of the 27th International Conference on Neural Information Processing Systems - Volume 2*, pages 2672–2680, 2014. 108

Y. Gorodnichenko, T. Pham, and O. Talavera. Social media, sentiment and public opinions: Evidence from #brexit and #uselection. Working Paper 24631, National Bureau of Economic Research, May 2018. 10, 96

D. Gunetti and C. Picardi. Keystroke analysis of free text. *ACM Transactions on Information and System Security*, 8(3):312–347, 2005. 36, 37

R. Hadsell, S. Chopra, and Y. Lecun. Dimensionality reduction by learning an invariant mapping. In *Proc. of the Computer Vision and Pattern Recognition Conference*, 2006. 29

M. Harbach, E. Von Zezschwitz, A. Fichtner, A. De Luca, and M. Smith. It's a hard lock life: A field study of smartphone (un)locking behavior and risk perception. In *Proc. of the 10th Symposium On Usable Privacy and Security ({SOUPS})*, pages 213–230, 2014. 9

E. Heremans, E. Nackaerts, S. Broeder, G. Vervoort, S. P. Swinnen, and A. Nieuwboer. Handwriting impairments in people with parkinson's disease and freezing of gait. *Neurorehabilitation and neural repair*, 30(10):911–919, 2016. 86

J. Hernandez-Ortega, R. Daza, A. Morales, J. Fierrez, and J. Ortega-Garcia. edBB: Biometrics and Behavior for assessing remote education. In *Proc. of the AAAI Workshop on Artificial Intelligence for Education (AI4EDU)*, 2020a. 7, 8, 122

J. Hernandez-Ortega, J. Fierrez, A. Morales, and D. Diaz. A comparative evaluation of heart rate estimation methods using face videos. In *Proc. of the IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1438–1443, 2020b. 7

J. Hernandez-Ortega, A. Morales, J. Fierrez, and A. Acien. Detecting age groups using touch interaction based on neuromotor characteristics. *IET Electronics Letters*, pages 1–2, September 2017. 13

J. Huang, D. Hou, S. Schuckers, and Z. Hou. Effect of data size on performance of free-text keystroke authentication. In *Proc. of the IEEE International Conference on Identity, Security and Behavior Analysis (ISBA 2015)*, pages 1–7, 2015. 37

H. Hyyro. Bit-parallel approximate string matching algorithms with transposition. *Journal of Discrete Algorithms*, 3(2):215–229, 2005. 55

D. Impedovo and G. Pirlo. Dynamic handwriting analysis for the assessment of neurodegenerative diseases: a pattern recognition perspective. *IEEE reviews in biomedical engineering*, 12: 209–220, 2018. 86

B. Inhelder and J. Piaget. The psychology of the child (vol. 5001), 1969. 74

A. Jain, K. Nandakumar, and A. Ross. Score normalization in multimodal biometric systems. *Pattern recognition*, 38(12):2270–2285, 2005. 76

A. K. Jain, K. Nandakumar, and A. Ross. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern recognition letters*, 79:80–105, 2016. 4

I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882, 2016. 121, 123

K. S. Killourhy and R. A. Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. In *Proc. of the International Conference on Dependable Systems & Networks*, pages 125–134, 2009. 38, 43

J. Kim and P. Kang. Freely typed keystroke dynamics-based user authentication for mobile devices based on heterogeneous features. *Pattern Recognition*, 108:107556, 2020. 36, 38, 52

C. Kotsavasiloglou, N. Kostikis, D. Hristu-Varsakelis, and M. Arnaoutoglou. Machine learning-based classification of simple drawing movements in parkinson's disease. *Biomedical Signal Processing and Control*, 31:174–180, 2017. 86

G. Li and P. Bours. A mobile app authentication approach by fusing the scores from multi-modal data. In *Proc. of the 21st International Conference on Information Fusion (FUSION)*, pages 2091–2097, 2018a. 60, 61, 63, 107

G. Li and P. Bours. A novel mobilephone application authentication approach based on accelerometer and gyroscope data. In *Proc. of the International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–4, 2018b. 59, 60, 65

G. Li and P. Bours. Studying wifi and accelerometer data based authentication method on mobile phones. In *Proc. of the 2nd international conference on biometric engineering and applications*, pages 18–23, 2018c. 6, 59, 60, 64, 65, 107

Y. Lin, S. Cheng, J. Shen, and M. Pantic. Mobiface: A novel dataset for mobile face tracking in the wild. In *Proc. of the 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–8, 2019. 7

X. Liu, C. Shen, and Y. Chen. Multi-source interactive behavior analysis for continuous user authentication on smartphones. In *Proc. of the Chinese Conference on Biometric Recognition*, pages 669–677, 2018. 60, 61

C. C. Loy, C. P. Lim, and W. K. Lai. Pressure-based typing biometrics user authentication using the fuzzy artmap neural network. In *Proc. of the 20th International Conference on Neural Information Processing (ICONIP)*, pages 647–652, 2005. 39

X. Lu, Z. Shengfei, and Y. Shengwei. Continuous authentication by free-text keystroke based on CNN plus RNN. *Procedia Computer Science*, 147:314–318, 01 2019. xxii, 28, 36, 38, 45, 50, 51, 54, 55

U. Mahbub and R. Chellappa. PATH: Person authentication using trace histories. In *Proc. of the IEEE 7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pages 1–8, 2016. 6, 59, 60

U. Mahbub, J. Komulainen, D. Ferreira, and R. Chellappa. Continuous authentication of smartphones based on application usage. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 1(3):165–180, 2019. 6, 60

U. Mahbub, S. Sarkar, V. M. Patel, and R. Chellappa. Active user authentication for smartphones: A challenge data set and benchmark results. In *Proc. of the IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8, 2016. 20, 22

S. Marcel, M. S. Nixon, J. Fierrez, and N. Evans. *Handbook of biometric anti-spoofing: Presentation attack detection*. Springer, 2019. 61

D. Martín-Albo, L. A. Leiva, J. Huang, and R. Plamondon. Strokes of insight. *Inf. Process. Manage*, 52(6):989–1003, 2016. 8

M. Martinez-Diaz, J. Fierrez, and J. Galbally. Graphical password-based user authentication with free-form doodles. *IEEE Transactions on Human-Machine Systems*, 46(4):607–614, 2016. 9

M. Martinez-Diaz, J. Fierrez, R. P. Krish, and J. Galbally. Mobile signature verification: Feature robustness and performance comparison. *IET Biometrics*, 3(4):267–277, 2014. 62, 65, 76

L. McKnight and B. Cassidy. Children's interaction with mobile touch-screen devices: experiences and guidelines for design. In *Social and organizational impacts of emerging mobile devices: Evaluating use*, pages 72–89. IGI Global, 2012. 11, 73

J. McLennan, K. Nakano, H. Tyler, and R. Schwab. Micrographia in parkinson's disease. *Journal of the neurological sciences*, 15(2):141–152, 1972. 11

R. G. Meulenbroek and G. P. Van Galen. The acquisition of skilled handwriting: Discontinuous trends in kinematic variables. In *Advances in psychology*, volume 55, pages 273–281. Elsevier, 1988. 75

J. V. Monaco. Robust keystroke biometric anomaly detection. *arXiv preprint arXiv:1606.09075*, June 2016. 36, 44

J. V. Monaco and C. C. Tappert. The partially observable hidden markov model and its application to keystroke dynamics. *Pattern Recognition*, 76:449–462, 2018. xxii, 36, 37, 50, 54, 55, 60

S. Mondal and P. Bours. A study on continuous authentication using a combination of keystroke and mouse biometrics. *Neurocomputing*, 230:1–22, 2017. 8

S. Mondal, P. Bours, and S. S. Idrus. Complexity measurement of a password for keystroke dynamics: Preliminary study. In *Proc. of the 6th International Conference on Security of Information and Networks*, pages 301–305, 2013. 39, 42

F. Monrose and A. Rubin. Authentication via keystroke dynamics. In *Proc. of the 4th ACM Conference on Computer and Communications Security*, pages 48–56, 1997. 36

J. Montalvão, E. O. Freire, M. A. Bezerra Jr, and R. Garcia. Contributions to empirical analysis of keystroke dynamics in passwords. *Pattern Recognition Letters*, 52:80–86, 2015. 39, 42

A. Morales, J. Fierrez, and J. Ortega-Garcia. Towards predicting good users for biometric recognition based on keystroke dynamics. In *Proc. of the European Conference on Computer Vision Workshops*, volume 8926 of *LNCS*, pages 711–724. Springer, September 2014. 39, 47

A. Morales, J. Fierrez, R. Tolosana, J. Ortega-Garcia, J. Galbally, M. Gomez-Barrero, A. Anjos, and S. Marcel. Keystroke biometrics ongoing competition. *IEEE Access*, 4:7736–7746, 2016. 20, 36, 39, 42, 57, 65

M. Muaaz and R. Mayrhofer. Smartphone-based gait recognition: From authentication to imitation. *IEEE Transactions on Mobile Computing*, 16(11):3209–3221, 2017. 3

J. Mucha, V. Zvoncak, Z. Galaz, M. Faundez-Zanuy, J. Mekyska, T. Kiska, Z. Smekal, L. Brabenec, I. Rektorova, and K. Lopez-de Ipina. Fractional derivatives of online handwriting: A new approach of parkinsonic dysgraphia analysis. In *Proc. of the 41st International Conference on Telecommunications and Signal Processing (TSP)*, pages 1–4, 2018. 85

C. Murphy, J. Huang, D. Hou, and S. Schuckers. Shared dataset on natural human-computer interaction to support continuous authentication research. In *Proc. of IEEE/IAPR International Joint Conference on Biometrics (IJCB)*, pages 525–530, 2017. 36, 37, 53

J. C. Neves, R. Tolosana, R. Vera-Rodriguez, V. Lopes, H. Proenca, and J. Fierrez. GANprintR: Improved fakes and evaluation of the state of the art in face manipulation detection. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):1038–1048, August 2020. 103

M. O'Neal, K. Balagani, V. Phoha, A. Rosenberg, A. Serwadda, and M. E. Karim. Context-aware active authentication using touch gestures, typing patterns and body movement. Technical report, Louisiana Tech University, 2016. 65

C. O'Reilly and R. Plamondon. Development of a sigma–lognormal representation for on-line signatures. *Pattern recognition*, 42(12):3324–3337, 2009. 74

K. Palin, A. Feit, S. Kim, P. O. Kristensson, and A. Oulasvirta. How do People Type on Mobile Devices? Observations from a Study with 37,000 Volunteers. In *Proc. of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, 2019. xxvi, 20, 21, 37, 45, 52

V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello. Continuous user authentication on mobile devices: Recent progress and remaining challenges. *IEEE Signal Processing Magazine*, 33(4):49–61, 2016. 3, 6, 61, 73

L. Pei, R. Chen, J. Liu, T. Tenhunen, H. Kuusniemi, and Y. Chen. Inquiry-based bluetooth indoor positioning via rssi probability distributions. In *Proc. of the 2nd International Conference on Advances in Satellite and Space Communications*, pages 151–156, 2010. 6

P. Perera and V. M. Patel. Quickest intrusion detection in mobile active user authentication. In *Proc. of the IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8, 2016. 27

P. Perera and V. M. Patel. Efficient and low latency detection of intruders in mobile active authentication. *IEEE Transactions on Information Forensics and security*, 13(6):1392–1405, 2017a. 27

P. Perera and V. M. Patel. Towards multiple user active authentication in mobile devices. In *Proc. of the 12th International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 354–361, 2017b. 73

P. A. Pérez-Toro, J. C. Vásquez-Correa, T. Arias-Vergara, N. Garcia-Ospina, J. R. Orozco-Arroyave, and E. Nöth. A non-linear dynamics approach to classify gait signals of patients with parkinson's disease. In *Proc. of the Workshop on Engineering Applications*, pages 268–278, 2018. 89

R. Plamondon. A kinematic theory of rapid human movements. *Biological Cybernetics*, 72(4): 295–30, 1995. 8, 25, 98

R. Plamondon, C. O'Reilly, C. Rémi, and T. Duval. The lognormal handwriter: learning, performing, and declining. *Frontiers in psychology*, 4:945, 2013. 122

S. Rosenblum, M. Samuel, S. Zlotnik, I. Erikh, and I. Schlesinger. Handwriting as an objective tool for parkinson's disease diagnosis. *Journal of neurology*, 260(9):2357–2361, 2013. 11

A. A. Salah, T. Gevers, *et al. Computer analysis of human behavior*. Springer, 2011. 3

A. P. Saygin, I. Cicekli, and V. Akman. Turing test: 50 years later. *Minds and Machines*, 10: 463–518, 10 2000. 95

M. Schuster. Speech recognition for mobile devices at google. In *Proc. of the Pacific Rim International Conference on Artificial Intelligence*, pages 8–10, 2010. 7

A. Serwadda, V. V. Phoha, and Z. Wang. Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms. In *Proc. of the 6th international conference on biometrics: theory, applications and systems (BTAS)*, pages 1–8, 2013. 76

D. Shanmugapriya and G. Padmavathi. A survey of biometric keystroke dynamics: approaches, security and challenges. *arXiv preprint arXiv:0910.0817*, 2009. 43

C. Shen, Z. Cai, X. Guan, and R. Maxion. Performance evaluation of anomaly-detection algorithms for mouse dynamics. *Computers & Security*, 45:156–171, 2014. 19, 20

W. Shi, J. Yang, Yifei Jiang, Feng Yang, and Yingen Xiong. Senguard: Passive user identification on smartphones using multiple sensors. In *Proc. of the IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 141–148, 2011. 7, 60, 61

T. Sim and R. Janakiraman. Are digraphs good for free-text keystroke dynamics? In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 36

T. Sim, S. Zhang, R. Janakiraman, and S. Kumar. Continuous verification using multimodal biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):687–700, 2007. 8

E. J. Smits, A. J. Tolonen, L. Cluitmans, M. Van Gils, B. A. Conway, R. C. Zietsma, K. L. Leenders, and N. M. Maurits. Standardized handwriting to assess bradykinesia, micrographia and tremor in parkinson's disease. *PloS one*, 9(5):e97614, 2014. 11

R. Snelick, U. Uludag, A. Mink, M. Indovina, and A. Jain. Large-scale evaluation of multimodal biometric authentication using state-of-the-art systems. *IEEE transactions on pattern analysis and machine intelligence*, 27(3):450–455, 2005. 43

R. Spreitzer. Pin skimming: exploiting the ambient-light sensor in mobile devices. In *Proc. of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, pages 51–62, 2014. 6

A.-S. Suleyman, A.-J. Naseer, and H. Sellahewa. User-age classification using touch gestures on smartphones. *International Journal of Multidisciplinary Studies*, 2(1), 2015. 74, 75

L. Sun, D. Zhang, B. Li, B. Guo, and S. Li. Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations. In *Proc. of the International conference on ubiquitous intelligence and computing*, pages 548–562, 2010. 6

Y. Sun, H. Ceker, and S. Upadhyaya. Shared keystroke dataset for continuous authentication. In *Proc. of the IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2016. xxvi, 38, 52, 53

Z. Syed, S. Banerjee, Q. Cheng, and B. Cukic. Effects of user habituation in keystroke dynamics on password security policy. In *Proc. of the IEEE 13th International Symposium on High-Assurance Systems Engineering*, pages 352–359, 2011. 39

C. Taleb, M. Khachab, C. Mokbel, and L. Likforman-Sulem. Feature selection for an improved parkinson's disease identification based on handwriting. In *Proc. of the 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, pages 52–56, 2017. 86

M. Tavakolian, C. G. B. Cruces, and A. Hadid. Learning to detect genuine versus posed pain from facial expressions using residual generative adversarial networks. In *Proc. of the 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–8, 2019. 7

P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen. A survey on touch dynamics authentication in mobile devices. *Computers & Security*, 59:210–235, 2016. 37, 50

M. Thomas, A. Lenka, and P. Kumar Pal. Handwriting analysis in parkinson's disease: current status and future directions. *Movement Disorders Clinical Practice*, 4(6):806–818, 2017. 11

R. Tolosana, P. Delgado-Santos, A. Perez-Uribe, R. Vera-Rodriguez, J. Fierrez, and A. Morales. DeepWriteSYN: On-line handwriting synthesis via deep short-term representations. In *Proc. of the Conference on Artificial Intelligence (AAAI)*, February 2021a. 122

R. Tolosana, J. C. Ruiz-Garcia, R. Vera-Rodriguez, J. Herreros-Rodriguez, S. Romero-Tapiador, A. Morales, and J. Fierrez. Child-computer interaction: Recent works, new dataset, and age detection. *arXiv preprint arXiv: 2102.01405*, 2021b. 11

R. Tolosana, R. Vera-Rodriguez, and J. Fierrez. Biotouchpass: Handwritten passwords for touchscreen biometrics. *IEEE Transactions on Mobile Computing*, 19(7):1532–1543, 2020a. 4

R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia. Benchmarking desktop and mobile handwriting across cots devices: the e-biosign biometric database. *PLOS ONE*, 5(12), 2017. 9

R. Tolosana, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia. Exploring recurrent neural networks for on-line handwritten signature biometrics. *IEEE Access*, 6:5128–5138, 2018. 9, 61

R. Tolosana, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia. Reducing the template aging effect in on-line signature biometrics. *IET Biometrics*, 8(6):422–430, June 2019. 9

R. Tolosana, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia. BioTouchPass2: Touchscreen password biometrics using Time-Aligned Recurrent Neural Networks. *IEEE Transactions on Information Forensics and Security*, 2020b. 28, 45

R. Tolosana, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia. Deepsign: Deep on-line signature verification. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2021c. 28, 45, 61

R. Tolosana, R. Vera-Rodriguez, R. Guest, J. Fierrez, and J. Ortega-Garcia. Exploiting complexity in pen-and touch-based signature biometrics. *International Journal on Document Analysis and Recognition (IJDAR)*, 23(2):129–141, 2020c. 9

R. Tolosana, R. Vera-Rodriguez, J. Ortega-Garcia, and J. Fierrez. Preprocessing and feature selection for improved sensor interoperability in online biometric signature verification. *IEEE Access*, 3:478–489, 2015. 9

P. Tome, J. Fierrez, R. Vera-Rodriguez, and M. Nixon. Soft biometrics and their application in person recognition at a distance. *IEEE Transactions on Information Forensics and Security*, 9(3):464–475, March 2014. 3

Y. Tousignant-Laflamme, N. Boutin, A. M. Dion, and C.-A. Vallée. Reliability and criterion validity of two applications of the iphone to measure cervical range of motion in healthy participants. *Journal of neuroengineering and rehabilitation*, 10(1):1–9, 2013. 7

I. Traore. *Continuous Authentication Using Biometrics: Data, Models, and Metrics: Data, Models, and Metrics*. Igi Global, 2011. 4

C. M. Travieso, J. B. Alonso, J. R. Orozco-Arroyave, J. F. Vargas-Bonilla, E. Nöth, and A. G. Ravelo-García. Detection of different voice diseases based on the nonlinear characterization of speech signals. *Expert Systems with Applications*, 82:184–195, 2017. 89

R.-D. Vatavu, L. Anthony, and Q. Brown. Child or adult? inferring smartphone users' age group from touch measurements alone. In *Proc. of the IFIP Conference on Human-Computer Interaction*, pages 1–9, 2015a. 79

R.-D. Vatavu, G. Cramariuc, and D. M. Schipor. Touch interaction for children aged 3 to 6 years: Experimental findings and relationship to motor skills. *International Journal of Human-Computer Studies*, 74:54–76, 2015b. 20, 22, 74, 75

R. Vera-Rodriguez, J. Fierrez, J. Ortega-Garcia, A. Acien, and R. Tolosana. e-biosign tool: towards scientific assessment of dynamic signatures under forensic conditions. In *Proc. of the IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–6, 2015. 9

R. Vera-Rodriguez, R. Tolosana, R. Plamondon, A. Marcelli, and M. Ferrer. Modeling the complexity of signature and touch-screen biometrics using the lognormality principle. In *The Lognormality Principle and its Applications*. World Scientific, 2019. 5, 26

K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009. 29

C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl. Sampling matters in deep embedding learning. In *Proc. of the IEEE International Conference on Computer Vision*, pages 2840–2848, 2017. 122

Y. Xie and S.-Z. Yu. A large-scale hidden semi-markov model for anomaly detection on user browsing behaviors. *IEEE/ACM Transactions on Networking*, 17:54–65, 02 2009. 96

T. Yasseri, R. Sumi, A. Rung, A. Kornai, and J. Kertesz. Dynamics of conflicts in Wikipedia. *PLOS ONE*, 7(6):1–12, 2012. 8