UNIVERSITAT D'ANDORRA

Programa de doctorat de la Universitat d'Andorra

Identifying users using Keystroke Dynamics and contextual information

Identificació d'usuaris mitjançant cadència de tecleig i dades contextuals

Aleix Dorca Josa

Direcció: Dr. Jose Antonio Morán Moreno i Dra. Eugènia Santamaría Pérez
Identificador: TD-049-100018/201710
Data de defensa: 5 de febrer de 2018

UNIVERSITAT D'ANDORRA

Programa de doctorat de la Universitat d'Andorra

Identifying users using Keystroke Dynamics and contextual information

Identificació d'usuaris mitjançant cadència de tecleig i dades contextuals

PhD Thesis

Tesi doctoral

Aleix Dorca Josa

supervised by

Eugènia Santamaría Pérez

Jose Antonio Morán Moreno

October 6, 2017

To my wife Nina and sons Aleix & Jordi

Sorry...

"So long, and thanks for all the fish"

Douglas Adams

Contents

List of Figures

List of Tables

List of Listings

List of Abbreviations

AUC Area Under the Curve. 22, 37

CCTV Close Circuit Television. 2

DBMS Database Management System. 235

EER Equal Error Rate. 21, 22, 36, 38–41, 101, 107, 155, 156, 159, 160, 162, 166, 187

FAR False Acceptance Rate. 20–22, 35–38, 40–42, 47, 48, 101, 106, 107, 155, 156, 159

FMR False Match Rate. 21

FNMR False Non-Match Rate. 21

FRR False Rejection Rate. 20, 21, 35–38, 40, 41, 47, 48, 101, 106, 107, 155, 156, 159

FTA Failure to Acquire. 21

FTE Failure to Enroll. 21

HTTP Hypertext Transfer Protocol. 60

KD KeyDown. 12, 14, 45, 71

KU KeyUp. 12, 14, 71

LCMS Learning Content Management System. 57, 59, 60

MCCV Monte Carlo Cross-Validation. 103, 112, 114, 115, 124, 125, 133, 147, 150, 151, 155, 161

ME Margin of Error. 151–153, 161, 163, 164, 167–169, 174–176

PCA Principal Components Analysis. 23

PCIS Percentage of Correctly Identified Sessions. 106, 115, 116, 151, 160, 162, 165, 167

ROC Receiver Operating Characteristic. 22, 101, 106, 156

SMS Short Messaging System. 1

SVM Support Vector Machine. 20, 31, 37, 40–42, 90

Declaration

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Signed: ..... **Date:** .....

Abstract

Biometric identification systems based on Keystroke Dynamics have been around for almost forty years now. There has always been a lot of interest in identifying individuals using their physiological or behavioral traits. Keystroke Dynamics focuses on the particular way a person types on a keyboard.

The objective of the proposed research is to determine how well the identity of users can be established when using this biometric trait and when contextual information is also taken into account. The proposed research focuses on free text. Users were never told what to type, how or when. This particular field of Keystroke Dynamics has not been as thoroughly studied as the fixed text alternative where a plethora of methods have been tried.

The proposed methods focus on the hypothesis that the position of a particular letter, or combination of letters, in a word is of high importance. Other studies have not taken into account if these letter combinations had occurred at the beginning, the middle, or the end of a word.

A template of the user will be built using the context of the written words and the latency between successive keystrokes. Other features, like word length, minimum number of needed words to consider a session valid, frequency of words, model building parameters, as well as age group and gender have also been studied to determine those that better help ascertain the identity of an individual.

The results of the proposed research should help determine if using Keystroke Dynamics and the proposed methodology are enough to identify users from the content they type with a good enough level of certainty. From this moment, it could be used as a method to ensure that a user is not supplanted, in authentication schemes, or even to help determine the authorship of different parts of a document written by more than one user.

Keywords: Keystroke Dynamics, context, free text, identification, authentication

Resum

Els sistemes d'identificació biomètrica basades en la cadència de tecleig fa gairebé quaranta anys que s'estudien. Hi ha hagut molt interès en identificar les persones a partir de les seves característiques fisiològiques o de comportament. La cadència de tecleig és la manera en la que una persona escriu en un teclat.

L'objectiu de la recerca proposada és determinar com de bé es pot arribar a identificar un individu mitjançant aquesta característica biomètrica i quan també es prenen en consideració dades contextuals. Aquesta recerca es basa en text lliure. Als usuaris mai se'ls va dir què, quan o com havien d'escriure. Aquest camp de la cadència de tecleig no ha estat tan estudiat com l'alternativa de text fix on un gran ventall de mètodes s'han provat.

Els mètodes d'identificació proposats es basen en la hipòtesi que la posició d'una lletra, o combinació de lletres teclejades, en una paraula és de gran importància. Altres estudis no prenen en consideració aquesta informació, és a dir, si la combinació de lletres s'ha produït al principi, al mig o al final de la paraula.

Es crearà una empremta de l'usuari tenint en compte el context de les lletres en les paraules escrites i les latències entre pulsacions successives. Altres característiques com la mida de les paraules, el nombre mínim de paraules necessari per considerar una sessió vàlida, la freqüència de mots, els paràmetres de construcció dels models, així com el grup d'edat i el gènere també s'han estudiat per determinar quines són les que millor ajuden a identificar un individu.

Els resultats de la recerca proposada haurien de permetre determinar si l'ús de la cadència de tecleig i els mètodes proposats són suficients per identificar els usuaris a partir del contingut que generen, sempre amb un cert marge d'error. En cas afirmatiu es podria introduir la tècnica proposada com un mètode més per assegurar que un usuari no és suplantat, en sistemes d'autenticació, o fins i tot per ajudar a determinar l'autoria de diferents parts d'un document que ha estat escrit per més d'un usuari.

Paraules clau: Cadència de tecleig, context, text lliure, identificació, autenticació

Acknowledgments

I would like to express my sincere gratitude to Dr. Eugènia Santamaría Pérez and Dr. Jose Antonio Morán Moreno for the continuous support during these three years, for their patience, for their motivation in the hardest moments and for keeping me focused in the easier ones. I could not have imagined having better supervisors for this study.

I would like to thank the members of the committees that helped define the present work: Dr. Ferran Virgós Bel, Dr. Juan Alberto Sigüenza Pizarro, and Dr. Carlos Monzo Sánchez, for their insightful comments and encouragement, but also for their questions regarding the research which encouraged me to widen it from various perspectives.

My sincere thanks also go to Dr. Miquel Nicolau i Vila and Dr. Virginia Larraz Rada, who provided me with the opportunity to use the University of Andorra as my lab without ever saying no to any demand. Without their precious support, it would have not been so easy to conduct this research.

I thank my fellow workmates: Víctor Llorente Vaquero, Sergio Gil Ovejero, Rui Filipe Marques Fernandes, and Roc Duran Martinez for their help and for all the fun we have had these last three years. Sorry for the noise, you guys! I also thank my colleagues and friends at the University of Andorra, especially those part of the Doctoral Program: fellow PhD students, faculty members, and PhD graduates, you have all been very kind to me and my research and I thank you deeply for your support.

Special thanks go to Dr. Cristina Yáñez de Aldecoa for the immense and endless support, and to Dr. Betlem Sabrià Bernadó because once, during a casual conversation, she made a comment that made me realize I had been doing something wrong all the time. We never know enough, but we keep pushing!

To my friends who, at all times, supported my work on this Thesis: Jaume Ribolleda Bernad, and Oriol García Ruiz, thank you, mates.

Last but not the least, I would like to thank my family: my loving wife Nina, my two stupendous kids Aleix and Jordi, my parents Alejo and Marisun, and my brothers Albert and Marisun for supporting me unconditionally throughout unending nights, long weekends, summer or winter holidays, or whenever there was a moment to perform some kind of stupid test on no matter how ridiculous a theory might be. You are the world to me.

1 | Introduction

Traditionally, the *password* has been the most popular method of authenticating a user when accessing a protected resource. The reason behind this fact is quite simple: it is convenient, simple and cheap. There are other ways to protect resources like, for instance, using Short Messaging System (SMS) account verification, but these are not as practical as the login/password combination. Recent security related events that exposed sensible data to attackers, and the tendency to disregard the importance of a secure password, have shown that having just a simple password (many times written on a post-it and tagged to the wall or under the keyboard) may not be enough to guarantee access to protected resources and sensitive information.

Some popular Internet services like Apple's iCloud[1], Google's web services[2] or Microsoft Live accounts[3], to name a few of the *big* ones, have adopted the second factor option (also known as two-step verification). From the moment this option is enabled and configured, when users want to access their private and sensitive data they also have to provide some extra information to prove that they are who they claim to be. This extra information can either be (though not limited to) a telephone number, an answer to a previously stored question that only the user should know, or a code sent to the user with a limited time-frame validity. Similar to this approach is the concept of latching an account on mobile devices[4].

In the majority of these cases, users supply information based on what they know (when answering questions or replying SMS messages with a security code), or on something they have (a swipe card, a code card, or some other kind of *token*). These second factor options, though, are also far from perfect: there is always the possibility that questions are inadequate (close people to the user may know the answer or can try an obvious answer), or that tokens may be lost or stolen.

Another of the weaknesses of using only a login/password combination is the fact that it *only* grants access to a system but, afterwards, it does not say anything about who is really using it. The same can be said about tokens or some other second factor

[1] `https://support.apple.com/en-us/HT204152`. Last accessed: September 30, 2017
[2] `https://www.google.com/landing/2step/`. Last accessed: September 30, 2017
[3] `https://goo.gl/DS6tVB`. Last accessed: September 30, 2017
[4] `https://latch.elevenpaths.com`. Last accessed: September 30, 2017

options. The question is if there is a way to know who the user behind the actions performed after being authenticated and granted access is. The first idea that comes to mind, probably, is to use a system based on some kind of Close Circuit Television (CCTV) surveillance system, but it is simply unfeasible to implement this solution in large scale environments with numerous users and devices. A suitable and feasible option is to use biometric techniques.

This PhD Thesis focuses on this particular field related to Computer security and, more specifically, on Keystroke Dynamics. Trying to identify users that have been working on a computer system, even after having been authenticated is going to be one of the main goals of the study presented in this document. At the same time, the proposed methods will also be evaluated to determine if the possibility of, not only identifying users, but also authenticating them is feasible. To do so, a new way of organizing samples, based on contextual information, will be analyzed and evaluated.

1.1 Justification and research context

Biometrics refers to a physiological or a behavioral characteristic associated to a person. Classic examples are fingerprint, iris or palm scanning; the way a person types on a keyboard, walks, talks or writes, among many others. These techniques have been around for many years now but their adoption has been mainly restricted to environments with access to substantial financial resources and the need to secure access beyond the simple login/password scheme. The use of biometric techniques, though, is not limited to granting access or verifying user behavior on computer systems. It is also being widely used, for example, to grant access to restricted areas within the enterprise. Yet another example is the use of biometric features in a passport to identify travelers. More and more, in the user's everyday life, such features are being implemented to ease the use and access to technological resources, but even more, to sensitive information.

Biometrics, historically, have presented a problem: they tend to be rather expensive for the average end user [46, 63, 95]. Only now, fingerprint scanning is starting to become standard on high(er)-end personal computers, and is becoming a *de facto* standard in smart-phones and other hand-held devices to grant access to these devices. On recently released mobile devices, it is also possible to access the system using face recognition. This method of authenticating users takes advantage on the fact that these devices have an incorporated camera. On the other hand, cheating these devices has been proved to be as simple as showing the camera a still photo of a valid user [43].

Other methods like iris or hand geometry scanning, or thermal imaging are still science fiction in the realm of the traditional home or small office user. It should also be

stated, though, that usability is a main factor when implementing biometric techniques in end-user devices. Transparency, reliability and accuracy go hand in hand to ensure user adoption.

Keystroke Dynamics, the focus of this research, uses the natural rhythm that a user has when typing on a keyboard. It has been widely discussed that this rhythm tends to be unique for each person and that it can be a valid method of identifying, authenticating, constantly monitoring, or even classifying them. Compared to other biometric systems, Keystroke Dynamics is quite easy to implement, and most important of all, it is not expensive at all. All that is needed is an *off-the-shelf* keyboard and the possibility to determine the latency between successive keystrokes when the user is typing, something that any modern operating system will allow.

In Figure 1.1, Keystroke Dynamics is shown as part of the some different biometrics options that deal with Computer security. This figure shows only a *very* small part of the field of Computer security. It even only focuses on a small selection of the available biometric techniques but it should help position the technique discussed in this document within the topic of Computer access and security.

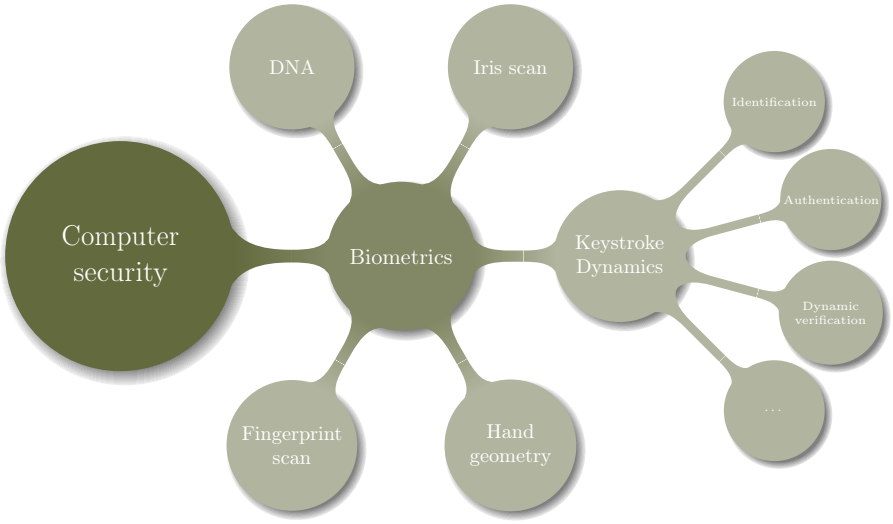

Figure 1.1: Keystroke Dynamics in the field of Computer Security

Since the beginning of the 20$^{\text{TH}}$ century, when Morse code was transmitted over the wire, some people said they were able to identify the other party by the way, or by the rhythm, the messages were transmitted, even before the other end had sent its proper identification [84]. More recently, in the late nineteen seventies, a new research path was taken to determine if Keystroke Dynamics could be a good enough method to classify users and thus, identify or authenticate them.

The main objective of the proposed research in this study is to determine if a new method of classifying samples based on contextual information is valid to identify, or even authenticate, users behind a computer system. The proposed methodology will

try to identify a user using the largest sequence of letters of a word and the latency associated with each keystroke. Other strategies will include finding the better suited building parameters to obtain quality models, evaluating which is the most accurate search criteria or determining behavioral features that yield the most discriminating features of the users being evaluated. These include, also, the age group and the gender a user belongs to. Previous research on this field has mainly focused on *n-graphs* frequency methodologies. Some research has been carried out on *wordgraphs* but it has not been the norm. Even less research has been performed on contextual information.

The samples for the present study have been collected over a period of three semesters from the use of the virtual campus at the University of Andorra. The possibility of using the proposed methods in this study in online learning environments will be evaluated to determine their feasibility. At the same time, if the models prove to be valid and useful, the applications could go well beyond the simple process of identifying the author of pieces of text sent to the online learning platform. Users could also be authenticated using this process when accessing private resources. The possibility of finding out if a user has been supplanted could be also a possibility. Users could be verified before performing an exam just by writing a short paragraph of text. Determining who wrote what on written assignments submitted by a user but authored by different ones could also be a valid practice. Tracking how users follow the course: who waits to write everything at the end, or who has a more regular pace throughout the semester, could also be evaluated. Even if the applications are easy to spot, ethical concerns could arise from the fact that students may feel like being spied upon. These concerns are discussed later in this document.

It is interesting to note that the proposed research is focused on free text. Even more interesting is the fact that the users that provided samples were not told *what*, *how* or even *when* to write. The samples were not tailored, modified or adapted in any way. At the same time, a minimum number of words per sample was never required. This includes the possibility that users may have used different languages, different devices (desktop computers in different environments, like the university labs or at their home, or even mobile devices), different times of day... The fact that this research uses samples from a non-controlled environment and that it tries to see if the models for each user are valid enough to identify or authenticate them on different situations, environments or emotional states should help understand its relevance.

1.2 Document structure

This PhD Thesis document is structured as follows:

- Chapter 2 summarizes the current State of the Art. It gives an overview of the

research that has been carried out on the field of Keystroke Dynamics since its inception in the late seventies. A good deal of effort has been put into describing the theory behind this biometric technique. Both the fixed text and free text methodologies are analyzed, as well as the leading methods to classify users that have been studied and the results these have given.

- Chapter 3 details the proposed Objectives and Hypotheses. These have been identified after carefully studying the State of the Art and after evaluating those topics that had not been previously studied in depth.

- Chapter 4 describes the Methodology used in this study. This chapter details not only the followed steps to develop the necessary tools to collect samples and analyze them, but also which groups of users were formed and why. It also details how the proposed tree models evolved from their inception, and the methods that have been used to test new user samples against these models and the strategies followed to identify the users who had authored them.

- Chapter 5 shows not only the results but also the procedures and experiments that led to obtaining them. These include the results from all the different tests that were performed with the available dataset as well as a number of tests performed using a popular *n-graph* methodology using Relative and Absolute distances. This has been done to have a comparable frame of work with the current State of the Art. The results should help decide if the proposed methods are valid to identify or authenticate users based on their typing behavior and other related contextual information.

- Chapter 6 shows the Conclusions based on the results from the previous chapter. These go hand in hand with the suggested Objectives and Hypotheses in Chapter 3.

- Chapter 7 outlines some future work ideas that could be implemented to bring the research some steps further.

- The Bibliography lists all the referenced publications in the text. These are sorted by author Surname and Name and not by their first apparition in the document.

- Finally, the Appendices contain additional material related to this PhD Thesis. These include contributions to congresses, the Python application help menu, the MySQL database schema for the persistent layer, and additional interesting references. The code, both in Python and R, developed for this study has not been included in this document due to its length, but it is available upon request.

2 | State of the Art

Before starting any kind of research, the current State of the Art on the selected subject has to be analyzed and, also, has to be fully understood. At the same time, a good knowledge of the background on the topic at hand is necessary. In all cases, the main idea is not to repeat what others have already done, and most importantly, to find those subjects or questions that have not been fully studied or addressed. The present chapter describes the State of the Art related to the field of biometrics and, more specifically, to the particular technique known as Keystroke Dynamics.

The first section focuses on the basics of biometrics, in general. The most important concepts are described as well as the most common techniques to identify individuals. Keystroke Dynamics gets then the spotlight and is described centering the efforts on both general and specific concepts based on the research that has been carried out in the last forty years. Following this section, other subjects related both to biometrics, but mostly centered on Keystroke Dynamics are also presented. These include: biometric evaluation, methodology, classification, fusion, and weighting techniques. To end this chapter, a summary of the most relevant publications and results in both free text and fixed text methodologies as well as in authentication, identification or continuous verification methods is presented.

2.1 Biometrics

This section details the principal features of Biometrics understood as the possibility of identifying an individual based on their distinguishing physiological or behavioral characteristics [66].

2.1.1 Introduction

Reports of the use of body measurements as a biometric technique date as far as the mid 19^{TH} century. Even though the use of biometric techniques has been highly related to law enforcement and criminal identification, nowadays these techniques are used, more and more, as a means to recognize users in common daily applications [70]. There are many human characteristics or traits that can be used as a biometric identifier.

These traits fall into one of these two categories [78]:

- Physiological traits: These are biological or chemical traits that are innate, or characteristics that a person has grown to. Examples of these are: the iris, the DNA, the hand palm, the ear, or the face geometry, among others.
- Behavioral traits: These characteristics are either trained or acquired over time. Examples of these could be: a person's signature, Keystroke Dynamics, that is the particular rhythm a user has when typing on a keyboard, the particularities of the voice, a user's handwriting, among others.

The described traits and their related techniques have also been commonly classified as Soft or Hard biometrics. Soft biometric traits are those characteristics or features, usually associated to behavioral traits, that provide some information about the individual, but lack the distinctiveness and permanence to sufficiently differentiate any two individuals [67]. On the other hand, Hard biometric traits, are considered better in terms of distinctiveness and permanence, like the fingerprints or the geometry of the face, and can give better results when differentiating individuals.

It has been discussed that any of these traits should satisfy, to a greater or lesser degree, the following requirements to be considered a valid biometric identifier [70]:

- Universality: How commonly a characteristic is found individually.
- Distinctiveness: Any two individuals should be distinct enough for a given characteristic.
- Permanence: The characteristic should be invariant through time.
- Collectability: The characteristic should be *easily* collected and measured.
- Performance: Any characteristic should be recognized fast and accurately.
- Acceptability: Determines how good users will accept the acquiring of an attribute.
- Circumvention: The system should not be easy to *trick*, *cheat*, or *spoof*.

2.1.2 Basic biometric steps

Any kind of biometric system has to go through the following steps [63] (these will be further explained and applied to the Keystroke Dynamics context later in this chapter):

- Capture: A physical or behavioral sample is captured by the system during the initial enrollment phase.

- Extraction: Unique features are extracted from the samples and a template for each user is created.
- Comparison: At a later stage, during either authentication, identification or verification processes, new samples are collected. These are then compared to the stored templates, using different possible methodologies, depending on the chosen identification process.
- Match/Non-Match Evaluation: Decide whether the features extracted from the new sample are a match or a non-match when compared to the selected templates.

These steps provide a common framework that all biometric systems tend to use. When focusing on particular techniques, or when deciding among the different possibilities these techniques will be applied to, some of the described steps may be adapted to reflect the particularities of the chosen applications. An example that relates to Keystroke Dynamics could be the process of capturing information. While on other biometric techniques this may be a once-in-a-lifetime process (the example of a DNA sample comes to mind), the capturing process when using Keystroke Dynamics may never stop, especially when the need to have up-to-date and accurate models is required.

2.1.3 Common biometric techniques

Some of the most commonly known biometric techniques are briefly described in this section. It should be noted that these are not the only available ones. New and current techniques are constantly researched and studied. There is a lot of interest by governments, military, security related corporations, among others, in finding reliable and cheap methods of accurately identifying individuals [68]. Below are some of the most common techniques and their main defining characteristics (adapted from [59, 61, 68, 120]):

- Fingerprint scanning: A fingerprint is the pattern of ridges and furrows on the surface of a fingertip. They are so distinct that even fingerprints of identical twins are different. This technique has been used for centuries and its validity has been well-established.
- Face recognition: Typically, this technique focuses on recognizing the global positioning and shape of the eyes, eyebrows, nose, lips, and chin of the face of an individual. Applications using identification based on face geometry range from the static, where users are still in front of non-variable backgrounds to dynamic, uncontrolled face identification with dynamic backgrounds.

- Retinal scan: The pattern formed by the veins beneath the retinal surface in an eye is stable and unique. Digital images of retinal patterns can be acquired by projecting a low-intensity beam of visual or infrared light into the eye and capturing an image of the retina using optics similar to a retinascope.

- Iris scan: The iris is the annular region of the eye bounded by the pupil and the sclera (white of the eye) on either side. The visual texture of the iris stabilizes during the first two years of life and its complex structure carries very distinctive information useful for identification of individuals.

- Hand geometry: This biometric technique focuses and the shape of the hand, including the length of the fingers and their respective width. The technique is very simple, relatively easy to use, and inexpensive. Unfortunately, the physical size of a hand geometry-based system is too big for applications in laptop computers. At the same time, the use of the shape of the hand as an authentication is totally viable, but using it to continuously verify a user may not be feasible.

- Signature recognition: Each person has a unique style of handwriting. However, no two signatures of a person are exactly identical. The identification accuracy of systems based on this highly behavioral biometric is reasonable but does not appear to be sufficiently high to lead to large-scale recognition. This is a typical example of a Soft biometric technique.

- DNA samples: Most of the DNA humans have is highly similar between different individuals, but there are portions that are different enough to be able to use it as a biometric technique. DNA does not change during a person's life or after their death. It has a double helix structure and it gives the most reliable result for offline personal identification excluding the case of identical twins. As opposed to the previous technique this one is usually described as a Hard biometric technique.

- Speech: The little variance in the individual characteristics of human speech is primarily due to relatively invariant shape/size of the appendages (vocal tracts, mouth, nasal cavities, lips) synthesizing the sound. Again, it has been argued that this technique may not be strong enough to use without another one complementing it.

Other techniques not that usually present in the user's daily life include, among others: the way users walk, the particular rhythm when typing on a keyboard, ear geometry, or handwriting.

Table 2.1 (adapted from [70]) shows a comparison of the previously described techniques, together with some other, also known, biometric methods. In this table,

the seven requirements, previously described, have been classified using the following three categorical values: *High*, *Medium* and *Low*.

Technique	Universality	Distinctiveness	Permanence	Collectability	Performance	Acceptability	Circumvention
DNA	H	H	H	L	H	L	L
Face geometry	H	L	M	H	L	H	L
Fingerprint	M	H	H	M	H	M	M
Hand geometry	M	M	M	H	M	M	M
Iris	H	H	H	M	H	L	L
Keystroke Dynamics	L	L	L	M	L	M	M
Palm print	M	H	H	M	H	M	M
Signature	L	L	L	H	L	H	H
Voice	M	L	L	M	L	H	H

H – High; M – Medium; L – Low

Table 2.1: Comparison of common biometric techniques

The scores assigned to each of these techniques allows the possibility of determining which would be better suited in different situations or applications. In the end, though, the most important feature, and the one that is most looked for, is accuracy. Of course, this may lead to the system being impossibly expensive to put into production. Usually, a compromise has to be taken between the needs and possibilities when choosing a particular biometric technique in order to achieve a win-win situation.

2.1.4 Multimodal biometric techniques

So far, all the presented techniques are considered to be unimodal in the sense that these are centered solely in a particular physiological or behavioral feature. Unimodal biometric techniques can yield excellent results for most users, especially when dealing with Hard biometrics, but may be inadequate for others (some users may lack the necessary trait to be analyzed or be in no condition to submit it). Also, and especially when using Soft biometrics, errors can be too high to differentiate certain users.

To overcome the limitations that using a sole biometric technique may present it is common to gather different features from users using different techniques and combine them. When only using Soft biometrics this could become mandatory if high accuracy is required. A system, for example, may require both a fingerprint scan and a voice recognition sample. While it may feasible to *spoof* them both, the moment the number

of samples from different techniques increases, the chance of cheating a multimodal scheme becomes less and less attainable.

The moment different readings from different traits are available, these can be combined using different techniques known as fusion. This combination of features not only expects to reduce problems in identifying users when only a single defining trait is used, but also to improve results by taking advantage of the inherent characteristics when each trait is taken into account. Many articles have focused on multimodal strategies [52, 103, 111, 120], and at the same time, the use of fusion techniques is common [13, 126]. An interesting project, led by the Open University of Catalonia[1], is TeSLA. It combines Face and voice recognition with Keystroke Dynamics techniques to provide an adaptive trust e-assessment system for online and blended environments[2].

Another requirement to choose a multimodal scheme over a unimodal one could be the price to implement them. Having, for example, three inexpensive Soft biometric techniques instead of a very expensive Hard biometric alternative can lead to having acceptable enough results that render the whole system usable without having to spend that extra money.

2.1.5 Keystroke Dynamics over other techniques

In the context where biometrics are applied in this study, that is, the identification and authentication of users in online learning environments it was thought that Keystroke Dynamics was the best suited option from the available alternatives. It perfectly suits the purpose in terms of transparency, usability and economic costs.

Students and teachers alike interact with the virtual campus using all kinds of devices, either at the university, the library, the cafeteria, at home... In order to have realistic samples users were not told or aware that timing intervals were being collected. This allowed for the study to be much more realistic.

Initially, having the users use hardware devices to improve biometric identification using a multimodal strategy was considered. Webcams to add a visual confirmation, the use of special keyboards with pressure sensors, the study of mouse movements, or even helmets to detect levels of stress, were ideas that were evaluated. The problem was that the use of such devices rendered the study non-realistic enough, even if some of the ideas could have been perfectly suitable to bring the study of Keystroke Dynamics and other related techniques a step further. The aim of the study, again, was not to test different techniques but to be able to identify users with minimum intrusion and ensuring transparency to the user. All these ideas were discarded right away as soon as they affected normal user behavior or if it meant that users had to act in unrealistic

[1]Universitat Oberta de Catalunya: `https://www.uoc.edu`. Last accessed: September 30, 2017
[2]TeSLA: `http://tesla-project.eu`. Last accessed: September 30, 2017

ways. That left the possibility of using mouse movements as a technique that could provide a good multimodal scheme. This was discarded because, again, it needed for the computer where the user was working, to have special software installed to capture such movements. The software that allowed the capturing of keystroke intervals was already available in all web browsers, be it in desktop computers or mobile devices, without the need of adding anything else.

It was found that implementing a Keystroke Dynamics scheme was cheap, straightforward and, in the end, more reliable than initially thought. To sum up, no multimodal scheme was used. Next section goes deeper into the main characteristics of the chosen biometric technique.

2.2 Keystroke Dynamics

Keystroke dynamics refers to the habitual patterns or rhythms an individual exhibits while typing on a keyboard input device. These rhythms and patterns of typing are idiosyncratic, in the same way as handwritings or signatures, due to their similar governing neurophysiological mechanisms [44, 140].

Keystroke Dynamics (also known as Keystroke Biometrics or Typing Dynamics) can also be defined as the detailed timing information that describes when each key was pressed (KeyDown (KD)) and when it was released (KeyUp (KU)) as a person is typing on a computer keyboard. This also includes dwell times (the time interval a key is pressed down), and flight times (the duration between keystrokes), typing speed, frequency of errors, use of modifier keys, use of numpad. . . .

The principal idea behind this biometric measurement is that every user has a particular way of typing and that, like any other behavioral biometric system, it allows the identification, authentication or classification of these users.

Given the rise in use and apparent ubiquity of mobile devices, some rather recent studies have also applied this technique to these devices, with logical keyboards, obtaining rather good results [29, 34, 35, 65, 72, 88].

In Table 2.1, Keystroke Dynamics has not been presented as one of the best in terms of Universality, Distinctiveness, Permanence or even Performance. On the other hand, though, it is fairly easy to collect and users tend to accept it better than other techniques. In any case, it seems to be clear that if one had to be chosen, based on the information on the table, Keystroke Dynamics would not be the first option.

Keystroke Dynamics has some advantages that users tend to appreciate, though. Keystroke Dynamics are non-intrusive and transparent. Users do not have to be afraid of exposing their eyes to a scanning machine or be afraid to touch a reader that may have been used by countless other users.

The numerical results shown in an article by Yampolskiy & Govindaraju [136], when compared to other techniques, rank Keystroke Dynamics only as an average technique. To make matters worse, these results have to be analyzed from a critical point of view. It should be taken into account that the experiments performed to achieve these results are *usually* highly tailored and controlled. The numbers tend to be even worse once this biometric technique is applied to real world and uncontrolled data, as is the case of the research presented in this study.

Whether applied to desktop computers, laptops, mobile devices or any other device with a keyboard, either physical or logical, there has been a lot of interest and research done in the field of Keystroke Dynamics mainly because the method is not expensive and it is fairly easy to implement. This technique, though, is not the easiest to deal with. Other traits like, for instance, hand or face geometry are easier to collect. Then again, it is easy to collect them once, but it becomes rather difficult if this capturing process has to be continuous. When using Keystroke Dynamics, an *off-the-shelf* keyboard and a computer system capable of logging all the pressed keys and their associated timings is all that is needed. This latest characteristic is important because some studies have suggested the possibility of using special keyboards that can also measure the pressure applied to every key so that classification can be more accurate [86, 100]. These keyboards have not been used in this research since they are not the most common keyboards available to common users.

Going a bit further into what can be inferred when looking at Table 2.1, Keystroke Dynamics has a medium Acceptability and Circumvention. One of the problems the collection of this trait has is the fact that users are reluctant to installing key loggers on their machines, especially when dealing with continuous verification. Users do not like being *spied upon* when they will be constantly typing passwords to access protected resources or submitting other sensible information. Users would be less reluctant if the key-logging software could be disabled, something that, when not used correctly, would completely defeat the purpose of this biometric technique [45].

Circumvention is graded as medium. Some studies have focused on debunking the methodology that most other studies use when identifying impostor users [106, 107]. These authors refer to this methodology as zero-effort attacks. This means that, when checking a sample against a model using, for example, a cross-validation approach, there is no effort at all in trying to mimic the stored model of the user being attacked or impersonated. These researchers believe that this is a non-realistic approach. Others have also applied timing attacks to obtain information of the stored templates and exploit it [105, 119]. On the other hand, though, *traditional* studies try to prove that Keystroke Dynamics is a *valid* biometric technique, not that it is *unbeatable*.

Keystroke Dynamics has another significant problem: Persistance. Users can

improve their typing over time, can get hurt, or be in different emotional states that render their use of a computer different [45, 79]. How to deal with this issue is not trivial. Some studies have focuses only on how to retrain the system over time so that stored templates remain usable [10, 48, 75, 81, 110]. Other biometric techniques do not present such problems. The Iris or the DNA of a person will not change over time, but then again, both the economic costs and resources needed to implement these biometric techniques are, usually, prohibitive [46].

2.2.1 Feature selection

When Keystroke Dynamics has been used as a biometric technique many different features have been extracted from a user's particular rhythm. The most common ones are discussed in the following section. The proposed study will try to also use other contextual information as an alternative feature.

Common features in Keystroke Dynamics

As can be seen in Figure 2.1, adapted from [14, 122], these are some of the most common extracted features when using Keystroke Dynamics:

- Dwell time: The latency between the pressing (P_i – KD_i) and releasing (R_i – KU_i) of a key. This is the most basic feature that can be extracted. In Figure 2.1 D_1, the time between P_1 and R_1 is and example of dwell time ($200ms$ in this case). Among others, this feature is also known as hold time or keystroke duration.

- Flight time: The period of silence between successive keystrokes. In Figure 2.1 F_1, the time between R_1 and P_2 is an example of flight time ($200ms$ in this case). It should be noted that this feature can also hold negative values. This happens when a key has not yet been released and a new one is pressed. See F_2 in the figure for an example. This feature is also known as latency time, or inter-key time.

- *n-graphs*: The delay between any n number of KD, KU or a combination of both events is known as an *n-graph*. Digraphs and trigraphs are particular examples of this feature. In Figure 2.1 the timing represented by Di_1 is an example of digraph and Tri_1 is an example of trigraph.

- wordgraphs: The distance from the first to the last KD, KU or a combination of events on a single word. This could be interpreted as a particular example of *n-graphs*.

Even though these are not the only combinations, other timings are usually just slight alterations or modifications of the ones here outlined. It should be noted that the definitions given for the different types of intervals are the ones that have been used in this study. Other researchers may give different meanings to these terms, specially for the Flight time feature, that may have different interpretations [116, 122].

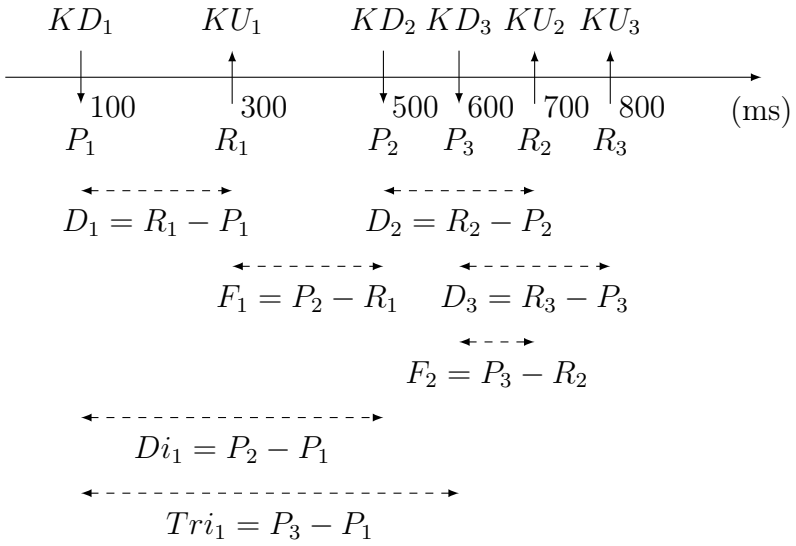

Figure 2.1: Frequently timing features used

The use of dwell times together with statistical methods like the mean, the median, and the standard deviation to determine the pattern of a user are good examples of the research that was done in the beginning [21, 47, 73, 131]. In this initial research, most of the efforts were centered on authenticating or identifying users using short pieces of information. Soon enough, though, the possibility of using longer texts was also considered [83].

Research then moved to seeing if grouping keystrokes into digraphs would provide more information and thus, yield a better user classification. This led to the question: are digraphs good enough? Why not trigraphs, *n-graphs*, or *wordgraphs* [55, 92]. In [117], this idea was also brought to discussion within the free text methodology. As soon as the number of considered events increased it was argued that these *n-graph* schemes were not solid enough.

Another feature that was studied was the language the user typed in [6, 56, 57]. It was found that building the model in a language and later verifying a user typing in a different language had little relevant effect in the decision process. These studies used languages that shared the same alphabets to prove their hypotheses. To this date, and to the best of our knowledge, no study has been performed on different alphabets like for example: roman or latin vs. cyrillic.

Other researchers have also tried partitioning the keyboard into different zones

and study which hand was used at any particular moment. It was claimed that these features also helped classify the users [87, 109, 133].

One fact that can be thought as surprising, or peculiar to say the least, is that most studies only use correct forms of input. This means that errors (the use of the *backspace* key, for instance) are usually discarded. Also discarded is the use of special keys like SHIFT or CONTROL. Researchers say that these offer little added information and that can bias the results. This is not always the case though even if studies focused on modifier keys are only a few [15, 33]. In this study, the patterns users create when they make mistakes will also be evaluated. These mistakes are considered contextual information related to user behavior and will be part of the studied behavioral features. The proposed research will try to determine if a user will type a word incorrectly most of the time, and at the same time, if this particular user will always follow the same steps to correct it.

Contextual information as a feature

The Merriam-Webster[3] dictionary provides the following definition for the word *context*:

> ***context***: noun – con·text – \'kän-tekst\
>
> 1. the parts of a discourse that surround a word or passage and can throw light on its meaning.
> 2. the interrelated conditions in which something exists or occurs.

For the present study, the best intended meaning is the second one. Contextual features, when applied to the present study, can range from characteristics directly related to Keystroke Dynamics, like for instance, the position of the letters in a particular word, or be somewhat related to user behavior, like for example, the frequency a word is typed, or the age and gender of the user.

For the first case, this study uses contextual information of the written words to identify the users as opposed to other well-known techniques like *n-graphs* frequency schemes. Most studies in this area have used some sort of data structure of two, three or more graphs to classify the samples. Usually, these samples are organized without taking into account their position in the original typed word. It will be discussed if the natural rhythm of a particular user is the same when they type, for example: IS, IRIS, THESIS or DISAPPEAR. When working with digraphs, the combination of letters IS would be grouped in a common data structure without considering if it had appeared at the beginning, the middle, or the end of the word. This idea of contextual information

[3]Merriam-Webster dictionary: `https://www.merriam-webster.com`, Last accessed: September 30, 2017

in free text environments has not been thoroughly studied before even though it has been hinted as a possible line of work [24, 118].

2.2.2 Fixed text vs. Free text

There are two main fields of research when dealing with Keystroke Dynamics. On the one hand, there are the studies focused on fixed, and usually short, text while, on the other hand, there are those studies focus on free text (some studies may have explored both, but these are a minority). Each field has its own applications and, of course, advantages and disadvantages.

These are the main characteristics of each of these methodologies:

- Fixed text: With this methodology, the user is asked to type a predefined text a number of times. This text is always the same (like maybe a *password*, a given string like *Name.Surname.Login.Password* or any other fixed string). This input, when using Statistical methods or Machine learning techniques will be used to build a model or to train a system labeling which samples belong to the user and which do not. Later, when users have to be identified they will enter the *same* text again and this new sample will be compared to the previously stored template or fed to the Machine learning algorithm.

- Free text: In this case, users type either *long* portions of text that simulate the idea of free text or, in other cases, type whatever they want whenever they want, without restrictions. Using this input, it is the job of the chosen algorithm to extract relevant features and build a model for each user. When, later, users have to be identified they can enter the same or a completely different text. It should be a matter of their choice. The chosen algorithm should determine if these new samples are valid or not, that is, if these belong to the user claiming authorship.

As previously pointed, these two approaches have advantages and disadvantages depending on the scheme where they are applied. For example, if users are being authenticated using Keystroke Dynamics and a fixed text approach, user access policies can be applied *only* during the authentication phase (an example can be seen at Coursera[4]). This means that the user *may* have authenticated correctly, but nothing can be said of the user that is really using the computer once this authentication phase is completed. On the other hand, the job of capturing Keystroke Dynamics information *only* during this initial phase is much simpler than when using a free text approach. Free text can be used to continuously monitor the user using the system and apply different policies if a change in the Keystroke Dynamics pattern is detected. This

[4]Coursera: `https://www.coursera.org`. Last accessed: September 30, 2017

method has the disadvantage that the user has to be monitored all the time and this may consume system resources and, at the same time, invade the privacy of the user when performing certain tasks involving personal or confidential information.

2.2.3 System vs. Application data recollection

Invading the privacy of the users is something that should not be taken lightly [70]. Keystroke Dynamics is a non-intrusive and transparent technique to the user but this does not mean that it is exempt of ethical and privacy issues. When performing authentication processes, the capture phase is limited to the login screen. At this point, when using Keystroke Dynamics, users *know* they are being monitored. Once the authentication phase is over, capturing also stops. Users do not have to be conscious about the fact that their actions may be analyzed locally or, what could be even worse, sent to a remote server for evaluation.

When continuously monitoring a user working on a computer, in verification or identification schemes, the capturing phase of the natural rhythm of a user never stops. This means that all, or most, keystrokes are evaluated and it would be very easy for a user to stop thinking about this. In fact, this should be a premise to allow users to show their real natural rhythm. Once the user has stopped thinking about a process continuously monitoring their actions, they can author documents, browse the web, access bank accounts, access shared resources, write potentially sensible information..., without being aware that *someone* is watching. While the user is performing these tasks the Keystroke Dynamics module is recording or analyzing the information.

The question of what should be captured becomes then a debate about the possibility of switching off the capturing module or, on the other hand, only capture those keystroke timing events on certain applications or environments. In his PhD Thesis, Marsters proposed a solution where the information gathered was stored in a matrix structure without an ordered log, improving the privacy of the captured data [90]. This methodology though, may not always be possible to implement. Should the capture process be limited only to one application, or only during certain periods of time, or should this be determined by the user?

When trying to evaluate who the author of a document is, it seems normal to think that the application used to author the document *should* be monitored but, what about the rest of the system? Does the user perform in the same way when typing on an editor than when surfing the web? Is this relevant to the template? All in all, it seems obvious that these questions should be answered and studied before implementing a Keystroke Dynamics monitoring system.

2.2.4 Authentication, Verification and Identification

As hinted on previous sections, there are important differences between the methods known as Authentication, Verification and Identification. Each of these methods has a field of research associated to it, even if these may sometimes overlap.

- Authentication: A template of the typing rhythm of a user is created. Successive samples are compared to the stored template. This method can be performed using both fixed text and free text approaches. This line of investigation focuses, as the name implies, on the job of authenticating users. This process is usually performed at the beginning of a session and, from then on, the user is considered to be valid or authenticated [122]. This technique is also known as static authentication, or dynamic authentication.

- Verification: The technique verifies that the user does not change during the whole time it is logged on a computer system, or while using the monitored application. When continuously verifying a user, there may be not a previous template [24]. This means that, while users are submitting data, the template is built in real time as soon as they start typing. If an anomalous behavior is detected (there has been a good deal of research to studying the possibility of detecting these changes [2, 24]), the system has to act accordingly applying a policy previously set by an administrator. In general, this method involves only free text approaches. This technique is also known as continuous verification, dynamic verification, or reauthentication [122].

- Identification: This third possibility involves having a number of previously built templates from a number of users. Either free text or fixed could have been used to build these templates. New samples are compared to the chosen templates and which user authored the new sample is determined. This technique has been applied using different methodologies (see [55]): a user could be identified to the closest template (using distance measurements), or remain unidentified until a minimum distance threshold is achieved.

2.2.5 Gender recognition

Gender recognition using biometric traits is a field that has also been studied using Keystroke Dynamics. Even if it has been merely testimonial, some studies have also tried to determine the gender of users based on this biometric technique [7, 53]. Another study tried to find the gender by analyzing behavioral patterns when surfing the web [80].

These studies have used Machine learning techniques as well as statistical methods. The idea of separating users by their natural typing rhythm renders a two-class problem that can be, thus, analyzed using a Support Vector Machine (SVM), for example. The results of these initial studies show promising results with an accuracy around 90%.

The authors of these studies suggest that knowing the gender of users could be of interest. A typical example could focus on advertising campaigns. Even if the study of the present research does not focus on gender recognition *per se*, it is argued that knowing the age and the gender of users could be useful to building better models. This is considered part of the contextual features analyzed in this study. Separating users by age group and gender and analyzing the accuracy of both identification and authentication is the goal of one of the experiments performed in Chapter 5.

2.3 Biometric evaluation

When it comes to evaluating the effectiveness and accuracy of biometric systems different measures have been used throughout the literature. The methodological approach used (fixed text or free text and authentication, verification, or identification) *usually* determines the way results are presented.

2.3.1 Accuracy

When dealing with identification schemes, and when a threshold may not be always an available parameter, the accuracy of the system has been measured using the percentage of effectiveness. This value is obtained as the proportion of correctly identified elements compared to the totality of elements.

For example, if m is the number of correctly identified messages from a total of M messages, the accuracy A of the system would be represented as $A = {}^{m}/_{M} \cdot 100$. The same methodology can be applied to the number of correctly identified users, for example.

2.3.2 FAR and FRR

In most studies regarding biometrics the False Acceptance Rate (FAR) and False Rejection Rate (FRR) rates have been used extensively [22]. In particular, when dealing with authentication these rates tend to be the most used. Below is the formal definition for each of these terms:

- FAR: Measures the percentage of impostors that are allowed to access the system.

It would be desirable for this value to be always as low as possible.

$$FAR = \frac{Number\ of\ false\ matches}{Total\ number\ of\ impostor\ match\ attempts}$$

- FRR: Measures the percentage of legitimate users that have not been given access to the system. A lower value is always desired so legitimate users are always granted access.

$$FRR = \frac{Number\ of\ false\ rejections}{Total\ number\ of\ genuine\ match\ attempts}$$

The ideal values, in both cases, would be zero but this situation does not happen often. In a perfect world, no impostors would be allowed into the system and all legitimate users would be granted access. It has been observed, though, that reducing one of the rates, by means of hardening or softening a threshold, tends to increase the other. The objective then becomes keeping these rates as low as possible (see Figure 2.2).

The European standard for access-control systems (EN-50133-1) specifies that a FRR of less than 1%, with a FAR of no more than 0.001% [31] should be achieved to consider a biometric technique production ready.

According to [22, 135] these rates should not be confused with False Match Rate (FMR) and False Non-Match Rate (FNMR). FAR and FRR refer to a Biometric Application and are the more conventional pattern recognition terminology while FMR and FNMR refer to a Core Biometric Matcher.

It is also common to see the use of other acronyms like Failure to Acquire (FTA), Failure to Enroll (FTE). The FTA rate is the percentage of the target population that does not possess a particular biometric. In general this would mean that the user does not possess the biometric that is needed for enrollment [22]. They may be missing a finger or an eye, for example. In Keystroke Dynamics, this could refer to the fact that the user does not possess a particular rhythm when typing on a keyboard. Non-proficient users like kids, who are not yet used to type on keyboards could be a good example. The FTE rate, on the other hand, refers to the percentage of the population that somehow cannot be enrolled because of limitations of the technology or procedural problems [22].

2.3.3 Equal Error Rate

The Equal Error Rate (EER), also known as Common Error Rate, has also been used to present the results of biometric studies. It is determined as the value where the FAR and FRR values are equal. The lower the EER value the better the classification

method is considered.

An example of EER is depicted in Figure 2.2 (adapted from [46]). The horizontal dotted line depicts the Equal Error Rate (approximately at 10%). The threshold value (x axis) determines which samples will be accepted as valid. If the threshold value is *tight* there will be almost no False Acceptances but the False Rejection value will be unacceptably high (something that tends to annoy the users because they are asked to submit the samples again). On the other hand, if the threshold value is too *loose* all valid users will be correctly given access but, at the same time, the FAR will be unacceptably high. It has been argued that having valid users rejected (even if angry) is better than having false users accepted [46].

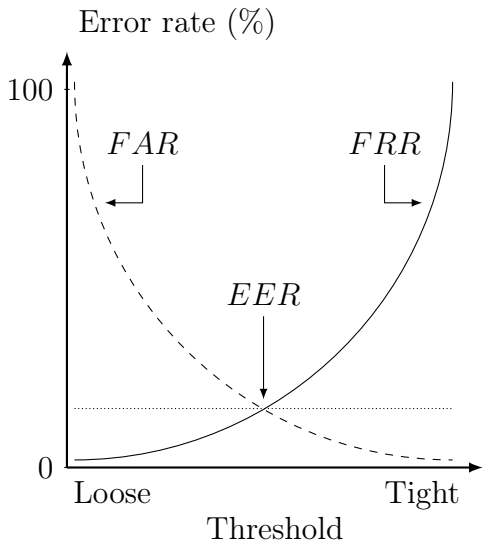

Figure 2.2: Equal error rate

2.3.4 Receiver Operating Characteristic curves

Yet another method to present the results has had much popularity recently in the field of Keystroke Dynamics. The use of Receiver Operating Characteristic (ROC) curves and the associated Area Under the Curve (AUC) value seem to have become sort of standard when measuring how good a classifier is (see Figure 2.3).

This technique has also been widely used in Data Mining methods and studies. As in the EER figure, this plot also responds to the variation of the threshold value. The curve is created by plotting the true positive rate against the false positive rate at various threshold settings. It would be desirable to have a AUC value as close as possible to 1.0. The best possible prediction method would yield a point in the upper left corner or coordinate (0.0, 1.0) of the ROC space, representing no false negatives and no false positives. The (0.0, 1.0) point is also called a perfect classification.

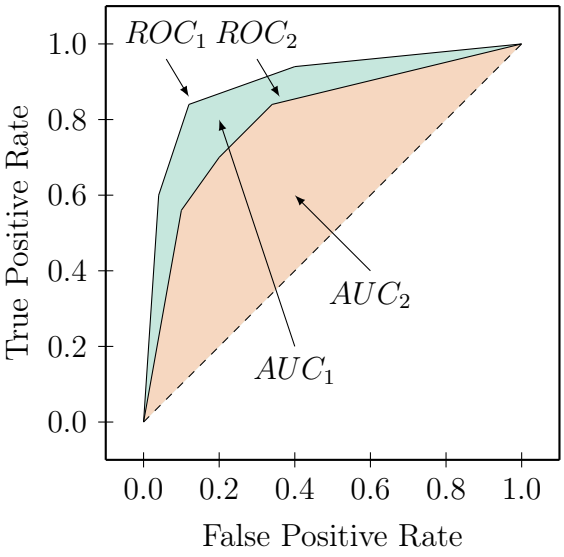

Figure 2.3: ROC and AUC example

2.4 Methodology applied to Keystroke Dynamics

At the beginning of this chapter the typical methodology when dealing with biometrics has already been hinted at. Capture, Extraction, Comparison and Evaluation, and in this particular order, are the common steps in biometrics. In-depth description of each of these steps is given below. This time though, the focus is given to how these steps are carried out when using Keystroke Dynamics:

- Information recollection: The capture process is considered to be, in general, the first process and an essential one. As the name implies, it consists in collecting user biometric samples. In the case of Keystroke Dynamics this is referred to the timing of the keystrokes on the keyboard produced by the user. It is also known as the data acquisition step.

- Extraction of relevant data and training: Once the data has been obtained it is time to choose which features will be used. This can range from using only keystroke timing features to all kinds of combinations to try to be as accurate as possible. This decision may have a serious impact on performance and training time when dealing with Machine learning techniques. In the end, the number of features can be excessive and some of these may only provide little added information. Some studies have focused on applying a Principal Components Analysis (PCA) to choose the most relevant features [134]. During the training phase, the models built from these samples and features will be stored in a form of persistent layer for later use.

- Classification: When it comes to comparing samples with the ones stored in the

previously built template, different techniques can be applied. These may range from simple Statistical methods to Distance measurements, Machine learning techniques or any other form of classifying elements.

- Evaluation: In this step, an action should take place based on the classification result from the previous step. If a login process is being performed, then the action could be to grant access or not. On the other hand, when dealing with verification or identification, an action regarding the validity of the user or the sample should take place. For instance, if there have been a number of invalid classifications, users could be forced to log out, or asked for a confirmation of their identity.

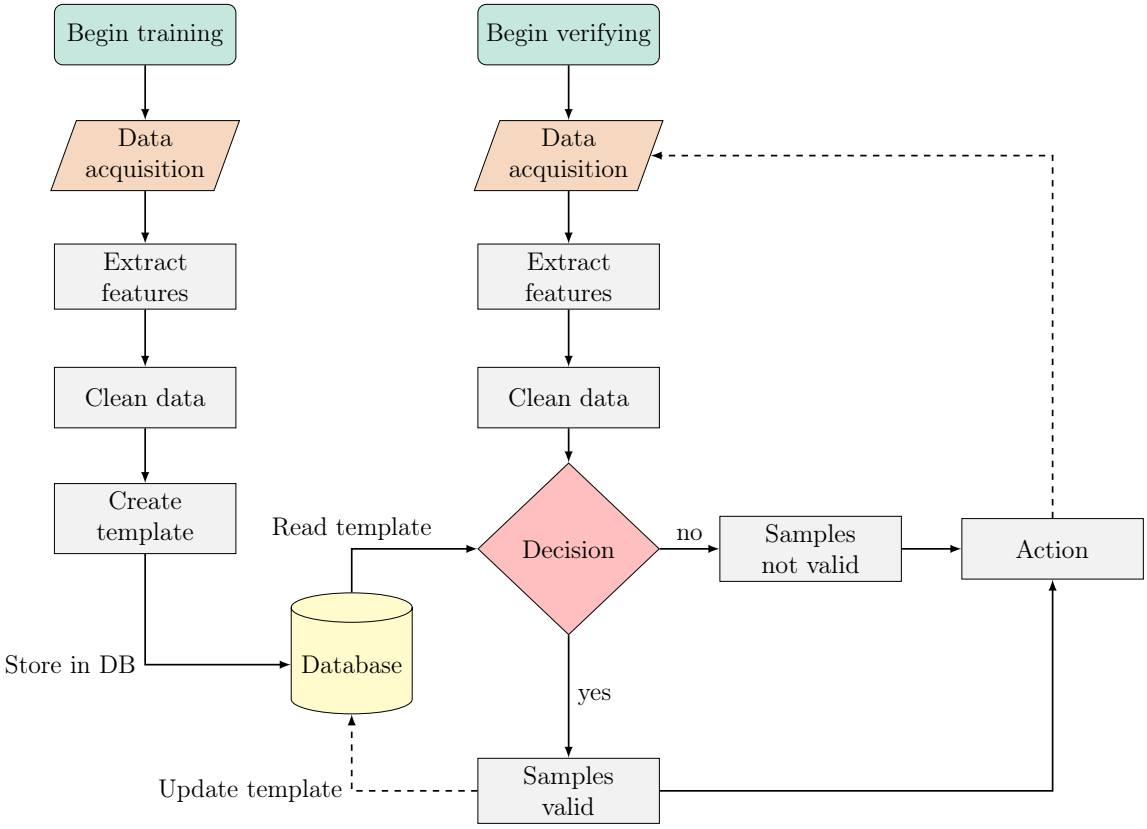

Figure 2.4: Typical biometric methodology

Figure 2.4 depicts the described steps. Two main phases or itineraries are shown:

- Training: This first phase, in the majority of cases (some methods may try to verify a user without a previous template), consists in training the classifier or building a statistical template of the gathered samples. In this step, samples are collected, cleaned (normalized and/or treated for *outliers*) and a template for the user is created and stored in some sort of persistent layer.

- Verification: This second phase verifies new collected samples against the previously stored model. The new samples are usually treated in the same way. The same features are extracted and these are also cleaned before being compared to the template.

As can also be seen in the figure there is an *optional* step (labeled as Update template) that deals with the adaptability and retraining of a user's template. What happens if a user has suffered an injury and cannot write like the day the template was created? Or what if the user is in a different mood [45]? This step tries to solve, or mitigate, this problem. Some studies have focused mainly in this area [10, 48, 75, 81, 110]. Retraining or adapting a user profile consists in incorporating the latest verified samples into the model to make sure that successive samples (if these have perceptibly varied over time from the initial model) will also be accepted as valid for the current user. There are basically two methods of doing so [3, 48]:

- Using the growing window method, the number of patterns is not fixed and it increases when new samples from the user are verified. As soon as the number of samples grows it may cause performance problems.
- Using the sliding window method, old patterns are discarded when new ones are incorporated into the model to adapt to the *new* or *adapted* rhythm of a particular user.

2.5 Classification techniques

Over the years there have been several different approaches on how to classify the rhythm a user has when typing on a keyboard. These include, among many others, Statistical, Distance measurements or Machine learning techniques. Some of these methods, mainly in the Machine learning area *can* have a feasibility problem in a way that they may need a lot of time and computer resources to be trained when models are big, something that it may not be always available in certain environments.

When following the steps described in the previous section, many researchers have worked in highly controlled environments. This means that all users used the same equipment and typed the same short or long text again and again. This may seem pretty far from reality but having these controlled environments allowed the researchers to determine what were the true factors that determine the results leaving out other factors that could bias them. Some studies, on the contrary, have focused on applying already well-known techniques to free text and real-life situations [71]. Having, again, a perfectly real and uncontrolled environment has been one of the main goals, and also one of the strengths, of the research presented in this document.

2.5.1 Statistical

The first written articles on the matter of Keystroke Dynamics used statistical methods [21, 47, 73, 83, 131]. The use of the mean, the median and the standard deviation was enough to provide excellent results (see Table 2.3). To date, these techniques are still widely discussed, improved and implemented. The present study focuses mainly on using these methods and statistical measurements.

Related to statistics is also the use of probabilities to classify keystrokes. The use of *t-tests* [8] or *chi-squared* [99] methods have also been explored. Clustering methods, like *k-means* and *fuzzy c-means*, have also been used with promising results [64, 89, 104, 130]. These, of course, are not the only techniques that have been used in the field of statistics. Many other approaches, to many to list them here, have been attempted with different results. Some of the studies centered on providing a survey on Keystroke Dynamics show more examples of the use of these other techniques [4, 122].

2.5.2 Distance measurements

Especially when dealing with free text, the use of distance measurements tends to be the preferred technique (see Table 2.2). This does not mean that these techniques have not also been used in static authentication studies, on the contrary. It seems, though, that nowadays when fewer samples are available and these are *known*, meaning that these do not come from a free text environment, researchers tend to favor Machine learning techniques.

Distance measurements determine how *far* a sample is from a previously stored one. Once a new sample is to be verified against the model, a vector is built and the distance between the two is calculated. The distance between a stored sample and a new one from the same user should be close to zero or below a given threshold. If compared to other models then the value should be the minimum to determine if the sample is valid.

The study presented in this document uses distance measurements to determine how far a given sample is from a previously built model.

Common distance measurements

Some common distance measurements that have been proved to be highly efficient in the literature are described below. In all these examples $\vec{X}$ and $\vec{Y}$ are sample vectors in the form of $\vec{X} = (x_1, x_2, x_3)$ and $\vec{Y} = (y_1, y_2, y_3)$:

- Euclidean: $D_E(\vec{X}, \vec{Y}) = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2}$

- Manhattan or city block: $D_M(\vec{X}, \vec{Y}) = \sum_{i=1}^{n} |X_i - Y_i|$
- Canberra: $D_C(\vec{X}, \vec{Y}) = \sum_{i=1}^{n} \frac{|X_i - Y_i|}{|X_i| + |Y_i|}$
- Chebyshev: $D_{CH}(\vec{X}, \vec{Y}) = max_{i=1}^{n} |X_i - Y_i|$

Relative and Absolute distances

Figure 2.5 shows an example of a Relative distance measurement (or R measure). This technique was used in an excellent paper by Gunetti and Picardi [55], even if its first appearance was in a paper by Bergadano et al. [18]. This method has had much popularity in the literature, especially when dealing with free text. The example has been adapted from the cited publication.

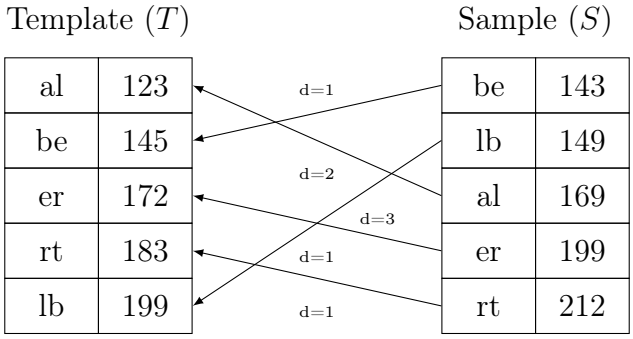

Figure 2.5: Example of a Relative distance measurement

To obtain the R measure, a series of *n-graphs* from a stored template and from a newly collected sample are used. These samples are sorted from high to low using the value in milliseconds (ms) of the Press–Press interval. The distance between two samples is then obtained measuring the relative positions between equal elements. It is worth noting that the specific values in ms of each keystroke are not relevant, only the relative position of each keystroke. In this example, the R distance would be:

$$R_2(S, T) = 1 + 2 + 3 + 1 + 1 = 8$$

When using this technique to identify users comparing the origin sample to many templates, the user with the lowest R value would be determined as the owner of the sample. The Relative distance presents a problem, though, as described in the paper by Gunetti and Picardi. In the case, for example, where the typing speed of each *n-graph* in a sample is exactly twice the typing speed of the same *n-graph* in a template sample, the Relative distance would be zero. This means that the Relative distance fails to discriminate between the typing samples of two users that have very similar typing rhythms, even if one of them is much faster than the other [55].

The Absolute distance (or A measure) is a similar measurement but it focuses on similar *n-graphs* instead. The Absolute distance only considers the absolute value of the typing speed of each pair of identical *n-graphs* in the two samples under comparison. Two *n-graphs* are similar if $1 < max(d1, d2)/min(d1, d2) \leq t$. In this formula, t is a value that the researchers determined empirically and set at 1.25, but it could be adapted to any value to get finer results. The Absolute distance for *n-graphs* is then defined as:

$$A_n^t(S_1, S_2) = 1 - \frac{number\ of\ similar\ n\text{-}graphs\ between\ S_1\ and\ S_2}{total\ number\ of\ n\text{-}graphs\ shared\ by\ S_1\ and\ S_2}$$

Using the previous example used to show how Relative distances worked, in this case the visual representation would be as depicted in Figure 2.6.

Template (T)		Sample (S)
123	al	169
145	be*	143
172	er*	199
183	rt*	212
199	lb	149

Figure 2.6: Example of an Absolute distance measurement

For each of the pair values from each *n-graph* and for $t = 1.25$, similar graphs would be determined by: $169/123 = 1.37$, $145/143 = 1.01$, $199/172 = 1.15$, $212/183 = 1.15$, $199/149 = 1.33$. The values below $t = 1.25$ are considered similar graphs (in Figure 2.6 these are marked with an asterisk), thus:

$$A_2^{1.25}(S, T) = 1 - \frac{3}{5} = 0.4$$

Further possibilities that included fusion or the combination of different *n-graphs* in the same measure were proposed in [55]. In their results, the use of such combinations gave even better results. The formula to obtain such combinations is pretty straightforward, and for each of the measurements it would be determined like this (n, m and p would be different graph lengths and N, M, and P would be the number of shared graphs for each of these lengths):

$$R_{n,m,p}(S_1, S_2) = R_n(S_1, S_2) + R_m(S_1, S_2) \cdot \frac{M}{N} + R_p(S_1, S_2) \cdot \frac{P}{N}$$
$$A^t_{n,m}(S_1, S_2) = A^t_n(S_1, S_2) + A^t_m(S_1, S_2) \cdot \frac{M}{N}$$

Finally, these measurements could be combined by simply adding them, as the original paper suggested: $R_{2,3} + A_2$.

Other distance measurements

Other distance measurements are also available (see below for some additional examples), and even the ones presented in this document can be further modified to include other values like, for instance, the standard deviation. This is the case, for instance, of the Scaled Manhattan distance. Another of the goals of the present research has been to find a good distance measurement to help identify the users and evaluate if there are significant differences when choosing one measurement over another.

- Czekanowski: $D_{CZ}(\vec{X}, \vec{Y}) = \frac{\sum_{i=1}^{n} |X_i - Y_i|}{\sum_{i=1}^{n} (X_i + Y_i)}$
- Gower: $D_G(\vec{X}, \vec{Y}) = \frac{1}{n} \sum_{i=1}^{n} |X_i - Y_i|$
- Intersection: $D_I(\vec{X}, \vec{Y}) = \frac{1}{2} \sum_{i=1}^{n} |X_i - Y_i|$
- Kulczynski: $D_{CK}(\vec{X}, \vec{Y}) = \frac{\sum_{i=1}^{n} |X_i - Y_i|}{\sum_{i=1}^{n} min(X_i, Y_i)}$
- Kulczynskis: $D_{CKS}(\vec{X}, \vec{Y}) = \frac{\sum_{i=1}^{n} min(X_i, Y_i)}{\sum_{i=1}^{n} |X_i - Y_i|}$
- Lorentzian: $D_L(\vec{X}, \vec{Y}) = \sum_{i=1}^{n} ln\,(1 + |X_i - Y_i|)$
- Minkowski: $D_{MK}(\vec{X}, \vec{Y}) = \sqrt[p]{\sum_{i=1}^{n} (X_i - Y_i)^p}$
- Scaled Manhattan: $D_{SM}(\vec{X}, \vec{Y}) = \sum_{i=1}^{n} \frac{|X_i - Y_i|}{\alpha_i}$
- Mahalanobis: $D_M(\vec{X}, \vec{Y}) = \sqrt{\sum_{i=1}^{n} \frac{(X_i - Y_i)^2}{\alpha_i}}$

These, and some additional ones, were evaluated in a study that used fixed text, and both an authentication and an identification scheme to compare up to nineteen different distance measurements [112]. To this date, no study has tried all this distance measurements on free text environments. This could be considered as future work.

2.5.3 Machine learning

Machine learning techniques have also been the focus of many studies in the Keystroke Dynamics area but mostly when using the fixed text methodology. A recent study that did a survey upon the research carried out using a free text methodology showed little use of the techniques described in this section [4].

Neural Networks and Support Vector Machines have achieved great results but tend to be more difficult to implement, maintain and, most of important of all, train than statistical methods. Other approaches that have been studied, among many others, include Decision Trees [8, 58, 139], Fuzzy Logic methods [39, 64], Genetic algorithms and Particle Swarm Optimization [76, 138] or Ant Colony Optimization [77].

Almost all these techniques require training before they can be used. A Machine learning algorithm can be trained using what is known as Supervised training or, on the other hand, it can learn as new samples are available using the Unsupervised learning alternative [19].

- Supervised learning: This method implies that there are a number of samples from which the outcome of the algorithm is known. These training samples are known as labeled data and for any input given to the system, the expected output is also known. The system *learns* from these samples and adapts its behavior to be consequent with the inputs and the expected results. When a new unknown sample is to be evaluated it uses what it has previously learned to give an answer. In general, the larger number of samples that are used to train a classifier the better it can later perform. Common examples of supervised learning algorithms are: Neural networks, Support Vector Machines, Decision Trees or the k-nearest neighbor algorithm.

- Unsupervised learning: This method, on the contrary, does not know any valid samples from which to learn, all data is unlabeled. It learns from the samples as they are feed to the algorithm. An example of unsupervised learning can be the Bayesian classifiers. The outcome can be ambiguous if the initial samples are misleading. Clustering algorithms can be another good example of unsupervised learning.

No further description of Machine learning methods is given due to the fact that in the free text methodology these are seldom used and, in the particular research performed in this study, only Decision Trees were briefly evaluated and soon discarded due to the poor performance in the given scenario.

2.6 Other techniques

The two following techniques are not exclusively related to Keystroke Dynamics, or even biometrics, but have also been used extensively in many studies in this area.

2.6.1 Fusion

When using fusion, as previously discussed in Section 2.1.4, the results of different methods can be combined to achieve a better overall result. For instance, if a sample is evaluated against a trained Neural Network and also against a trained SVM, the results of these classifications could be combined to determine a new value to accept or reject the new sample, for example, by using a voting method. There are many other different possibilities, though. Some of these have been studied in relation to Keystroke Dynamics in [123, 124, 126].

Below are some examples of such fusion techniques:

- Sum rule: $S_f = \frac{s_1+s_2}{2}$
- Weighted sum rule: $S_f = w_1 s_1 + w_2 s_2$
- Product rule: $S_f = \frac{s_1 \cdot s_2}{2}$
- Max (or min) rule: $S_f = max\,(s_1, s_2) \mid S_f = min\,(s_1, s_2)$
- OR Voting rule: $valid = \begin{cases} 0 & S_1 < thr, S_2 < thr \\ 1 & otherwise \end{cases}$
- AND Voting rule: $valid = \begin{cases} 1 & S_1 > thr, S_2 > thr \\ 0 & otherwise \end{cases}$

The main idea when using fusion methods is that the combination of results obtained using different methods can improve the overall classification by favoring the best results of each classifier. A classifier could outperform the others when evaluating certain features but be a poor one in other situations. The combination tries to make them all better as a whole.

2.6.2 Weighting features

Assigning weights to different features is also something quite normal across the literature [12, 69]. A weight can be applied in different stages across the biometric evaluation. Different features from the way users type may have different importance

when it comes to evaluating if new samples belong to these users. Which features have more importance is something that can be determined empirically or throughout a, for example, *leave-one-out* procedure. Others may apply weighting techniques to fusion techniques as it has been previously shown in the Weighted sum rule [52, 127].

An example of weighted features may help understand how it is normally used. When evaluating a vector of features, these may contain information about digraphs and trigraphs. A researcher could choose to give more importance to digraphs because these appear with greater frequency. This could be expressed like: $R = w_1 \cdot di + w_2 \cdot tri; w_1 > w_2$. Again, from these digraphs, maybe those that contain two consonants could be even given more importance. The same could happen with words: if a user types a given word more frequently then it could be given more weight.

In this research, one of the methods studied to determine the rightful owner of a session uses weights to give more importance to the distances that are closer to zero. Another example where this methodology has been used in the present work is studying if the frequency of words is a valid feature, giving higher weights to those words users type more often.

2.7 Bibliography analysis

For the present study, close to 300 references focused on Keystroke Dynamics have been analyzed. Other material related to Biometrics, Machine learning theory, Classification methods, Implementations... has also been consulted and referenced in Appendix G. This section focuses on publications that deal with Keystroke Dynamics directly. Figure 2.7 shows relevant information about the distribution of these publications over the years.

It is interesting to see that the majority of studies work with the fixed text methodology. A lot of effort has been put into having reliable methods to authenticate or identify users using short pieces or bursts of information. The right-hand side of Figure 2.7 shows that free text studies had their moment during the second half of the first decades of the two thousands but these were always a minority. Nowadays both kinds of studies seem to be a bit in decline. Keystroke Dynamics had most of its research performed from the nineties onward with a peak from 2000 to 2010. Every year, though, a good number of publications on the matter at hand are still published.

The number of publications that focus on mobile devices has seen an increasing interest since the year 2000. If studies on the matter of Keystroke Dynamics are still relevant it is mostly due to the interest on hand-held devices, how user interact with them, and the possibility of using multimodal approaches using the sensors onboard [54]. On the other hand, it seems that free text studies based on mobile devices

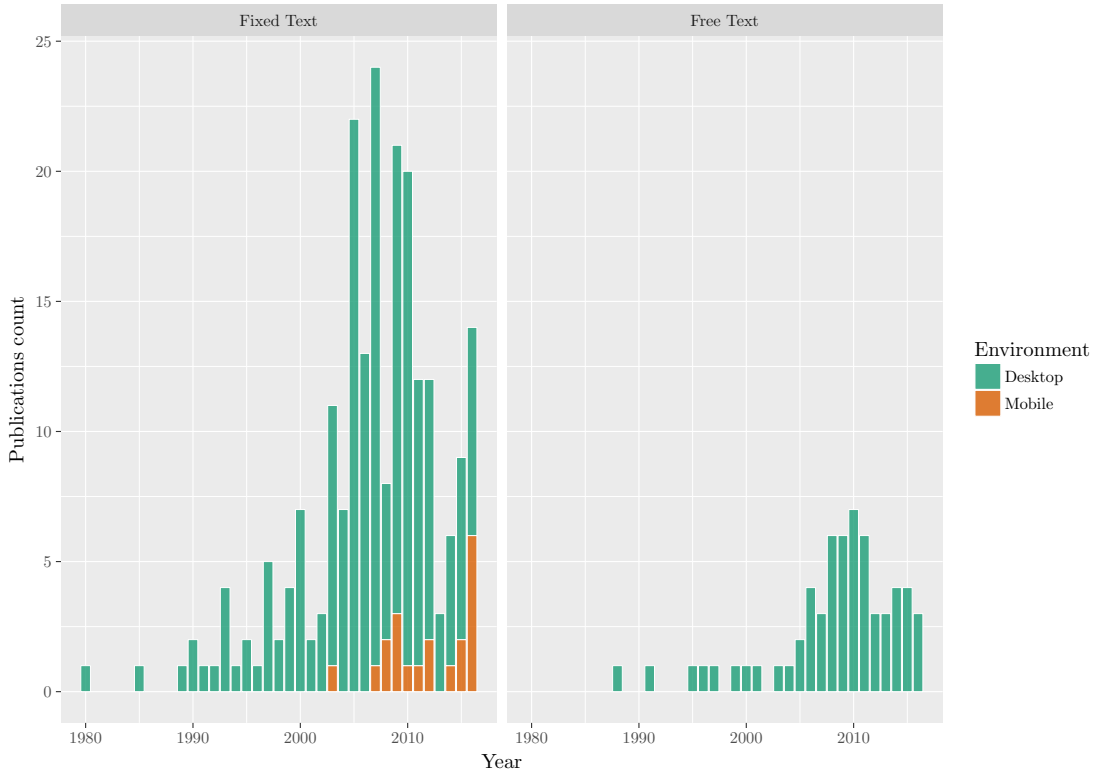

Figure 2.7: Publications distribution

are still a minority of the minority. Taking into account that the use of social networks on mobile devices is ubiquitous, and that many users use their devices to write messages to communicate with each other with popular applications like Whatsapp[5], Telegram[6], or Signal[7], among others, it is strange that there are no studies that deal with the possibility of assessing the user typing these messages, that can be clearly classified as free text, using Keystroke Dynamics.

As per the number of citations, Figure 2.8 shows that, as expected, older relevant publications have the highest number of citations. There are both, free text and fixed text centered publications, that have presented very relevant and interesting results. It can also be seen that as soon as the date is closer to the publication of this PhD Thesis, the number of citations per publications lowers[8]. The most cited work from the free text methodology is *Keystroke dynamics as a biometric for authentication* by F. Monrose and A. Rubin [95]. The most cited work that uses the fixed text methodology is *Password hardening based on keystroke dynamics* by F. Monrose, M. K. Reiter & S. Wetzel [93].

In terms of the most popular journals, or congresses where articles, papers and

[5]Whatsapp: `https://www.whatsapp.com`. Last accessed: September 30, 2017
[6]Telegram: `https://telegram.org`. Last accessed: September 30, 2017
[7]Signal: `https://signal.org`. Last accessed: September 30, 2017
[8]The number of citations shown in the figure was obtained from the Google Scholar database. These values were obtained on January 2017, they may have changed since.

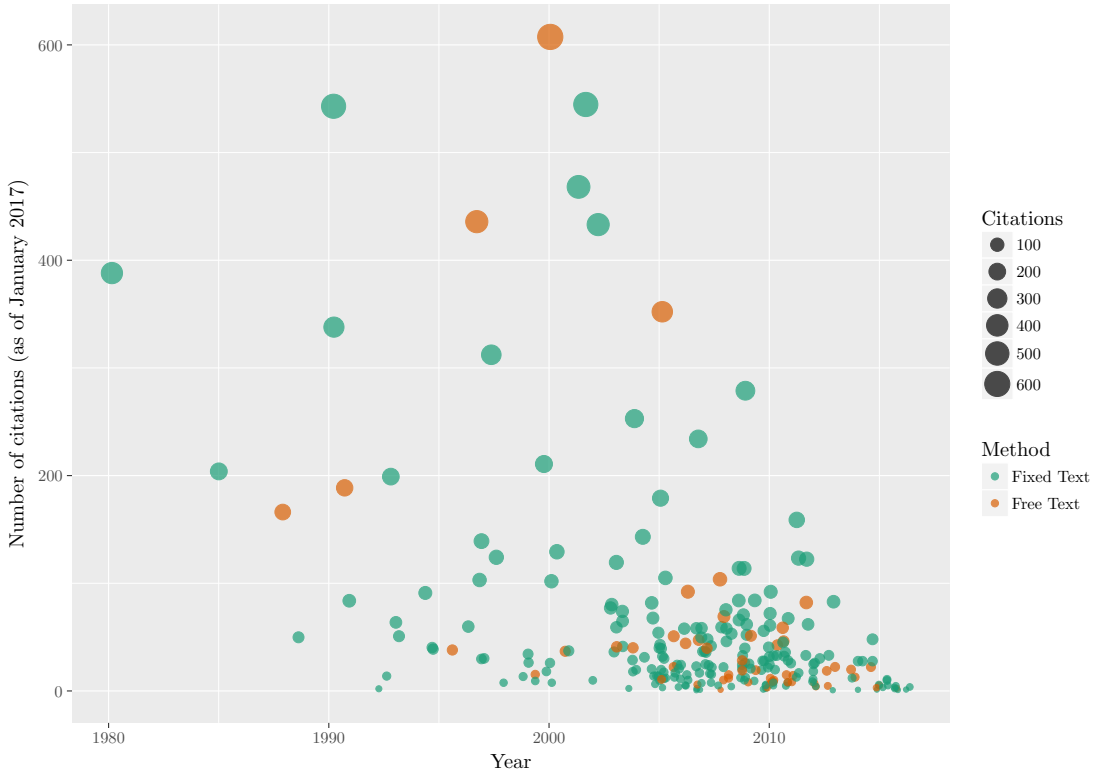

Figure 2.8: Citations per publication

other contributions have been published, these are the most relevant: Computers & Security (Elsevier - ISSN: 0167-4048), Transactions on Systems, Man and Cybernetics (IEEE), International Journal of Man-Machine Studies (Elsevier - ISSN: 1071-5819), International Journal of Information Security (Springer - ISSN: 1615-5262), or Advances in Biometrics (Springer), to name a few. As per congresses these are common: International Joint Conference on Biometrics (IJCB), Biometrics: Theory, Applications and Systems (BTAS), Computer Vision and Pattern Recognition (CVPR), International Joint Conference on Neural Networks (IJCNN), or Systems Science & Engineering, Human-Machine Systems, Cybernetics (SMC), among others.

2.8 Relevant results from previous research

The field of Keystroke Dynamics had a period between 2000 and 2010 when studies were popular and numerous. Since then, even if the number of studies has decreased, a good number of articles and conference papers are still published every year. This section focuses on relevant results that have been published since the inception of Keystroke Dynamics back in the early eighties. These results have been organized in Tables 2.2 and 2.3, for free text and fixed text studies respectively (see pages 47 and 48). In these tables results are sorted alphabetically by author. In this section, though, results are presented chronologically.

2.8.1 Free text studies

There have been a good number of studies on the field of free text, if again, these are a minority when compared to the fixed text alternative. This section focuses on some of the relevant studies centered on free text. Even if these studies came after fixed text had already been studied, these are shown first because free text is the main focus of the proposed research in this document.

In 1997, Monrose & Rubin [94] performed the first study that combined both free text and fixed text. When using free text the results were not encouraging. Only a 23% accuracy was achieved. For their experiments, a group of 31 users was evaluated. On the other hand, when they performed similar tests using fixed text, the accuracy went up to 90%. These accuracy results should not be generalized, though, as other studies proved later on. The difference in accuracy between fixed text and free text has not been that radical. In their study, different methods were evaluated. The main feature used was dwell times and the distance measurement used was the Euclidean distance.

Later, in 1999, Gunetti & Ruffo [58], using digraphs and Decision trees (a C4.5 to be more specific), studied the possible application of Keystroke Dynamics when analyzing commands executed on a system to detect intrusion. Over a period of three months, 10 users submitted samples. Overall, different users got different accuracy results. The best obtained results were of 90% accuracy. The researchers reported these results still inadequate for a fielded intrusion detection system. It can be seen that, despite the differences in setting parameters compared to the previous study, the increment in accuracy is more than obvious.

Dowland, Singh, & Furnell, in 2001, presented a preliminary investigation of user authentication using continuous Keystroke Analysis [42]. In their study, they had a sample size of only 4 subjects, even if 10 were profiled. They used filtered (per count of digraphs) and unfiltered digraphs (with latency between 40 and 750ms) collected over a period of weeks (they did not specify how many). Using weighted statistical measurements and Data mining techniques, as well as other classification algorithms that included Neural Networks, Nearest Neighbor classification and Decision trees, the accuracy results were close to 50%.

In 2005, a key study carried out by Gunetti & Picardi was published [55]. In their study they focused on free text Keystroke Dynamics techniques to identify, authenticate and verify users using a computer system. Many users submitted samples, both real users whose samples were used to build the templates (40 users providing 15 samples each), as well as users that submitted impostor samples (165 users providing one sample each). Their study was focused on Relative and Absolute distances. Their results were very good: when authenticating users these went up to a 0.005% FAR and 5% FRR; when identifying users their results were higher than 99% accuracy. These

are impressive when compared to previous research. It has to be taken into account that much more samples from many more users were available, something that helped detecting better features and a performing a better analysis.

Filho & Freire performed an interesting study in 2006. They evaluated both fixed and free text methodologies. The free text alternative used a simplified Markov chain model. 15 users, that had submitted a total of 150 samples were evaluated. Again, as with most studies to date, only digraphs were evaluated. They proposed a technique based on the equalization of Keystroke Dynamics timing histograms. When applying this technique, they obtained an EER of 12.7%, compared to the 41.6% value when no equalization was performed.

That same year Villani et al. proposed a study based on the Euclidean distance and a Nearest Neighbor classification algorithm [133]. 118 users submitted a total of 2.360 samples. What is most interesting about this study is that they compared different settings in both real environments and under supervised conditions. At the same time, different computer models (desktop and laptop) were compared and evaluated. Different situations provided wild different results. These ranged from 44.2% to 99.6%. The best were achieved when laptops were used to train and test the model.

The following year, Janakiraman & Sim published a paper based on the Bhattacharyya distance [71]. In their study, they had information collected from 22 users. Their focus was set to generalize the use of Keystroke Dynamics from fixed text authentications schemes in free text. They presented a measure of goodness to determine how good a word was based on its Universality, Accuracy, and Expectancy. After finding which were the best words in a particular language, they identified a person based on a common list of fixed strings derived from analyzing user keystroke logs. The proposed methodology is of interest when compared to what is proposed in this research, specially when dealing with behavioral features. Sim & Janakiraman, also in 2007, were also the authors of an interesting study that suggested that digraphs, when used in free text environments, may not be the best structure to organize typing samples [117]. They suggested the possibility of improving the results by using specific *wordgraphs*. Much of what is presented in this study is inspired by the idea they proposed.

In 2008, different studies centered on the free text technique were published. One of these was the one published by Ahmed, Traore, & Ahmed [1]. Much like what Gunetti and Ruffo had attempted previously, their study was focused on forensics. After an attack had been perpetrated, analyzing the Keystroke Dynamics information could be enough to determine the author of the attack. Digraphs and a Neural Network were used to test the samples from 22 users from 13 to 48 years, collected over a period of 9 weeks. This may be one of the few studies that features such young users. The results were a FAR of 0.0152% and a FRR of 4.82%.

That same year, Buch et al. used different features that included digraph latency, and duration and percentage of special characters [27]. They collected 650 samples from 36 users. Their methodology was based in the work of the previous commented article by Villani et al. [133]. They proposed a new and better method using the Euclidean distance to improve accuracy, which peaked at 100% accuracy in a particular case. The worst case was much better, in any case, than what Villani et al. had found.

Also in 2008, Hempstalk, Frank, & Witten proposed the use of a one-class classifier method based on a density estimator with a class probability estimator. They used a bagged unpruned C4.5 Decision tree with Laplace smoothing as the probability estimator [60]. At the same time, they also evaluated the possibility of using SVM if only for the sake of completeness. For their test, they used 150 samples from 10 different users. Their AUC results ranged from 0.540 to 0.941. From the different one-class methods they evaluated none performed always better than the others.

Hu, Gingrich, & Sentosa, used *n-graphs* in a study that also used the Relative and Absolute distances previously presented in the article by Gunetti & Picardi [55] to compare it to a Nearest Neighbor algorithm to classify users [62]. 36 users submitted 36.554 samples. One of the conclusions of their article was that they had proposed an alternative method that solved the scalability problems the Gunetti & Picardi method had while achieving the same good performance in terms of FAR and FRR. This was the first attempt to try to improve the methods proposed by Gunetti and Picardi.

In 2009, Bours & Barghouthi published a paper based on a distance measurement adapted to use penalty and reward functions [25]. They tested both static authentication and continuous authentication. This penalty and reward function kept track of the behavior of the user over time and decided on locking out a user or not when a C value went below a minimum threshold. 25 users were evaluated in a real environment. Their results showed that between 79 and 348 strokes were necessary to block intruders. They reported that this showed that an intruder would be locked out fairly quickly.

That same year, Samura & Nishimura also presented a free text study based on a weighted euclidean distance [114]. Their user base was of 112. The study is interesting because, for the first time, it is based on Japanese writing, something that defers much from the *traditional* alphabet used previously. What they found, after separating the users into three groups based on their typing speed, is that better trained users were easier to classify or identify, with an up to 100% accuracy in some particular cases.

Messerman, Mustafic, Camtepe, & Albayrak, in 2011, published a paper focused on identifying users using free text and real-time environments [92]. 55 different user submitted samples. The researchers used a model based on *n-graphs* of increasing length. The distance measurement they proposed was a measure of the similarity between the users' expected behavior B_E and the determined behavior B_D. It could be

argued that their model was close to what is presented in this study, even if contextual information was never used. Also, they used the idea of an increasing number of vector of graphs, that could be similar to the idea of the Forest of trees that is discussed in Chapter 4. Their results were good, with a FAR of 2.02% and FRR of FRR of 1.84%.

That same year, Stewart et al. published a paper using a k-nearest neighbor classifier. Data was collected from 40 students from the business school of a four-year liberal arts college. 30 were used for the study. Using their proposed model, they got an EER value of 0.5%.

Also in 2011, Rahman, Balagani, & Phoha performed a study based on the degree of disorder of the collected samples [106]. What is most interesting about this paper is that the proposed attack on Keystroke Dynamics challenges the zero-effort methodology that had been used previously. The success rate of forgery attempts created using snooped information (stolen keystroke timing information) had was alarming. Their results showed a 87.75% intrusion success rate.

Bours, in 2012, published a very good study on continuous Keystroke Dynamics [24]. The proposed system would continuously monitor the typing behavior of a user and determine if the current user was still the genuine one or not. This publication shares much similarity with a previous commented study of the same author [25]. A similar system of reward and penalties is used in this study. In this case, the distance measurement used was the Scaled Manhattan distance. The average number of keys that an attacker could type against a genuine template is 182 before being detected.

Also in 2012, Chantan, Sinthupinyo, & Rungkasiri used a Bayes Network classifier and fuzzy logic [32]. Their model was based on Keystroke Dynamics, Location, and IP address used by the users to connect to the internet. Unfortunately, not much information is available on the dataset they used. It is commented that the data in the training and testing sets was generated using a bootstrap method. The results reported show perfect classification in some cases.

Alsultan & Warwick performed a study on free text in 2013 in which they incorporated the concept of keyboard partitioning that had already been attempted in fixed text scenarios [5]. They used the Euclidean distance to obtain distances of key pairs based on their position on the keyboard. Something interesting that is relevant to the present research is that they found that flight times were more relevant than dwell times, something that is also evaluated in the present study. The best results were obtained when using the features Press–Press (PP) + Release–Press (RP) + Release–Release (RR) with a 21% FAR and a 17% FRR.

One of the most interesting articles was published by Brizan et al. [26]. In their study, they tried to identify the gender of the studied users with an 82.2% accuracy when samples were at least 50 words long. They used many features that could be

compared to the contextual ones that are studied in this research.

Ceker & Upadhyaya researched the use of Gaussian Mixture Models with 30 users that had entered a minimum number of characters [30]. Their results showed an accuracy up to 93.8% when using a Gaussian Mixture Model. These two previous publications show some conclusions that are also of interest in the present study.

Another article on free text was published in 2015 by Matsubara, Samura, & Nishimura [91]. Their study focused on finding if different keyboards had an effect on identifying users. Their results determined that if users are proficient, there is no relevant effect on changing keyboards. Otherwise, users should specify which machines they are using so templates generated from other computers can be generated accordingly.

A great article was published by Kang & Cho in 2015 [74]. Their study focused on free text authentication and the results showed that it worked way better with PC vs. other Soft devices. Excellent EER values were obtained when many samples were used.

Also, close to the methodology that will be proposed in this study, is the work of Morales, Fiérrez Vera-Rodriguez, & Ortega-Garcia [97]. They studied 64 students and using different distance measurements, digraphs and trigraphs obtained an accuracy over 90% when identifying users in online learning environments. They did not use contextual information, though.

Alsultan, Warwick & Wei, for the first time, tried to authenticate users using the Arabic language [6]. Their results, using Decision Trees and Support Vector Machines as classifiers, were promising. What was most interesting was their comparison between English-Arabic results. These were better when the user was familiar with the language, Arabic in this case.

In 2016 and 2017, two studies by the author and supervisors of this document were presented at different congresses [40, 41]. These showed the initial results of applying the contextual information methods proposed in this document. The results were still far from optimal but showed promising results. The proposed methodologies are studied in depth in the present document and results have been improved substantially.

2.8.2 Fixed text studies

Fixed text is the methodology with the largest number of studies, when compared to free text, in the field of Keystroke Dynamics. Many of the articles commented below have been already cited throughout this chapter. This section focuses on the relevant studies in this particular field, again, sorted chronologically. Since the present study does not focus on fixed text, for the sake of completeness, a rather small selection is commented in this section. The Bibliography and the Appendix G list all the

publications that have been accessed during the elaboration of this document.

Fixed test was the methodology that was first studied. In 1980, Gaines, Lisowski, Press & Saphiro, published an interesting study focused on authenticating users using their typing rhythm [47]. Their results were called preliminary even if these were rather good. They had access to the Keystroke Dynamics information of 7 users. After applying a statistical methodology based on probability distributions on the most common digraphs in paragraphs they got very promising and encouraging results. Alsultan & Warwick, in [4], reported on a value of 95% EER.

The following years, interesting studies related to typing were published. In 1985, Umphress & Williams published a study on the possibility of verifying the identity of a user through keyboard characteristics [131]. 17 subjects submitted samples that were grouped into digraphs. An statistical approach based on scoring was used. Their results showed a 6% FAR and a 12% FRR.

In 1990, Bleha, Slivinsky & Hussein published a paper on authentication using a Bayesian classifier. They tried different texts to be used as passwords, some of which included short phrases [21]. They used data from 10 subjects for authentication and 26 for verification purposes. In the first case, they got an error of 1.2% (which is roughly an accuracy of 98.8%) and in the case of verification, the error was of 8.1% FRR and 2.8% FAR. This work is a continuation of what Bleha had already worked on in his PhD Thesis [20].

Also in 1990, Joyce & Gupta did a study on 33 subjects focused on a statistical methodology [73]. Using the mean and the standard deviation, and after applying an outlier cleaning process, they got results of 0.25% FAR and 16.36% FRR among different experiments.

In the period between 1993 and 1997, Obaidat, Macchiarolo & Sadoun, published a series of papers centered on the use of neural networks as the main classifier of samples [101, 102]. In their conclusions, they stated that this Machine learning method (they tried both supervised and unsupervised alternatives) performed better that previous statistical methodologies. In the article published in 1997 they got to a 0% EER when testing samples from 15 subjects.

In 2003, Yu & Cho also used Machine learning algorithms to classify samples [137]. The basis of their experiments was the use of SVM. They also used Neural Networks and, at the same time, the use of fusion was also evaluated. When testing samples from 21 users they got to a 0% FAR, and a 0.814% FRR in particular cases, but it is interesting to see that their method called *FS-Ensemble* (the one that used fusion) was the one that performed best.

Another approach was performed by Nonaka & Kurihara. They used special keyboards that allowed for the pressure information of each keypress to be recorded [100].

They used a Time Frequency Analysis approach on a few participants but, unfortunately, no numerical results were provided. It is interesting to note that the use of such keyboards could help in authentications processes as was later tested, in 2005, by Loy, Lai & Lim [86]. In this second case, the use of pressure capable keyboards improved the results up to 70%.

Also in 2005, Sheng, Phoha & Rovnyak, used Machine learning techniques to classify samples [115]. They used a Parallel Decision Tree combined with Wavelet analysis. They argued that the proposed technique needed less computer resources than the Neural network alternative. The data came from 43 students that typed a sequence of 37 characters 9 times. To have even more data, a Monte Carlo method was used. The results proved to be promising, with a 9.62% FRR and a 0.88% FAR.

In 2007, Kang, Hwang & Cho, evaluated the possibility of adapting the models over time [75]. They tested different methods, already commented in this chapter: the moving window and the growing window methodologies. Using the information of 21 subjects, dwell times and flight times, and a K-Means algorithm to classify the samples, they concluded that retraining a model always performed better, no matter the scheme chosen, than without retraining the model. Their best result was 3.8% EER, a 1% lower than when using a fixed window alternative.

The research group formed by Giot, El-Abed & Rosenberger have published a good number of interesting papers over the years in the field of Keystroke Dynamics. In 2009, they proposed the GREYC public keystroke database, focused on having a common set of short inputs of data (i.e. *passwords*) were Keystroke Dynamics methods could be tested, and thus, results could be comparable [49]. At the same time, they tested different Statistical and Machine learning methods on the proposed dataset (using information from 100 subjects) and found that, again, SVM performed rather well (6.9% EER). They also proved that EER performance is dependent on the database size, and especially on the number of users.

In 2011, Li et al. also used SVM, Gaussian Models and k-Nearest Neighbor methods on data from 117 subjects [85]. What is interesting is that, as Giot et al. had previously done, they also proposed a public keystroke dataset to be used in Keystroke Dynamics studies. Their results showed that the best method was the SVM alternative with an 11.83% EER. Again, it was proved that different environments could provide significant different results, with ranges of up to a 10%.

As Rahman, Balagani & Phoha had done using free text [106], Tey, Gupta & Gao also performed a rather complex imitation attack on Keystroke Dynamics authentication schemes [129]. Their article combined different distances (Euclidean and Manhattan) and used different approaches to imitate a user typing rhythm. They used a group of 84 participants playing the role of attackers. Two passwords of different difficulty were

also used. In optimal conditions, a 0.99 FAR value was achieved for both passwords and the 14 best attackers. Due to this fact, they concluded that keystroke biometrics based authentication systems were unusable.

In 2014, Antal, Szabó & László tested the possibility of using Keystroke Dynamics in mobile platforms [8]. They used information from up to 42 users. They were asked to type the *famous* password .tie5Roanl 30 times. They extracted up to 71 features from the collected data, not only using keystroke information, but also other features related to touch and pressure. Using different classification methodologies provided by the WEKA software platform, they obtained a 93.04% accuracy when using Random forests. One of the interesting conclusions of their study is that considering the features related to pressure improved the results significantly.

Also in the mobile device field, the following year, Lee & Lee evaluated the possibility of verifying users using not only Keystroke Dynamics, but also a wide range of available sensors on the device [82]. These included the accelerometer, the orientation, and the magnetometer. Their classification method was based on SVM. An accuracy of 90.23% was achieved and they determined that the orientation sensor was not as important as the other two sensors.

A new idea when collecting samples with results over 90% accuracy was tried by Venugopalan, Juefei-Xu, Cowley & Savvides [132]. The use of an *electromyograph* scan to capture the movement of the muscles showed potential even if it increases drastically the complexity of collecting samples.

In recent years there has been an increasing number of surveys related to Keystroke Dynamics and Mobile Devices [28, 82, 108, 113, 125, 128]. There seems to be an ever-growing interest in studying user behavior when using tablets and smartphones.

2.9 Keystroke Dynamics applications

The applications of Keystroke Dynamics are many. Most of the applications commented below have already appeared previously in this document. This section is provided as a summary of the most common applications where the use of Keystroke Dynamics can be relevant.

- Authentication: Using Keystroke Dynamics users can be authenticated. The combination of a *login*, a *password* and the biometric signature provides a means of accessing protected resources.
- Verification: Keystroke Dynamics allows the possibility of continuously verifying a user using a computer system by constantly checking their way of typing against a template.

- Identification: Even if there has not been a proper previous authentication a user could still be identified by doing a one-to-many check against a database of templates.
- User change control: By constantly monitoring the input of a user it can be determined at which moment the user has been supplanted by another [2, 24].
- Exam control: Keystroke Dynamics could be used to detect abrupt changes on the template of a user and conclude that another user is taking the exam. Apart from [2], another article that comments on the use of Keystroke Dynamics to control remote users when doing exams is found in [121].
- Password hardening: Protecting access to highly sensible resources can be achieved by adding a Keystroke Dynamics signature to the *password*. Examples where this has been studied and applied can be found in [16, 93, 109].
- Authoring: When users submit information that has been controlled using a Keystroke Dynamics technique, it is possible to determine who wrote the entry by doing an identification check on the data. This provides a means of detecting fraud, for example, when contributions are sent to an e-learning environment. At the same time, when a report has been authored by more than one person, using the same technique it could be determined which part was written by each student.
- Emotion detection: Some studies have proven that, using Keystroke Dynamics, it is possible to detect some emotions [45, 79].
- Attacks: Keystroke analysis can lead to sophisticated timing attacks. By *listening* to a user typing, either via a network sniffer or using a key-logger, it is possible to learn how the user types or, even identify what users are typing over encrypted channels [105, 107, 119, 129].
- Online user identification: Information could be captured to recognize users on subsequent visits to a website and improve the user experience using marketing techniques. Also, users could be identified using Keystroke Dynamics when surfing the Internet to prevent crime.

2.10 Advantages of using Keystroke Dynamics

There are two main criticisms against Keystroke Dynamics. The first one focuses on the technique itself: the question whether the results are good enough to apply this technique in production environments is often asked. The second problem relates

to methodology, being the size of the tested population the main problem. Despite these two issues, Keystroke Dynamics is a very interesting technique, mainly due to its simplicity and the possibility of being ubiquitous, to be used in online learning environments. This section outlines the features this biometric technique presents that have been the basis for choosing it over the alternatives when elaborating this study.

Keystroke Dynamics requires training, as any other biometric technique. The problem is that training in free text environments can be slow. This technique shows its true potential once a minimum number of samples have been captured. With this in mind, it should be perfectly normal to find that accuracy is low during the initial phases of enrollment. If this technique is applied in online learning environments, it is easy to understand that as the student progresses throughout their studies the accuracy will increase over the semesters.

Keystrokes Dynamics is very easy to implement. This should be one of the key features that have to be considered when evaluating the possibility of choosing this technique. This is not only related to the hours needed to implement a solution that can capture the timing intervals users have (see Appendix C for the code developed to capture samples for this study), but also to the economic resources needed to implement it. As has been previously commented in this chapter, using simple pieces of software that, together with the operating system, can capture such timing intervals is enough to be able to create a template of the particular rhythm a user has. It should be noted that the code needed to capture the rhythm should be ubiquitous and not require user intervention or the possibility of disabling it. Having a user install an application, plugin, or add-on that will spy him is not going to bring too many costumers. On the other hand, if this is already part of the learning application, users will feel more comfortable with it.

Another of the key features that makes this technique attractive is the fact that it does not need fancy hardware to work. Be it a desktop or notebook computer, or a mobile device, all that is needed is an off-the-shelf keyboard or the built-in onscreen keyboard. It has been established that the rhythm a user has may depend on the device being used, meaning that a user may have different rhythms when using traditional keyboards as opposed to onscreen ones but, nonetheless, the rhythm is present anyway. This should point to the possibility of having different models depending on the source of information.

From this previous point, it is also easy to see that the possibility of using this biometric technique everywhere the user is, at any moment, is highly interesting. Using other biometric measurements, the need of a physical sample capturing device may be mandatory, something users may not always take with them. On the other hand, a keyboard, when using any kind of computer device, is always present. Again, the idea

that the capturing software does not depend of the environment the user is working on, and that it does not require installation intervention from their end, is very important to ensure the maximum number of samples can be acquired and that the identification or authentication processes can happen anywhere on any device.

The use of this technique is fairly transparent to the user, something that is key when implementing a biometric solution. This means that users do not have to be even aware that such biometric technique is being enforced. When using other alternatives, like for instance face recognition, a camera is always staring at the user, something that may make them uncomfortable. Users tend to forget that their typing rhythm is being captured, something that is also very important to be able to have relevant samples from the users. At the same time, if the Keystroke Dynamics module is only enabled in specific parts of the application, the fear of having sensitive information captured also decreases. Related to this, the fact that users are identified when they use their own devices should also be considered important, because they do not have to touch recognition devices that may have been used by countless other users.

As per the obtained results, it has been commented that a good approach would be that of a multimodal technique while models are not robust enough. The results presented in this chapter, obtained by previous research, have shown that this technique can be perfectly valid as a sole biometric technique. The tests carried out in this study should help determine if the new proposed method of organizing samples can be a good option to identify or authenticate users, always with a small margin of error, in different schemes using Keystroke Dynamics.

Another interesting feature of Keystroke Dynamics, this one related to research, is the possibility of accessing public keystroke databases [17, 50, 85]. These are sources of information that contain timing intervals from keystroke sessions. In general, these tend to be focused on short texts, mainly to test authentication. Other publicly available databases that contain samples captured in free text environments like the one used in [55] have the problem that only KD events are available. Recently, a very interesting initiative, similar to what is being done in Kaggle[9], is the KBOC: Keystroke Biometrics Ongoing Competition[10] [98]. A public database is given to participants and using the methods researchers may prefer, the objective is to correctly identify as many as users as possible. This database is focused on fixed text without mistakes being allowed.

To sum up, the main advantages this technique offers are: transparency to the user, easiness of implementation, cost effective technique, and good enough results for non-critical applications.

[9]Kaggle: `https://www.kaggle.com`. Last accessed: September 30, 2017

[10]KBOC: `https://sites.google.com/site/btas16kboc`. Last accessed: September 30, 2017

2.11 Summary

In this chapter, Keystroke Dynamics past and current research has been presented. A brief description of key theoretical concepts on both Biometrics and Keystroke Dynamics has also been provided. The most used techniques have been analyzed. Both the fixed text and the free text methodologies have been described and reviewed. The most common used data analysis techniques in Keystroke Dynamics (Statistical analysis, Distance measurements, and Machine learning) have been commented upon. Previous research where these methodologies and techniques have been applied has been reviewed.

The following chapter presents the Objectives and Hypotheses set to study and prove in this PhD Thesis. The proposed research topics have been identified after studying the current State of the Art.

Study	Features	Classifier	Subjects	Samples	Performance
Ahmed et al. [11]	DG[1]	NN[2]	22	-	0.015% FAR, 4.82% FRR
Alsultan et al. [6]	DG, KP	DT, SVM	21	180	0.169 FAR, 0.423 FRR
Alsultan & Warwick. [5]	DG, KP[3]	ED[4], Fusion	15	380	21% FAR, 17% FRR
Bours [24]	DG, KD[5]	MD[6]	25	-	182 keystrokes
Bours & Barghouthi [25]	DG, KD	Distance	25	-	79 - 348 keystrokes
Brizan et al. [26]	Contextual features	Logit, SMO[23], NB[21]	486	300	82.2% Accuracy
Buch et al. [27]	DG, KD, PSC[18]	ED	36	650	100% - 98% Accuracy
Ceker et al. [30]	DG	GMM[22]	30	500	93.8% Accuracy
Chantan et al. [32]	DG	Bayes	-	-	0% EER
Dowland et al. [42]	DG	Statistical	4	-	50% Accuracy
Filho & Freire [96]	DG	HMM	15	150	12.7% EER
Gunetti & Picardi [55]	NG[7]	RD[8], AD[9]	205	765	0.005% FAR, 5% FRR
Gunetti & Ruffo [58]	DG, Commands	DT[10]	10	-	90% Accuracy
Hempstalk et al. [60]	DG, KD, TS[11], ER[12], PRO[13]	One-Class	10	150	AUC 0.540 to 0.941
Hu et al. [62]	NG	RD, AD, KNN[14]	36	36554	0.045% FAR, 0.005% FRR
Janakiraman & Sim [71]	DG, KD	BD[15]	22		100% - 70% Accuracy
Matsubara et al. [91]	DG, KD	WED, RD	21 – 26	-	$\sim$ 99% Accuracy
Messermann et al. [92]	NG	SDM[16]	55	-	2.02% FAR, 1.84% FRR
Monrose & Rubin [94]	DG, KD	ED	31	-	23% Accuracy
Morales et al. [97]	DG, NG	KNN, MD, MHD[20]	64	500	90% Accuracy
Rahman et al. [106]	FT	Degree of disorder	50	-	87.75% Attack accuracy
Samura & Nishimura [114]	DG, KD	WED[17]	112	-	67.5% - 81.2% Accuracy
Stewart et al. [121]	DG	KNN	30	-	0.5% EER
Villani et al. [133]	DG, KD, TS, PSC, EP[19]	ED, KNN	118	2360	99.8%, 44.2% Accuracy

[1] Digraphs [2] Neural Network [3] Keyboard partitioning [4] Euclidean distance [5] Key duration [6] Manhattan distance
[7] n-graphs [8] Relative distance [9] Absolute distance [10] Decision Tree [11] Typing speed [12] Error rate [13] P-R Ordering
[14] k-Nearest Neighbour [15] Bhattacharyya distance [16] Spearman's foot-rule distance-metric
[17] Weighted Euclidean distance [18] Percentage of special characters [19] Editing patterns [20] Mahalanobis distance
[21] Naive Bayes [22] Gaussian Mixture Model [23] Sequential Minimal Optimization

Table 2.2: Free text studies results (own elaboration and adapted from [4])

Study	Features	Classifier	Subjects	Performance
Antal & Szabó [8]	DT[1], FT[2]	Statistical, SVM[3], Neural Network, DT[4]	42	93.04% Accuracy
Bleha & Slivinsky [21]	FT	Statistical, Distance	26	2.8% FAR, 8.1% FRR
Gaines & Lisowski [47]	FT	Statistical	7	95% EER
Giot & El-Abed [51]	DT, FT	Statistical, Distance, SVM	100	6.96% EER
Joyce & Gupta [73]	FT	Statistical	33	0.25% FAR, 16.36% FRR
Kang & Swang[75]	DT, FT	Clustering, Distance	21	3.8% EER
Lee & Lee [82]	Sensors	SVM	4	93.8% Accuracy
Li & Zhang [85]	DT, FT	SVM	117	11.83% EER
Loy & Lai [86]	Pressure	Neural Network	-	0.87% FAR, 4.4% FRR
Nonaka & Kurihara [100]	Pressure	Time series analysis	-	-
Obaidat & Sadoun [102]	DT, FT	Neural Network	15	0% EER
Sheng et al. [115]	DT, FT	Decision Tree, Monte Carlo	43	0.88% FAR, 9.62% FRR
Tey & Gupta [129]	DT, FT	Statistical	84	0.99% FAR*
Umphress & Williams [131]	DI, DT	Statistical	17	6% FAR, 12% FRR
Venugopalan & Juefei-Xu [132]	Electromyograph	PCA[7], UDP[8], kNN[9], CFA[10]	14	$\sim$ 90% Accuracy
Yu & Cho [137]	DT, FT	Neural Network, SVM	21	0% FAR, 0.814% FRR

* Attack study [1] Dwell time [2] Flight time [3] Support Vector Machine [4] Decision Tree
[5] Digraph [6] Shift Key [7] Principal Components Analysis [8] Unsupervised Discriminant Projection [9] Nearest Neighbor
[10] Class-Dependent Feature Analysis

Table 2.3: Other studies results (own elaboration and adapted from [78, 122])

3 | Objectives and Hypotheses

This chapter presents the proposed Objectives and Hypotheses of the research. These have been defined after a careful analysis of the current State of the Art.

3.1 Objectives

The main objective of the proposed research is:

> Determine if the use of Keystroke Dynamics and models based on contextual information and behavioral features allows the possibility of identifying or authenticating users with a small margin of error.

These below are some sub-objectives that expand the proposed main objective:

- Determine if the proposed methodology of classifying samples based on contextual information is useful enough to identify users using a computer system.
- Identify a user using the largest sequence of letters of a word and the latencies associated with each keystroke.
- Determine the model building and searching parameters that better help during the identification process.
- Find out if the proposed methodology is also a good candidate to authenticate users instead of only identifying them.
- Determine if other behavioral features, such as mistakes users make, word or sentence frequency, or word delimiters, are also valid features to identify users or, on the other hand, these should be discarded.
- Determine if gender or age group present particularities that can be useful to build better models.

3.2 Hypotheses

The following Hypotheses and sub-Hypotheses are proposed:

Hypothesis 1. *The global size of the model and the number of samples are highly relevant when building quality models.*

Hypothesis 1.1. *If more samples are collected from users, the template will be better and the chances of identifying them with a smaller error will improve.*

Hypothesis 1.2. *If a good number of samples per user are available, the proposed method will perform better than using n-graphs frequency models.*

Hypothesis 2. *It is possible to identify a user on a computer, with a small margin of error, using Keystroke Dynamics, contextual information, and behavioral features.*

Hypothesis 2.1. *Not all model building parameters are equally relevant, some will be more suited to better identify users.*

Hypothesis 2.2. *Behavioral features such as mistakes users make, word and sentence repetition, and the use of particular key combinations can also be important features when identifying users.*

Hypothesis 3. *The proposed methodology can be valid to authenticate users instead of only identifying them.*

Hypothesis 4. *The gender and the age group a user belongs to can be useful to build better models and improve accuracy.*

3.3 Summary

The Objectives and Hypotheses presented in this chapter suggest that there are different lines of study. The first line tries to establish how much information is needed to build valid and robust models. The second line suggests a study on the features that better help identify individuals, either from the building and searching parameters or from behavioral features. A third line of study will try to apply the proposed methods to authenticate users. Finally, there is a line of study focused on the age group and gender of the users submitting samples.

The following chapter details the phases that have gone through to implement the different experiments to test these Objectives and Hypotheses.

4 | Methodology

This chapter focuses on the methodological particularities that have been followed to determine if the proposed Objectives and Hypotheses from the previous chapter are valid or achievable. This study combines procedures from both the typical biometric methodology described in Section 2.4 and from basic Data Mining techniques to test samples against models and establish accuracy. Outlined below is a description of the steps that have been followed throughout this study. The following sections in this chapter explain in more detail each of these steps.

These are the the phases the study presented in this document has been through:

1. Define contextual information and relevant behavioral features: The term *context* can be misinterpreted or in need of a more in-depth clarification. A short introductory section explains what is meant when studying the relevancy of contextual information related to Keystroke Dynamics. At the same time, the chosen behavioral features are also enumerated.

2. Collect user samples:

   - Develop a keystroke sample collector and install it accordingly: For this goal, a combination of different segments of code written in PHP and Javascript (that combine AJAX and jQuery as well) has been used. The goal is to capture the timing intervals from users when they submit messages to the Discussion forums at the Virtual Campus of the University of Andorra.
   - Configure a persistent layer to store the collected information: For this, a simple MySQL database has been used. Appendix E describes the characteristics and shows the commands used for its creation. During the period in which samples were collected, close to 7.5 million events were stored from more than $10,000$ sessions and from close to 500 users.
   - Collect keystroke information: This is the first real step in the classic biometric methodology and an essential one. This process was performed during a limited period of a year and a half, from October 2015 until February 2017.

- Select the most relevant users to work with: Even if close to 500 users submitted messages to the Discussion forums, many of these users only sent a couple of messages or only a few events. Not all users have been found equally relevant in terms of quality of the submitted samples. Different strategies have been followed to select consistent groups of users to work with.

3. Data analysis:

   - Develop software to analyze the collected information: For this task, different tools have been written in both Python[1] and R[2]. A Python application has been developed to build the different types of models (later described in this chapter), taking into account the chosen parameters, to compare new samples to these models, and to obtain distance measurements. These distances measurements are usually written to an output file to be later treated using the R statistical programming language. This process consists in applying different strategies to identify the owner of a session and determine which are the best suited configurations and parameters. The possibility of authentication has also been implemented using R.
   - Build a series of models for each user: The models evaluated consist in logical tree models and in *n-graph* frequency models.
   - Analyze the captured samples and models built to determine:
     - The optimal size and quality of the proposed models: Different groups of users may have different sized tree models depending on the submitted number of sessions and events. Determining if the size of the model improves the results is the objective of this experiment. Another goal it to determine the optimal building parameters to have models of a certain quality.
     - The relevant parameters linked to word searching: Different parameters are evaluated to improve the accuracy of the system when words are searched in the tree models. These include the minimum number of words needed to consider a session valid, the type of recursion used when searching words in the models, and the length of words, among others.
     - The best method to identify users: This includes evaluating different distance measurements and different methods to identify the owner of a session. The main goal is to increase the accuracy of the system up to standard values, as reported by the current State of the Art.

[1]Python: `https://www.python.org`. Last accessed: September 30, 2017
[2]R Project: `https://www.r-project.org`. Last accessed: September 30, 2017

  - The effect of particularities related to user behavior: In this case, mistakes users make, frequency of words and sentences, and the delimiters used to determine particular key combinations are evaluated to find whether such features are relevant when identifying users. These features are going to be used to weight distances previously obtained from the models.
  - The possibility of authenticating users instead of only identifying them.
  - Whether the age group and gender a user belongs to has an effect when identifying or authenticating them.
  - How the proposed methodology compares to classical *n-graphs* frequency models: For this, a model using Relative and Absolute distances and various graph lengths is going to be used. The results from both methodologies are going to be compared to determine the environments in which each methodology performs better.
- Apply weighting techniques to improve the results: In some tests, weighting techniques are applied to favor relevant features or to give more importance to those distances closer to zero. Also, when scaling distances as per word frequency or successive word usage, weighting is also applied.
- Apply fusion techniques to improve global results: This technique is applied in some of the methods to determine the owner of a session. The goal is to use the features that better classify users and combine them in a way that improves global accuracy.

4. Elaborate and present the results: This step is presented in Chapter 5. This chapter contains the description of every performed experiment as well as the results these have given.

5. Present the conclusions and propose future work ideas: This last step is presented in Chapters 6 and 7. The conclusions obtained from the results in relation to the proposed Objectives and Hypotheses are presented in Chapter 6, and future work proposals are outlined in Chapter 7.

Figure 4.1 shows a visual representation of the steps described and the order in which these have been carried out. The following section begins with the description of how *context* is understood in this research study.

4.1 Contextual information and behavioral features

This section explains how contextual information, linked to Keystroke Dynamics, is used in this study. Also, a description of the chosen behavioral features is given.

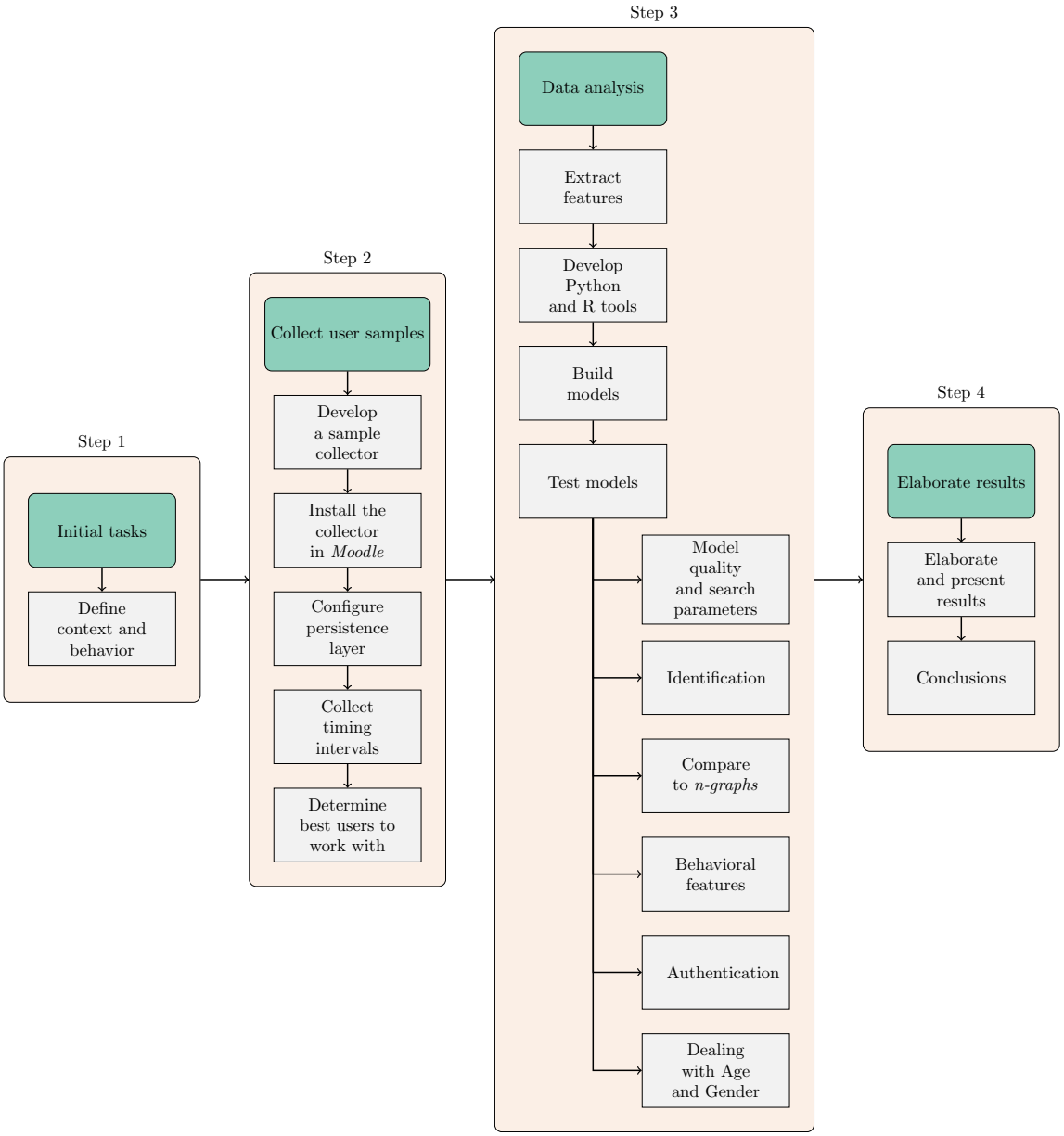

Figure 4.1: Followed methodology

4.1.1 Context applied to Keystroke Dynamics

The proposed research in this document presents a new way of structuring the collected samples so they retain contextual information related to the position of each typed letter. In the majority of studies that have been carried out to date, where a free text methodology is used, samples have usually been sorted out as *n-graphs* in dictionary styled logical structures where the information of the position where each letter has been typed is not kept or considered relevant.

The following words: IS, IRIS, THESIS and DISAPPEAR, have something in common: all these words share the IS combination of letters. If *n-graphs* were to be used, these words would be sliced into groups of n letters. In the case of digraphs, or 2-graphs, the unique combinations (with their total number of appearances) would be:

IS (4), IR, RI, TH, HE, ES, SI, DI, SA, AP, PP, PE, EA, AR

If the same was to be performed using trigraphs, or 3-graphs, this would be the result:

IS, IRI, RIS, THE, HES, SIS, DIS, ISA, SAP, APP, PPE, PEA, EAR

This could grow up to any n number of graphs. As soon as n increases, the number of common graphs decreases dramatically if not enough samples are available. In the proposed example, when using trigraphs, no combination of letters is ever repeated. On the other hand, when using digraphs 4 IS samples are found. From the repeated IS combination, a *fairly* good idea of the user's way of typing this particular pair of letters can be established. The same, of course, applies to the rest of letter combinations, but since these appear only once, the template could not be that defining.

What is most interesting is that no importance is given to the fact that the IS combination of letters appears in different positions in each of the examples. IS is a word with only these two letters; IRIS and THESIS present the combination at the end of the word; and DISAPPEAR has the letter combination in the middle of the word.

One of the main goals of this research it is to determine if using the information related to the position of any combination of letters instead of grouping them all in a common dictionary of *n-graph* values is of relevance when identifying or authenticating users. The typical *n-graph* dictionary logical structure, where each graph has a list of timing intervals for each appearance, is of no use when the preservation of the information related to the position of any letter combination is needed. In this research, a tree of words is proposed in Section 4.3. A hash dictionary could have also been used, but in order to achieve greater speeds when searching new samples and all the possible

derived words from a common root, a logical tree was thought to be better suited and more versatile.

Newly collected samples will be tested against the proposed tree model of words. The main objective is not to find the exact same words in the model, as a *wordgraph* methodology would propose, but fair approximations that have the same root letters. The *longest* similar partial words will be used to compare and find minimum distances between testing samples and models.

4.1.2 Behavioral features

When talking about behavior, the features that are going to be evaluated are not directly related to Keystroke Dynamics *per se*, but are going to be used to modify the distances obtained using the timing intervals associated to the rhythm of each user.

These are the different behaviors that are evaluated in this study:

- Mistakes users make: This feature evaluates if incorporating recurrent mistakes users make into the model improves the accuracy of the system. The idea behind this behavioral feature is that users make the same mistakes over and over again when typing specific words and, at the same time, they may follow the same steps to correct them.
- Similar to the previous feature, the possibility of detecting other key combinations users may use commonly is the goal of the evaluation of the delimiters used to detect words. If these delimiters are kept to a minimum, allowing more information into the model, accuracy may improve.
- Word and sentence frequency: This feature evaluates if users have a common *bag-of-tricks* where they choose words from and, at the same time, they build similar sentences from. If a user always uses the same words, the distances obtained from comparing a testing session to a model will be rewarded. The same will be done when evaluating common sentence constructions.

4.2 The Dataset

This section provides information about how the samples used in this study have been collected over a period of three semesters. At the same time, the creation of different sized groups of users for the different proposed experiments is also outlined. Each of these differently sized user groups has the specific goal in mind of increasing the statistical relevance of the obtained results.

4.2.1 Software developed to collect samples

This research focuses on free text. In order to perform any kind of user identification or authentication, it is necessary to collect, or have access to, samples of the users' keystroke natural rhythm. Also, it would be highly desirable if these samples came from a free and uncontrolled environment. When dealing with Keystroke Dynamics it is common to develop software in two stages of the research: when capturing the rhythm of a particular user, and when analyzing the captured data. During the second stage of the research, though, the development may be optional if existing software is used. Some researchers have used tools like Matlab[3] or RapidMiner[4] which already have plenty of classification algorithms available.

Historically, when capturing information from the rhythm a user has, the developed software solutions have ranged from simple C/C++ programs [94], to Java applets for the key-logging of web events [112], mobile applications [9] or even full-fledged Windows™ key-logger desktop applications with added survey functionality [45].

Whatever the chosen method to capture information is, it can present an ethical problem: what should or should not be captured? Should the key-logger code be system-wide or application-specific? How can it be determined if something is sensible or private data? Sometimes, the different solutions proposed by the literature solve these problems by allowing the users to choose what to capture. On the other hand, samples can also be collected on a controlled environment when users are fully aware that this process is being taken place. Having users choose if the sample gathering application is enabled or not, of course, limits the research in ways that are not always desirable. On the other hand, having a too much controlled environment can also lead to having not realistic enough templates.

One of the goals of this research is to see if it is possible to apply the described methods on e-learning environments, and at the same time, on an environment as free and real as possible. It would be of great interest to determine, for example, that an entry submitted to a Learning Content Management System (LCMS) has been written by the logged in user and not by an impostor. It would also be of great interest that users were not aware that their rhythm is being captured, so that their behavior is not affected. On the other hand, users have to be informed at all times that the Keystroke Dynamics samples collection will eventually take place.

With this in mind, the first objective has been to develop a *snippet* of code that can be enabled only on the LCMS and, more specifically, on the Discussion forum modules of the e-learning environment. The collector has not been enabled anywhere else in the system or in any other web page. In this particular case, and taking into

[3]Matlab: `https://mathworks.com/products/matlab.html`. Last accessed: September 30, 2017
[4]RapidMiner: `https://rapidminer.com`. Last accessed: September 30, 2017

account the possibility of using the Virtual Campus of the University of Andorra as a lab, code for the *Moodle*[5] open source learning platform has been developed. This code has been developed using the PHP[6] language according to the latest *Moodle* development standards. The documentation page for this software application was thoroughly studied[7].

Appendix C shows the full code listing that has been developed for this purpose. This Appendix includes both the code executed on the client's side as well as the code executed on the server's side. The code is shown as a proof of how easy it is to collect timing intervals of any individual and send them to a remote server using only a few lines of code.

The client's side of the code is written using PHP. The code prepares a Javascript script that, together with AJAX, collects the timing information of the user's way of typing and sends it over the network, using a POST method, over a secure connection, to a remote server where it is stored in a MySQL database for later analysis. The server side code has also been developed in PHP and its only job is to store the received information into the database.

Taking a deeper look at the collector code, the most relevant parts are the following:

```
1 ...
2 addLoadEvent(function() {
3   setInterval(sendData, 5000);
4   require(['jquery'], function($) {
5     $(document).on('keydown', function(event) { record(event); });
6     $(document).on('keyup', function(event) { record(event); });
7   });
8 });
9 ...
```

Listing 4.1: How events are recorded

Listing 4.1 shows the definition of the function called *record(event)* that is executed every time either a *keydown* or a *keyup* event is detected. This is the basis of the samples gatherer and it is important to note that both events are recorded and stored, instead of just *keydown* events as is the case of studies previously carried out [55].

```
1 ...
2 function phd_theData(usrid, session, event) {
3   if (isMobile()) {
4     var date = new Date();
5     this.timeStamp = date.getTime();
6   } else {
7     this.timeStamp = event.timeStamp;
```

[5]Moodle: `https://moodle.org`. Last accessed: September 30, 2017
[6]PHP: `http://www.php.net`. Last accessed: September 30, 2017
[7]`https://docs.moodle.org/dev/Main_Page`. Last accessed: September 30, 2017

```
8     }
9     this.usrid = usrid;
10    this.session = session;
11    this.keyCode = event.keyCode;
12    this.type = event.type;
13    this.altKey = event.altKey;
14    this.ctrlKey = event.ctrlKey;
15    this.metaKey = event.metaKey;
16    this.shiftKey = event.shiftKey;
17  }
18  ...
```

Listing 4.2: Data structure for the recorded events

Listing 4.2 shows the information that is stored for every detected event. These are the most relevant captured fields:

- timestamp: This value contains the instant a key has been *pressed* or *released*. Every browser or device handles this value in a different way but, in the end, whatever the format, two successive events can always be subtracted to obtain a timing interval in milliseconds.
- usrid: The *Moodle* LCMS identifier for the current user. It is used to verify that samples belong or not to a particular user.
- session: A random identifier for the particular message being typed. Each session is either used to build the user model or to be tested against it, but never for the two processes at the same time. This would defeat the goal of having different data in the model from the one being tested, something that would give erroneous and biased results.
- keyCode: The code of the pressed or the released key. The list of Key Codes and their values can be easily obtained on the Internet, and also the *madness* that is not having a standard for every OS out there to treat commonly the particular combinations that can occur with modifier keys[8]. See Appendix F for a list of common Key – Key Codes.
- type: This value can only present two possible states. It specifies if the event is either a *keydown* or a *keyup* event. The otherwise also possible events known as *keypressed* have not been used in this study.
- altKey, ctrlKey, metaKey, and shiftKey: These flags allow the possibility of knowing if a particular combination of keys have been pressed at the same time.

[8]JavaScript Madness: `http://unixpapa.com/js/key.html`. Last accessed: September 30, 2017

For example, if a CTRL+C combination has been typed, the entry for the C letter with key code 67 would have this flag enabled. Also, the *keydown* and *keyup* events for the CONTROL key would have been recorded separately.

The gathered data in the browser using Javascript is sent regularly (once every 5 seconds) to the database server using an Hypertext Transfer Protocol (HTTP) POST method. This means that other information related to the client's browser environment can be also collected. For every session, the following information has also been collected and stored in the persistent layer:

- Origin IP address: This allows the possibility of knowing if users change working environments often or, on the contrary, they are usually in the same working place, using the same device.
- Browser agent: This, in combination with the previous parameter, allows the possibility of knowing if a user always uses the same device or, on the other hand, uses different devices more often than not. This could be used to determine if the effectiveness of the proposed methods depends on the devices (as studied in [133]).
- Language: This parameter is stored to know the language preference of the client's browser. It was thought to be used to test the feasibility of comparing different languages, but in the end, it was not used due to the fact that almost all entries submitted to the Discussion forums at the UdA's LCMS were in Catalan.

At this point, no information is available regarding age group or gender of the users sending events to the server. To obtain this information the University of Andorra was asked for access to the records of the users identified by the *usrid* field on the *Moodle* platform. With their consent, information of age (relative to the date when data collection ended) and gender was also added to the database.

4.2.2 Samples gathering

The process of collecting samples was performed from October 2015 until February 2017. This includes, roughly, three academic semesters. As explained in the previous section the developed code was installed in the *Moodle* LCMS at the University of Andorra, and the process of collecting data run non-stop during the specified period.

The goal has always been to have as many samples as possible from a wide variety of users. It is true that, during this period, some users may have started or finished their studies. In such cases, the number of collected samples, based on the number of

messages, may have been too low and, in most cases, these samples have not been used in this study.

The possibility of comparing the results using other known methods or the use of public datasets was also considered. The problem was that, for example, some of these third-party publicly available datasets had only taken into account *keydown* times [55]. When evaluating these public datasets, it was thought that missing features were a drawback to the proposed research.

The environment in which samples are collected is also important and has been a matter of study in the past [133]. It has been established that the keyboard and the computer (either a desktop or a laptop) can influence greatly the way a person types. The proposed research and its possible applications to e-learning environments could be affected by this fact. Since there were no restrictions or impositions on computer type or model, keyboard type, the moment the samples were captured or any other external factor, results could be affected. It has always been the goal of this research to collect samples in an environment as close as possible to real life, without any kind of tailoring or other form of intervention. In the end, it is believed that this has been fully accomplished.

4.2.3 Ethics in samples gathering

The code showed in Section 4.2.1 only captures the keystrokes when a message to a Discussion forum module is written. This is not specified in the code *per se*, but the capture code was only enabled in this particular module. Since this could be regarded as an act of spying users, and trying to be as ethical as possible, the following message was added, after being approved by the *Junta Acadèmica de la Universitat d'Andorra*[9], to all the Virtual Campus *Moodle* pages:

> Les dades d'ús dels recursos tecnològics de la Universitat d'Andorra
> podran ser utilitzades per a estudis de la mateixa Universitat.

This roughly translates to:

> The data originating from the use of technological resources at the
> University of Andorra can be used for studies of the same University.

Even if this meant that users were informed that their use of the web applications at the University of Andorra could be used for studies, it is firmly believed that they were not aware of the implications of having these keyboards events being recorded.

[9] Academic Board of the University of Andorra

Samples were obtained in real time as soon as the user typed text on any *textarea* of the Discussion forum message submission page. This posed an ethical dilemma. As soon as the user typed any kind of information, this would be immediately sent to the database. If then, the user decided to delete this data, either because they had changed their mind, or for whatever other reason, the previously entered information would have already been stored in the database. New, or corrected information would be also sent to the database, complementing the model, but never replacing the old one.

Even if all the collected information has never been used for anything else than the present study, it could be argued that it contained information that users had never intended to send and thus, this should have been treated accordingly. Solutions to avoid the possibility of rebuilding the original messages and their word sequence were thought of. Initially, it was argued that what was interesting to the research were only the words typed. The order in apparition of the words in a sentence was, *a priori*, not important (this was later debunked, though, when the evaluation of sentence construction was evaluated). The following two methodologies were proposed:

- Store the words in the database without preserving the order: This implied that the PHP script that inserted the information into the database had to be aware of the logic of detecting words, something that depended on many parameters that were also a matter of study. The script, then, could not perform such action without losing valuable information.
- Scramble the key code information: Another possibility that was evaluated was to avoid having the key codes in the clear. This way, using a reversible cipher, the information in the database could be hidden to curious eyes. It was though, though, that a simple frequency attack could be used to break this system, and it was also discarded.

In the end, none of these solutions were implemented due to their complexity, little added value and the possibility of messing with valuable features from the collected information, like for example, the idea behind frequent words scaling or successive words scaling. On the other hand, policies to ensure the security, privacy and integrity of collected information were enforced.

4.2.4 Keystroke dataset

Tables 4.1 and 4.2 show a summary of the collected data, and the associated distribution. Figures 4.2 and 4.3 depict the information regarding users from these tables. It can be seen that user distribution is heterogeneous, something that has been highly regarded in this kind of studies, and at the same time, something that is not always easy to

achieve. Data from all age and gender ranges between 18 to 69 years is available even if the group that submitted most information is the group of the youngest users, and from these, the women group. It could be argued that women write more than men, but even if this was true (this is not the objective of the present research), the distribution could be explained because most women belong to studies where the use of the Virtual Campus is more widespread and encouraged than in studies, like Computer Science, where the use of the Discussion forum modules is more limited. The percentage of men and women in their respective studies is highly biased. In Computer Science studies, at the University of Andorra, the vast majority of students are men, while on the contrary, the almost totality of students doing Nursing studies are female. The rate of male/female students at the University of Andorra is approximately 40%/60%.

The age groups have been determined manually trying, as much as possible, to have similar groups in terms of best users submitting samples. This will be relevant when performing tests related to age group and gender. In order to perform such tests, the moment users were separated by age group or gender, the number of suitable users in each group decreased dramatically. To ensure similar groups in terms of quality of the samples this age group separation is proposed: Young users, from 18 to 34 years; Middle age users, from 35 to 45 years; Senior users, from 46 to 69 years. These groups have been labeled $(18, 34]$, $(34, 45]$, and$(45, 69]$, respectively.

Age group	Men	Women	Total
(18, 34]	96	186	282
(34, 45]	40	68	108
(45, 69]	42	39	81
Total	178	293	471

Table 4.1: Users in the dataset, totals with age group and gender separation

Total number of forum posts (sessions)	10,649
Average number of posts per user	22.60
Median number of posts per user	8
Standard deviation of posts per user	52.19
Total number of events	7,440,935
Average number of events per forum post (session)	698.74

Table 4.2: Session information from the dataset used in this study

As per the number of captured keyboard events, Table 4.2 shows a statistical summary of the collected information. These numbers, even if informative, are highly misleading because they come from the average of all collected events from all 471

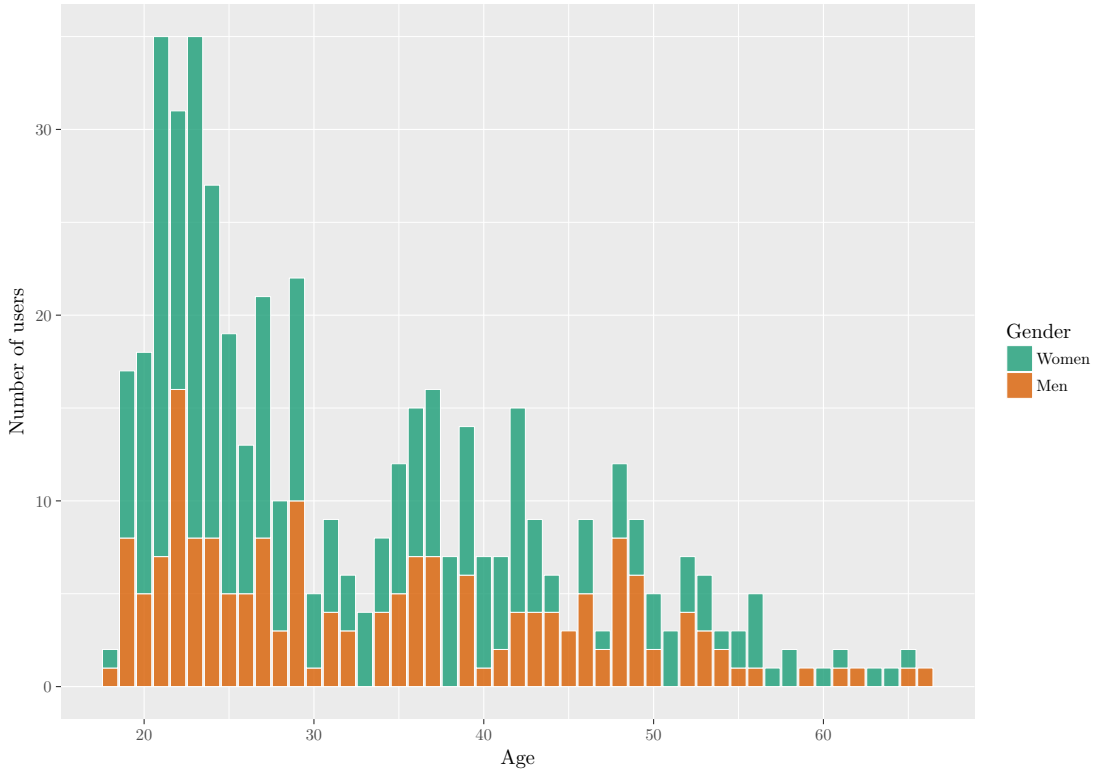

Figure 4.2: Users distribution

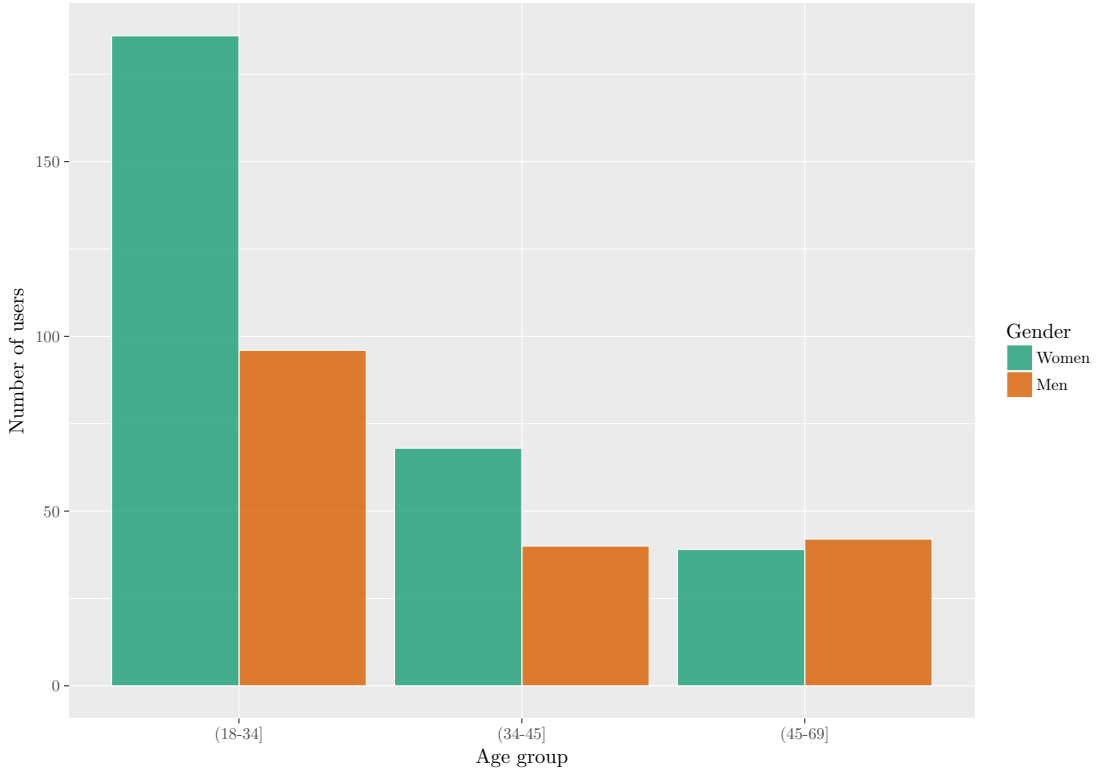

Figure 4.3: Users distribution by age group

users. Figure 4.4 shows, in comparison, the real distribution of the first 60 ranked users from the number of events submitted. This figure also shows the difference between men and women. Again, it can be seen that women are predominant in the group of users that sent most information, and from these, there is a great difference between a small group of ten users that submitted more than $100K$ events.

When choosing a group of users to perform the experiments with, it was argued whether it would be better to focus only on those users that had sent most events regardless of the gender, thus having an abnormally biased set towards women produced samples or, on the other hand, have an equal number of men and women samples even if this meant discarding better suited users and using many others that had not sent so many samples. In Figure 4.5 a representation of what is meant by this can be seen. The differences are obvious when compared to Figure 4.4.

In the end, it was decided to focus solely on the number of events as a measure to determine the best users to perform the tests. It was thought that, in real life, the chances of having a perfectly distributed set would be very low and since one of the aims of the study was to perform it in an as real as possible environment this approach was thought optimal.

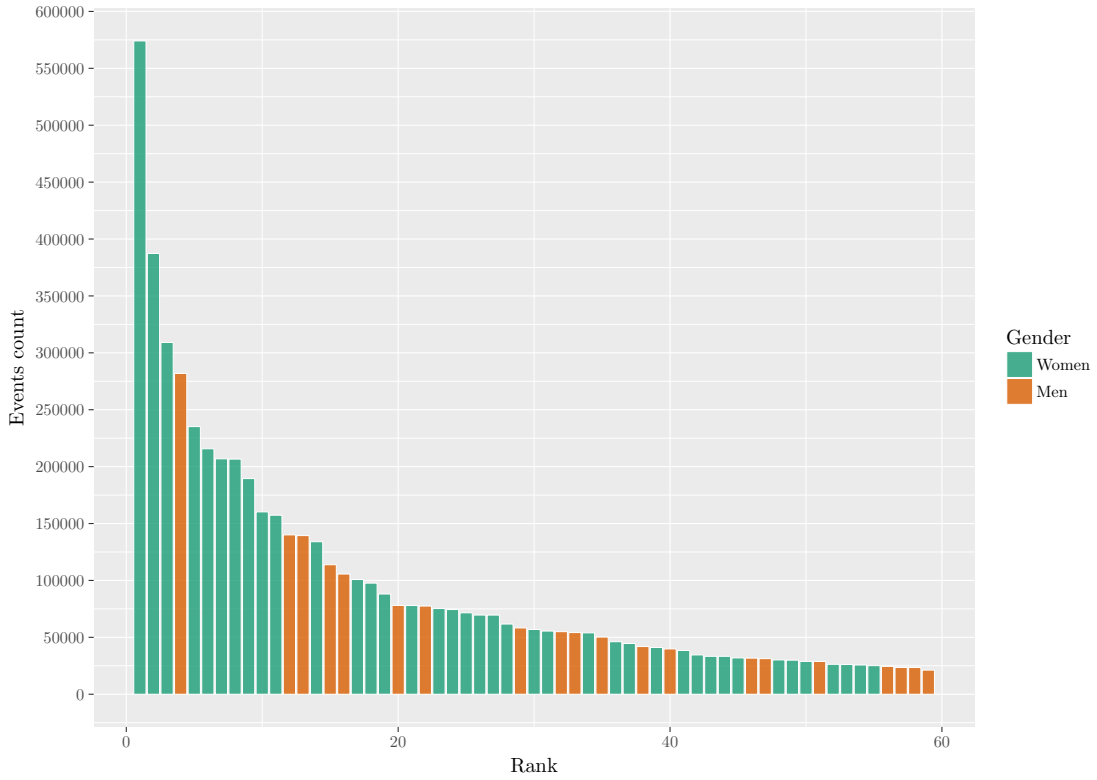

Figure 4.4: Ranked users from the number of submitted events

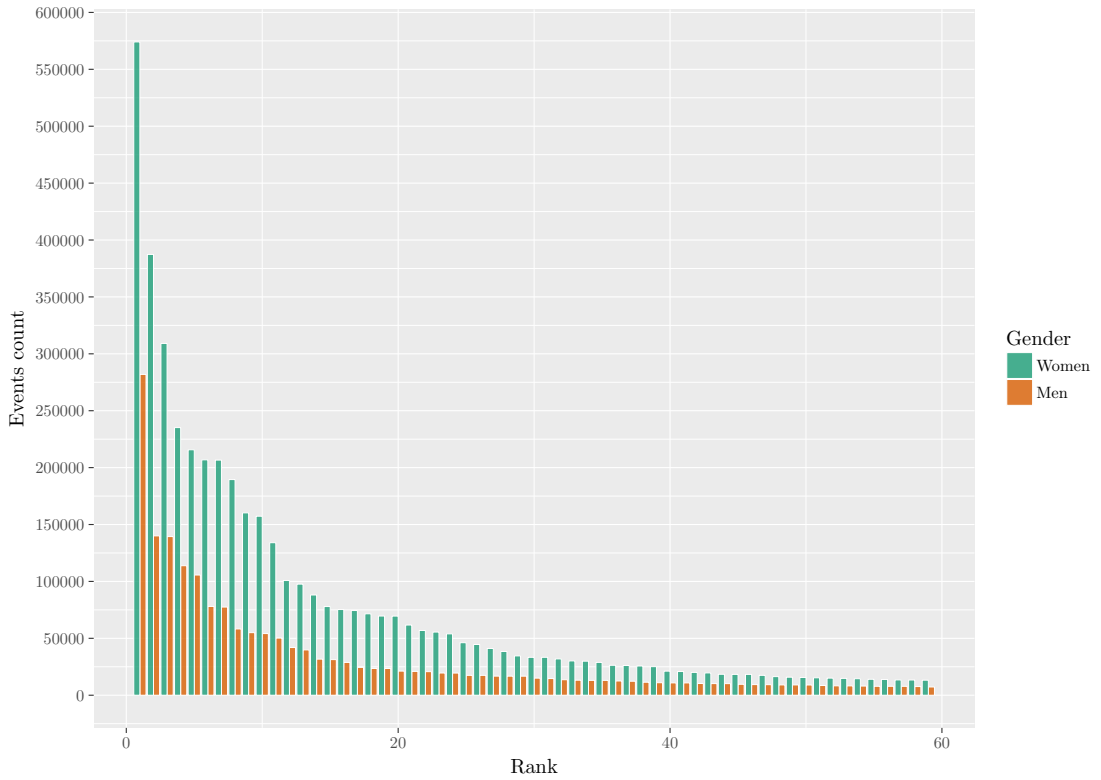

Figure 4.5: Ranked users from the events submitted grouped by gender

4.2.5 Selecting users and groups

Even if 471 users had submitted unique messages, many of these posts (or sessions, as these are known in this document) were of little use since the number of events, or even the number of words, was not above a certain relevant threshold. This threshold has never been established in the strict sense of the word, on the contrary, most of the time tests are performed choosing from the pool of the users that have submitted the largest number of events. Also, it could well happen that a user prepares the post offline and then pastes it into the Discussion forum module. In this case, the only sequences captured is that of the CTRL+V keys. Other similar behaviors have also been detected and treated accordingly or, otherwise, the samples have simply been discarded.

For this study, users have also been separated into periods, depending on the date the samples were submitted, and also, the 60 best users from each period have been selected, always taking into account the number of events submitted.

Periods

To perform the experiments, the information available on the keystroke database has been partitioned into four different periods. Each of these four periods corresponds to the following partitions:

- Period 0 ($P0$): All the available data, no partition whatsoever
- Period 1 ($P1$): Data belonging to the Autumn 15–16 semester
- Period 2 ($P2$): Data belonging to the Spring 15–16 semester
- Period 3 ($P3$): Data belonging to the Autumn 16–17 semester

These periods have a specific goal in mind: to test the samples both in the real setting where these were collected, without mixing sessions from different semesters, and to have a global period of samples that can be compared to having a larger dataset from a single period. This larger dataset *could* have come from an environment where much more samples had been submitted in a single semester. The possibility of comparing the results from different periods where users, session count, and events are very different is thought to give statistical relevance to the results.

Age group and gender users per period

Table 4.3 shows user distribution regarding age group and gender after the 60 best users of each period have been selected. It can be seen that women, again, are the most prolific users. In general, all age groups are more or less equal with a skewness towards the Younger age group. Also interesting to comment upon is that some groups present a rather low number of users. Such is the case, for instance, of the number of male users in the $(34, 45]$ age group of Period 1, where only 2 users are available. This could affect the accuracy and statistical relevance of the tests performed with these groups. One of the reasons different Periods are used is to have examples of all possibilities and, ensuring that there are always large enough group, that results are relevant.

The fact that some groups are rather small has been taken into account when the age group and gender experiments have been carried out. In the results of these experiments, the margin of error related to sample size has also been included. The formula that has been used to determine the margin of error with a 95% of confidence ($z^* = 1.96$) has been the following:

$$ME = 1.96 \cdot \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

Initial groups based on the number of events

Groups of valid or suitable users based on the total number of events stored in the database have been selected for the initial tests. The initial research, focused on determining the effects of the quality and size of the model, is performed using 3 groups

Period	Age group	Men	Women	Total
	(18, 34]	8	14	22
$P0$	(34, 45]	4	15	19
	(45, 69]	8	11	19
	(18, 34]	14	11	25
$P1$	(34, 45]	2	12	14
	(45, 69]	10	11	21
	(18, 34]	7	16	23
$P2$	(34, 45]	5	16	21
	(45, 69]	8	8	16
	(18, 34]	9	18	27
$P3$	(34, 45]	6	9	15
	(45, 69]	11	7	18

Table 4.3: Users in the selected periods, with age group and gender separation

of 20 users each. The number of users per group is something that was largely discussed. After reading some articles where this issue had also been studied [36, 92], it seems obvious that the smaller the group the better the results are. Trying to be as close to reality as possible, a group of 20 users is though to be ideal to be able to have 3 full groups with enough samples to build reasonable models.

The first group (identified as A) has the users with the largest count of keyboard events. This one is known as the one with the *Rich* models. It is thought that having so many events available will yield the best models and, at the same time, the best results. The second group (identified as B) has, what is called, *Normal* models since the number of sessions and events is somewhat in between groups A and C. At the bottom of the scale is the third group (identified as C) where the number of keyboard events is the lowest. This is known as the group with the *Poor* models. Of course, *a priori*, it seems that this group will yield the worst results. Table 4.4 shows the total number of sessions and events for each of these groups.

It is important to note that the data of the three partial periods in Table 4.4 does not add to the total data in Period 0 ($P0$). This is because, for each period, the relative 60 best users to the period are used, and these are not always the same due to students' rotation or varying Virtual Campus usage per period.

Table 4.4 shows another interesting fact. When compared to the average number of sessions shown in Table 4.2, it can be seen that there is a big difference when all users are used to get this value and when only a small group of 20 users is used. Seeing that the 20 best users have an average of almost 200 sessions per user comes to show

Group	Period	Sessions	Avg. Sessions	Events	Avg. Events
A: *Rich* models	$P0$	3,966	198.30	3,921,156	196,057.80
	$P1$	987	49.35	773,317	38,665.85
	$P2$	1,815	90.75	2,720,176	136,008.80
	$P3$	1,594	79.70	1,102,861	55,143.05
B: *Normal* models	$P0$	1,492	74.60	1,174,928	58,746.40
	$P1$	396	19.80	260,011	13,000.55
	$P2$	754	37.70	499,628	24,981.40
	$P3$	458	22.90	397,916	19,895.80
C: *Poor* models	$P0$	694	34.70	569,305	28,465.25
	$P1$	159	7.95	127,356	6,367.80
	$P2$	330	16.50	229,777	11,488.85
	$P3$	405	20.25	247,799	12,389.95

Table 4.4: Initial user groups

that, on the other end, there are many users with a very small number of submitted sessions. This poses a dilemma when it comes to implementing the proposed methods: from which moment should models considered robust enough to compare new samples against? How many events should be used to create these models? Unfortunately, at this point, no easy answer can be given. Figure 4.6 depicts the number of events from the selected users per period. Again, important differences are detected regarding the number of events per period and group. In general, Periods 0 and 2 ($P0$ and $P2$) have the most number of events. At the other side of the spectrum Periods 1 and 3 have a rather low number of events.

As per the size of the sample when performing the initial quality of the model test, as is described later in this document when talking about the cross-validation technique used, 70% of the sessions have been used to build the models and the other 30% have been used to test the models and obtain the results. To have an idea of what this means, from Table 4.4, from Period 0 and Group A, approximately 2,776 sessions have been used to build the models, while the remaining 1,190 have been used to train the system. The same procedure has been used for all periods and all groups.

Random users

After the initial research, useful to determine if the size of the model has an impact to the number of correctly identified sessions, a different methodology to select users has been taken, with the aim of bringing it closer to real-life situations.

Instead of having groups of 20 users based on the number of events submitted, 40

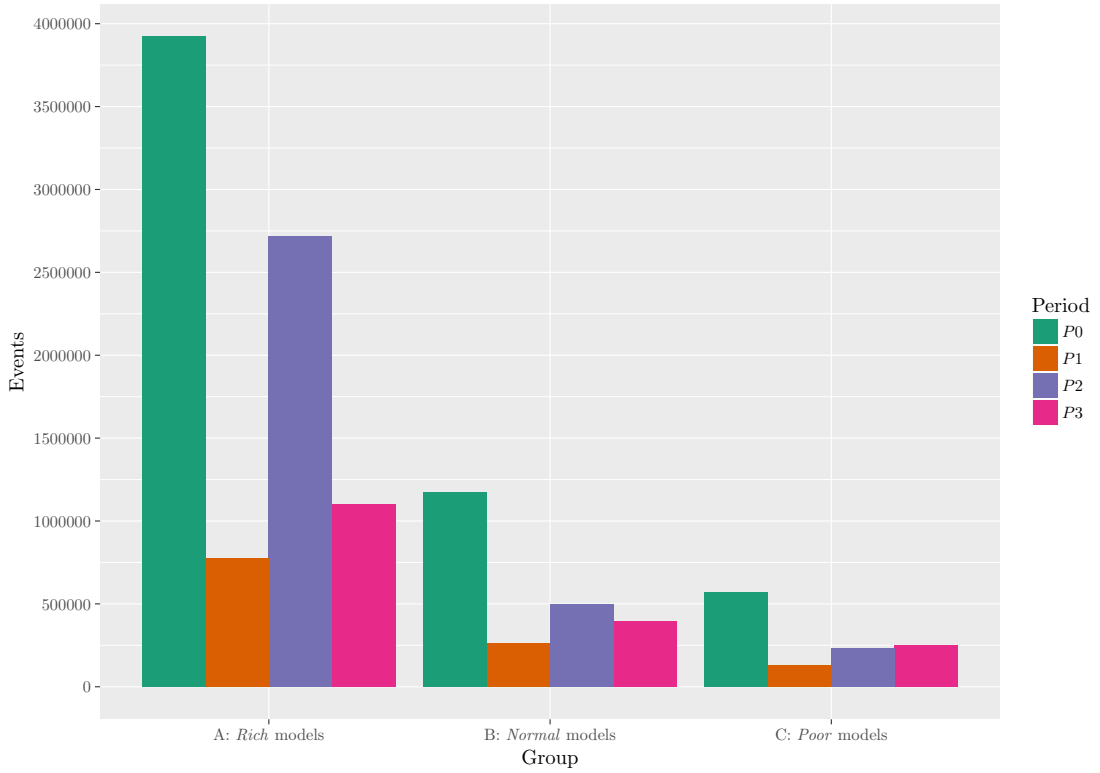

Figure 4.6: Initial user groups per period

random users are chosen among the 60 best from each period. Table 4.5 shows the information on the number of events and sessions per period when no group separation is performed. With this group of users, it is thought that the results give more relevant and realistic results since no classification based on the quality of the model is done, and also, the number is more in concordance with other studies in the State of the Art. This group of 40 users is used not only to identify users but also to test the possibility of authenticating users.

Period	Sessions	Avg. Sessions	Events	Avg. Events
$P0$	6,152	102.53	5,665,389	94,423.15
$P1$	1,542	25.7	1,160,684	19,344.73
$P2$	2,899	48.31	3,449,581	57,493.02
$P3$	2,457	40.95	1,748,576	29,142.93

Table 4.5: Sessions and events from the best 60 users per period

Finally, a test has been performed to determine what is the best group size. Different groups of users have been chosen randomly, with group size values ranging from 2 to 60 users. When performing this experiment, and when having small groups of users, the number of compared sessions has been rather low. As is the case of the age group and gender groups, the margin of error values are also included in the results to be

able to have a good idea of the statistical relevancy of the presented results.

Regarding the size of the sample when performing these tests, 70% of the sessions have been used to build the models and the other 30% have been used to test them, as previously explained. In this case, this means, from Table 4.5, from Period 0 and when selecting 40 users, approximately ⅔ of 6,152, that is 3,691 sessions have been used. From these 2,583 have been used to build the models, while the remaining 1,107 have been used to train the system. The same procedure has been used for all other periods. Of course, these numbers are just an approximation because not all users had submitted the same number of sessions, but it serves as an example to put things in perspective.

The test that tries to determine the best size of the groups also includes the margin of error values for each sample size. These values have been used in this test, and also in the test regarding age group and gender, to demonstrate the effect of the number of sessions used in the margin of error values, and to have a better understanding of the results portrayed. These margin of error values have not been shown in other tests in order to keep results clean and readable.

4.3 Model description

This section focuses on the definition of the logical tree models that are proposed in this research. These tree models contain, not only the timing intervals of every keyboard event, but also the contextual information of where a particular event has happened in a word.

4.3.1 Interval analysis

The analysis of a session consists in working with the different KD and KU events (Press and Release, respectively) and finding the timing intervals between successive events. This also allows the possibility of finding the information of the Press–Release (dwell time or PR) and Release–Press (flight time or RP) intervals for every pressed key, as well as the Press–Press (PP) and the Release-Release (RR) information.

The process of detecting words is performed taking two features into account: known delimiters (see Table 4.6) and a maximum time interval of silence. These stop-keys have been selected among the most common in the Catalan language. When studying if mistakes are a relevant behavioral feature, the code 8 key (the *backspace* key) will be removed from the list. The use of the *backspace* key will then be included as part of the models to see if users always type combinations of letters that include the mistakes and, at the same time, the steps to correct them. Another experiment, also related to

behavioral features, that has been performed is to only leave the space as the unique word delimiter. This way, all key combinations are taken into account and used to build the model. This has been tried to see if the use, for example, of navigation keys, improves the percentage of correctly identified sessions.

The silence threshold has been set empirically at $300ms$ and is left as a parameter to be studied in the future. It could be discussed if this value should be set specifically for every user.

Key	Key Code
Unknown	0
Backspace	8
Tab	9
Enter	13
Caps Lock	20
Escape	27
Space	32
Navigation keys	33 – 40
Insert	45
Delete	46
Colon	186
Comma	188
Period	190

Table 4.6: Word delimiters (stop-keys)

Figure 4.7 shows an example of the time intervals described for the words: *THE SUN*. The first word (*THE*) is formed by the following dwell intervals (PR): $PR_1 = 54$, $PR_2 = 28$ and $PR_3 = 18$. The flight intervals (RP) are: $RP_1 = 25$ and $RP_2 = 5$. When a word separator is detected (a *space* key event, in this case) the intervals of that event are discarded. RP_3, PR_4 and RP_4 in this case. The second word (*SUN*) is formed by the following PR intervals: $PR_5 = 32$, $PR_6 = 38$ and $PR_7 = 28$ and of the following RP intervals: $RP_5 = 29$ and $RP_6 = 33$.

It is easy to see that from this information other combinations like PP, RR or whole *wordgraphs* intervals can be also easily obtained. The first word would have two PP intervals: $PP_1 = 79$ and $PP_2 = 33$, and also two RR intervals: $RR_1 = 53$ and $RR_2 = 23$. The second word would have two PP intervals as well: $PP_5 = 61$ and $PP_6 = 71$ and two RR intervals: $RR_5 = 67$ and $RR_6 = 61$. The two *wordgraphs* in this example are: $wg_1 = 130$ and $wg_2 = 160$.

Any N letter word has N PR intervals and $N - 1$ RP intervals. One-letter words do not have PP, RP, or RR intervals, only PR time intervals.

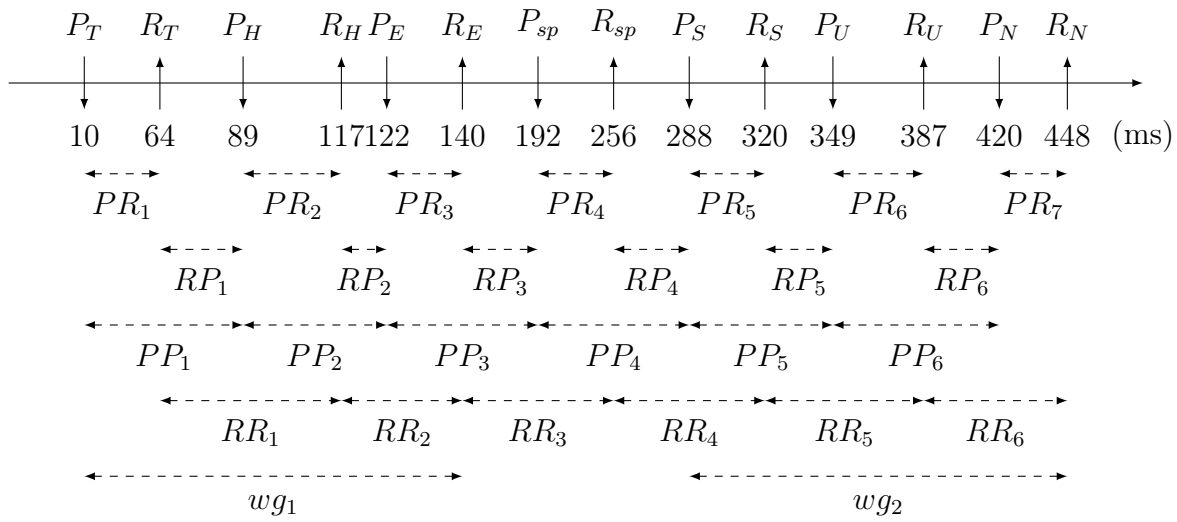

Figure 4.7: Time intervals for the words: *THE SUN*

4.3.2 Straight tree model

All detected words have been stored in a logical tree model like the one shown in Figure 4.8. In this example, the following words have been added to the tree: *ALL*, *ALBERT*, *THE*, *THERE*, *THIS*, *WORD* and *WIT*.

As can be seen in the figure, each of the nodes *can* have PR and RP (first and second list respectively) time intervals information. A node will have this information or not only if the user has typed that particular *whole* word. The timing information is always stored on the node corresponding to the last letter of the detected word. This is a very important characteristic of this model and the one that allows the possibility of studying the context reliably.

If a word is detected more than once there will be a different PR and RP list for each instance of the word (i.e. *ALL*). If a detected word is a sub-word of an already stored word there will be PR and RP timing information in a non-leaf node (i.e. *THE* – *THERE*).

This model is highly versatile. Even though some information may be lost regarding the number of times a particular *n-graph* has been typed, the information regarding the position of every typed letter in respect of a word is gained.

4.3.3 Inverted tree model

The Straight tree model shown in the previous section stores the information from the beginning to the end of each word. Close to the root of the tree are the letters at the beginning of each word and the tree grows in depth as the words increase, also, in length. As an example, the word *THIS*, present in the tree model in Figure 4.8, has a

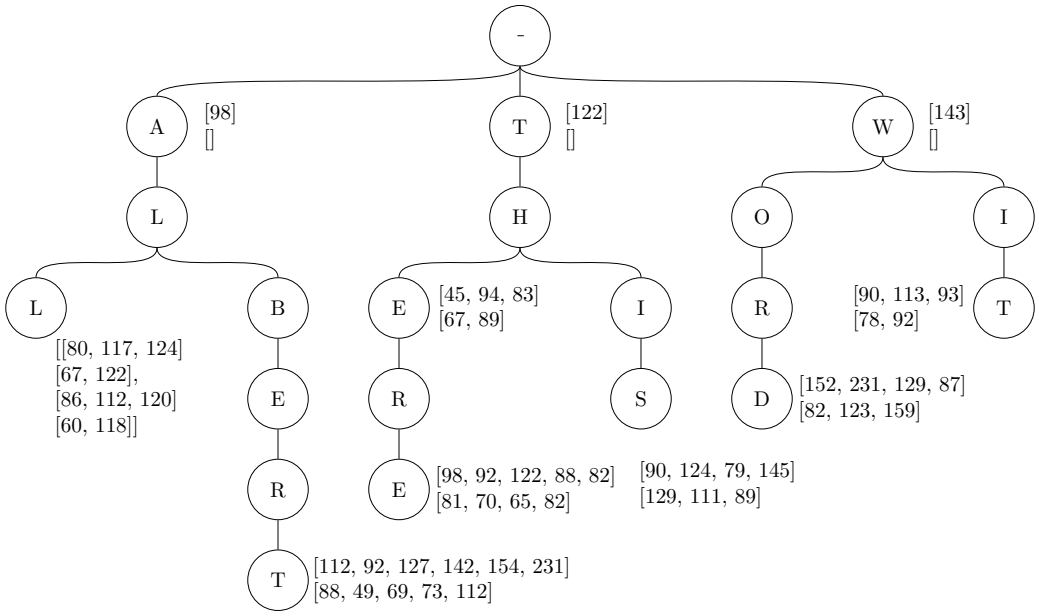

Figure 4.8: Straight tree model

depth of four letters, and the depth increases with each letter following the letters of the word: $T_1 \rightarrow H_2 \rightarrow I_3 \rightarrow S_4$.

Another model that has been used in this study is an Inverted tree model. The tree structure is the same but, in this case, the model is built from the end to the beginning of the words. Using the same example, the word *THIS* is stored as *SIHT*: $S_1 \rightarrow I_2 \rightarrow H_3 \rightarrow T_4$. The first letter S is the closest to the root of the tree. The same methodology is used for all other words inserted into the tree (see Figure 4.9 for an example). In this Inverted tree model the following words have been added: *T*, *THIS*, *HIS*, *IS*, *AXIS*, *ROBERT*, *ALBERT*, *WORD* and *CARD*.

As expected, the words that appear in both trees have the same time intervals but these are inverted in the lists on the node where they are attached to. For example, the words *ALBERT*, *THIS* and *WORD* are in both trees.

It has been observed that when comparing sessions against the Straight model many words are found only to a certain depth because the user has, previously, typed a different word with the same root letters. It is normal to discard a lot of information from the endings of these partially found words. The idea to use an Inverted tree model is to be sure that most of the contextual information available is properly used.

4.3.4 Combined tree model

Each tree model, either Straight or Inverted, has its own logical structure in computer memory. These two tree models are built separately and each word from a new session is compared to the Straight tree model and also to the Inverted tree model after having

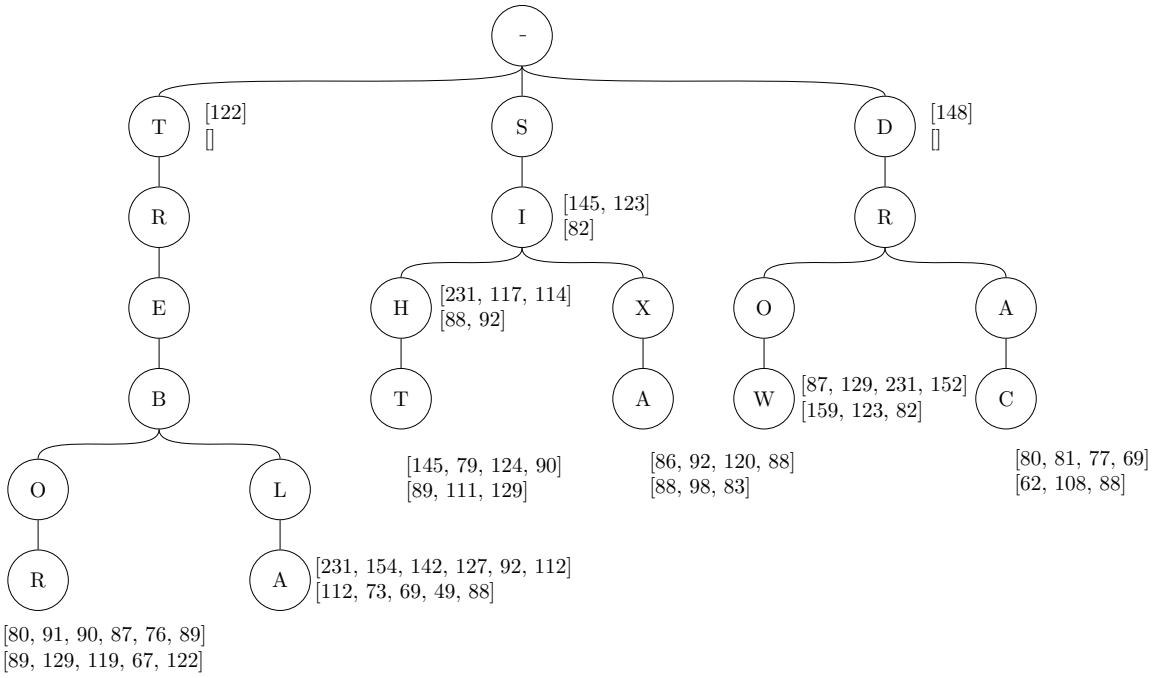

Figure 4.9: Inverted tree model

the source word reversed. For example, if the word *HOUSE* is detected in a testing session the word is searched in the Straight model normally, but on the Inverted model the searched word is: *ESUOH*.

Searching a word in either model returns the distance from the origin word to that stored in the tree, always up to the deepest node of a word with the same root letters. If a word is completely found in the tree, the Combined tree model has two identical values, one for each logical tree. A column in the results shows if the distance measurement has been obtained from the Straight or the Inverted tree. An example of this can be seen in Section 4.4.4.

At the moment of determining the owner of a session this column is used to filter the data returned from the Straight tree model only, the Inverted tree model only, or by discarding this column altogether, use the combined information from both models.

4.3.5 Forest of trees model

Most of the work done in the field of free text involves one form or another of categorizing the samples gathered into *n-graphs* frequency dictionaries, and these, into groups based on the entries submitted by the users. Following this methodology, each user would have M models from each of the M messages or sessions submitted. Each model M_i would have a collection of *n-graphs* in a sort of dictionary structure.

To be consequent with previous research the same methodology has been attempted in this study. The followed methodology is to build a Forest of trees model using a

tree (like the ones previously described) for each of the sessions instead of having a unique tree with the information from all sessions from a user. New testing sessions are compared to this Forest of trees. It has been argued that having smaller trees in a forest can improve the global accuracy of the system.

4.3.6 *n-graph* frequency model

Another completely different model that has been used, in order to compare the use of contextual vs. non-contextual models, is that of an *n-graph* frequency model. Since most of the research performed on Keystroke Dynamics focuses on some sort or other of *n-graph* dictionary it was though that a comparison between both methodologies had to be attempted. Like in the case of the Forest of trees model, this one has been used to be able to compare results with previous research available.

It is worth noting that the efforts have not been set on improving existing *n-graph* frequency methods but only to evaluate if, by using contextual features, the results imply a step backwards in terms of performance, accuracy and reliability, or on the contrary, these yield and improvement.

Compared to the tree models (Straight, Inverted and Combined) and the Forest of trees model previously described, building an *n-graph* model is a rather simple task.

The basis of an *n-graph* model, in the case of this research, is a dictionary of graphs. These graphs can be of any n length. Digraphs, trigraphs and 4-graphs have been used. An example of this logical structure, applied to digraphs, would contain the following for the word SUN:

$$(S, U) = (83, 85) \rightarrow [(87, 54), 98]$$
$$(U, N) = (85, 78) \rightarrow [(122, 165), 76]$$

In this particular example, the key codes for the letters S, U, and N are 83, 85, and 78 respectively. The list, pointed by the arrow are the dwell times $(87, 54)$ for S and U and the 98 corresponds to the flight time between these to events. The same can be said for the pair U and N: the dwell times are $(122, 165)$ and the flight time are 76.

In the case where more than one instance of a graph is found, as it is common, the list of dwell and flight times increases without limitations.

4.4 Testing the models

This section outlines the methodology used to perform the different tests on the available keystroke dataset. Each of these tests focuses on a particular feature of the proposed logical tree model and on the different alternatives evaluated to identify or authenticate users. The experiments carried out fall into one of these categories:

- Size, quality of the model and searching parameters
- Behavioral features
- Comparing new samples to the model
- Determining the owner of a session
- Authentication
- Age group and gender particularities

The rest of this chapter is focused on detailing the different methodologies used in each of these areas in greater depth, as well as, at the end, the cross-validation technique that has been used to ensure statistical relevance.

4.4.1 Size, quality and searching parameters

The main goal behind the research presented in this study is to determine if contextual information is useful when identifying or authentication users. At the same time, some behavioral features highly related to Keystroke Dynamics are also studied. Many features, parameters and configurations have been evaluated. These affect greatly how the models are built and how new testing samples are compared to these models. The parameters described below are grouped by their effect on the process of building and analyzing new samples against the logical tree models.

Type of model

These are the different types of models that have been proposed earlier in this chapter:

- Single tree model: Either Straight or Inverted.
- Single combined tree model: Combining the distance measurements obtained from searching both the Straight and the Inverted tree models.
- Forest of trees: Using a single tree for each session. These could be, again, either Straight, Inverted or Combined.

Choosing a model will not only have a relevant effect on performance and optimization, but will also help determine how relevant contextual information is. The idea of using *all* available information will be recurrent in different experiments. In such cases, using Inverted trees could help or not in establishing the most accurate procedure to identify users.

Removal of *outlier* values

When users type on a keyboard they are not always consistent. Many external factors can affect their performance. Since the proposed logical tree models accumulate each of the instances of a typed word, it is normal that some of these may be way off the *normal* rhythm of a user. When using these *strange* samples the distances obtained could be highly affected. In previous research, it has been observed that if *strange* samples are discarded from the models results may improve. The question is what method has to be used to discard samples in the newly proposed models. In the end, it has been decided to use a method that discards samples based on the number of standard deviations the sample is outside of the mean value of all available samples per word. This will be, then, one of the parameters related to model building.

Other possibilities that deal with adapting the model like discarding old samples or keeping only the best (even when incorporating new instances of a word into the models) have, unfortunately, not been tried. The possibility of adapting the models over time is left as future work.

Model size and quality

The following parameters may affect model size and quality of the time intervals stored:

- Number of words allowed in a model: This parameter tries to establish the effect of limiting the maximum number of words allowed in a model, creating abnormally small templates. A performance or scalability driven scheme could suggest or force doing so.
- Number of instances per word in a model: Related to the previous parameter, this one tries to establish the effect of limiting the number of instances of a previously detected word in a model.
- *Outlier* detection and removal: Once different instances of a word have been added into a model error may increase due to *outlier* values. The process of removing these *outlier* values is evaluated to see if more precise and accurate models can be built.

To test these parameters an incremental process is used, allowing the models to increase in size progressively evaluating the accuracy in each iteration. This is explained in depth in the results chapter, more specifically, when performing the first experiment.

Session evaluation

Once the selected models have been built with the training samples it is possible to compare testing sessions against them and try to establish the owner of these sessions. The typical process is to try to find every word from the new training session in the models and obtain the distances between them.

When following this process one of the following situations would be encountered:

- The word is not found in the model. The word is simply discarded. In this case, the possibility of having *not found* words punish the owner of the session was initially thought of, but later discarded. It was argued that a user would probably use a common set of words.
- The word is completely found in the model and the last letter is that of a leaf node in the tree, or contains timing intervals information. The distance between the two samples can be immediately obtained.
- The word is partially found but the node in which the last letter of the word is found does not have time intervals information because this is the first time the user has typed this particular word. In this case, there are two choices: the word can be simply discarded and treated as if it had not been found or partial timings from the leafs hanging from the node of the last letter can be determined. These two options have been evaluated to choose which method behaves better.
- The word is partially found in the model but there are still letters left because the user has only entered shorter words with the same root letters in the past. In this case only the timings of the partial word found are used. Also, the partial sub-word not found still contains keystroke information that may be useful. How this information is to be treated is also a matter of study in this research. Three different options have been studied regarding the level of recursion used:
  - Search the partial sub-word again in the model as if it was a whole new word. If not found, the first letter of the partial sub-word is discarded and the process is repeated again until all letters have been used or a sub-word has been found. This method uses the highest level of recursion and is also the most exhaustive.
  - Search the partial sub-word again as if it was a whole new word and discard it if not found. Only partial recursion is used.

– Discard the sub-word. No recursion is used whatsoever.

A possible alternative to deal with non-found words would be to use recursion as is the case of partially found words. This methodology could suggest the possibility of discarding letters at the beginning of words and try to find the remaining sub-words. This method has not been evaluated in the current study and could be suggested as future work.

Parameters related to session analysis

The following parameters may affect session analysis when words are searched in the logical tree models:

- Length of found words: This parameter analyzes if the length of a typed word is relevant and if all lengths have the same importance. This is of interest, not only in terms of performance and model optimizing, but also to determine if users have a natural tendency to be more consistent in their typing during a limited number of keystrokes. The study of this feature was suggested by the idea related to fragmentation that Bor stated in his study where short pieces of information are easier to remember [23].

- Recursion when searching partial sub-words: This parameter has been described in Section 4.4.1. The effect of using different types of recursion when searching partial words is analyzed with this parameter.

- Discard child times: When a word is found only up to a certain depth, and when that node has no time intervals information, the intervals can be obtained from the leafs from that particular node. Independently to the fact of using recursion or not, this parameter discards these words if the node where the last letter has been found has no timing information.

- Number of words found in the model: Limitations may appear when the number of words in a session is too low. It can happen that a user only accesses the Discussion forums to contribute with a few words. At the same time having abnormally small models can lead to incorrect identification because the template of a user does not have enough information to be properly defined. This parameter tries to mitigate this problem by establishing a minimum number of words to be found to consider a session valid.

When dealing with *n-graph* models and Relative and Absolute distances the following parameters have also been analyzed:

- Minimum number of graphs: Similar to the minimum number of words found, if not enough graphs are found on the dictionary the session is discarded.
- Maximum number of sessions to compare to: This parameter has to do with performance and scalability problems related to Relative and Absolute distances. These methods can be very resource demanding and comparing a testing sample to an unlimited number of sessions can render the method unfeasible.

A practical example

As an example of the different situations that can be encountered when searching words in the model, Figure 4.10 shows a possible logical tree with some words in it, similar to the Straight tree model previously shown. The tree depicted in this figure can be helpful to understand how words are searched, how recursion is performed, and also how the concept of discarding child times has been implemented.

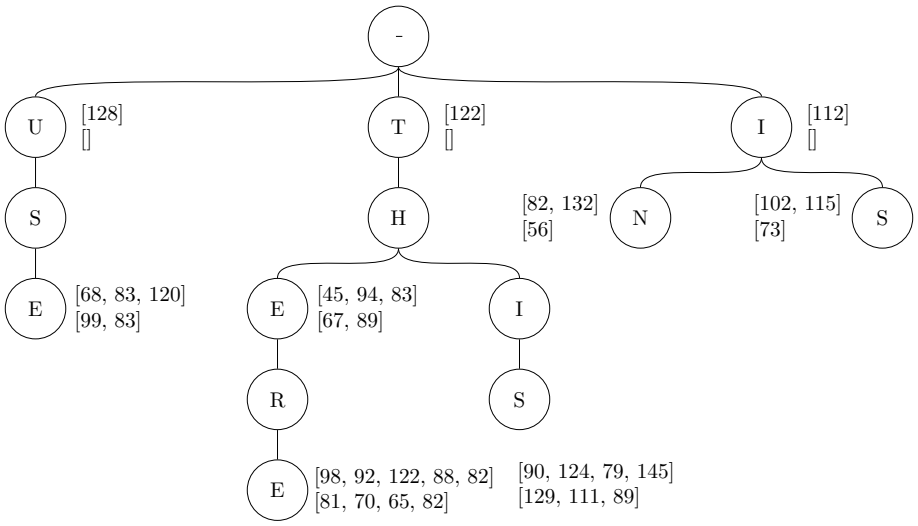

Figure 4.10: Straight tree model for the practical example

Let us suppose that the following words are searched in the tree model: *HOUSE*, *THIS*, *THUS*, *USEFUL*, and *THEREIN*. The easy cases are those where words are not found in the model (like the word *HOUSE*) and those words that are *exactly* found (like the word *THIS*). In the first case, the word *HOUSE* would be simply discarded. No recursion is ever performed on those words that not even a single letter has been found. This is a decision that could be revised in further studies. In the second case, the word *THIS* does not present any problems at all. The values returned are those present in the node of the last letter ($[90, 124, 79, 145], [129, 111, 89]$). If more than one instance of timing intervals is available in the node, the mean value of all timing intervals is returned.

The rather more interesting cases are those when words are partially found. The

word *THUS* can be found in the model using different strategies. The letters T and H are found successively hanging from the root but, from this point, the letter U is not found. Let us stop here before deciding what to do with the tail of the word not yet been found. For the two initial letters, different approaches have been evaluated in this study. The first is to simply discard the partial letters found because in the node where the last letter has been found (H) there are no timing intervals. This would be considered as a not found word. The other approach has been to use the information from the leafs hanging from this node. In this case there are timing intervals from the words *THERE* and *THIS*. These last letter nodes of each of these words have timing intervals of five- and four-letter words respectively. In the case of the sub-word *TH* only timings of the first two letters are necessary. These would be: $[98, 92][81]$, and $[90, 124][129]$ respectively. From these two the mean value is returned: $[94, 108][105]$. It should be noted that the timing intervals in the node for the letter E (part of the *THE* stored word) are not used, only those in the leafs. This was a decision taken during the implementation that could be revised in further studies.

For the two letters still left to be found (US), depending on the recursion setting, the algorithm could perform a new search, again from the root, using only these letters. Of course, if no recursion is used whatsoever, these two letters would be discarded. In the case when recursion is used, these two letters are found in the word *USE*. Again, the methodology to treat nodes without timing intervals would have to be evaluated. If child times are used, then the returned timing intervals would be: $[68, 83][99]$, those from the leaf node (E), using the information of only the first two letters. If child times are discarded, then, since the S node has no timing intervals, this sub-word would be discarded.

The following word to be analyzed is *USEFUL*. In this case, in the tree, the successive letters U, S, and E are found. This sub-word would return the values stored in the E node: $[68, 83, 120][99, 83]$. The letters F, U, and L are still left to be found. Without recursion, these would be simply discarded and only the timings of the partial sub-word *USE* would be used. Using partial recursion, the new sub-word *FUL* would be searched in the model, and it can be seen that it would not be found, thus becoming a not found word. The process when using partial recursion would end here. On the contrary, when using the exhaustive recursion method, the first letter would be discarded, leaving a new sub-word to be searched (UL). Again, this word is not found in the tree model. The process continues with the last sub-word (L). This one is also not present in the model, so it would be discarded too. It can be seen that the exhaustive recursion method is the one that requires more computer resources and the slowest one.

The next word, *THEREIN*, is interesting because it is pretty similar to the case of *THUS* but in this case the sub-word is found when all letters present in the

tree model have been exhausted from a shorter word. The sub-word *THERE* is found completely and the last node has timing information that would be returned: $[98, 92, 122, 88, 82][81, 70, 65, 82]$. The rest of letters (*IN*), using recursion, would be searched again in the model. In this case these are completely found too. The algorithm would then return the values: $[82, 132][56]$. If no recursion is used the sub-word *IN* would be discarded.

Other particular situations could be encountered when searching words in the tree model, but the ones described in this section are the most common ones and should help understand how searching the model works.

Even if this example has only focused on the Straight tree model, the procedure to search words in the Inverted tree model is exactly the same. The only particularity is that source words are inverted previous to being searched in the tree.

4.4.2 Behavioral features

This section comments upon the behavioral features that have been analyzed in this study, and how the process of detecting and using them has been performed.

Mistakes users make

One of the objectives of this research is to determine if the mistakes that users make are an influential feature when it comes to identifying them. This section details how these mistakes are treated and what tests have been performed in order to see if the overall results improve when this feature is considered.

A typical word has L letters, and usually, users *always* type it in the same manner, and with the same rhythm. This is the basis of Keystroke Dynamics. The idea behind the feature that evaluates the mistakes is that this also includes the words in which users make mistakes. For example, if a user types the word *WEIGHTED* and as a rule of thumb they always invert the T and the H, the user could type the word using the following sequence of keystrokes ($\leftarrow$ being the *backspace* key):

$$W\ E\ I\ G\ T\ H\ E\ D \leftarrow \leftarrow \leftarrow \leftarrow H\ T\ E\ D.$$

This is just one of the ways a user may correct the mistake. Users can always use the mouse and right-click the word to let the application auto-correct functionality fix it. In this case, there is nothing that ends up in the proposed models since no keys are pressed.

The *backspace* (key code: 8) has been defined as a word breaking delimiter. When it is detected it is automatically discarded and when new, different, letters are found again these are treated as part of a new word. The process described before would not

end up in the tree model as a single word but as two: *WEIGTHED* would be the first detected word and *HTED* the second. In no way, the correct word is ever stored in the model. This also means that all sub-words coming from mistake corrections that share the same letters do not end up in the same branch of the tree without having real relation to the context of the written words. With this in mind, what is evaluated is the possibility of removing the *backspace* code from the list of word delimiters.

The idea that different types of users, be it in gender or age group, type rather differently suggested the possibility of also evaluating the proposed methodology to treat mistakes separating user per age group and gender. An experiment is specifically dedicated to this idea.

Other behavioral features

Other features related to user behavior that have been analyzed are the following:

- The possibility of using particular keys as word delimiters gives also the possibility of studying the effect of only using the space key as a word delimiter. With this parameter, all other key codes are used to create the models. This means that not only words are used in the models but also, for example, navigation keys or other particular key combinations.
- Frequency scaling: This parameter allows the possibility of studying the effect of giving more importance to those words that users use the most. The distance between a word from a new sample and the one stored in the model can be scaled to a better value if that particular word is frequently used.
- Successive words scaling: Using the same idea, this parameter analyzes the effect of the use of successive words. The idea is to study if the use of common constructions of sentences is a differentiating feature. If a word is followed by another one in a new sample and the same construction has also been used when building the model, then the samples involved can also be given more relevance in the distance measurements.

Extending the models to include behavioral features

Adding information related to behavior means that models have to be adapted to store such data. Different approaches have been considered. To implement such modifications, the simplest approach has been used, even if this meant that performance could be affected negatively.

The frequency scaling proposed method, when related to the number of words users use most, needs no adaptation of the model whatsoever. The number of words, and

the number of instances of a word is easy to obtain counting the number of elements in the list of instances of a particular node. A function that does exactly this has been implemented. It could be argued that instead of searching the tree, an index of the words in a tree and their frequency could be available. This would increase the performance of the algorithm. For the tests in this research this improvement has not been implemented and is left as future work to build better and optimized models.

On the other hand, the behavioral feature related to word sequences needs the tree models to be adapted to store the information regarding successive words. The proposed solution is rather simple and uses a dictionary to store references to all previous words that have been already been stored for a given word. This dictionary, as with time intervals, is always stored in the node of the last letter of a word.

When the tree model is built, or when new words are added to the tree, the information about previous words is also added or updated. To do so, for every new word, the reference to the last node (the last letter) of a word is temporarily stored. Then, when the next word is inserted, in its last node information, this last node reference is stored in the dictionary. This way, every word in the tree has the list of words that have preceded them.

Later, when words are searched in the tree, to detect successive construction of words, the list of previous words is consulted. The procedure is straightforward. The previous word reference, initially, is set to *null*, because the first word searched will not have a previous word. After the first word is searched, the previous word reference is set to the last node reference of this word. When the last node of a second word is found, the node reference of the last node of the first word is looked upon the dictionary of previous words. If it is found, weighting is performed.

For example, if the words *THIS* and *WORD* have been added to the tree in this order (see Figure 4.8), a memory reference to the last node of the first word would be stored. In this case, the last letter of the first word is *S*. The memory reference to the node containing the S could be, for example: *4338740848*. When the second word (*WORD* in this example) is added to the tree, the last node is updated to include the reference to a previous word. This means that in the list of previous words the value of the previous word reference (*4338740848*) is added.

When words are later searched in the model from a new sample, if these two words are again searched in this particular order, the process is similar: when the last node of the first word (*THIS*) is accessed the value of the memory reference of the last node (the *S* letter node, with value *4338740848*) is stored temporarily. When the second word is searched and the last node is found, the list of previous words is consulted to determine if the value *4338740848* is present. If it is the case, weighting policies may be applied.

4.4.3 Comparing new samples to the model

Once the described models for a number of users have been built it can be said that the system is ready to identify users. From this moment on, new sessions can be compared to these models to try to establish conclusions on the author, or owner, of the session.

To compare new sessions to the models a distance based approach is used. The user with the minimum distance to sample is determined as the author of the sample. This is, again, used to either identify or authenticate them.

Distances analyzed

Five different distance measurements have been used in this study. These have been chosen among the most used in the State of the Art. Even if some articles have focused solely on trying almost every distance measurement available, a smaller number has been used in other publications. Knowing if the distance measurement chosen is an important parameter is also one of the goals of this study, tied to the parameters and features described in the previous section.

The distance measurements chosen for this study are the following:

- Euclidean: $D_E(\vec{X},\vec{Y}) = \sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2}$
- Manhattan (a.k.a. City block): $D_M(\vec{X},\vec{Y}) = \sum_{i=1}^{n}|X_i - Y_i|$
- Canberra: $D_C(\vec{X},\vec{Y}) = \sum_{i=1}^{n}\frac{|X_i - Y_i|}{|X_i|+|Y_i|}$
- Chebyshev: $D_{CH}(\vec{X},\vec{Y}) = max_{i=1}^{n}|X_i - Y_i|$
- *wordgraph*: $D_{wg}(\vec{X},\vec{Y}) = |\sum_{i=1}^{n}X_i - \sum_{i=1}^{n}Y_i|$

In every case there is a source vector ($\vec{X}$) and a target vector ($\vec{Y}$). The source vector is the list of time intervals from the word being searched from the newly obtained sample. This list includes the following intervals: Press–Release (PR), Press–Press (PP), Release–Press (RP) and Release–Release (RR). Each of these same values are available for the target vector (that of the intervals found in the model). If more than one instance of a word is available in a node of a tree, the mean value of all available instances is used. As previously commented, one letter words only have Press–Release (PR) values. In this case the distance measurements are only evaluated for this feature.

Just as an example, if $\vec{X} = (78, 93, 63)$ was the template vector of dwell times associated to the word THE and $\vec{Y_1} = (90, 87, 88)$ and $\vec{Y_2} = (60, 120, 103)$ were two vectors from newly collected samples, the result of obtaining the distances previously shown, in both cases, would be:

- Euclidean distance:

$$\begin{aligned} D_E(\vec{X},\vec{Y_1}) &= \sqrt{(78-90)^2+(93-87)^2+(63-88)^2} \\ &= \sqrt{144+36+625} = 28.37 \\ D_E(\vec{X},\vec{Y_2}) &= \sqrt{(78-60)^2+(93-120)^2+(63-103)^2} \\ &= \sqrt{324+729+1600} = 51.50 \end{aligned}$$

- Manhattan or city block distance:

$$\begin{aligned} D_M(\vec{X},\vec{Y_1}) &= |78-90|+|93-87|+|63-88| \\ &= 12+6+25 = 43 \\ D_M(\vec{X},\vec{Y_2}) &= |78-60|+|93-120|+|63-103| \\ &= 18+27+40 = 85 \end{aligned}$$

- Canberra distance:

$$\begin{aligned} D_C(\vec{X},\vec{Y_1}) &= \frac{|78-90|}{|78|+|90|}+\frac{|93-87|}{|93|+|87|}+\frac{|63-88|}{|63|+|88|} \\ &= \frac{12}{168}+\frac{6}{180}+\frac{25}{151} = 0.2703 \\ D_C(\vec{X},\vec{Y_2}) &= \frac{|78-60|}{|78|+|60|}+\frac{|93-120|}{|93|+|120|}+\frac{|63-103|}{|63|+|103|} \\ &= \frac{18}{138}+\frac{27}{213}+\frac{40}{166} = 0.4981 \end{aligned}$$

- Chebyshev distance:

$$\begin{aligned} D_{CH}(\vec{X},\vec{Y_1}) &= max(|78-90|,|93-87|,|63-88|) \\ &= max(12,6,25) = 25 \\ D_{CH}(\vec{X},\vec{Y_2}) &= max(|78-60|,|93-120|,|63-103|) \\ &= max(18,27,40) = 40 \end{aligned}$$

- *wordgraph* distance:

$$\begin{aligned} D_{wg}(\vec{X},\vec{Y_1}) &= |(78+93+63)-(90+87+88)| = 31 \\ D_{wg}(\vec{X},\vec{Y_2}) &= |(78+93+63)-(60+120+103)| = 49 \end{aligned}$$

In these five cases the first sample vector, $\vec{Y_1}$, is *closer* (since the result of the distance measurement is smaller) to the stored sample $\vec{X}$ than the second sample vector

$\vec{Y_2}$. The user who had submitted $\vec{Y_1}$ would be considered a better candidate to be identified as the valid owner or author of the testing sample.

A graphical example of the distances obtained from different users is shown in Figure 4.11. In this case the dark green line belongs to the previously built model from user *Real*. The orange line belongs to a new sample from the *Real* user, and the other three belong to the timing intervals for the 'a word' sequence of three *Other* users. It can be seen that the minimum distances between any user and the Model line belongs to the user *Real*. This is the expected behavior when comparing the same words between samples and different models. The shorter distances should belong to the *Real* user, that is, the owner of the session.

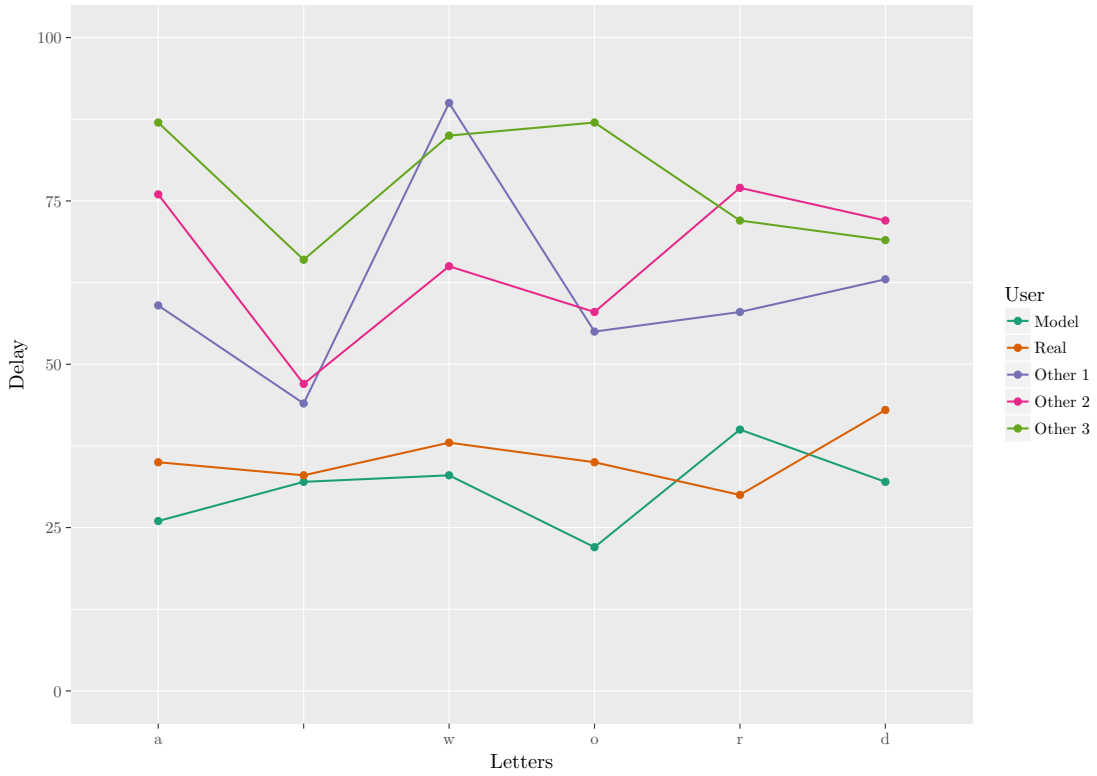

Figure 4.11: Timing intervals for 'a word' from different users

Figure 4.12 shows another interesting graphical example of what it would be desirable to find in terms of density of distances belonging to the owner of a particular session. In this case a session belonging to user 24 has been evaluated and the distances between all words from a testing session and a number of different models (8 in this case) has been obtained. In the figure, user 24 has the largest density of short distance values between almost 0 and 50. The other models this session has been compared to show a more irregular distribution ranging roughly from 20 to 300ms.

Even if this is the expected and desired result of comparing a session against many models, the densities shown in this figure are not always these. Errors in identification are present and efforts should be focused on minimizing them by identifying those

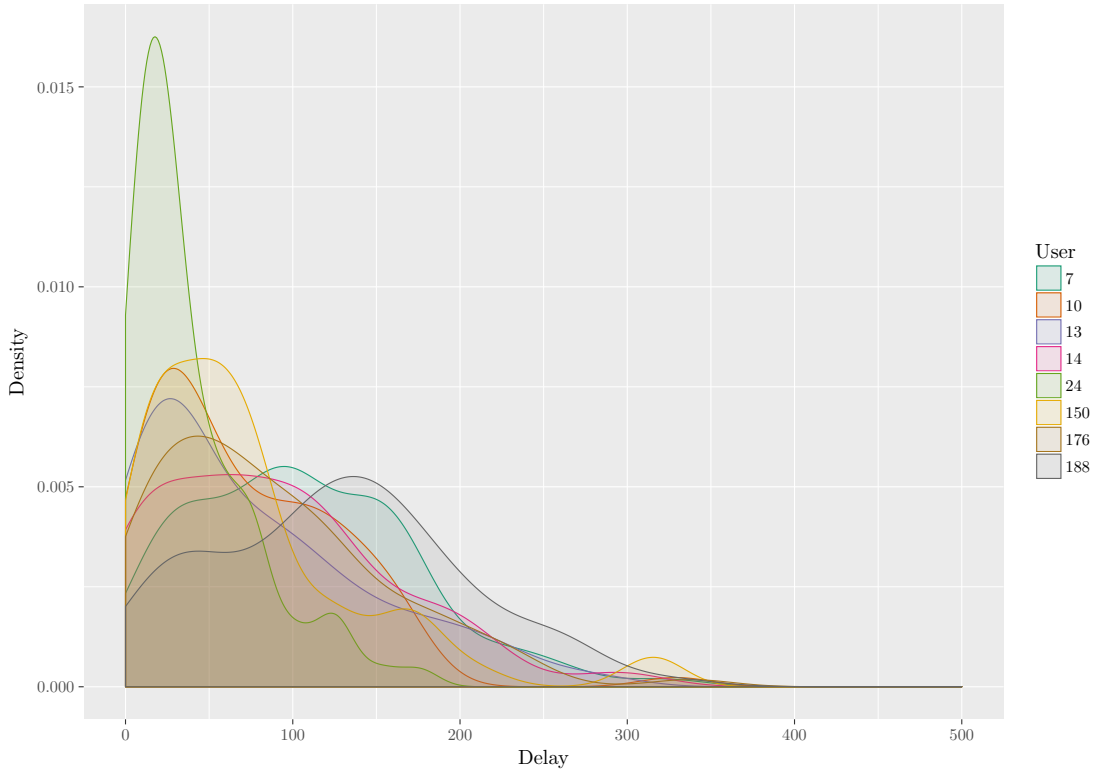

Figure 4.12: Density of distances

features that better differentiate individuals.

Scaling distances

The distance measurements proposed in the previous section return a value for the difference between every word from a new sample being compared and the same word stored in the model. The features proposed in subsection 4.4.2 include Frequency scaling and Successive words scaling. These take advantage of the particular structure of the tree models to try to find additional information that can helpful when identifying users.

Scaling distances means that the values returned by the distance measurements *can* be modified to bring them closer to zero (improve them) or, on the other hand, move them away (demote them). The goal is to give more relevance to those measurements that have additional relevant information. In any of these cases a modifier m is determined and then multiplied to the distance measurements previously obtained.

In the case of the Frequency scaling method, the relation of the number of instances of a word (WC) in respect of the total words in a model (TW) is obtained (PW). The modifier m is then obtained by using these equations:

$$PW = \frac{WC}{TC} \times 100 \qquad\qquad m = \frac{1}{1 \pm PW}$$

The modifier m can either add or subtract the PW value. It is added when the word searched has been stored at least once (time intervals are present in the node where the word has been found). If, on the contrary, the word is not present in the model, but still time intervals have been obtained from a partial match, then PW is subtracted. This way, completely found words are given more importance, and even more if these are frequently used. On the other hand, partial words seldom used, are given less importance and demoted.

In the case of the Successive word scaling method, the idea is much simpler. Successive words that have been searched and found in the same order that once were typed are given more importance. The distance measurements are divided by 2. This method, on the contrary, does not punish those words that have no relation to others on the model.

If these methods are found to be relevant and indeed improve the accuracy of the system, a more in-depth study could be performed to determine the best modifiers m that better help classify users.

4.4.4 Determining the owner of a session

Different methods, both Statistic and Machine learning techniques, have been tried on the distances returned by the data search procedures. The Machine learning techniques evaluated (including SVM, Decision trees, and Neural networks) have been discarded due to their ineffectiveness in classifying users in the proposed scenario.

This section explains the different methods that have been evaluated to classify users. Each of these methods is a sort of evolution of the previous one, and each of them tries to improve the results progressively, even if by just a little. All methods explained in this section are accompanied with examples that should help understand the basic methodology behind each of them.

Sample data is needed to perform any kind of evaluation. Table 4.7 shows a basic dataset of example values that could have been obtained after having a distance measurement obtained between a testing sample and a target model for a different number of users[10]. 5 different models are used in this example (these models belong to users in column *User–Test*) to test an origin session. The origin session belongs to user 192 (column *User–Real*). 4 words have been compared between the origin session and

[10]The data in this table is merely informational. It does not come from real data from real collected samples and it is only shown to describe how each of the proposed methods to identify a user works.

the models. Four time intervals features are also shown and used in these examples (PR, RP, PP and RR). No importance is given to the distance measurement chosen or to what model structure has been used to obtain the values shown in this table. The proposed methods in this section work the same way if these values come from the Euclidean, the Manhattan, or any other distance measurement. At the same time, it does not matter if a Straight, Inverted or Combined tree model has been used or any of the other models described in previous sections. For the sake of completeness, the *Tree* column shows a value of S if the distance has been obtained from a Straight tree or a value of I if it comes from an Inverted tree. The Combined tree model is used by default when this column is entirely discarded. The Depth column indicates, as the name implies, the depth at which the last letter of a word searched has been found.

Just as an example, to clarify what the values for each feature mean, when the word *here* typed by user 192 (the owner of the session) has been searched in the model belonging to user 3207 the distance obtained when taking into account the Press–Release (PR) feature has been 69. In this case, the Depth at which this word has been found is 4, because the origin word has four letters and it has been completely found in the model. The last column states that this distance has been obtained from a Straight tree model.

The following sections describe the methods that have been applied on the available information to determine the owner of a session. These methods are the following:

- Mean of distances: This method obtains a mean value of all the distances obtained for every searched word and every feature. These values are then combined into a global mean per user. The user with the minimum value is determined as the owner of the session.

- Median of distances: This method is exactly as the previous one with a slight modification. Instead of using the mean to obtain a single value for each feature, this method uses the median. These median values are then combined, again, into a global mean per user. The user with the minimum value is determined as the owner of the session.

- Weighted mean of distances: A different version of the Mean of distances method is evaluated, but in this case, the distances are weighted depending on their closeness to zero. Due to the inner workings of the weighted mean procedure, the weighted median alternative was not implemented.

- Higher number of minimum distances: Instead of obtaining a global mean like in the previous methods, a voting fusion method is performed on the mean values obtained from all the words searched and every feature.

Word	Feature				Depth	User		Tree
	PR	RP	PP	RR		Test	Real	
here	69	144	176	99	4	3207	192	S
sun	67	19	48	21	3	3207	192	S
there	56	135	145	93	5	3207	192	I
moon	88	33	66	30	4	3207	192	I
here	84	200	163	124	4	37	192	S
sun	71	16	58	74	3	37	192	S
there	72	187	145	110	5	37	192	I
moon	66	25	70	60	4	37	192	I
here	23	11	16	20	4	192	192	S
sun	15	15	14	23	3	192	192	S
there	34	20	13	18	5	192	192	I
moon	20	30	15	28	4	192	192	I
here	71	13	43	59	4	56	192	S
sun	48	31	24	17	3	56	192	S
there	80	22	55	48	5	56	192	I
moon	56	40	40	25	4	56	192	I
here	60	120	155	140	4	78	192	S
sun	30	15	10	45	3	78	192	S
there	52	112	163	132	5	78	192	I
moon	33	5	3	38	4	78	192	I

Table 4.7: Distances after comparing a session against 5 different models

- Weighted mean of distances, revised: This last method combines all previous methods and adds the Depth at which the word has been found to try to improve the global results. To do so, the mean of each feature is weighted, but instead of doing it by feature it is done by word (instead of evaluating columns by feature, rows by word are treated). Then, two different global values are found and combined using a fusion method. For this method, the use of fusion is discussed because the fact that only two global values are used poses a dilemma when trying to establish a threshold value for the voting scheme.

The following sections describe in more detail each of the methods that have been enumerated in the previous list.

Mean of distances

This method performs a straight computation of different mean values of some chosen features. In Table 4.7 there is an entry for each of the distances obtained between words from an origin session and the different target models. In the example, the origin session belongs to user 192, and is compared against models belonging to users 3207, 37, 192, 56, 78.

The following is a rather quick and informal description of how this method works. For each *User–Test* user, the mean value for all words and features F_j is obtained (see columns $\bar{x}_{F_j}$ in Table 4.8). Then, a new global mean is obtained from the values of each row (column $\bar{x}_{gm}$). The user with the minimum $\bar{x}_{gm}$ value is determined as the owner of the session. Formally this would be written down as follows.

An origin session S from a user U has W words. Each of these words is searched in different models M. Each word W_i is a vector of values $\vec{X}$. This vector may include a combination of dwell times and flight times from the recorded timing intervals depending on the chosen feature F_j being analyzed. In the example, F_j can be one of the following: PR (Press–Release), RP (Release–Press), PP (Press–Press), and RR (Release–Release).

Every word W_i is searched in the model M belonging to user U_k (M_{U_k} with $k \in 1..5$ in this example). These models may have a vector $\vec{Y}$ of previously collected timing intervals corresponding to the word being searched. From these two $\vec{X}$ and $\vec{Y}$ vectors a distance can be determined:

$$\forall W_i \in S, D(W_i, M_{U_k}) = D(\vec{X}, \vec{Y}) \tag{4.1}$$

The distance $D(W_i, M_{U_k})$ is obtained for each feature being considered. From these, the mean value $\bar{x}_{F_j}$ can be obtained:

$$\forall F_j, \bar{x}_{F_j} = \bar{x}(D(W_i, M_{U_k})_{F_j})$$

Finally, using these $\bar{x}_{F_j}$ values, a global mean value for each user U_k is obtained :

$$\forall F_j, \bar{x}_{gm} = \bar{x}(\bar{x}_{F_j})$$

The user with the minimum $\bar{x}_{gm}$ value is determined as the owner of the session. In the example shown in Table 4.8, the minimum $\bar{x}_{gm}$ value is **19.69**. The *User–Test* user is identified as 192. This user would be determined as the owner of the session and it can be seen that the *User–Real* user is also identified by 192. This would count as a correctly identified session. If columns *User–Test* and *User–Real* had been different it would be counted as an incorrectly identified session.

Feature				$\bar{x}_{gm}$	User	
$\bar{x}_{PR}$	$\bar{x}_{RP}$	$\bar{x}_{PP}$	$\bar{x}_{RR}$		Test	Real
70.00	82.75	108.75	60.75	80.56	3207	192
73.25	107.00	109.00	92.00	95.31	37	192
23.00	19.00	14.50	22.25	**19.69**	192	192
63.75	26.50	40.50	37.25	42.00	56	192
43.75	63.00	82.75	88.75	69.56	78	192

Table 4.8: Mean of distances method

In this example, four timing interval features have been used, but this is not mandatory or limited to a particular number. In the real tests, all possible combinations of these four features ($2^4 - 1 = 15$) have been evaluated to determine if there is a combination that always yields the best results. It is worth noting that when only one feature is being evaluated, the step where the $\bar{x}_{gm}$ is obtained is meaningless since it always corresponds to the value of the feature being analyzed.

Median of distances

This method is exactly as the Mean of distances method, but instead of using the mean, the median value is used. From the previous definition, only the following changes:

$$\forall F_j, Md_{F_j} = Md(D(W_i, M_{U_k})_{F_j})$$

Using these Md_{F_j} values, a global mean value for each user U_k is obtained:

$$\forall F_j, \bar{x}_{gm} = \bar{x}(Md_{F_j})$$

The user with the minimum $\bar{x}_{gm}$ value is again determined as the owner of the session. In the example shown in Table 4.9, the minimum $\bar{x}_{gm}$ value is **20.63**. The *User–Test* user is identified as 192. This user would be determined as the owner of the session and it can be seen that the *User–Real* user is also identified by 192.

Feature				$\bar{x}_{gm}$	User	
Md_{PR}	Md_{RP}	Md_{PP}	Md_{RR}		Test	Real
68.00	84.00	105.50	61.50	79.75	3207	192
71.50	106.00	107.50	92.00	94.25	37	192
28.50	17.50	15.00	21.50	**20.63**	192	192
63.50	26.50	41.50	36.50	42.00	56	192
42.50	63.50	82.50	88.50	69.25	78	192

Table 4.9: Median of distances method

Why these two methods have been chosen is explained graphically in Figure 4.13. In this figure, for a couple of users (identified as 7, being the rightful owner, and 14), both the mean and the median of their samples per word is shown in vertical lines (the dotted lines on the right correspond to the mean values, while the dashed lines on the left correspond to the median values). The median seems to be a much better measurement if the distribution is skewed, as tends to be the case with the collected samples.

Weighted mean of distances

This method is a revision of the Mean of distances method. A $\bar{x}_{gm}$ global mean value of all considered features is obtained. Previously, though, the distances of each of the words are weighted using a criterion based on how close their distance to zero is. Distances far from zero would be scaled even further. On the contrary, distances closer to zero would be kept close to zero. This initial idea was revised choosing different possible weights.

For this example, the values in Table 4.7 have been modified by a weight. The weighted value is obtained using the following modifiers: all values up to 100 have a modifier of 1; values between 100 and 200 have a modifier of 2; and values between 200 and 500 have a modifier of 3. Values above 500 are discarded.

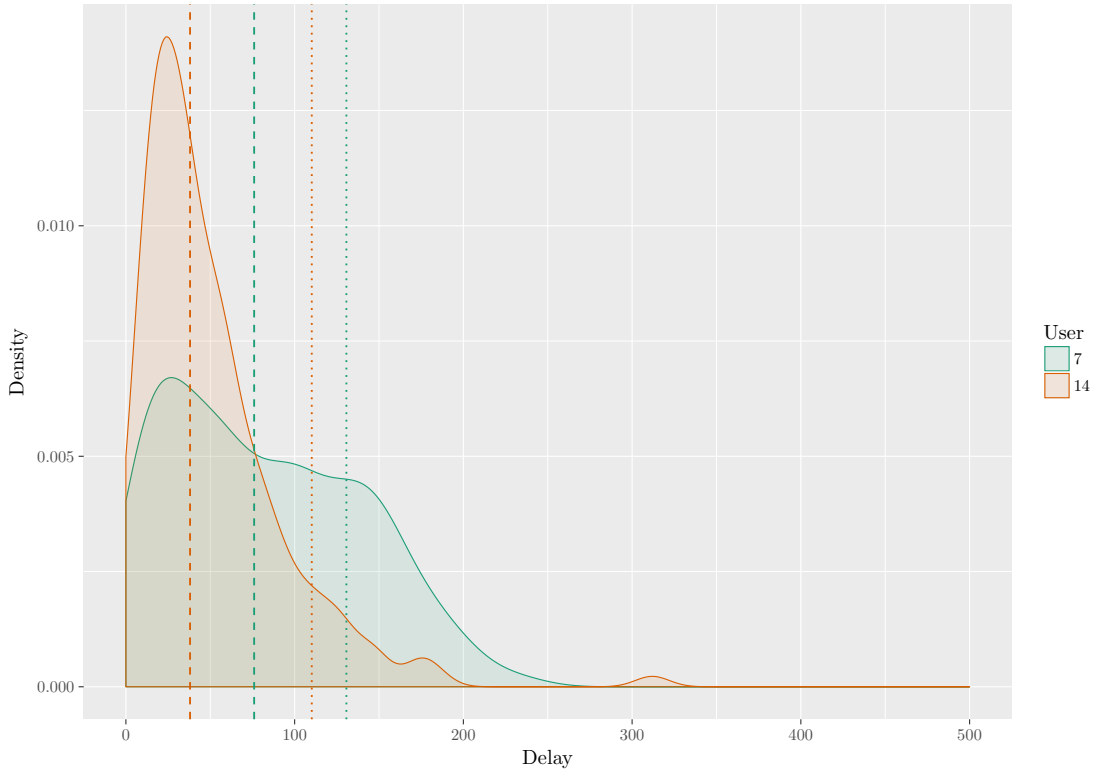

Figure 4.13: Mean and Median of distances

Just as an example, the 101.67 value from Table 4.10, from the $\bar{x}_{RP}$ feature and test user 3207 was obtained using the following procedure $(144 \cdot 2 + 19 \cdot 1 + 135 \cdot 2 + 33 \cdot 1)/6 = 101.67$.

The result of applying this method to the example table, after obtaining all means like in the first method, is shown in Table 4.10. Again, the minimum global mean value is **19.69**, but it is interesting to see that other $\bar{x}_{gm}$ values in this table have a larger range if distances are far from zero.

Feature				$\bar{x}_{gm}$	User	
$\bar{x}_{PR}$	$\bar{x}_{RP}$	$\bar{x}_{PP}$	$\bar{x}_{RR}$		Test	Real
70.00	101.67	126.00	60.75	89.60	3207	192
73.25	135.83	124.00	100.33	108.35	37	192
23.00	19.00	14.50	22.25	**19.69**	192	192
63.75	26.50	40.50	37.25	42.00	56	192
43.75	80.67	108.17	104.50	84.27	78	192

Table 4.10: Weighted mean of distances method

Higher number of minimum distances

This method is yet another variation of the previous ones. Instead of obtaining a $\bar{x}_{gm}$ global mean value for all features, once the mean value, the median value, or the weighted mean value for each feature $\bar{x}_{F_j}$ has been obtained, the total number of features with a minimum value for each user is obtained. The user with the higher number of minimum values is chosen as the owner of the session. This is basically a voting fusion method between all evaluated features. Formally, there is much similarity with the previous methods, only the last step varies.

In Table 4.11, the Votes column shows the total number of occurrences where the $\bar{x}_{F_j}$ value is a minimum value when compared to the same feature of the other users. A session is determined as owned by a particular user by selecting the one that has the larger Votes value. In the example in Table 4.11, user 192 obtained 4 votes and thus it is determined as the owner of the origin session.

Feature				Votes	User	
$\bar{x}_{PR}$	$\bar{x}_{RP}$	$\bar{x}_{PP}$	$\bar{x}_{RR}$		Test	Real
70.00	82.75	108.75	60.75	0	3207	192
73.25	107.00	109.00	92.00	0	37	192
23.00	19.00	14.50	22.25	**4**	192	192
63.75	26.50	40.50	37.25	0	56	192
43.75	63.00	82.75	88.75	0	78	192

Table 4.11: Higher number of minimum distances method

The number of features taken into account can be a problem as happens in the example shown. The possibility of having a tie in the number of minimum values could well happen. The algorithm would choose either user with the highest number of minimum values, and this could mean that incorrectly identified sessions could increase in number.

Weighted mean of distances, revised

This method shares most of what has already been described in previous ones with an important difference: instead of obtaining a mean for each feature F_j in a column, the mean is obtained by row first, and then global values are obtained. At the same time, it combines the voting fusion method with the weighted and non-weighted values from all previous methods. Formally, this method is described as follows:

An origin session S from a user U has W words. Each of these words is searched in different models M. Each word W_i is a vector of values $\vec{X}$. This vector may

include a combination of dwell times and flight times from the recorded timing intervals depending on the chosen feature F_j being analyzed. In the example, F_j can be one of the following: PR (Press–Release), RP (Release–Press), PP (Press–Press), and RR (Release–Release).

Every word W_i is searched in the model M belonging to user U_k (M_{U_k} with $k \in 1..5$ in this example). These models may have a vector $\vec{Y}$ of previously collected timing intervals corresponding to the word being searched. From these two $\vec{X}$ and $\vec{Y}$ vectors a distance can be determined:

$$\forall W_i \in S, D(W_i, M_{U_k}) = D(\vec{X}, \vec{Y}) \tag{4.2}$$

From these distances two values are then calculated: the mean md and the weighted Mean wmd for all Features. The md and the wmd values make use of the Depth d at which each W_i is found. The Weighted Mean value is obtained using the following weights: all values up to 100 have a weight of 1; values between 100 and 200 have a weight of 2; and values between 200 and 500 have a weight of 3. Values over 500 are discarded.

$$\forall F_j, md(W_i) = \frac{\bar{x}(D(W_i, M_U)_{F_j})}{d}$$
$$\forall F_j, wmd(W_i) = \frac{w\bar{x}(D(W_i, M_U)_{F_j})}{d}$$

At this point, there is an $md(W_i)$ and a $wmd(W_i)$ value for every Word W_i searched in the model M. The final global distance gd between a Session S and the Model M is composed of two values (gd_{med}, gd_{wmed}) calculated using the following method:

$$\forall md(W_i), gd_{med} = Md(md(W_i))$$
$$\forall wmd(W_i), gd_{wmed} = Md(wmd(W_i))$$

As an example of this method, Table 4.12 shows the original table after having calculated a distance measurement between the words of an origin session and target models. This table also include columns md and wmd with the calculated values for each word.

For the first row of user 3207 the Mean value would be: $(69+144+176+99)/4 = 122$. Similarly, the Weighted Mean value would be: $(69 \cdot 1+144 \cdot 2+176 \cdot 2+99 \cdot 1)/6 = 134.67$. These two values would be then divided by the depth at which the last letter of the word was found: $md = 122/4 = 30.50$ and $wmd = 134.67/4 = 33.67$.

Word	Feature				Depth	User		*md*	*wmd*
	PR	RP	PP	RR		Test	Real		
here	69	144	176	99	4	3207	192	30.50	33.67
sun	67	19	48	21	3	3207	192	12.92	12.92
there	56	135	145	93	5	3207	192	21.45	23.63
moon	88	33	66	30	4	3207	192	13.56	13.56
here	84	200	163	124	4	37	192	35.69	37.79
sun	71	16	58	74	3	37	192	18.25	18.25
there	72	187	145	110	5	37	192	25.70	27.31
moon	66	25	70	60	4	37	192	13.81	13.81
here	23	11	16	20	4	192	192	4.38	4.38
sun	15	15	14	23	3	192	192	5.58	5.58
there	34	20	13	18	5	192	192	4.25	4.25
moon	20	30	15	28	4	192	192	5.81	5.81
here	71	13	43	59	4	56	192	11.63	11.63
sun	48	31	24	17	3	56	192	10.00	10.00
there	80	22	55	48	5	56	192	10.25	10.25
moon	56	40	40	25	4	56	192	10.06	10.06
here	60	120	155	140	4	78	192	29.69	31.79
sun	30	15	10	45	3	78	192	8.33	8.33
there	52	112	163	132	5	78	192	22.95	24.74
moon	33	5	3	38	4	78	192	4.94	4.94

Table 4.12: Results table with mean values

From each of these md and wmd values and for each user U the final two values gd_{med}, gd_{wmed} are then calculated. Table 4.13 shows these final values for the proposed example. Again, as an example, for user 3207, $gd_{med} = (12.92, 13.56, 21.45, 30.50) = 17.51$

Feature		Votes	User	
gd_{med}	gd_{wmed}		Test	Real
17.51	18.60	0	3207	192
21.98	22.78	0	37	192
4.98	4.98	**2**	192	192
10.16	10.16	0	56	192
15.64	16.54	0	78	192

Table 4.13: Final values for the proposed method

Finally, in Table 4.13, the Votes column shows the total number where each of the gd values is a minimum value when compared to other users. It has been observed that when evaluating sessions using these gd values, there are some incorrectly identified sessions but most of the time these errors are not reported by all the gd values at the same time. It has been decided to use a fusion method to try to improve the global rate of identification using a voting scheme, but this one is stricter: a session is determined as owned by a particular user only if all features have selected the same user. In the example in Table 4.13, user 192 obtained 2 votes and thus it is determined as the owner of the session. If these two votes had been assigned to different users the session would have been marked as unidentified, and from this, considered badly identified.

In the results section, when this method has been evaluated, the possibility of not using fusion, and choosing only one global value as the result, has also been considered.

4.4.5 Authentication

The previous section has focused on the different methods to determine the owner of a session. The possibility of authenticating users is also analyzed in this study. This means that, by using the methodology proposed up to this point, that is, the proposed logical tree models, the optimal building and searching parameters, the best distance measurement, and the best method to identify a session, authentication is also attempted.

To authenticate a user, different methodologies can be tried. In general, the common way of doing it is by establishing a threshold value below which the user is considered authenticated. This threshold value has to be determined basing the decision on the distance measurement and the different methods available so that it better fits the current environment. For example, if the vast majority of distances obtained from

the logical tree model, when using a particular distance measurement, range between 20 and $50ms$, the threshold value should be somewhere in between these two values. It is possible that there are also other values outside this range, especially above the upper value, but these are normally considered *outliers*. Determining which is the best threshold value can be a problem as has been previously discussed in Section 2.3.3. If the idea is to be as strict as possible, a value close to the lower limit would be chosen. This would mean that only those users with very tight typing patterns, pretty close to the stored models would be granted access. This would also mean that *many* valid users with a typical distance value to the model close to the threshold would be denied access. On the other hand, having a threshold value close to the upper range would satisfy most, if not all, valid users but impostors and invalid users would be given access. This behavior, of course, is not desired.

In this study, all training sessions are going to be evaluated against a group of previously built models from different users. The distance values for *all* training sessions to all the available models will be recorded as well as the identified user, exactly as if it was an identification process. In this particular case, though, an increasing threshold value will also be used to determine how many users are correctly granted access or not depending whether the obtained distance measurement is below the given threshold. This is known as a zero-effort attack because all impostors are not *really* trying to impersonate other users.

A particular example of this methodology is shown in the following example. Let us suppose that a user has authored a message and this is compared against $M = 5$ number of models. In this particular case, the first model M_1 happens to belong to the user author of the message (this is not mandatory if only impostor users are being evaluated in an intrusion attack). Let us also suppose that the obtained distances D to the models are: $[23.5, 18.7, 45.3, 82, 30.1]$, and that the given threshold value is 25. From the 5 authentication attempts D_{M_1} and D_{M_2} are below the threshold value. This means that $2/5$ attempts would be granted access, while $3/5$ would be denied access. Since the first model belongs to the user author of the message, the FRR would be 0 and the FAR would be $1/4 = 0.25$, from the attempt of this user to authenticate as the user owner of the second model. This procedure is going to be repeated for all sessions from all users and for different threshold values.

To measure the accuracy when authenticating users, the EER value will be used. This value corresponds to the rate where both the FAR and FRR values are equal. This measure has been widely used to evaluate authentication schemes throughout the literature related to biometrics. Also, ROC curves will be shown as another further measure to test the feasibility of the proposed authentication method.

4.4.6 Age group and gender

Age group and gender have also been considered as contextual features. These are also evaluated in the following chapter with an experiment entirely dedicated to these particularities. The goal is to determine whether the age group and gender a user belongs to has an effect when processes such as identification or authentication are performed.

The methodology that has been followed in these experiments is the following: one experiment is centered on evaluating gender, while another is focused on age groups. A third experiment, evaluates the effect of having mistakes users make incorporated in the model as well. This one has a particular methodology that is discussed later in this section.

For the first experiment, centered on gender, users and their models are separated in two groups: male and female. All testing sessions are then compared to groups of models where only male or female users are present. This comparison is performed exhaustively: all testing sessions are tested against all model groups. The idea behind this test is to determine if men and women have different typing patterns that can lead to the possibility, for example, of identifying the gender of the user submitting information.

The same approach has been considered for age groups. The age groups already described in the current chapter have been used as groups of models against which testing sessions have been evaluated. In this case, a session belonging to a user of the Young age group, for instance, has been tested against all three groups (Young, Middle age, and Senior). This has also been performed exhaustively, with all sessions being tested against all age group models. As with the gender test, the possibility of identifying the age group a user belongs is analyzed.

This two features can lead to the possibility of building better and more robust models. At the same time accuracy could be improved when testing sessions are only compared to those age ranges that better classify individuals.

For these two tests, the number of users per group has had to be lowered to a value that ensures comparison of the different genders and age groups. The size of the gender groups has been decreased to 20 users, while the group size for the age groups has been decreased to 15 users.

The last experiment combines both age group and gender and compares the results in identification when mistakes users make are included in the model and when these are not. This is the test with the tiniest groups (only 10 users per group). In the results of all these experiments the margin of error related to sample size has also been included.

4.4.7 Cross-validation methodology

To perform the experiments that evaluate all the proposed parameters, features, and methods, different approaches regarding cross-validation have been used and taken into account to increase statistical significance.

For the Quality of the model experiments, the selected users have been divided into three groups of 20 users as it has been previously described in Section 4.2.5. The process to evaluate the percentage of correctly identified sessions is that of a typical Data mining study. The available sessions in each group are partitioned into two groups, one for training the models and one for testing them. The partition used has been 70/30% respectively. The procedure follows a Monte Carlo Cross-Validation (MCCV) technique where a random number of sessions are selected without replacement. This process is then repeated multiple times, generating new training and testing partitions each iteration. A problem this approach may have is that since partitions are created independently for each run, the same session can appear in the test set multiple times or, if a user has only a few sessions available, the testing sessions can overlap in different runs.

The process is repeated 10 times to avoid choosing *lucky* samples. The different models are then created with the selected sessions from every user. Then, each session from the testing samples is evaluated against all available models of the group the user belonged to. The number of correctly identified sessions is then recorded.

For the rest of the experiments centered on establishing a good method to identify users, instead of having three groups of 20 users, a single group of 40 random users from the pool of the 60 best per period has been used. The rest of the cross-validation process has not been altered, using 70% of the sessions to build the models and the remaining 30% to test them. Again, the process has been repeated 10 times to ensure statistical relevancy.

For the experiment centered on the optimal group size against which a testing session is compared to, the group size has been evaluated with values ranging from 2 to 60 users. On the other hand, for the experiments focused on age group and gender different group sizes have been used as well. In this case, the size of the groups has been established from the need to have groups as similar as possible.

4.5 Summary

In this chapter, the methodology that has been followed to study the particular rhythm users have has been outlined. To begin with, contextual information and the chosen behavioral features have been defined. At the same time, how these are interpreted

in this study has also been detailed. Then, the steps followed to develop software to capture the necessary user features have been described. From this point, the obtained dataset has been analyzed. The distribution of users regarding age group, gender and the number of events recorded has also been shown. At the same time, the different groups that have been selected in terms of quality have been described.

Two sections take most part of the rest of the chapter. These are: the definition of the proposed model, and the different chosen methods to determine the owner of testing sessions. The proposed model is based on a logical tree of words where latency is stored taking into account the position of letters, thus keeping contextual information. It is thought that this idea is relevant instead of grouping graphs in non-contextual dictionaries. The different outlined methods use simple statistical methods, weights, and fusion to try to determine the owner of a session using most of the contextual information available in the models.

5 | Results

The following chapter describes the experiments that have been carried out on the Keystroke Dynamics dataset collected during a period of three semesters and the results that have been obtained. Different tests, as explained in the previous chapter, have been performed. In this chapter, these tests have been organized in their own sections, and each of them focuses on particular parameters, or features, being analyzed. These tests show the evolution of the quality of the proposed tree models in comparison to previous methodologies, more specifically to an *n-graphs* frequency scheme that uses Relative and Absolute distance measurements.

The list of performed tests in relation to the sections of this chapter is the following:

- Test 1 – Quality and size of the model: This initial test focuses on proving the effect of modifying the underlying characteristics of the proposed tree models, such as the model structure, the number of words, the number of instances, and how the cleaning of the included samples is performed. It also tries to determine which type of tree model (Straight, Inverted, or Combined) is the best to evaluate new samples, and if using a Forest of trees, instead of a single tree, is a better suited alternative.

- Test 2 – Most relevant model parameters: Having established the parameters that help build quality models, this test focuses on the information that ends up in the tree models. These parameters deal with the depth at which words are found, if recursion to analyze more data is necessary, or if less but better information improves the results. Finally, the recommended minimum number of words found in the models is also evaluated and established.

- Test 3 – Distances and methods to classify users: Having established the parameters that help build quality models and those that better help classify them, this test focuses on the proposed methods to identify users and improve the results obtained in previous tests and also when using an *n-graph* methodology.

- Test 4 – Features related to user behavior: With all distances measurements and proposed methods evaluated, this test focuses on behavioral features that can be adapted to the rhythm pattern of a user. These include the mistakes users make,

the frequency with which users repeat words or sentences, and the delimiters that are used to detect words on the submitted messages.

- Test 5 – User group sizes: Once all parameters have been evaluated, and a valid methodology has been established to compare a new session against a group of models, this test evaluates different user group sizes to find out where the threshold for a reliable system should be set, and if all sizes behave the same way when different underlying building parameters are chosen.

- Test 6 – Authenticating users: This test evaluates the possibility of using the proposed methods to authenticate users instead of just identifying them. The results are presented using FAR and FRR values as well as ROC curves. It is discussed if using Keystroke Dynamics as a way of authenticating users is reliable and robust enough taking into account that users may have to type a minimum number of words for the results to be good enough.

- Test 7 – Dealing with age group and gender: In this test, different parameters are evaluated. First, gender separation is tried. The idea behind this test is to see if different models separated by gender help determine who a user is. The test has been performed comparing all sessions with models of the same origin gender, and again, with models containing the opposite gender. The second test is similar, but in this case, the separation is based on age group. Three age groups have been used: Young, Middle age, and Senior. All sessions, again, have been compared against the three different age groups and the accuracy has been evaluated. Finally, a last test regarding age group and gender combined, evaluates the accuracy when mistakes are also used as part of the models.

In order to have a frame of reference to evaluate the results presented in this chapter similar tests have been performed using an *n-graph* frequency methodology. The method proposed by Daniele Gunetti and Claudia Picardi in their excellent paper *Keystroke Analysis of Free Text* [55] has been implemented and evaluated with the available dataset. The results obtained using their method are presented in the next section and these set the bar to which other results from the proposed tests will be compared.

In general, the results in this chapter are presented in terms on accuracy showing the Percentage of Correctly Identified Sessions (PCIS). When a new testing session is tested against a model it can be either correctly identified as belonging to its rightful owner or, on the other hand, identified as belonging to a different user. The first case is treated as a *success*, while the second one is treated as a *failure*. The PCIS is the total number of successes in relation to the total number of samples evaluated. The inverse of the PCIS is the percentage of the number of failed evaluations. At the same time,

FRR, FAR, and more specifically, EER values are used when testing authentication. All these evaluation measures have been described in Chapter 2, more specifically in Section 2.3.

5.1 Using Relative and Absolute distances

In 2005, Daniele Gunetti and Claudia Picardi published a paper in ACM Transactions on Information and Systems Security [55]. This paper was relevant in terms of accuracy and the proposed methods on how to deal with free text. This was the first time that the Relative (R) and the Absolute (A) distances, or measures, were put to test even if the basic idea for the Relative distance has already been introduced by Bergadano et al. [18]. In their article, the accuracy when performing tasks of Identification, User classification, and Authentication was evaluated with rather excellent results.

The impact of the paper was important. To date, this is one of the most cited papers centered on free text and Keystroke Dynamics[1]. Other articles [37, 38, 92], have either used, adapted or modified the proposed measurements to try to improve results in particular situations. Some work has also been carried out to try to minimize problems in scalability. Be that as it may, the excellent results reported by the use of their distances set a standard when dealing with Keystroke Dynamics and free text.

Their proposed methods to identify users have been implemented for the study presented in this document. In their case, the number of samples from users was very different from what the dataset available contains. At the same time, no samples of impostors are available. To make matters worse, the great amount of data available presents a rather serious problem in terms of needed computer resources.

To be able to present comparable results to the research proposed in this document, a number of decisions have been taken. First, the number of tested users has been established. As explained in Chapter 4, two different sets of users have been chosen to perform the initial tests. These sets are:

- Set 1: Three groups of users containing the 20 best users from the 60 best in number of events.
- Set 2: 40 random users selected from the 60 best users, also, in number of events. This set is pretty much similar, in terms of users, to the user set used by the original paper.

In their study, 40 volunteers submitted 15 samples each. These were used to build the models. At the same time, 165 users submitted impostor samples. The samples

[1] 372 cites on Google Scholar when accessed on the 3rd of April, 2017.

obtained had a length varying between 700 and 900 characters. From this number of characters, it is not easy to establish the number of different *n-graphs* each sample contained. Even if a mean value was to be used, probably more than 200 digraphs per sample would be available. This is much more, in general, than the typical sample from the dataset available in this research. In this study, the number of samples per user varies a lot. Not all users have submitted a fixed minimum number of samples. At the same time, the number of *n-graphs* in a sample is something that can be very different to the settings they used on their experiments.

To make matters even worse, and just as an example, if the group with the most number of events is used to evaluate the R (Relative) and A (Absolute) distances, having only 20 but as many as 100–300 sessions per user the computational requirements to obtain the proposed distances make the tests impossible to complete. A subset of the available data had to be chosen. To be consequent with their and this research, the following restrictions have been applied:

- Use a maximum number of sessions, chosen randomly. The tests have been performed setting this value to 15 (the minimum used by the original paper), 35 and 50 sessions.
- Use a minimum number of *n-graphs* in every session. Again, the tests have been performed with the following values: 100 and 135 *n-graphs*. Why these values have been chosen is something that will make sense later in this chapter, but in the end, the goal is to have a minimum number of quality origin samples to perform comparisons to the model, and thus, improve results.
- Set of users: Both, the three groups of 20 users set, and the random set of 40 users have been evaluated.
- Period: Four different partitions of the available keystroke dataset have been used to perform the *n-graph* tests. Three partitions have been selected from the samples belonging to each available semester (labeled as $P1$, $P2$ and $P3$ respectively) and the last partition has all the information available from all semesters (labeled as $P0$).

As per the distances obtained, as in their study, both the R distance and the A distance have been implemented and evaluated for 2, 3, and 4 *n-graphs*. It is worth noting that, in their study, they combined the results, for example, of R_2 and $A_{2,3}$, to obtain values to better discriminate users in the form of $R_2 + A_{2,3}$. In the case of this study it has been found that the best results have been always obtained when using the values R_2 and A_2. Even better are the results when these two values are added in the form of $R_2 + A_2$. For the results presented in this section this last value is the one

that has been chosen. As a side note, it should be said that the results in the current study have never been as good as the ones they obtained. This could be explained because of the setting in which the samples for this study were gathered. In this case, it seems to be much more open to randomness in terms of number of sessions per user and number of events per session.

5.1.1 Results using the *n-graph* methodology

This section presents the best results obtained when using Relative and Absolute distances applied to the dataset available for this study. Table 5.1 shows the results when the parameters described in the previous section have been evaluated. It should be noted that, to obtain these results, as in the original paper, only the Press–Press (PP) time interval feature has been used.

Group of users	Period	Accuracy				
		15–∞[1]	15–100	35–100	35–135	50–135
A: *Rich* models	$P0$	84.56	97.63	99.06	99.06	**99.26**
	$P1$	84.65	96.81	98.10	**98.53**	98.34
	$P2$	82.91	97.80	98.66	98.77	**98.79**
	$P3$	84.16	96.21	97.94	97.44	**97.94**
B: *Normal* models	$P0$	74.38	95.52	96.45	**96.69**	95.55
	$P1$	68.49	83.49	85.22	**89.83**	**89.83**
	$P2$	73.00	92.66	94.49	95.14	**96.61**
	$P3$	78.68	91.32	94.63	**95.50**	**95.50**
C: *Poor* models	$P0$	78.65	96.90	97.56	**97.96**	**97.96**
	$P1$	85.92	97.84	97.84	**98.79**	**98.79**
	$P2$	56.30	82.45	80.61	**87.61**	**87.61**
	$P3$	55.68	82.71	80.82	**86.18**	**86.18**
Random 40	$P0$	71.42	93.49	95.57	**96.67**	96.60
	$P1$	75.03	89.75	89.58	91.48	**93.53**
	$P2$	61.44	89.38	92.05	93.46	**93.99**
	$P3$	62.89	88.66	89.59	92.34	**92.53**

[1] The first value is the maximum number of sessions; the second value is the minimum number of graphs when building the models.

Table 5.1: Results when using Relative and Absolute distances: $R_2 + A_2$

From the presented results, it can be seen that the more samples are used to build the model, the better the model behaves. This goes in concordance with what had

already been reported in the conclusions of the original paper. At the same time, if the origin sessions are selected among those that have a minimum number of *n-graphs* the results improve considerably too.

On the other hand, it should be said that the tests performed to present these results have been chosen among those that can be considered feasible in terms of computer resources restrictions. Above 50 sessions, the time needed to obtain the distances has been considered unfeasible.

The minimum number of *n-graphs* a session should have to be considered valid is a parameter that is discussed later in this chapter. When setting this parameter high, the number of sessions that are discarded, because this requirement is not met, increases rapidly. At the same time, using only samples of greater quality improves the results. This is something that should be taken into account when establishing the parameters for the system. Unfortunately, in the environment where these measures were evaluated, that is, an online free text virtual campus environment, the minimum number of *n-graphs*, letters, events, words or whatever other measurement of the quality of a sample could not be determined or established. This parameter should be taken into account and discussed to set a minimum threshold above which samples are used, even if going against the main objective of correctly identifying as many samples as possible.

Finally, a rather interesting fact is that as soon as the number of users these sessions are compared to, the accuracy of the system decreases. This is something that has been previously commented on the literature. One of the key parameters when using a Keystroke Dynamics system should be the size of the group against which a sample is tested. The bigger the group, the more robust the system should be considered. This, again, goes against the resources needed to perform such calculations, which should always be rational to the time a user can wait to be authenticated or identified. Of course, if such identification is not performed in real time, as could be the case of an offline verification of a previously submitted assignment, the time needed to obtain a result can be stretched, though never infinitely.

From this point, the accuracy obtained when using *n-graphs* is used as a threshold to compare the results from the following sections, focused on contextual information and words searched on logical trees.

5.2 Test 1 – Quality and size of the model

This is the first test that has been performed on the collected Keystroke Dynamics dataset that evaluated contextual information. The goal of this experiment is to prove that the information, be it in quantity or quality, stored in the model affects the

accuracy of the whole system significantly. It is worth noting that the goal of this test is not to obtain great results in accuracy but to evaluate the effects of adapting the parameters when building the different proposed tree models.

5.2.1 Model building methodology

As discussed in Section 2.4, the typical sequence to build a model of a user consists in the following steps: Data acquisition; Extract features; Clean data; and Create template. In this case, the procedure has been altered to have the models cleaned after all training samples have been incorporated into the model.

These are the steps that have been followed to build the models and the particularities that have been decided upon for this adapted process:

1. Data acquisition: How this task has been carried out has been described in Chapter 4. In the end, data from three different semesters is available.

2. Extraction of features: When building the models, the features that have been stored are only the timing intervals between Press–Release and Release–Press events. From these, other features like Press–Press or Release–Release are also obtainable.

3. Store samples in the model: This should have been the last step of the process and it should have been performed after the cleaning of samples. Instead, the collected events and their related timings have been inserted into the models unaltered.

4. Clean stored samples: When different timing intervals of the same word are available, these can be more or less similar. How the term *similar* is understood, in this case, can be misleading. It is not easy to determine when a user has typed a word like they *always* do or, on the other hand, the collected sample is *way off* the normal. To do so, all available samples have been added into the tree model and then these have been cleaned keeping only those in a range of a number of standard deviations treated as a parameter[2].

Figure 5.1 shows a graphical description of the process detailed when adapted to the methodology performed in this experiment.

The samples used to test the models have been treated somewhat differently. In this case there has been no storing or cleaning of samples used for testing. All words,

[2]For the Forest of trees model, a cleaning parameter of 2 standard deviations has been chosen as well as no limitations to the number of words or graphs inserted in the model. These parameters have been chosen after evaluating the results obtained from evaluating the same parameters using a single tree model.

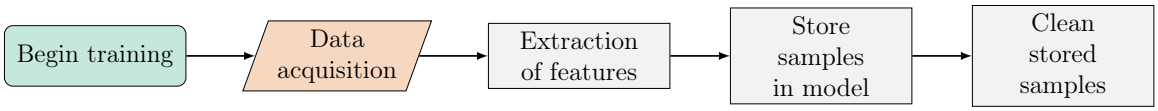

Figure 5.1: Followed procedure to build the tree models

even if repeated, have been treated individually. There has been no grouping by words to discard those outside the limit established by the standard deviation parameter. This has been done this way to have the most information from the origin samples unaltered.

For this test, a number of different models have been built: a Straight tree model, an Inverted tree model, the Combined tree model that uses the previous two as a whole, and a Forest of trees model where each session has its own set of single tree models (Straight, Inverted and Combined).

5.2.2 Samples verification methodology

The same four partitions of the keystroke dataset, discussed in the previous section, have been used to perform this test. Each available semester has its own set of users (labeled as $P1$, $P2$ and $P3$ respectively) and the last partition contains all the information available from all periods (labeled as $P0$).

For this test, the 60 users that have sent the largest number of events (not the largest number of sessions), from each period, have been selected to build the models. As previously described in Section 4.2.5, three groups of 20 users have been created. The first group (labeled as A) contains the 20 users that have submitted the most number of events. The second group (labeled as B) has the 20 next users in terms of events submitted. Finally, the third and last group (labeled as C) has the users that submitted the lowest number of events during the considered period.

Table 4.4, in the previous chapter, shows the total and average number of events from each of these user groups and periods. It is easy to see that these groups are far from homogeneous. This is considered to be a good feature since it will help to see how the models behave in highly different situations. There are some interesting facts in Table 4.4 that are worth commenting upon because these will be relevant when the results are discussed. In terms of events, the second period is considerably larger than the other two partial periods, especially for groups A and B. At the same time, the largest period is, of course, the one that groups all periods into one. It is also worth noting that group A is considerably larger, in general, than the other two groups. With this in mind, it should be expected that results for group A are substantially better.

Finally, to build and test the proposed models, the MCCV commented in Chapter 4 cross-validation methodology has been used. This technique separates the dataset

into two partitions: one for training the model, and the other to test it. In this case, a 70/30% partition has been used. 70% of the sessions have been used to build the models, and the remaining 30% have been used to test them. The cross-validation procedure has been repeated 10 times to improve statistical relevance. The results come from the mean value of the different repetitions performed.

5.2.3 Evaluated parameters

These are the parameters that have been studied in this experiment:

- Group of users (G): A (Rich), B (Normal), or C (Poor), depending on the ranked number of events per user. The question is if the number of events is directly related to the accuracy of the system.
- The period the sample has been captured (P): Four periods have been taken into account, three for each semester available in the dataset ($P1$–$P3$), and one for the totality of the data collected ($P0$). What is interesting here is to determine the relation with the number of events and the accuracy.
- The maximum number of words allowed in the model (MW): This parameter is allowed to increase with the following values: 100, 500, 1000, 2000, 4000, and Unlimited or ∞. Some users with a low number of events do not always take full advantage of this parameter, though. This means that these users will end up with smaller models even if the possibility of going beyond the number of stored samples is possible.
- The maximum number of instances per word allowed in the model (MI): Again, this parameter is allowed to increase with the following values: 1, 2, 3, 5, 10, and Unlimited or ∞. The same particularity about the size of the models is present when evaluating this parameter: not all users, especially those in group C, have enough instances per word to take full advantage of this parameter.
- The number of standard deviations to keep multiple instances of a word ($STDS$): For each word that has multiple instances, the process of removing those outside the normal value has been evaluated. In this case, the different values evaluated range from 0 (not cleaned at all) to 4.
- Type of tree model: For this parameter three different tree models have been studied. The Straight tree model that contains words stored from beginning to end; the Inverted tree model with words stored from end to beginning; and the Combined tree model that uses the information from both trees at the same time.

- Structure of the tree model: Finally, with this parameter, the structure of the model has also been evaluated. Two options have been tried: a Single tree model, and a Forest of trees model where each training session has its own tree model.

5.2.4 Number of independent tests performed

The total number of different possibilities when all these parameters are combined is: $3G \cdot 4P \cdot 6MW \cdot 6MI \cdot 5STDS \cdot 10 = 21.600$ different tests. Due to computational and performance restrictions, not all combinations have been tried. The chosen approach has been to test the parameters incrementally. First, the number of maximum words has been evaluated. Then, once an unlimited number of words is allowed into the model, the number of maximum instances has been allowed to increase. The same has been done for the standard deviations parameter.

All in all, the number of independent tests for this experiment, for each period, have been $3G \cdot (5MW + 5MI + 4STDS) = 42$. This has been repeated 10 times using the MCCV technique and 4 times as per each period. The grand total of tests performed for this first experiment has been: $42 \cdot 4P \cdot 10 = 1,680$ tests.

The Forest of trees alternative has only been tested for groups B and C due to computer resources restrictions. At the same time, not all combinations used with the single tree model have been tried on the Forest of trees. This model structure presents serious scalability problems. It is not strange to see that some users end up with more than 100 tree models. To deal with this problem, only the parameters that have yielded the best results with the Single tree structure have been evaluated. The test has been carried out for the 4 periods with a total of $2G \cdot 4P \cdot 10 = 80$ tests.

5.2.5 Determining the owner of a session

This initial test is focused on the quality and size of the model rather than on the evaluation of distance measurements or methods to correctly identify the owner of a session. This will be the goal of other experiments in this chapter. With this in mind, only the Euclidean[3] distance has been used. The method to identify the owner of a session has been the Mean of distances method.

The time interval features that have been taken into account are: Press–Release (PR), Release–Press (RP), Release–Release (RR) and Press–Press (PP). To determine the result from the Mean of distances method, all possible combinations of these 4 features have been considered. There are 15 ($2^4 - 1$) different combinations.

Figure 5.2 shows different examples of results after establishing some building

[3]Euclidean distance: $D_E(\vec{X}, \vec{Y}) = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2}$

parameters, finding the distances between test sessions and the proposed tree models, and applying the Mean of distances method to identify the owners. The parameters for these example results have been set as shown in Table 5.2.

Parameter	Example			
	1	2	3	4
Group of users	A	C	B	B
Period	$P0$	$P1$	$P2$	$P3$
Maximum number of words	∞	500	∞	∞
Maximum number of instances per word	∞	1	5	∞
Standard deviations	3	0	0	3

Table 5.2: Chosen parameters for the example results

The chosen parameters for these examples are highly different from one another. Group of users A has the best users in terms of events while group C has the worst. The values for the building of the models show a wide diversity to give a first impression of the different results that can be obtained. These have been selected to show somewhat different results but, at the same time, to point out the similarities that all share.

After analyzing the results from these four executions, the first that comes to the attention is that the percentages of correctly identified sessions can be very different. The PCIS in these examples range from $\sim 26\%$, to $\sim 75\%$. These values are far from being great but show how very different samples can yield very different results. It is interesting to see that if the models are limited in size the results are even worse than if, on the other hand, the tree models are allowed to grow in size and are cleaned.

Also, the Feature that seems to give the worst results in all these examples is PRE (Euclidean Press–Release) while some show that RPE (Euclidean Release–Press) can be an interesting Feature to use in its own. On the other hand, the rest of combinations of Features seem to be, more or less, equally relevant or adequate. None seems to outperform the others. In Figure 5.2, the horizontal dotted lines depict the mean values for all Features for each of the tree models considered.

As per which tree model is better (Straight, Inverted, or Combined), no conclusion can be determined yet. It seems obvious that, analyzing only these four examples is far from statistically relevant to obtain meaningful conclusions. What is done next is take the mean values for all Features considered and find the mean values of all tests performed from the MCCV repetitions. Next section shows the global results for this experiment.

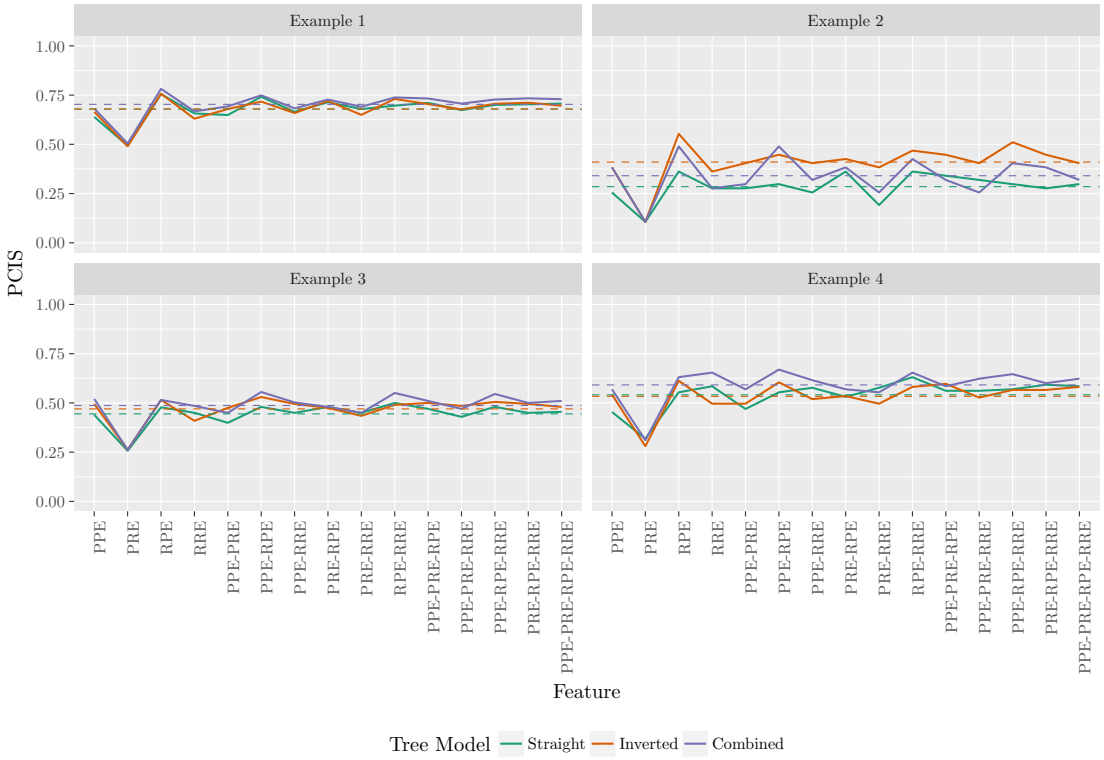

Figure 5.2: Result examples for the quality of the model

5.2.6 Results for the Quality and size of the model test

Table 5.3 in page 122 shows the results when all parameters have been evaluated for the different groups and periods. At the same time, Figures 5.3, 5.4 and 5.5 show the evolution of the PCIS after modifying the building parameters and comparing all testing sessions against the models. In Table 5.3, the bold fields represent the best result for a group of users of a period. At the same time, the red fields represent the best tree model when compared between Straight, Inverted, or Combined.

Both, the table and the figures, show relevant results that should be commented upon. To do so, the different parameters previously described are analyzed below:

- Group of users: The results show that the better the model in terms of events, the better the results. It is interesting to see that periods 0 and 2 share quite similar results when models are good (groups A and B). This could propose the possibility of limiting the number of words in a tree after a certain threshold. In both cases, the results top at $\sim 75\%$ accuracy even if the number of events is almost a half (see Figure 4.6). When to stop adding samples has not been fully addressed. From the presented results, it seems that once a number of events, or words, have been added to the tree results do not improve anymore. Some ideas that come to mind deal with the possibility of adding new words as soon as these are available for the first time, but stop adding more instances of words

from a certain moment. This goes hand in hand with the adaptability of a model discussed in Section 2.4.

- Period: Periods where more events have been captured provide better results. This was expected. It should be pointed out, though, that having many words in the model building samples may not be always the best solution to obtain good results. Having many repeated words, to be able to create a meaningful template should be favored.

- Maximum number of words allowed in the model (MW): This is a parameter that helps improve the results, no doubt, but not in a way that could be considered outstanding. It should be taken into account that many small models (mainly from group of users C) do not take full advantage of this parameter. This should be considered when building smaller models.

- Maximum number of instances per word (MI): This parameter helps define the way a user types, and it can be clearly seen that as soon as more instances per word are allowed the results improve significantly. Unfortunately, smaller models cannot take full advantage of the use of this parameter.

- Number of standard deviations to clean the model ($STDS$): This is a parameter that greatly helps define the rhythm of a user. It seems that beyond 2 standard deviations the results do not improve that much, but these improve greatly once the cleaning of stored samples is taken into consideration. In order to build strong models a value of 2 or 3 standard deviations should be considered as *essential*.

- Feature: When testing all 15 feature combinations, two of these Features have given the best and worst results, respectively, most of the time. In terms of the best combination of features, the RPE (Euclidean Release–Press) feature has been the best 45.23% of times for the Straight tree, 54.76% for the Inverted tree, and 60.17% for the Combined model. The worst feature has been the PRE (Euclidean Press–Release) feature 84.94% of the times when using the Straight tree, 90.47% when using the Inverted tree, and 96.60% of the times when using the Combined tree. This information could be helpful in order to choose a Feature to work with and increase performance, without having to use 15 different Feature combinations. A quick test has been performed using *only* the RPE feature. The execution performance has been more than 15 times faster since only one combination of Features is used. When using this unique feature, the results in Table 5.3 have increased up to a 10% in accuracy. This is considered to be significant and in need of more testing.

- Type of tree model: In all cases, the best suited tree model is the Combined one, taking into account the words searched from beginning to end and those searched from back to fore. This hints at the importance of contextual information and the importance of having high hit rates. The second test will go deeper into this result, discussing how words are searched in the model.

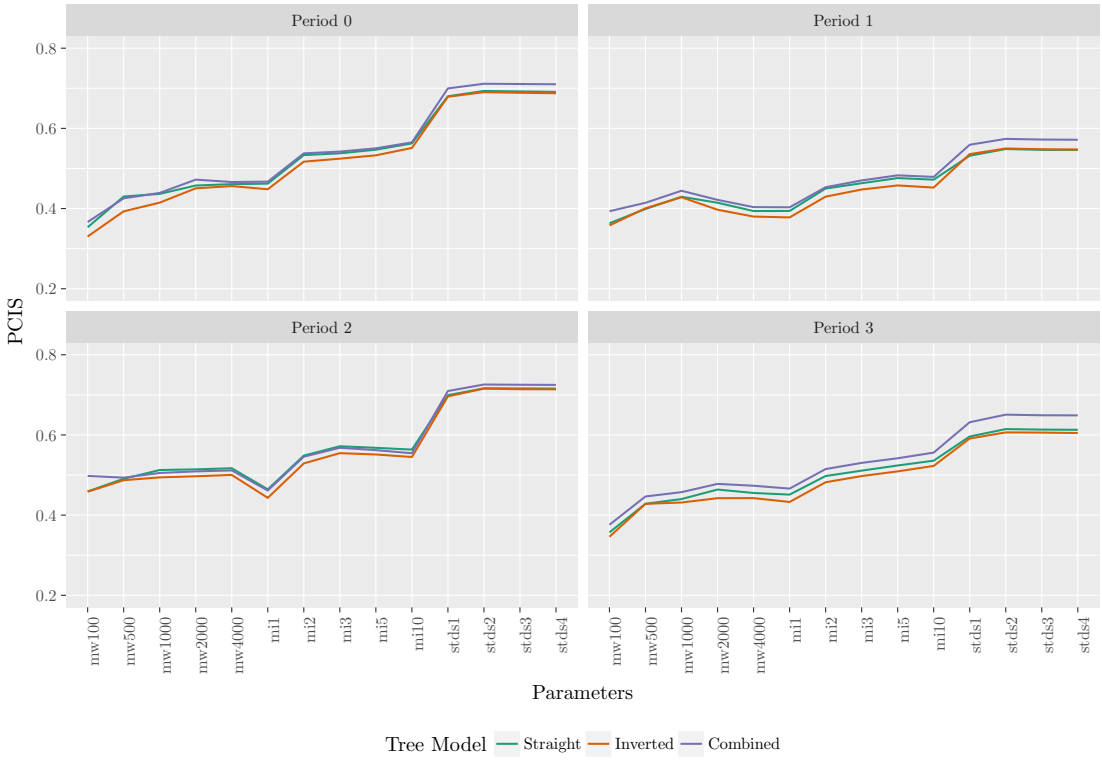

Figure 5.3: Results for Group A per period

Of all these parameters, the maximum number of words allowed in the tree and the maximum number of instances per word, when models are small, are inefficient because not enough samples are available. This presents a limitation to the proposed model, where hit rate can be too low when not enough samples have been captured. In such cases, adopting an *n-graph* alternative can provide better results due to the inherent model characteristics. In the long run, though, when more and more samples are available, the proposed model should be considered to improve performance.

As per the tree structure, as can be seen in Table 5.3, the results from using the Forest of trees technique, in some cases, improve the results that the alternative single tree model provide. In particular, it seems that with smaller models (Group C and period $P3$, in this case), the results improve. The same cannot be said once the number of events increase.

The decision to use a Forest of trees model or not depends greatly on the time and computer resources available, but taking into account the little improvement on

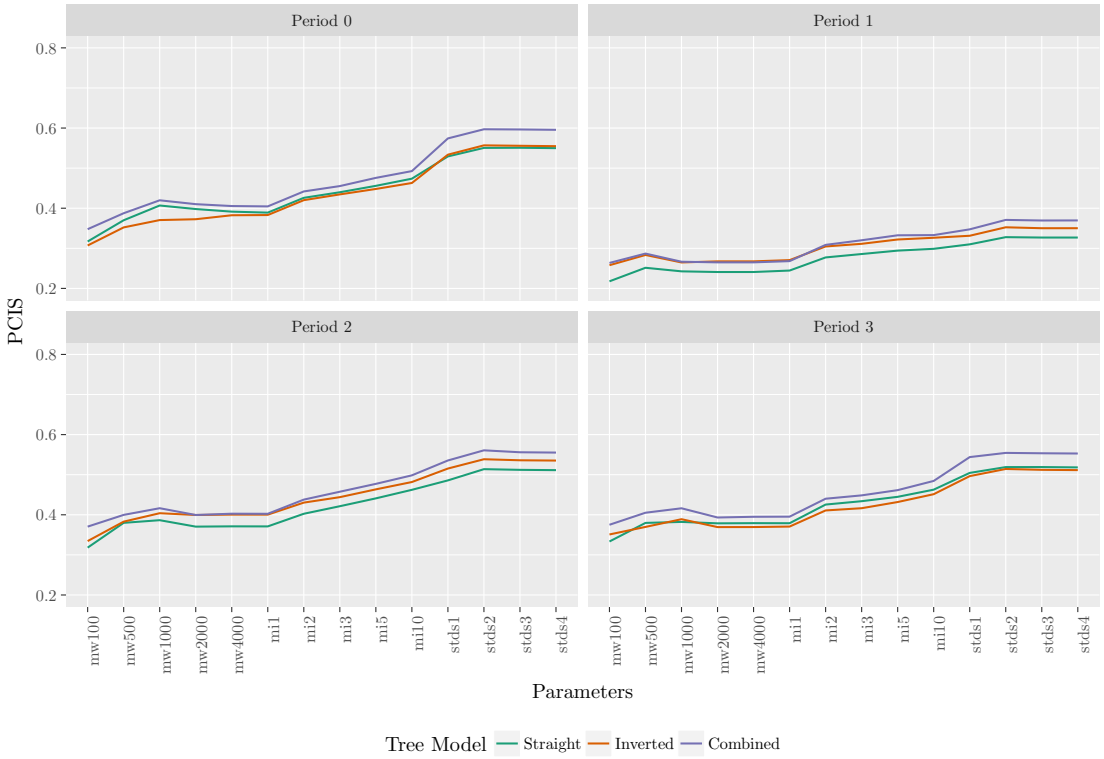

Figure 5.4: Results for Group B per period

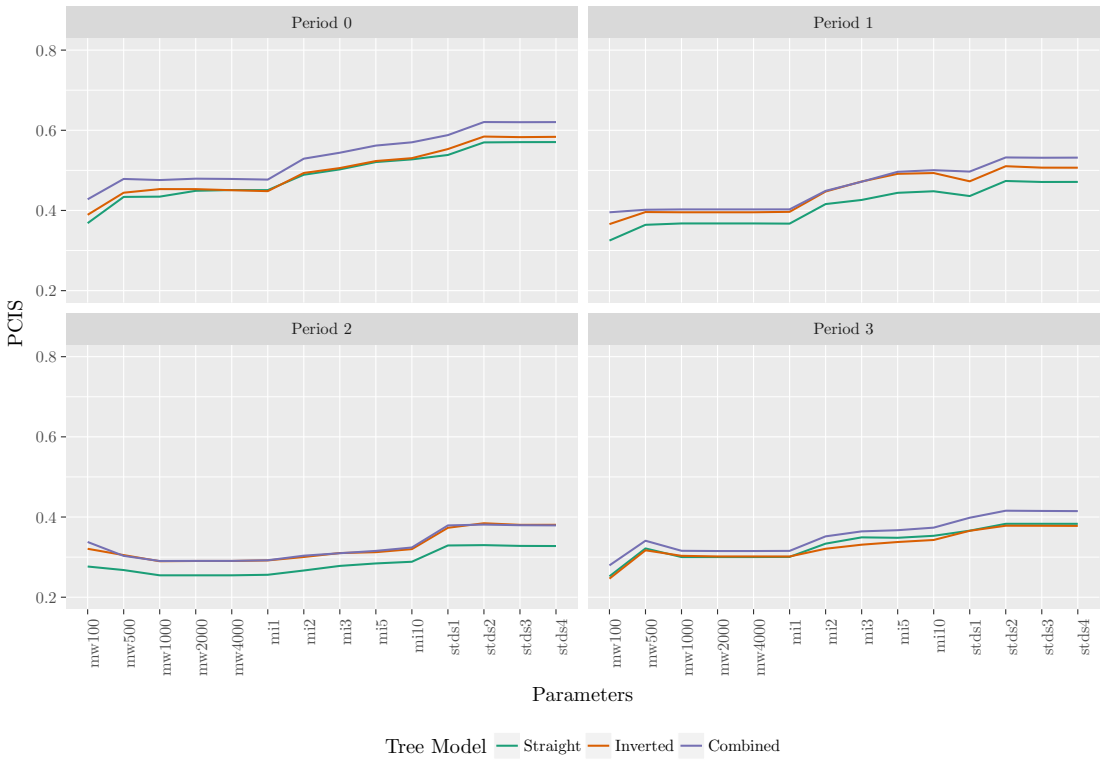

Figure 5.5: Results for Group C per period

the results it would be discouraged to use this structure. Another thing that has to be taken into account is the results size: when searching against a single tree, a word would have a single distance in the results file. On the other hand, when searching against many trees would mean having a distance value for each of these trees. The result files increase exponentially in size.

A possibility that has been thought about to reduce these scalability problems when dealing with a Forest of trees model has been to force the use of smaller models, selecting only a limited number of sessions to compare an origin session to. This has been intended only to face the computer and performance restrictions. This idea has been discarded for this test because this test is mainly focused on dealing with different model sizes, and since groups A and B have many sessions and events, it has been thought that abnormally decreasing the size of these models goes against the objective of the quality and size of the model test.

For the rest of the experiments in this document, a Combined single tree approach will be used, even though the possibility to test particular configurations on a Forest of trees scheme should not be totally discarded. The alternative to implement the proposed solution using a Forest of trees model is left as future work.

5.2.7 Performance evaluation

The building of models when using an *n-graph* frequency model is straightforward and this is clearly seen in terms of performance. Compared to the contextual tree model, the time needed to build and test new samples is considerably faster. If this was one of the parameters on which to focus a decision to choose one model over the other, the *n-graph* frequency models would be chosen without any doubt. In general, the *n-graph* frequency model is about 2.5 times faster than the tree model when using similar parameters.

On the other hand, once the model has been built, applying the Relative and Absolute distances to the samples has been *much* slower than using the proposed method of searching words in a logical tree and obtaining the final results using the Mean of distances method.

When dealing with a Forest of trees model, in the worst case, the tests have taken almost 20 times to complete when compared to the single tree alternative. Due to this fact, and the impossibility to perform these tests with the larger group, the use of Forest of trees, even with the promise of providing somewhat better results has been discarded.

5.2.8 Test 1 summary

Up to this point, the parameters regarding size and quality of the model have been studied. These parameters include the size of the model, the maximum number of words allowed in the model, the maximum number of instances per word, and how to proceed when cleaning the model of outlier values.

In general, it has been seen that the bigger the models the better the accuracy. Also, no limits should be put to the maximum number of words or instances in the model. The parameter that best worked to increase the accuracy of the system has been cleaning outlier values outside of 2 standard deviations, followed by the number of instances of a word.

It has also been proved that not all features have the same importance when identifying users. Determining which combinations work best is left as future work. For the test in this study all combinations will be tried to avoid losing relevant information.

As per performance, building logical trees is not as fast as building *n-graph* frequency dictionaries. Up to this point, also, the results when using logical trees that include contextual information, are far from optimal. Further experiments are necessary and more parameters have yet to be analyzed.

Max. words			100	500	1000	2000	4000					No limit					
Max. instances			1	1	1	1	1	1	2	3	5	10		No limit			
Std. deviations			Model not cleaned of outlier values										1	2	3	4	2
Group[1]	Period[2]	Tree model[3]															Forest[4]
A: *Rich* models	*P*0	S	35.32	43.00	43.66	45.75	46.09	46.21	53.34	53.78	54.68	56.25	68.01	**69.33**	69.22	69.10	–
		I	33.03	39.30	41.48	45.04	45.60	44.82	51.70	52.44	53.28	55.11	67.89	**69.02**	68.87	68.77	–
		C	36.65	42.55	43.90	47.22	46.62	46.71	53.77	54.21	55.02	56.51	69.97	**71.13**	71.07	71.00	–
	P1	S	36.33	39.89	42.95	41.45	39.38	39.40	44.99	46.32	47.60	47.21	53.18	**54.85**	54.64	54.62	–
		I	35.80	40.09	42.82	39.70	38.00	37.79	42.97	44.75	45.77	45.24	53.58	**54.99**	54.79	54.74	–
		C	39.34	41.44	44.44	42.17	40.36	40.31	45.32	47.02	48.29	47.90	55.92	**57.38**	57.21	57.16	–
	*P*2	S	45.89	49.07	51.25	51.42	51.69	46.44	54.84	57.18	56.79	56.35	69.92	**71.64**	71.60	71.57	–
		I	45.85	48.71	49.41	49.69	50.03	44.31	52.90	55.46	55.13	54.49	69.63	**71.55**	71.41	71.38	–
		C	49.78	49.34	50.52	50.91	51.15	46.14	54.57	56.80	56.21	55.44	70.95	**72.58**	72.52	72.49	–
	*P*3	S	35.69	42.84	44.01	46.38	45.53	45.12	49.76	51.10	52.38	53.58	59.62	**61.46**	61.34	61.28	–
		I	34.61	42.82	43.15	44.23	44.24	43.29	48.20	49.75	50.91	52.28	59.12	**60.64**	60.59	60.48	–
		C	37.60	44.66	45.73	47.79	47.33	46.62	51.49	53.03	54.18	55.59	63.18	**65.06**	64.91	64.88	–
B: *Normal* models	*P*0	S	31.70	37.00	40.68	39.80	39.15	38.90	42.59	43.98	45.58	47.37	52.92	55.05	**55.08**	54.99	44.96
		I	30.71	35.22	37.04	37.26	38.25	38.30	42.01	43.45	44.80	46.29	53.36	**55.68**	55.57	55.48	50.22
		C	34.77	38.75	41.98	41.01	40.54	40.46	44.19	45.53	47.57	49.26	57.41	**59.70**	59.65	59.56	50.32
	*P*1	S	21.77	25.15	24.25	24.09	24.09	24.45	27.73	28.58	29.43	29.88	30.99	**32.79**	32.69	32.69	**34.89**
		I	25.79	28.34	26.47	26.75	26.75	27.06	30.47	31.12	32.20	32.63	33.13	**35.24**	35.01	35.02	35.07
		C	26.36	28.69	26.65	26.50	26.50	26.80	30.86	32.01	33.25	33.29	34.74	**37.09**	36.94	36.96	**38.01**
	*P*2	S	31.81	37.99	38.65	37.05	37.12	37.11	40.26	42.14	44.10	46.23	48.58	**51.39**	51.21	51.14	44.91
		I	33.45	38.29	40.40	39.95	40.05	40.04	43.04	44.40	46.34	48.17	51.54	**53.86**	53.60	53.53	53.50
		C	37.05	39.98	41.64	39.95	40.28	40.25	43.79	45.76	47.72	49.84	53.55	**56.08**	55.60	55.52	52.70
	*P*3	S	33.34	37.98	38.22	37.87	37.92	37.90	42.55	43.39	44.48	46.27	50.45	51.90	**51.91**	51.82	51.80
		I	35.09	36.97	38.90	36.93	36.94	37.06	41.09	41.64	43.18	45.14	49.65	**51.43**	51.22	51.15	**55.69**
		C	37.51	40.52	41.62	39.33	39.50	39.53	44.01	44.84	46.15	48.45	54.39	**55.44**	55.35	55.29	**57.99**
C: *Poor* models	*P*0	S	36.83	43.38	43.45	44.90	45.09	45.07	48.91	50.24	52.07	52.74	53.84	56.98	57.04	**57.06**	**58.07**
		I	38.91	44.43	45.33	45.32	45.05	44.82	49.35	50.58	52.33	53.05	55.31	**58.44**	58.29	58.36	**59.27**
		C	42.78	47.86	47.58	47.93	47.86	47.69	52.92	54.40	56.19	57.00	58.80	**62.05**	62.02	62.04	**62.66**
	*P*1	S	32.49	36.42	36.76	36.76	36.76	36.72	41.59	42.63	44.40	44.79	43.59	**47.35**	47.11	47.12	**51.69**
		I	36.59	39.60	39.55	39.55	39.55	39.66	44.72	47.22	49.14	49.34	47.27	**51.04**	50.68	50.67	**57.46**
		C	39.54	40.18	40.26	40.26	40.26	40.27	44.89	47.17	49.64	50.05	49.70	**53.22**	53.15	53.18	**59.55**
	*P*2	S	27.65	26.78	25.46	25.46	25.46	25.59	26.66	27.81	28.43	28.84	32.90	**32.99**	32.80	32.76	**34.16**
		I	32.08	30.50	28.97	29.04	29.04	29.18	30.03	31.02	31.23	32.00	37.32	**38.45**	38.05	38.06	**41.46**
		C	33.78	30.31	29.04	29.06	29.06	29.22	30.36	30.97	31.55	32.40	37.91	**38.12**	37.96	37.92	**41.72**
	*P*3	S	25.21	32.17	30.01	30.02	30.02	30.05	33.36	34.93	34.82	35.32	36.59	**38.31**	**38.31**	**38.31**	**41.91**
		I	24.63	31.70	30.33	30.17	30.17	30.16	32.09	33.10	33.77	34.28	36.56	**37.83**	37.82	37.79	**40.62**
		C	27.94	34.09	31.58	31.52	31.52	31.55	35.18	36.42	36.72	37.35	39.82	**41.58**	41.52	41.48	**45.06**

[1] Group: A: *Rich* models; B: *Normal* models; C: *Poor* models
[2] Period: 0: Full set; 1: Fall 15–16; 2: Spring 15–16; 3: Fall 16–17
[3] Tree model: S: Straight; I: Inverted; C: Combined
[4] Forest of trees: Unlimited models cleaned of 2 Std. deviations

Table 5.3: Model size and quality effect

5.3 Test 2 – Most relevant model parameters

The second experiment that has been performed on the available dataset deals with parameters that have to do with the information stored in the model. More specifically, the Depth at which words are found in the tree has been analyzed, as well as the level of Recursion needed to improve results when identifying users. At the same time, the Minimum number of needed words has also been studied, to determine if, from a certain number of words found in the model, the results can be considered optimal. This last parameter has much to do with the different parameters that have been evaluated when using the *n-graphs* model. In that case, it has been found that restricting the test to a minimum the number of graphs and sessions improves the results significantly. Finally, a particular feature, called Discarding of child times, that has to do with how time intervals are obtained from the nodes in the tree model is also analyzed.

5.3.1 Initial model parameters

To build the models for this test, the selected parameters have been the ones that have given the best results in the previous test. Throughout all the tests in this chapter, this incremental methodology will be used. This means that no limits have been imposed regarding the number of words in the models, as well as the number of instances per word. The cleaning of the model has been performed using a parameter of 2 standard deviations to remove *outlier* values. Finally, as per the type of tree model, the Single combined tree model has been used.

One of the goals of the first experiment has been to determine if the number of events per user has a relevant impact in the results. It has been proved that better groups of users obtain better results. At the same time, it has also been observed that those groups of users belonging to partial semesters have very different results depending on the activity on the Discussion forum modules. After having seen this, for the present and the following tests, and with the aim to go in concordance with having a relevant number of users to compare a session to, a single group of 40 random users will be used. These 40 users are chosen randomly from a pool of 60. These, still, are the users with the most number of events. This *limited* group of 60 users has been chosen due to the fact that if more users beyond this threshold were used, models could end up being very poor and distorting to the results. The 20 users sized groups will not be longer used as it depicts an unreal classification of users. This separation has been of great interest when analyzing model size, but from this point on, a more realistic approach is favored.

5.3.2 Samples verification methodology

The same four partitions of the keystroke dataset have also been used to perform this test. Again, these have been labeled as $P0$–$P3$. A single random group of 40 users has been used per period and repetition. This means that, for each considered period, the 60 best users have been selected, and from this, a random selection of 40, independently of the number of events submitted, have been used. It is worth noting, as in previous tests, that the best users in each period may not have been the same in each group.

To build and test the proposed models the same MCCV cross-validation methodology has been attempted. A 70% partition to train the models has been used, and the remaining 30% has been used to test them. The cross-validation procedure has also been repeated 10 times.

5.3.3 Evaluated parameters

The following parameters have been evaluated in this experiment:

- Length of found words (L): With the analysis of this parameter the question whether the length of a typed word is relevant and if all lengths have the same importance is evaluated. This could be of high interest, not only in terms of model optimizing, but also to have a better understanding of the factors that influence the rhythm of a user. The different values that have been evaluated are the following: Use all lengths (∞), ≥ 2, ≥ 3, $[2-5]$, $[2-7]$, $[3-7]$. This parameter evaluates depth and progressively discards shorter words. A typical distribution table of found words would be the one shown in Table 5.4[4]. It is easy to see that shorter words are the most common found words in the model. The question whether having higher counts of shorter or longer found words is relevant will be discussed by analyzing this parameter.

- Recursion when searching partial sub-words (R): This parameter has been described in Section 4.4.1. The effect of using different types of recursion when searching partial words in the tree model is analyzed with this parameter. The values for this parameter have been: $R0$ (exhaustive recursion); $R1$ (partial recursion); and $R2$ (no recursion at all).

- Discarding of child times (D): When a word is found only up to a certain depth, and when the node of the last letter found has no time intervals information, these can be obtained from all the leafs from that particular node or, on the other hand, this word can be considered as *not found*. This parameter tries to prove

[4]Longer words can certainly be stored in the tree models, and be found, but these are a minority, and are not shown in this table for formatting reasons

whether having real match contextual information from the tree is better than discarding information. This parameter is a simple Boolean, either *yes* or *no*.

- Number of words found in the model (W): This parameter establishes a minimum number of found words from the session being analyzed to see whether results improve if the quality of the origin samples is better. The different values tried have been: Use all words (∞), 10, 25, 50, and 75.

This last parameter has a lot to do with one of the parameters that has been evaluated to test sessions using the *n-graph* frequency methodology. In Section 5.1, due to computer resources limitations and the need to have a reasonable and comparable execution time when using Relative and Absolute distances, a minimum number of graphs has been used to discard poor sessions and at the same time improve accuracy. In that test, it has been proved that the better the origin session in terms of *n-graphs* the better the results.

Length	1	2	3	4	5	6	7	8	9	10
%	38.90	36.23	16.84	3.44	2.02	1.21	0.62	0.32	0.21	0.09

Table 5.4: Word length distribution example (in %)

As per the Number of words found in the model (W), and just as an example of what is the difference between different sizes, the following is an example text that contains 75 words (the other considered minimum number of words evaluated have been marked using parentheses):

> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque arcu (*10*) justo, posuere eget lorem ut, rhoncus aliquet urna. Duis vel efficitur ligula. Ut nec semper (*25*) lorem. Integer in maximus arcu, ut interdum mi. Sed massa est, fermentum eget nunc sit amet, dictum posuere risus. Cras sodales velit vel leo hendrerit (*50*), id volutpat ipsum efficitur. Suspendisse non tortor at augue blandit mattis. Phasellus sagittis ipsum feugiat magna fringilla iaculis. Nam ac magna at lorem efficitur placerat (*75*).

5.3.4 Number of independent tests performed

Following the same methodology as in the previous test, the number of independent tests for this experiment, for each period, have been $6L \cdot 3R \cdot 2D \cdot 5W = 180$. This has been repeated 10 times using the MCCV technique and 4 times as per each period. The grand total of tests in this experiment has been: $180 \cdot 4P \cdot 10 = 7,200$ tests.

5.3.5 Determining the owner of a session

This test is focused on determining the parameters related to searching the model that better help identify the users. The evaluation of distance measurements or methods to correctly identify the owner of a session is, still, a matter to be discussed in further tests. With this in mind, the Euclidean distance is used again and the method to identify the owner of a session is the Mean of distances method.

The time interval features that have been taken into account are: Press–Release (PR), Release–Press (RP), Release–Release (RR) and Press–Press (PP). To determine the result from the Mean of distances method, all possible combinations of these 4 features have been considered. There are $2^4 - 1 = 15$ different combinations.

As part of the results of this test, the combination of features that have given the best results *most of the time* is also discussed.

5.3.6 Results for the Most relevant model parameters test

The results for this test are presented taking into account the following parameters: the period used to test the samples, the minimum number of words found in the model, how the discarding of child times is performed, and the type of recursion used. The combination of all these parameters does not fit in a single table. Due to formatting issues, these are shown separated using the period and the child times discarding parameters as a reference. Tables 5.5 and 5.6, for example, show the results for Period 0 with both Discarding and Not discarding child times results, respectively.

From the results in these two tables some interesting facts can be commented upon. First of all, the fact that using no recursion whatsoever improves the results is of great importance. At the same time, the fact that using only partial short words is also of great relevancy. This two facts are important because they confirm that contextual information is relevant when classifying users using Keystroke Dynamics. At the same time, it has to be taken into account that most samples searched in the tree model have short distances.

It is easy to see that the best results in both tables come from discarding *a lot* of information and using only that that better helps to classify users.

Tables 5.7, 5.8, 5.9, 5.10, 5.11, and 5.12 show the results obtained for the rest of periods and for all the parameters taken into account.

After evaluating all the results presented in these tables, the best performance is achieved when the studied parameters are set to the following values:

- Length of words: It seems evident that, when using the Mean of distances method (and this is important as will be later seen) and the Euclidean distance

Recursion	Words	Length of words					
		All	≥ 2	≥ 3	$[2-5]$	$[2-7]$	$[3-7]$
	1	51.27	51.00	60.08	59.76	56.50	63.71
	10	54.04	53.75	62.62	63.16	59.60	66.44
$R0$	25	57.68	57.32	66.88	67.07	63.48	70.89
	50	60.45	60.13	70.09	69.78	66.26	74.17
	75	61.68	61.39	71.63	70.72	67.30	**75.61**
	1	51.20	50.97	60.08	59.78	56.54	63.66
	10	54.12	53.87	62.75	63.34	59.80	66.53
$R1$	25	57.67	57.36	66.96	67.14	63.56	70.92
	50	60.40	60.16	70.15	69.84	66.36	74.19
	75	61.61	61.43	71.69	70.74	67.37	**75.60**
	1	55.27	52.85	58.89	60.75	57.94	62.48
	10	59.36	56.68	62.32	65.37	62.27	66.21
$R2$	25	64.21	61.32	67.21	70.20	67.05	71.20
	50	67.03	64.13	70.59	72.56	69.75	74.64
	75	68.53	65.64	72.26	73.96	71.17	**76.24**

Table 5.5: Period 0 ($P0$) – Not discarding child times results

Recursion	Words	Length of words					
		All	≥ 2	≥ 3	$[2-5]$	$[2-7]$	$[3-7]$
	1	43.31	43.25	50.95	53.63	49.57	55.38
	10	46.91	46.83	54.12	58.22	53.80	58.90
$R0$	25	50.76	50.65	59.01	62.46	57.99	64.09
	50	54.09	53.87	63.77	65.67	61.42	68.93
	75	55.82	55.68	66.14	67.44	63.21	**71.31**
	1	44.03	42.95	48.91	53.19	49.11	53.44
	10	50.10	48.77	54.02	60.24	55.78	59.09
$R1$	25	54.67	53.29	59.29	64.88	60.64	64.62
	50	58.65	57.18	64.27	68.29	64.36	69.65
	75	61.16	59.65	67.02	70.50	66.76	**72.10**
	1	44.96	42.68	47.84	52.81	48.66	52.38
	10	52.24	49.52	53.51	61.01	56.41	58.69
$R2$	25	57.31	54.44	59.20	65.96	61.59	64.69
	50	61.65	58.78	64.64	69.38	65.80	70.02
	75	64.61	61.71	67.44	72.45	68.71	**72.80**

Table 5.6: Period 0 ($P0$) – Discarding child times results

Recursion	Words	Length of words					
		All	≥ 2	≥ 3	$[2-5]$	$[2-7]$	$[3-7]$
$R0$	1	31.93	32.26	38.66	44.59	38.87	44.38
	10	33.24	33.52	39.92	46.39	40.34	45.71
	25	35.19	35.39	42.34	49.06	42.62	48.74
	50	38.01	37.93	45.81	52.22	45.55	52.48
	75	39.51	39.65	47.42	53.17	46.86	**54.15**
$R1$	1	31.71	32.21	38.48	44.54	38.90	44.28
	10	33.10	33.60	39.72	46.57	40.57	45.60
	25	35.02	35.42	42.14	49.19	42.78	48.68
	50	37.84	38.05	45.71	52.48	45.82	52.58
	75	39.35	39.63	47.29	53.32	47.05	**54.25**
$R2$	1	32.91	30.88	35.25	43.43	37.62	41.05
	10	35.56	33.20	37.12	46.75	40.36	43.18
	25	38.61	35.96	40.22	50.65	43.71	46.95
	50	42.07	39.29	44.33	53.31	46.81	51.32
	75	44.07	41.10	46.55	55.15	48.70	**53.68**

Table 5.7: Period 1 ($P1$) – Not discarding child times results

Recursion	Words	Length of words					
		All	≥ 2	≥ 3	$[2-5]$	$[2-7]$	$[3-7]$
$R0$	1	23.20	23.58	25.38	35.72	30.13	31.05
	10	25.03	25.39	25.86	38.48	32.43	31.75
	25	27.01	27.38	27.83	41.56	34.95	34.31
	50	30.24	30.54	31.11	44.82	38.26	38.00
	75	32.10	32.33	33.85	**45.65**	39.60	40.84
$R1$	1	23.96	23.28	23.05	35.39	29.84	28.82
	10	27.39	26.55	24.55	40.44	34.07	31.01
	25	30.46	29.38	27.11	44.06	37.45	34.16
	50	34.49	33.43	31.26	47.40	41.07	38.78
	75	38.64	37.59	35.43	**50.86**	44.88	42.98
$R2$	1	23.95	22.05	21.89	34.32	28.83	27.56
	10	27.82	25.58	23.19	40.00	33.53	29.76
	25	31.44	28.87	26.21	44.19	37.42	33.50
	50	36.84	34.20	31.64	48.53	42.54	39.69
	75	40.62	38.04	35.42	**51.68**	46.30	43.75

Table 5.8: Period 1 ($P1$) – Discarding child times results

Recursion	Words	Length of words					
		All	≥ 2	≥ 3	$[2-5]$	$[2-7]$	$[3-7]$
$R0$	1	38.31	40.18	53.78	53.28	47.47	58.79
	10	40.41	42.40	55.82	56.61	50.32	61.07
	25	42.66	44.90	59.76	60.01	53.36	65.38
	50	44.61	46.98	63.56	62.55	55.82	69.42
	75	45.09	47.77	65.46	63.58	56.66	**71.31**
$R1$	1	38.31	40.38	53.62	53.34	47.67	58.58
	10	40.53	42.73	55.76	56.82	50.66	60.98
	25	42.78	45.19	59.63	60.17	53.68	65.20
	50	44.65	47.22	63.28	62.66	56.10	69.14
	75	45.16	48.07	65.34	63.79	57.02	**71.13**
$R2$	1	42.93	42.12	50.90	54.82	49.45	56.39
	10	46.65	45.67	53.71	59.95	53.83	59.68
	25	49.78	48.83	58.14	63.97	57.62	64.47
	50	52.25	51.43	62.07	66.98	60.60	68.46
	75	53.88	53.19	64.23	68.90	62.39	**70.48**

Table 5.9: Period 2 ($P2$) – Not discarding child times results

Recursion	Words	Length of words					
		All	≥ 2	≥ 3	$[2-5]$	$[2-7]$	$[3-7]$
$R0$	1	30.69	32.39	41.95	45.46	39.54	48.17
	10	33.25	35.13	44.02	49.66	42.99	50.80
	25	35.68	37.76	47.39	53.20	46.28	54.68
	50	37.50	39.87	50.84	55.82	48.80	58.38
	75	39.05	41.53	53.32	57.89	50.70	**61.04**
$R1$	1	33.50	33.62	39.87	46.54	40.89	46.16
	10	38.54	38.67	43.71	53.78	47.35	50.93
	25	41.72	41.89	47.45	57.92	51.16	55.09
	50	44.68	44.98	51.45	61.57	54.61	59.48
	75	45.99	46.44	53.74	62.73	56.24	**61.99**
$R2$	1	34.17	32.98	38.78	46.01	40.18	45.18
	10	39.94	38.52	42.68	54.05	47.28	50.15
	25	43.73	42.21	46.83	58.58	51.48	54.63
	50	47.23	45.71	50.75	62.32	55.43	59.17
	75	48.41	46.76	53.45	63.66	56.80	**61.83**

Table 5.10: Period 2 ($P2$) – Discarding child times results

Recursion	Words	Length of words					
		All	≥ 2	≥ 3	$[2-5]$	$[2-7]$	$[3-7]$
$R0$	1	39.23	39.18	43.82	48.68	44.04	48.50
	10	41.89	41.76	45.85	51.97	46.83	50.67
	25	45.15	44.99	49.83	55.76	50.36	54.90
	50	48.49	48.07	54.22	58.89	53.60	59.50
	75	49.65	49.24	56.00	59.77	54.65	**61.35**
$R1$	1	39.11	39.08	43.65	48.58	44.00	48.36
	10	41.93	41.82	45.76	52.08	46.98	50.64
	25	45.31	45.15	49.92	55.92	50.62	55.06
	50	48.34	47.98	54.19	58.83	53.58	59.52
	75	49.58	49.23	55.93	59.82	54.73	**61.30**
$R2$	1	39.72	37.63	40.40	47.90	42.89	45.43
	10	43.75	41.27	43.27	52.58	46.93	48.61
	25	48.47	45.66	48.25	57.90	51.85	53.86
	50	50.49	47.79	51.79	59.70	53.74	57.48
	75	52.26	49.45	53.62	60.81	55.36	**59.30**

Table 5.11: Period 3 ($P3$) – Not discarding child times results

Recursion	Words	Length of words					
		All	≥ 2	≥ 3	$[2-5]$	$[2-7]$	$[3-7]$
$R0$	1	31.08	30.77	32.67	40.31	35.67	37.01
	10	34.00	33.61	34.42	44.19	38.94	38.99
	25	37.43	36.93	37.52	48.14	42.70	42.46
	50	40.73	40.05	41.92	51.02	45.76	46.99
	75	42.51	41.83	44.51	**52.58**	47.40	49.54
$R1$	1	30.79	29.54	30.98	39.11	34.28	35.51
	10	36.00	34.46	34.14	45.73	39.99	39.11
	25	40.36	38.64	38.18	50.52	44.55	43.33
	50	43.75	41.88	42.75	53.18	47.74	48.23
	75	45.36	43.35	45.00	**53.82**	48.68	50.22
$R2$	1	30.45	27.85	29.96	38.14	33.10	34.36
	10	36.37	33.05	32.99	45.53	39.33	37.91
	25	40.86	37.37	37.05	50.47	44.18	42.32
	50	43.98	40.26	41.46	52.44	46.70	46.91
	75	47.16	43.53	44.59	**54.33**	49.33	49.95

Table 5.12: Period 3 ($P3$) – Discarding child times results

measurement, not all word distances are equally interesting, relevant or optimal. Two values have given the best results. On the one hand, poorer models tend to favor the $[2-5]$ length while, when more information is available in the models, the best results come from the $[3-7]$ size. In any case, it seems that longer words do no improve the results, and most important of all, even if there are a lot of one-letter words, these do not help a lot when identifying users.

- Recursion: The best results are achieved when no recursion is used whatsoever. This is of high relevance because it confirms the importance of contextual information. Taking the tails of words and searching them again as if these were new and complete words can be considered as *cheating* and so it has been proved by the obtained results.

- Discarding child times: If the node of the last letter of a word has no timing intervals in a tree, using those partial timing intervals from the mean values of the leafs, as opposed to discarding the word, improves the results. This improvement is substantial, proving, again, how important contextual information is.

- Number of words: This is a controversial parameter. The results prove, as in all tests performed up to this point, that the more number of words (or graphs as proved in the initial test), the better the results. The discussion of where to set the threshold could depend on the information available, the desired performance, and the accuracy expected as per the security policy established.

5.3.7 Feature selection

Table 5.13 sets the parameters to those that have given the best results in this test, but only focuses on the feature that *most of the time* has given the best results. As in the previous test, the features that have given the best and worst performance were the Release–Press (RP) feature a 74.16% of the times, and the Press–Release (PR) a 91.66% of the times, respectively.

As in the first test, it can be seen that the feature used is an important decision not to be left to chance. In this case, the improvement is of more than an 8%. Still, to be able to have a margin of decision all features are combined even if this goes against the accuracy of the whole system. The possibility of relying in only one feature, or a particular combination, is always open.

5.3.8 Performance evaluation

After having set the parameters using the results from Test 1, the building of tree models becomes rather straightforward. It certainly shows an improvement in performance.

Recursion	Words	Length of words					
		All	≥ 2	≥ 3	$[2-5]$	$[2-7]$	$[3-7]$
	1	59.60	59.60	67.55	67.53	64.82	70.72
	10	62.17	62.17	69.25	70.64	67.69	72.54
$R0$	25	66.03	66.03	73.58	75.20	71.89	77.14
	50	69.89	69.89	78.05	79.18	75.76	81.58
	75	71.83	71.83	80.01	81.06	77.72	**83.72**
	1	59.53	59.53	67.51	67.50	64.75	70.66
	10	62.20	62.20	69.28	70.72	67.70	72.55
$R1$	25	66.08	66.08	73.67	75.24	71.93	77.18
	50	69.99	69.99	78.19	79.21	75.86	81.70
	75	71.84	71.84	80.05	81.08	77.79	**83.73**
	1	59.76	59.76	65.54	67.06	64.48	68.53
	10	64.68	64.68	69.58	72.89	70.02	72.92
$R2$	25	70.30	70.30	75.40	78.65	75.77	78.91
	50	74.02	74.02	79.63	82.15	79.63	83.31
	75	76.16	76.16	81.07	83.99	81.36	**84.79**

Table 5.13: Period 0 ($P0$) – Not discarding child times results using the RP feature

At the same time, knowing which are the parameters that better classify users, can help a lot in building smaller models and performing *much* fewer calculations.

A clear example of this is the Recursion parameter. If no recursion is used the size of the results that have to be later studied is reduced by almost a half. This clearly speeds the global procedure and alleviates the system of having to perform calculations to obtain distance measurement that bring no improvement whatsoever to the results.

5.3.9 Test 2 summary

This second test has focused on the parameters used to store words in the Combined logical tree model. These parameters include the depth at which words are found in the model, whether recursion is needed, whether node values should be inferred from those in the leafs in case no information is found for a partially found word, and the minimum necessary number of words found in the model.

Many results have been obtained from the evaluation of these parameters. The fact that contextual information is of high relevance, as proven by the fact that recursion is not necessary when a minimum number of words are used, should be considered paramount. At the same time, the effect of limiting the minimum number of words found in the model is of high relevance because it improves the results significantly.

The depth at which words are found is of interest because it suggests that users show their natural rhythm once a certain number of keystrokes have been submitted. One-letter words bring little to the global accuracy of the system and, at the same time, discarding longer words tends to improve accuracy.

All in all, knowing which are the parameters that better help identify users improves the performance of the system radically since *many* useless calculations can be avoided.

5.4 Test 3 – Distances and methods to identify users

So far, all previous experiments using the logical tree models have been carried out using the Euclidean distance measurement and the Mean of distances method to classify users. The results these two methods have given are far from optimal when compared to the base results obtained using the *n-graphs* methodology. The purpose of this test is to try different popular distance measurements (described in Section 4.4.3) and, at the same time, try the different proposed methods to identify users (described in Section 4.4.4) to improve global accuracy.

5.4.1 Initial model parameters

A fair number of different parameters have been already analyzed. For the current test, the settings that have already given the best results are used to test the different distance measurements and the methods to identify users. More specifically, no limits will be set when building the Combined tree model, and all instances of words will be cleaned using 2 standard deviations. At the same time, and given the fact that no real consensus had been achieved as per the word length, both the $[2-5]$ and $[3-7]$ ranges will be tried. It is worth noting that when evaluating the Weighted mean of distances, revised method, the word length parameter also includes the $>= 2$ setting. This has been decided because this method is the only one that takes advantage of the Depth at which a word is found in a model. The minimum number of words found on the model will be 50 and 75, if only to see the differences with the quality in words per Session. Using higher values, of course, means that more Sessions may be discarded. No recursion is used and, also, no discarding of child times is performed.

5.4.2 Samples verification methodology

As with the previous test, only one single group of 40 randomly chosen users is used. Again, the tests are repeated 10 times using the MCCV method. For this test, a 70% partition to train the model is used, and the remaining 30% is used to test the different proposed parameters.

5.4.3 Distances and methods evaluated

Five different distances measurements (D) are evaluated in this experiment (examples of how these distance measurements are implemented can be found in Chapter 4):

- Euclidean: $D_E(\vec{X},\vec{Y}) = \sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2}$
- Manhattan (a.k.a. City block): $D_M(\vec{X},\vec{Y}) = \sum_{i=1}^{n}|X_i - Y_i|$
- Canberra: $D_C(\vec{X},\vec{Y}) = \sum_{i=1}^{n}\frac{|X_i - Y_i|}{|X_i|+|Y_i|}$
- Chebyshev: $D_{CH}(\vec{X},\vec{Y}) = max_{i=1}^{n}|X_i - Y_i|$
- *wordgraph*: $D_{wg}(\vec{X},\vec{Y}) = |\sum_{i=1}^{n}X_i - \sum_{i=1}^{n}Y_i|$

At the same time, the methods (M) to identify users are the following:

- Mean of distances
- Median of distances
- Weighted mean of distances
- Higher number of minimum values, for the mean, the median, or the weighted mean, using a voting fusion methodology
- Weighted mean of distances, revised

5.4.4 Number of independent tests performed

The samples are tested using all distance measurements and all methods to identify users. In total, 5 distances are tested using 7 different methods. Three subsets of word lengths (L) are used: $>= 2$, $[2-5]$ and $[3-7]$ and two minimum words found in the model settings (W) are also applied: 50 and 75. Each of the 4 periods has been used and the tests have been repeated 10 times using different sets of 40 randomly chosen users. The grand total number of tests is: $5D \cdot 7M \cdot 3L \cdot 2W \cdot 4P \cdot 10 = 8,400$ tests.

5.4.5 Results for the identification of users test

The results for all these different tests have been separated into sections taking into account the method to identify users being evaluated. The following sections have been structured depending on the characteristics of the chosen method to identify users. The first section comments upon the methods that do not use voting fusion schemes. The second section focuses on voting methods, and the last section deals with the Weighted mean of distances, revised method since it includes a combination of all previous methods evaluated.

Methods that do not use voting fusion schemes

The results in this section are presented per period and commenting upon the different parameters being evaluated. Table 5.14 shows the results for Period 0 ($P0$) for all the distances evaluated and the methods that do not include a voting scheme. These include the Mean of distances, the Median of distances, and the Weighted mean of distances methods. Respectively, Tables 5.15, 5.16, and 5.17 show the results for Periods 1, 2, and 3 ($P1$, $P2$, $P3$, respectively).

Method[1]	Words	Depth	Distance measurement				
			Euclidean	Manhattan	Chebyshev	Canberra	*wordgraph*
MD	50	$[2-5]$	72.56	68.19	72.74	**78.56**	73.82
		$[3-7]$	74.64	70.94	73.67	**78.81**	69.52
	75	$[2-5]$	73.96	69.77	74.12	**80.77**	74.91
		$[3-7]$	76.24	73.23	75.22	**81.36**	71.14
MMed	50	$[2-5]$	88.42	79.56	**90.83**	75.05	86.08
		$[3-7]$	88.42	79.58	**90.24**	77.75	76.87
	75	$[2-5]$	91.02	83.04	**93.18**	78.07	88.28
		$[3-7]$	91.00	83.11	**92.43**	80.74	79.94
WMD	50	$[2-5]$	88.78	79.60	**91.58**	80.37	87.05
		$[3-7]$	89.11	81.44	**91.14**	82.19	76.36
	75	$[2-5]$	91.35	83.12	**93.69**	82.75	89.13
		$[3-7]$	91.52	84.99	**93.24**	84.90	79.54

[1] MD – Mean of distances; MMed – Mean of medians; WMD – Weighted mean of distances

Table 5.14: Period 0 ($P0$) – Distances and Methods without voting

From the three methods proposed, the Mean of distances is always the one the yields the worst results. By comparing the use of the median and the mean methods it is confirmed that the first is far better than the second and it also confirms that when the distribution of values is highly skewed (as is the case) the median is a much better measurement to classify users.

As per the distance measurement that works better, the Chebyshev distance seems to behave better in the Mean of medians and the Weighted mean of distances methods, but not in the Mean of distances method. In this case, the Canberra distance measurement gives the best results.

Finally, the number of minimum words found on a session determines the best results once again, better sessions means better results, but an interesting fact is seen when looking at the Depth at which words are found in the model. The $[3-7]$ interval is not always the best one, especially when using the Chebyshev distance, which tends to favor the $[2-5]$ depth interval.

Method[1]	Words	Depth	Distance measurement				
			Euclidean	Manhattan	Chebyshev	Canberra	*wordgraph*
MD	50	$[2-5]$	53.31	43.86	**55.13**	54.80	53.82
		$[3-7]$	51.32	43.55	51.36	**52.33**	42.97
	75	$[2-5]$	55.15	45.73	**56.76**	56.21	53.17
		$[3-7]$	53.68	45.98	53.72	**54.33**	43.11
MMed	50	$[2-5]$	71.33	56.21	**76.77**	56.91	74.41
		$[3-7]$	67.10	52.80	**71.38**	53.65	53.66
	75	$[2-5]$	73.91	59.41	**79.13**	59.34	76.71
		$[3-7]$	70.00	55.50	**74.49**	56.09	55.69
WMD	50	$[2-5]$	72.82	58.42	**78.28**	58.46	76.99
		$[3-7]$	68.78	57.32	**73.70**	57.49	53.31
	75	$[2-5]$	75.85	61.98	**80.92**	60.52	79.16
		$[3-7]$	71.93	60.44	**76.99**	59.95	56.07

[1] MD – Mean of distances; MMed – Mean of medians; WMD – Weighted mean of distances

Table 5.15: Period 1 ($P1$) – Distances and Methods without voting

Method[1]	Words	Depth	Distance measurement				
			Euclidean	Manhattan	Chebyshev	Canberra	*wordgraph*
MD	50	$[2-5]$	66.98	53.67	**69.21**	67.09	67.67
		$[3-7]$	68.46	59.50	68.90	**70.16**	62.08
	75	$[2-5]$	68.90	55.50	**71.10**	69.13	69.16
		$[3-7]$	70.48	61.66	71.00	**72.99**	64.16
MMed	50	$[2-5]$	84.51	69.27	**88.26**	66.33	84.55
		$[3-7]$	81.72	69.72	**85.21**	69.73	70.91
	75	$[2-5]$	87.15	72.19	**90.60**	68.60	86.72
		$[3-7]$	84.87	72.91	**88.04**	72.70	74.20
WMD	50	$[2-5]$	85.22	70.61	**89.34**	70.09	85.72
		$[3-7]$	83.13	73.32	**86.25**	75.61	70.01
	75	$[2-5]$	87.79	73.53	**91.74**	72.32	87.76
		$[3-7]$	86.03	76.54	**88.92**	78.53	73.26

[1] MD – Mean of distances; MMed – Mean of medians; WMD – Weighted mean of distances

Table 5.16: Period 2 ($P2$) – Distances and Methods without voting

Method[1]	Words	Depth	Distance measurement				
			Euclidean	Manhattan	Chebyshev	Canberra	*wordgraph*
MD	50	$[2-5]$	59.70	51.55	**61.27**	60.21	60.00
		$[3-7]$	57.48	49.37	57.91	**59.16**	52.32
	75	$[2-5]$	60.81	53.09	62.45	**63.15**	61.53
		$[3-7]$	59.30	51.91	59.47	**63.30**	54.23
MMed	50	$[2-5]$	75.55	62.65	**80.13**	59.77	77.35
		$[3-7]$	73.48	60.67	**76.88**	61.12	62.31
	75	$[2-5]$	79.66	67.20	**84.14**	63.29	80.68
		$[3-7]$	77.40	64.86	**80.57**	65.51	65.91
WMD	50	$[2-5]$	76.63	64.44	**81.47**	63.90	79.13
		$[3-7]$	74.57	64.42	**78.40**	64.78	61.88
	75	$[2-5]$	80.49	68.89	**84.96**	67.36	82.76
		$[3-7]$	78.49	69.02	**82.26**	69.38	66.16

[1] MD – Mean of distances; MMed – Mean of medians; WMD – Weighted mean of distances

Table 5.17: Period 3 ($P3$) – Distances and Methods without voting

Methods that use voting fusion schemes

This section focuses on the Higher number of minimum values. This is a voting fusion scheme and it has been applied to the three methods tested in the previous section, that is, the Mean of distances, the Median of distances, and the Weighted mean of distances.

Table 5.18 shows the results for Period 0 ($P0$) for all the distances evaluated and the methods that include a voting scheme. Respectively, Tables 5.19, 5.20, and 5.21 show the results for Periods 1, 2, and 3 ($P1$, $P2$, $P3$, respectively).

From the obtained results, most of the conclusions from the methods that did not use a voting fusion scheme can still be applied. In this case, the results tend to be better, especially those where a voting scheme, the median value and the weighted values are used. Again, the worst results are obtained with the Mean of distances even if results improve a lot when using the voting scheme instead of combining the average value of all evaluated features.

It is difficult to choose a distance measurement that works best all the time. If only this test (using a voting fusion scheme) had been carried out, a sort of tie could be established between the Chebyshev, the Canberra and the *wordgraph* distances. On the other hand, the tests without a voting scheme have determined the Chebyshev as the best one. Again, for the tests where a voting scheme is used, it is the best in many cases.

The length of words is something that, again, catches the eye. The $[2-5]$ interval,

Method[1]	Words	Depth	Distance measurement				
			Euclidean	Manhattan	Chebyshev	Canberra	*wordgraph*
MIN	50	$[2-5]$	75.49	71.74	75.80	**83.43**	82.78
		$[3-7]$	78.39	74.95	77.65	**83.36**	80.48
	75	$[2-5]$	76.67	73.08	76.94	**85.29**	83.31
		$[3-7]$	79.79	77.22	78.98	**85.61**	81.41
MINMed	50	$[2-5]$	92.47	85.12	**94.53**	82.73	94.62
		$[3-7]$	92.14	85.12	**94.06**	84.08	89.15
	75	$[2-5]$	94.52	88.06	**96.20**	85.33	95.82
		$[3-7]$	94.17	88.22	**95.61**	86.74	91.20
WMIN	50	$[2-5]$	92.80	85.38	**95.07**	86.33	94.58
		$[3-7]$	92.89	86.61	**94.55**	87.56	88.36
	75	$[2-5]$	94.95	88.25	**96.56**	88.35	95.83
		$[3-7]$	94.73	89.70	**96.07**	89.64	90.64

[1] MIN – Voting with the Mean; MINMed – Voting with the Median; WMD – Voting with the Weighted mean

Table 5.18: Period 0 ($P0$) – Distances and Methods using fusion

Method[1]	Words	Depth	Distance measurement				
			Euclidean	Manhattan	Chebyshev	Canberra	*wordgraph*
MIN	50	$[2-5]$	56.49	47.65	58.23	62.37	**67.36**
		$[3-7]$	55.15	48.15	55.21	**59.30**	55.73
	75	$[2-5]$	58.42	50.02	59.63	63.52	**66.34**
		$[3-7]$	57.98	50.80	57.76	**60.91**	55.79
MINMed	50	$[2-5]$	78.34	63.57	**83.09**	65.85	87.72
		$[3-7]$	74.13	60.53	**78.37**	62.08	68.93
	75	$[2-5]$	80.22	65.99	**85.10**	68.09	89.32
		$[3-7]$	77.01	63.04	**81.22**	64.09	69.89
WMIN	50	$[2-5]$	79.29	65.14	83.62	66.98	**90.15**
		$[3-7]$	76.31	65.03	**80.17**	65.30	67.58
	75	$[2-5]$	81.99	68.37	85.92	68.60	**91.21**
		$[3-7]$	79.31	67.93	**83.20**	67.72	70.53

[1] MIN – Voting with the Mean; MINMed – Voting with the Median; WMD – Voting with the Weighted mean

Table 5.19: Period 1 ($P1$) – Distances and Methods using fusion

Method[1]	Words	Depth	Distance measurement				
			Euclidean	Manhattan	Chebyshev	Canberra	*wordgraph*
MIN	50	$[2-5]$	70.81	57.63	72.53	74.09	**77.58**
		$[3-7]$	72.39	64.73	72.70	**76.88**	74.88
	75	$[2-5]$	72.73	59.64	74.48	76.00	**78.26**
		$[3-7]$	74.35	66.89	74.55	**79.94**	76.67
MINMed	50	$[2-5]$	89.65	76.78	92.12	76.51	**93.30**
		$[3-7]$	86.33	76.86	**89.37**	79.23	83.94
	75	$[2-5]$	91.78	79.59	94.08	79.17	**94.61**
		$[3-7]$	89.51	79.73	**91.78**	82.51	86.82
WMIN	50	$[2-5]$	90.39	77.91	92.92	78.30	**93.66**
		$[3-7]$	87.74	79.67	**90.16**	82.95	82.64
	75	$[2-5]$	92.55	80.79	95.03	80.53	**95.04**
		$[3-7]$	90.43	82.67	**92.47**	85.65	85.36

[1] MIN – Voting with the Mean; MINMed – Voting with the Median; WMD – Voting with the Weighted mean

Table 5.20: Period 2 ($P2$) – Distances and Methods using fusion

Method[1]	Words	Depth	Distance measurement				
			Euclidean	Manhattan	Chebyshev	Canberra	*wordgraph*
MIN	50	$[2-5]$	62.91	55.50	64.66	67.81	**72.79**
		$[3-7]$	61.23	52.92	61.65	65.36	**66.38**
	75	$[2-5]$	63.92	57.13	65.97	70.97	**74.26**
		$[3-7]$	62.66	55.51	62.83	**69.53**	68.01
MINMed	50	$[2-5]$	81.63	69.29	85.63	68.80	**89.20**
		$[3-7]$	79.42	67.92	**82.71**	69.09	78.41
	75	$[2-5]$	85.47	73.68	89.24	72.24	**91.45**
		$[3-7]$	83.03	72.31	**85.79**	73.69	81.15
WMIN	50	$[2-5]$	82.74	71.32	86.63	71.67	**89.94**
		$[3-7]$	80.87	71.04	**83.73**	72.61	76.93
	75	$[2-5]$	86.40	76.04	90.00	75.05	**92.22**
		$[3-7]$	84.38	75.72	**87.12**	77.25	80.86

[1] MIN – Voting with the Mean; MINMed – Voting with the Median; WMD – Voting with the Weighted mean

Table 5.21: Period 3 ($P3$) – Distances and Methods using fusion

when not using the mean, is the best length interval. Also, the number of minimum words, when using a voting scheme, sets the bar to determine the best results: sessions with more words imply better results.

The weighted mean of distances, revised

The last section of this experiment focuses solely on the Weighted mean of distances method, revised. This method has been built on the experience gained from all the previous methods. This method uses only a small set of features for every test. For the current method, only the Press–Release (PR), Release–Press (RP), Press–Press (PP), and the Release–Release (RR) feature has been used. This means that the 4 values are used in a single combination to obtain the md and the wmd values as explained in Chapter 4.

Different types of fusion methods have been used when evaluating this method:

- Use fusion by setting two different thresholds: the first threshold value accepts the session if only one of the two values (gd_{med} or gd_{wmed}) accepts it as valid, while the more restrictive one needs both values to report the session as valid. These methods are labeled in the results as *Loose*, and *Strict* respectively.
- Do not use fusion and use only either the median or the weighted median values to report the number of correctly identified sessions. These methods are labeled in the results as gd_{med} or gd_{wmed} respectively.

The results, again, are shown per period, but in this case, as the Depth is also part of the methodology of the method to obtain the results, the possibility to use all depths from 2 upwards (≥ 2) has also been considered. Tables 5.22, 5.23, 5.24, and 5.25 show the different obtained values when using this method.

From the results presented in this final section the following can be established:

- The Weighted mean of distances, revised is the best method so far.
- When using this method, the best distance measurement, if sometimes tied with the Euclidean distance, is the Chebyshev distance measurement.
- At the same time, and since this method is the only one that uses the Depth as part of its methodology, using all relevant depths is a must.

It could be argued that only words with at least two letters are used. Seeing that depth is so important when using this method, why not use one-letter words? The answer to this question deals with the features chosen for this test. From the four

Fusion	Words	Depth	Distance measurement				
			Euclidean	Manhattan	Chebyshev	Canberra	*wordgraph*
Strict	50	$[2-5]$	**95.70**	92.56	95.66	88.69	93.14
		$[3-7]$	**94.21**	92.72	94.16	86.38	89.65
		≥ 2	97.24	92.24	**97.60**	82.52	94.19
	75	$[2-5]$	97.19	94.68	**97.20**	90.45	94.03
		$[3-7]$	95.75	94.51	**95.86**	88.19	91.34
		≥ 2	98.39	94.32	**98.55**	84.85	95.01
Loose	50	$[2-5]$	**96.75**	94.96	96.56	92.25	95.10
		$[3-7]$	**96.23**	95.60	95.56	94.02	92.97
		≥ 2	98.24	95.22	**98.44**	92.11	96.09
	75	$[2-5]$	**97.96**	96.44	97.87	93.76	95.76
		$[3-7]$	**97.29**	96.81	97.06	95.15	94.32
		≥ 2	99.08	96.55	**99.17**	93.70	96.67
gd_{med}	50	$[2-5]$	**96.48**	94.45	96.23	92.11	94.04
		$[3-7]$	**95.62**	94.51	95.07	93.55	91.40
		≥ 2	98.00	94.69	**98.22**	91.92	95.20
	75	$[2-5]$	**97.81**	96.12	97.60	93.61	94.83
		$[3-7]$	**96.88**	95.96	96.58	94.79	92.85
		≥ 2	98.95	96.22	**99.02**	93.49	95.88
gd_{wmed}	50	$[2-5]$	95.97	93.07	**96.00**	88.82	94.19
		$[3-7]$	**94.82**	93.82	94.65	86.85	91.22
		≥ 2	97.48	92.78	**97.82**	82.71	95.08
	75	$[2-5]$	97.33	94.99	**97.47**	90.60	94.95
		$[3-7]$	96.15	95.35	**96.33**	88.55	92.81
		≥ 2	98.52	94.65	**98.71**	85.06	95.80

Table 5.22: Period 0 ($P0$) – Weighted mean of distances, revised

Fusion	Words	Depth	Distance measurement				
			Euclidean	Manhattan	Chebyshev	Canberra	*wordgraph*
Strict	50	$[2-5]$	88.81	80.90	**91.14**	80.70	86.64
		$[3-7]$	83.39	76.46	**83.47**	70.91	74.57
		≥ 2	92.00	76.06	**94.15**	70.33	90.64
	75	$[2-5]$	90.84	83.30	**92.25**	81.58	87.18
		$[3-7]$	85.38	78.39	**85.99**	72.28	75.85
		≥ 2	93.59	78.56	**94.79**	71.47	90.29
Loose	50	$[2-5]$	90.87	85.53	**92.78**	84.95	91.22
		$[3-7]$	**87.58**	84.38	86.18	81.82	80.44
		≥ 2	94.60	83.10	**96.03**	81.29	93.57
	75	$[2-5]$	92.47	87.75	**93.67**	85.86	91.15
		$[3-7]$	**89.51**	86.66	88.42	83.03	82.17
		≥ 2	95.69	84.70	**96.12**	81.90	93.06
gd_{med}	50	$[2-5]$	90.39	84.32	**92.40**	84.89	89.06
		$[3-7]$	**86.32**	81.07	85.66	81.48	78.41
		≥ 2	94.31	82.04	**95.80**	81.17	92.39
	75	$[2-5]$	91.96	86.70	**93.27**	85.78	89.38
		$[3-7]$	**88.31**	83.23	87.96	82.74	79.93
		≥ 2	95.62	83.80	**95.83**	81.83	91.87
gd_{wmed}	50	$[2-5]$	89.30	82.10	**91.52**	80.76	88.80
		$[3-7]$	**84.65**	79.77	84.00	71.25	76.61
		≥ 2	92.29	77.12	**94.38**	70.45	91.81
	75	$[2-5]$	91.36	84.35	**92.65**	81.65	88.94
		$[3-7]$	**86.59**	81.81	86.45	72.56	78.08
		≥ 2	93.66	79.46	**95.08**	71.54	91.48

Table 5.23: Period 1 ($P1$) – Weighted mean of distances, revised

Fusion	Words	Depth	Distance measurement				
			Euclidean	Manhattan	Chebyshev	Canberra	*wordgraph*
Strict	50	$[2-5]$	94.94	91.52	**95.22**	87.38	91.16
		$[3-7]$	92.08	88.21	**92.63**	85.12	85.70
		≥ 2	96.17	89.22	**96.46**	79.76	92.12
	75	$[2-5]$	96.53	93.52	**96.72**	89.18	92.46
		$[3-7]$	94.41	91.07	**94.86**	87.63	88.24
		≥ 2	**97.52**	91.64	97.44	81.99	93.12
Loose	50	$[2-5]$	95.68	93.57	**95.76**	92.08	93.54
		$[3-7]$	**94.08**	91.64	93.91	91.85	89.54
		≥ 2	97.19	93.01	**97.36**	90.95	94.05
	75	$[2-5]$	**97.11**	95.15	97.10	93.88	94.49
		$[3-7]$	**95.97**	93.91	95.85	93.72	91.39
		≥ 2	**98.31**	94.99	98.27	92.84	94.82
gd_{med}	50	$[2-5]$	95.45	93.11	**95.62**	92.00	92.17
		$[3-7]$	93.42	90.59	**93.48**	91.47	88.41
		≥ 2	96.97	92.54	**96.98**	90.83	92.93
	75	$[2-5]$	96.92	94.80	**97.01**	93.79	93.24
		$[3-7]$	95.41	92.94	**95.52**	93.33	90.39
		≥ 2	**98.12**	94.64	97.90	92.71	93.64
gd_{wmed}	50	$[2-5]$	95.16	91.98	**95.36**	87.47	92.53
		$[3-7]$	92.74	89.26	**93.06**	85.49	86.84
		≥ 2	96.40	89.70	**96.84**	79.88	93.25
	75	$[2-5]$	96.73	93.88	**96.82**	89.27	93.71
		$[3-7]$	94.97	92.03	**95.19**	88.01	89.23
		≥ 2	97.71	91.99	**97.81**	82.12	94.29

Table 5.24: Period 2 ($P2$) – Weighted mean of distances, revised

Fusion	Words	Depth	Distance measurement				
			Euclidean	Manhattan	Chebyshev	Canberra	*wordgraph*
Strict	50	$[2-5]$	89.91	83.97	**90.64**	77.69	87.78
		$[3-7]$	84.98	81.20	**85.16**	74.71	79.94
		≥ 2	93.20	80.53	**93.48**	71.06	89.91
	75	$[2-5]$	92.75	87.97	**93.19**	80.58	89.58
		$[3-7]$	87.62	84.66	**87.98**	78.42	82.44
		≥ 2	95.35	85.04	**95.49**	74.35	91.12
Loose	50	$[2-5]$	91.61	88.11	**92.06**	84.40	92.03
		$[3-7]$	88.29	87.06	**88.00**	85.92	84.98
		≥ 2	95.00	86.92	**95.58**	83.26	93.35
	75	$[2-5]$	94.00	91.65	**94.22**	87.00	93.55
		$[3-7]$	**90.58**	90.01	90.28	88.73	86.92
		≥ 2	96.67	90.56	**96.84**	86.34	94.17
gd_{med}	50	$[2-5]$	91.23	87.10	**91.61**	84.09	90.61
		$[3-7]$	**87.63**	84.53	87.14	85.04	83.05
		≥ 2	94.62	85.60	**95.11**	83.11	91.90
	75	$[2-5]$	93.77	90.77	**93.97**	86.80	92.30
		$[3-7]$	**90.01**	87.74	89.57	88.00	85.13
		≥ 2	96.39	89.70	**96.60**	86.24	92.91
gd_{wmed}	50	$[2-5]$	90.28	84.98	**91.09**	77.99	89.20
		$[3-7]$	85.64	83.73	**86.01**	75.59	81.87
		≥ 2	93.58	81.85	**93.95**	71.21	91.36
	75	$[2-5]$	92.98	88.85	**93.44**	80.78	90.83
		$[3-7]$	88.19	86.94	**88.70**	79.14	84.24
		≥ 2	95.64	85.90	**95.73**	74.45	92.38

Table 5.25: Period 3 ($P3$) – Weighted mean of distances, revised

chosen features (Press–Release (PR), Release–Press (RP), Press–Press (PP), and the Release–Release (RR)), only the Press–Release (PR) can be obtained for one-letter words. The other three can only be obtained when at least two letters are available. With this in mind, only words from two letters and up have been used. Also, a previous experiment has already shown that using all lengths had not given the best results in any case, but then again, seeing how different parameters can affect different decisions on how to build the models and treat the samples a modification of the proposed method could be tried using only the PR value. This is left as future work.

All in all, when using this method, the Chebyshev distance, and Depths of 2 or more, results are found to be better than the *n-graphs* alternative. All that is needed to be discussed is the type of fusion that should be used when using the Weighted mean of distances, revised method. Table 5.26 shows a summary of the best results in this section. It can be seen that when restricting to those sessions with at least 75 words found in the model, the method behaves better, as has been the case all throughout the results chapter. Given the little increase in accuracy from going from 50 to 75 words, it should be discussed if it worth discarding all the sessions with less words found, but still relevant. This would probably be a decision of the production environment particularities where this biometric technique is implemented and the expected accuracy. Also, the differences in size of every period are apparent. The better the data, in size, of the period, the better the results. Smaller groups still get the worst results.

Fusion	Words	Period			
		$P0$	$P1$	$P2$	$P3$
Strict	50	97.60	94.15	96.46	93.48
	75	98.55	94.79	97.44	95.49
Loose	50	98.44	96.03	97.36	95.58
	75	99.17	96.12	98.27	96.84
gd_{med}	50	**98.22**	**95.80**	**96.98**	**95.11**
	75	99.02	95.83	97.90	96.60
gd_{wmed}	50	97.82	94.38	96.84	93.95
	75	98.71	95.08	97.81	95.73

Table 5.26: Summary of the method using the Chebyshev distance

As per the fusion method that should be used, if only the results shown in Table 5.26 were to be evaluated, it would be obvious that the best choice would be to use a *Loose* setting, where, if only one of the values, either the gd_{med} or the gd_{wmed} is reported

as a valid user, this one should be used. This setting, though, presents a problem. It cannot be known if one of the two is certain that the user is valid, so a random choice has to be done between the two. This could decrease the effectiveness of the method, so the *Loose* setting is rather misleading. The next in quality, in all cases, is to use only the values reported by the gd_{med} feature, without using any fusion at all. If this setting was to be used two things should be taken into account: no weighting of features and no fusion would be necessary. Only a simple median value of the mean of the 4 chosen features (Press–Release (PR), Release–Press (RP), Press–Press (PP), and the Release–Release (RR)). These results have been marked in bold in Table 5.26. Choosing these parameters simplifies greatly the calculations necessary to obtain the results and, at the same time, no big compromises regarding the number of discarded sessions have to be carried out.

5.4.6 Cleaning sessions of large values

After having tried all methods and distance measurements, and having a procedure that yields good enough results, the idea of initially cleaning the sessions of values that are way too far from zero has been evaluated. The main idea behind this approach can be explained by Figure 4.13. As can be seen in this figure, the vast majority of distance values obtained from the model range from 0 to 300.

A quick test has been performed using a minimum of 50 words found in the model and the Weighted mean of distances, revised method to determine if omitting the values above a given threshold also helps focus on the real matter at hand, that is, the distances closer to zero. The results of evaluating a threshold from 300 to 600 milliseconds samples are shown in Table 5.27.

Period	Threshold (ms)				
	None	300	400	500	600
$P0$	98.22	98.14	**98.23**	98.19	98.19
$P1$	95.80	96.02	**96.14**	95.91	95.97
$P2$	96.98	97.12	**97.22**	97.15	97.09
$P3$	95.11	**95.25**	95.22	95.14	95.17

Table 5.27: Omitting large sample values

The best results are achieved when using a threshold value close to 400 milliseconds. In this case, there is a slight improvement over the base results previously obtained in the test performed in this section, that should be taken into account when building quality and robust models.

5.4.7 Test 3 summary

Instead of studying model related features, as with previous tests, the focus of this experiment has been set on methods to increase accuracy by trying different distance measurements and the different proposed methods to determine the owner of a session.

Five different distances measurements and seven different methods to identify users have been evaluated. Three of these methods did not use fusion schemes, three did, and the last one combined what had been learned from the others.

The results have shown that there is no best distance measurement that outshines the others all the time. Different methods favor different distances. On the other hand, the best method has been the Weighted mean of distances, revised method. This method has been evaluated using fusion schemes and using only mean and median values. The best methodology when applying this method is to choose the median value.

When evaluating this method, the best distance measurement has been the Chebyshev one most of the time, even if the Euclidean alternative also yields good results. It is worth mentioning that, since this method made use of the Depth at which have been found in the model, the best results have been obtained when all word lengths, from two-letter words upward, have been used.

As with previous tests, having better sessions, understood as those that have a relatively high number of words, gives the best results. Also, as an additional test, the possibility of discarding those values above a certain threshold from the distances between a session and a model has also been evaluated. The results have shown an improvement if values above $400ms$ are suppressed.

The following tests, once a good method to determine the owner of a session has been established, center the efforts on finding whether *other* features related to user behavior help improve accuracy, especially in those cases where a smaller dataset is available. It should be noted that the first three tests have been rather exhaustive in terms of combination of parameters, methods and distance methods. Compared to the tests that follow this is going to be found rather scarce or simple. These tests have evaluated a large number of parameters and many results have been discussed to determine the optimal conditions in which the proposed model performed better. From this moment on, the proposed tests are simpler because a good method to identify users has already been established. The parameters that are tested from this point are evaluated to see if these improve the global accuracy of the system in a relevant way. An exception to this fact, though, is Test 7 that tries many different combinations to determine if age group and gender are relevant when identifying or authenticating users.

5.5 Test 4 – Features related to user behavior

Up to this point, a good method to classify users and the corresponding distance measurement has been identified. Many parameters have also been proved to work fine with the proposed setting and a particular number of words have been deemed optimal to have a good overall system accuracy.

The following test tries to determine if some easy to implement behavioral features could improve the accuracy of the system, especially in those cases where the number of sessions is limited, the data available is scarce or the chance to use sessions above a certain number of words is not possible.

The current test evaluates the following four different features:

- Incorporating the mistakes users make into the models, trying to see if the fact that a user always follows the same key sequence to correct a mistake, thus forming a *special* type of word, could help improve results.
- Using only the space key as word delimiter, trying to see if other key sequences, like navigation keys, are also part of the way a user is defined by their typing pattern.
- See if rewarding distances obtained from the model when words are used frequently also helps improve accuracy.
- Finally, determine if a user will use typical sentence constructions and, again, reward these successive words as a means of improving the overall accuracy of the system.

It could be argued that these last two features are not directly related to Keystroke Dynamics. These features are more related to user behavior rather than to their typing rhythm, but nonetheless, the fact that the possibility of detecting such behavior can lead to modifying the underlying features directly related to Keystroke Dynamics can provide a way of having a better method to classify users. Even if the proposed methodology for this test could be considered as a way of introducing a multimodal scheme, the chosen features are not considered to be part of any other biometric technique so the whole system is still considered to be unimodal.

5.5.1 Behavioral features

The effect of mistakes as a feature

For this particular test, the errors or mistakes users make are incorporated into the model as if the letter sequence to type a word, commit a mistake and correct it, was a

single word. If this pattern is constantly repeated a more accurate way of identifying users could be provided. The problem when applying this methodology can be that hit rate may be too low when searching these *special* words. How much the accuracy of the system can be improved by using this methodology is what this test tries to find out.

Using only the space as a word delimiter

Instead of only adding the *backspace* key into the model, by modifying the delimiters that establish what a *word* is, this test tries to see whether only using the *space* key as a word delimiter the overall accuracy improves[5]. The user may use special key combinations, navigation keys... when authoring a message. These combinations, so far, have been always discarded in previous tests and models. The question, again, is if the hit rate will be high enough to make this feature relevant.

Frequency scaling

This test will reward (or punish) those words users type more frequently or, on the contrary, do not usually submit. At the same time, those sequences of words the user types more frequently will also be rewarded. Following the reasoning explained in Section 4.4.3, the user will benefit from using the same words and sentences over and over again.

5.5.2 Initial model parameters

For the tests in this section, two independent tests will be carried out. One with all the sessions available and the other only with those sessions that have at least 50 words found in the model (W). The distance measurement will be the Chebyshev one, and the method to classify users will be the Weighted mean of distances, revised using the gd_{med} feature. These parameters have been chosen because these are the ones that have given the best results in previous tests. At the same time, 50 words have been considered enough to have a valid system, without discarding too many sessions.

5.5.3 Number of independent tests performed

As with the previous test, only one single group of 40 randomly chosen users is used. Again, the tests will be repeated 10 times using the MCCV method. For this test, a 70% partition to train the model is used, and the remaining 30% is used to test the different proposed parameters.

[5]It should be noted that the $300ms$ silence interval is still used to determine other word breaks.

It is worth noting that an additional test combining all four behavioral features chosen is also performed, bringing the number of tests to evaluate Features (F) to 5. The purpose of this additional test is to see whether by having all suggested features in the model, the total hit rate can be added together to affect accuracy positively.

The grand total number of independent tests for the current experiment is $5F \cdot 4P \cdot 10 \cdot 2W = 400$ tests. F refers to the Feature being analyzed, and W refers to the sets of sessions delimited by the minimum word count.

5.5.4 Results when evaluating user behavior

The results for this experiment are shown in Table 5.28. It can be seen that the use of the proposed methods adds little to the accuracy already achieved by previous tests if it does not decrease it.

Period	Words	Scaling algorithm					
		None	Words	Sentences	Mistakes	Space	All
$P0$	1	85.60	**86.06**	85.59	84.22	82.20	82.56
	50	**98.22**	**98.22**	98.04	98.15	97.60	97.33
$P1$	1	81.65	80.94	**81.77**	78.97	76.79	76.46
	50	**95.80**	94.96	95.68	94.50	93.27	92.84
$P2$	1	82.50	**82.84**	82.59	80.98	79.65	79.75
	50	96.98	**97.01**	96.77	96.74	96.28	95.88
$P3$	1	**78.44**	78.31	78.34	77.35	74.60	74.17
	50	**95.11**	94.73	95.01	94.43	93.73	92.66

Table 5.28: Frequency scaling results

More specifically, the use of the Word frequency scaling algorithm does improve the results, in some cases, but only marginally and only when the hit rate is increased by a larger number of words (Periods 0 and 2). In periods where the number of words is lower, the proposed methods achieve nothing as initially intended. This goes against the goal that had been initially set.

The use of the Sentence scaling algorithm does not improve the results substantially. Again, this can be though as a problem of having enough samples that really take advantage of this method.

The final two methods, the use of mistakes, and the use of only the *space* key as a delimiter show an interesting fact. Both only accomplish one things: yielding worse results. It is interesting to see, though, that as soon as more and more *special* keys are

allowed in the model, the results worsen even more. This could suggest the possibility of going exactly the other way around and focusing the efforts on finding those keys that really influence the results for the better. For example, the use of numbers, should they be allowed in the model? What about upper case letters used in combination with the SHIFT key? Do these slow the rhythm of a user? The possibility of choosing a smaller set of keys is left as future work, but it is considered as a very interesting experiment that should be performed.

The worst results are obtained when all these modifiers are applied at the same time. As pointed before in this chapter, this is considered to be of high relevance. Having specific quality data is much more important than having lots of data.

In the end, as soon as more and more information is added into the tree, how words were previously detected may have been affected. In other words, some words that previously were detected as unique words are now separated into different nodes, reducing the count and valid information per node. This suggests that having such information renders the logical tree unclean, as opposed to the conclusions presented in the initial tests where gathering relevant information in similar words was more interesting.

5.5.5 Test 4 summary

The main goal behind this test has been to find if, by taking advantage of behavioral features, the chance of improving the accuracy of the system can be achieved. Two main features have been evaluated: frequency and word delimiters. The first tries to evaluate the impact of rewarding the words, or sentences, more frequently used. The second has to do with *other* keys user may use frequently and that initially had not been considered as part of the logical tree models.

In most cases, when these features have been evaluated by themselves or combined, the accuracy has decreased. Only in some cases, when the dataset was large enough, some methods have taken advantage and improved accuracy by a small margin. The main issue that has been identified with the proposed methodology is the fact that the hit rate is too low to be able to take advantage of the proposed features. Being this the case, it is proposed not to implement the proposed behavioral measures.

5.6 Test 5 – User group size

Once all parameters have been studied and a process to identify a user from a session has been determined, the question against how many models should a testing session be compared to appears frequently.

So far all tests have been carried out using either groups of 20 users, or random groups of 40. When using groups of 20 users, these have been selected using a ranking that favors the best. On the other hand, when using a group of 40 users, these have been selected randomly among the 60 best from each period.

The current experiment evaluates if there is a threshold number of users from which the results become intolerably bad. To perform this test all 60 users from each period have been considered and 10 different random groups of increasing users have been created to test the accuracy of the system. More specifically, the selected sizes have been: 2, 5, 10, 15, 20, 40, and 60 users.

At the same time, these tests have been performed using all words available or, as in previous tests, only using those sessions that have at least 50 words found in the models.

5.6.1 Number of independent tests performed

The parameters taken into account are the number of words (All, or >= 50) (W) and the size of the group of models (2, 5, 10, 15, 20, 40, and 60 users) (S). The tests have been performed using an MCCV methodology. For this test, though, since in one of the tests all 60 users are used to build the models, it has been thought that two MCCV procedures combined should be used. One of these is used to perform ten 70/30% tests and the other to select also ten different sets of users within each of these partitions to build the groups of models. Again, all four periods (P) have been used.

The total number of tests has been $4P \cdot 2W \cdot 7S \cdot 10 \cdot 10 = 5,600$ tests.

5.6.2 Results for the user group sizes test

Table 5.29 shows the evolution in accuracy once the size of the group of users increases. It can be seen that as soon as the number goes above 20 users, the global accuracy of the system begins to decrease fast. This problem is much more present if all words from all sessions are used. In this case, the accuracy has an important drop as soon as more than 5 users are used to compare sessions. The fact of having better quality sessions not only improves the results but radically slows the process of deterioration when the size of the groups increases.

It should also be pointed out that these results, when using quality sessions in rich periods, show a mean value of accuracy close to a $\sim 97\%$ when using groups of 60 users. This is not the case when all words are used with numbers going down to an unacceptable value of $\sim 80\%$ in accuracy.

These results can also be seen depicted in Figure 5.6. In this figure, it is clearly depicted that the number of users against a session is compared to has a relevant effect

Period	Words	Group size (PCIS)						
		2	5	10	15	20	40	60
$P0$	All	97.31	93.37	91.18	89.84	88.47	85.56	83.87
	50	99.69	99.29	99.17	99.09	98.82	98.19	97.70
$P1$	All	96.80	93.47	90.37	88.23	86.10	82.36	79.53
	50	99.51	98.84	98.37	97.86	97.23	95.83	94.47
$P2$	All	95.97	91.22	88.76	87.18	85.95	82.88	81.15
	50	99.62	98.52	98.23	98.01	97.76	97.35	96.89
$P3$	All	95.41	91.79	87.86	86.00	84.24	80.20	77.20
	50	99.51	99.03	97.97	97.42	96.93	95.51	94.26

Table 5.29: Groups sizes results

on the global accuracy of the system. Also interesting to see is that the size of the Period becomes more and more relevant as soon as the size of the group grows. Those periods with more events ($P0$ and $P2$) maintain a better throughout accuracy than those that have less ($P1$ and $P3$), even if the size of the group increases.

The question at which value should the system be used still remains unanswered. 20 is a good value to maintain a good accuracy, but it may be argued that this value is not secure enough. A group of 40 users has also been proved throughout all the tests in this chapter as a good choice of group size. When comparing this setting against what other researchers have done in the past, this seems to go in accordance with current standards. Using higher values is, of course, a valid possibility, but not only it decreases the global performance of the system but it also decreases the accuracy beyond a point where the biometric measure can no longer be trusted by itself. It could be recommended to include other biometric techniques, thus using a multimodal scheme, to increase the effectiveness without sacrificing the need to compare samples against a large enough number of models.

It would have been interesting to see how the proposed methods behaved with larger user groups. Such tests could have been helpful to determine the feasibility of implementing the proposed solutions in large online learning environments.

For this test, the number of mean Sessions evaluated against the logical tree models across all 10 different runs from the MCCV technique, mean Incorrect and mean Correct identification values, as well as the Margin of Error (ME) with a 95% of confidence are shown in Tables 5.30 and 5.31. The first table shows the values when all sessions have been used, while the second table shows only the results when sessions with at least 50 words found in the models have been used.

Period	Users	Sessions	Incorrect	Correct	ME (%)
	2	46.51	1.24	45.27	4.63
	5	139.80	8.81	130.99	4.03
	10	288.62	25.08	263.54	3.25
$P0$	15	446.79	44.87	401.92	2.79
	20	582.43	66.64	515.79	2.59
	40	1143.50	164.83	978.67	2.04
	60	1687.70	272.20	1415.50	1.75
	2	12.80	0.39	12.41	**9.42**
	5	34.60	2.35	32.25	8.38
	10	71.17	6.91	64.26	6.88
$P1$	15	111.35	13.04	98.31	5.97
	20	147.20	20.34	126.86	5.57
	40	292.28	51.60	240.68	4.37
	60	430.40	88.10	342.30	3.81
	2	22.44	0.65	21.79	6.94
	5	63.59	4.95	58.64	6.59
	10	133.78	14.29	119.49	5.23
$P2$	15	204.06	25.31	178.75	4.52
	20	265.68	36.36	229.32	4.13
	40	528.21	90.03	438.18	3.21
	60	776.60	146.40	630.20	2.75
	2	19.40	0.83	18.57	**9.01**
	5	53.28	4.07	49.21	7.13
	10	112.77	13.33	99.44	5.96
$P3$	15	175.50	24.27	151.23	5.11
	20	227.56	35.64	191.92	4.72
	40	455.10	90.05	365.05	3.66
	60	676.30	154.20	522.10	3.16

Table 5.30: Error when all words are used

Period	Users	Sessions	Incorrect	Correct	ME (%)
	2	29.50	0.09	29.41	1.99
	5	86.79	0.49	86.30	1.58
	10	184.23	1.33	182.90	1.22
$P0$	15	286.97	2.53	284.44	1.08
	20	374.84	4.25	370.59	1.07
	40	732.77	13.16	719.61	0.96
	60	1086.10	25.00	1061.10	0.89
	2	8.12	0.03	8.09	**4.19**
	5	21.79	0.23	21.56	**4.29**
	10	44.28	0.69	43.59	3.65
$P1$	15	69.63	1.44	68.19	3.34
	20	91.83	2.47	89.36	3.31
	40	183.59	7.69	175.90	2.90
	60	268.50	14.90	253.60	2.74
	2	14.21	0.03	14.18	2.40
	5	40.21	0.39	39.82	3.03
	10	86.92	1.30	85.62	2.55
$P2$	15	133.32	2.28	131.04	2.20
	20	173.70	3.45	170.25	2.07
	40	343.18	9.08	334.10	1.70
	60	505.60	15.90	489.70	1.52
	2	10.62	0.04	10.58	3.68
	5	31.51	0.26	31.25	3.16
	10	66.07	1.21	64.86	3.23
$P3$	15	103.55	2.55	101.00	2.99
	20	133.23	3.97	129.26	2.89
	40	265.66	11.84	253.82	2.48
	60	394.10	22.60	371.50	2.30

Table 5.31: Error when at least 50 words are needed

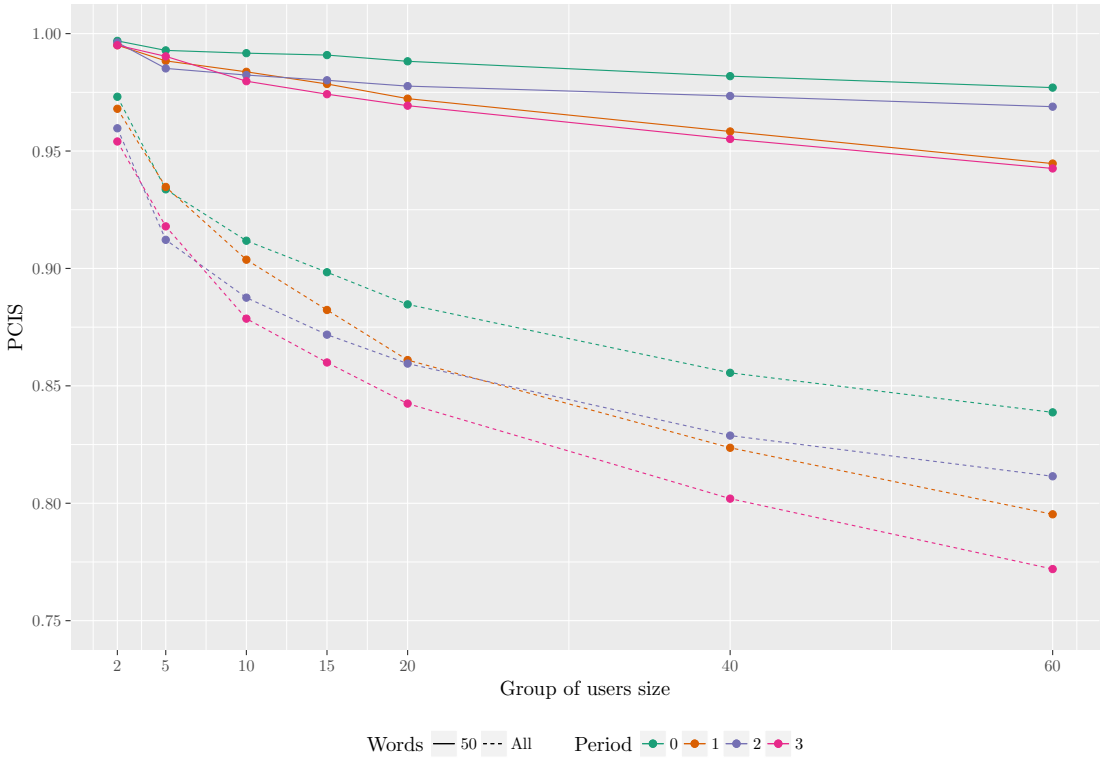

Figure 5.6: Group sizes results

These results are shown as evidence of the importance of having a high number of sessions to compare to the logical tree models. When the number of sessions is small, the margin of error increases, as expected. The most relevant results are obtained when the number of compared sessions is very large, as is the case of Period 0 and when large groups of users are used.

5.6.3 Test 5 summary

This test has evaluated what should be a good size in terms of the number of models a session should be compared to. As expected, the bigger the group against which a sample is tested the bigger the error. On the other hand, though, it has been observed that if the origin session is of a higher quality it maintains a much better chance of being correctly identified even if the size of the group increases up to 60 users. Also, the size of the dataset is determinant to obtain the best results. Periods 0 and 2, the largest, obtain always the best accuracy.

5.7 Test 6 – Authenticating users

This test evaluates the possibility of using the proposed methods, not only to identify users that have submitted a message to the Virtual Campus web application, but also

to authenticate them when accessing the web application. Of course, this same method could be used to authenticate users accessing any other protected resource as long as large enough sequences of words can be captured.

As with previous studies that have focused on authentication, this test tries to find a threshold value from which to grant or prohibit access to the application. The value of this threshold is determined trying a wide range of values from the obtained distances to determine at which point both the FRR and the FAR values are equal. This value is known as the EER and it has been used extensively in the past to determine the reliability of authentication schemes when using Keystroke Dynamics.

Once this value is determined it should be the choice of the administrator to set it above or below to increase security or, on the other hand, allow more users that are not correctly identified. As has been previously discussed in Section 2.3.3 something that annoys users is having to authenticate more than once because the underlying system has been unable to correctly identify them. On the other hand, when the threshold is too loose users can be falsely accepted, something that should be minimized, even if users are angry.

The threshold value depends greatly on the distance measurement used. For this test, the Chebyshev distance has been used. At the same time, to have a rather good value for each of the different periods, the whole set of 60 users has been used. From each of these periods all testing sessions have been evaluated against all the trained models from all users in the period, always following the MCCV technique. When using this particular distance measurement, the range of values went from ~ 4 to values well beyond 50. These values do not really represent a unit of milliseconds because weighting techniques may have been applied to modify such values.

Once all distance measurements from the training sessions to the models are available, a starting threshold value is set and increased progressively. When the threshold is very low only those users with very short distances to the model are given access. With a value this low, it is unusual seeing that any false users are given access. The problem is that many valid users are denied access. On the other hand, as soon as the threshold value increases the number of false user granted access also increases, even if almost all valid users are granted access correctly.

5.7.1 Number of independent tests performed

As with the case of the test regarding group sizes the parameters that have given the best results have been used for this test. At the same time, both tests using all words and only those sessions that have at least 50 words have been independently tested (W). The parameters for this test are, of course, the threshold that ranges from 10 to 25 in incremental values of 0.2 (75 different threshold values) (T), the 4 periods (P)

that have been used in all tests using groups of 60 users. All in all, the grand total of tests is: $75T \cdot 2W \cdot 4P \cdot 10 = 6,000$ tests.

5.7.2 Results for the authentication tests

The results for this test are presented using different measures than in previous tests. If for all previous tests the accuracy has been reported, in this case the EER value is used. Table 5.32 shows the FAR, FRR and EER values for each period considered when all words in all sessions have been and when only those sessions with at least 50 words have been used.

Period	Words	Threshold	FRR	FAR	EER
$P0$	All	17.60	9.59	9.54	9.57
	50	16.40	4.71	4.62	**4.67**
$P1$	All	18.20	9.81	9.78	9.80
	50	17.60	6.79	6.82	6.81
$P2$	All	18.20	11.46	11.51	11.49
	50	16.80	5.32	5.32	5.32
$P3$	All	18.60	12.74	12.81	12.77
	50	17.60	7.29	7.48	7.38

Table 5.32: EER when authenticating users

The values shown in Table 5.32 are depicted in Figure 5.7, for the test in which all words are used, and in Figure 5.8 for the test in which only sessions with at least 50 words found in the model are used.

As in previous tests, it can be observed that having a larger number of words is highly determinant when it comes to obtaining the best results. In all cases, the EER values are close to a 50% smaller when authenticating users, with a peak value of 4.67% when using a large dataset ($P0$) and at least 50 words in the sessions.

Figures 5.9, and 5.10 show the ROC curves for both cases, showing, once again, that the best value is achieved with large datasets and a minimum number of words found per session.

5.7.3 Test 6 summary

Authentication can be performed with the proposed models and methods. The results obtained show good EER values when compared to those obtained by previous research.

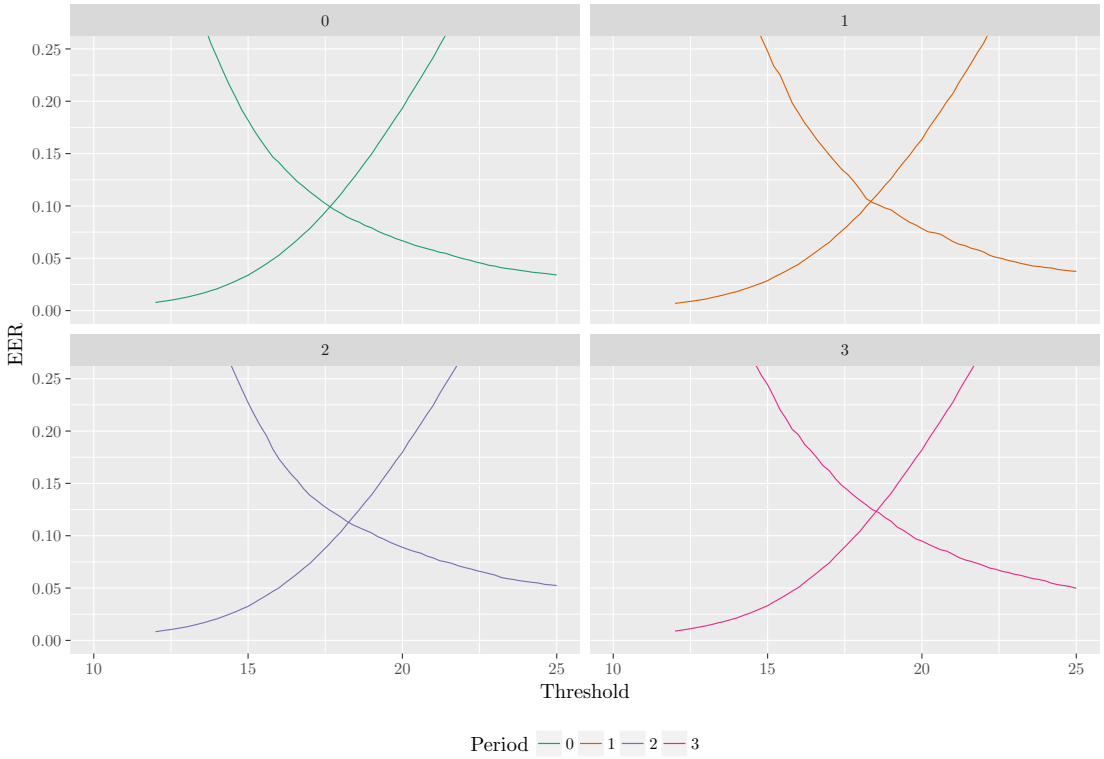

Figure 5.7: EER when authenticating users using all words

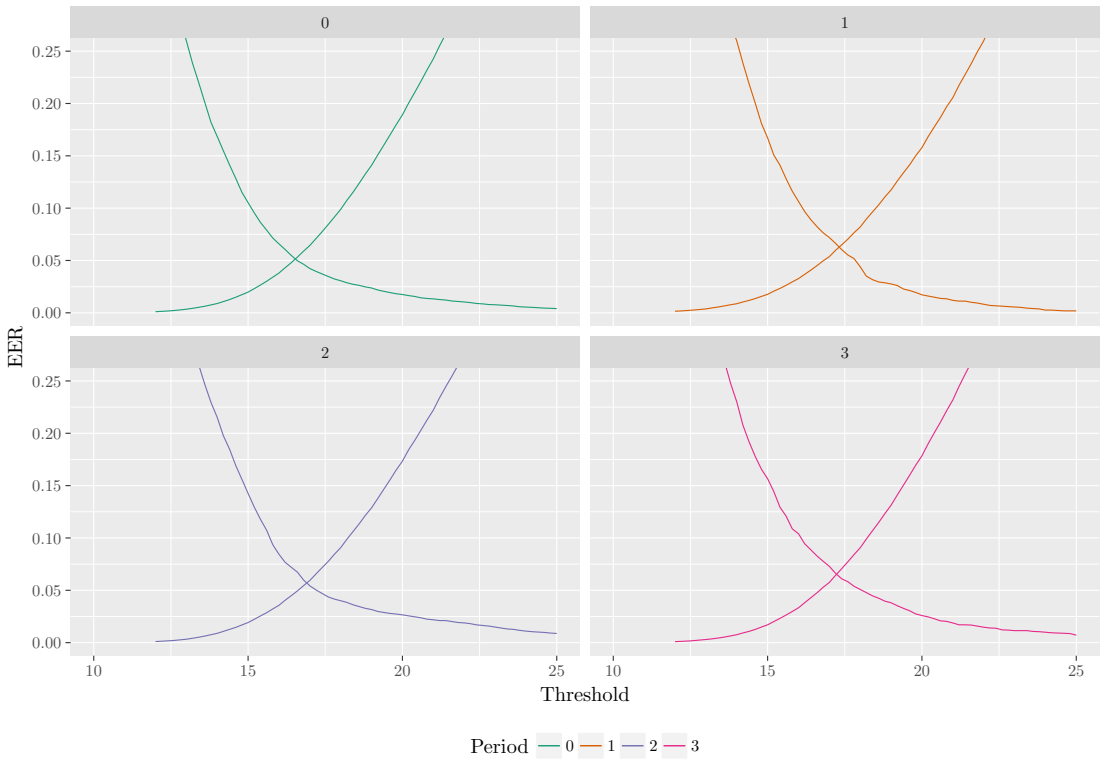

Figure 5.8: EER when authenticating users using sessions with at least 50 words

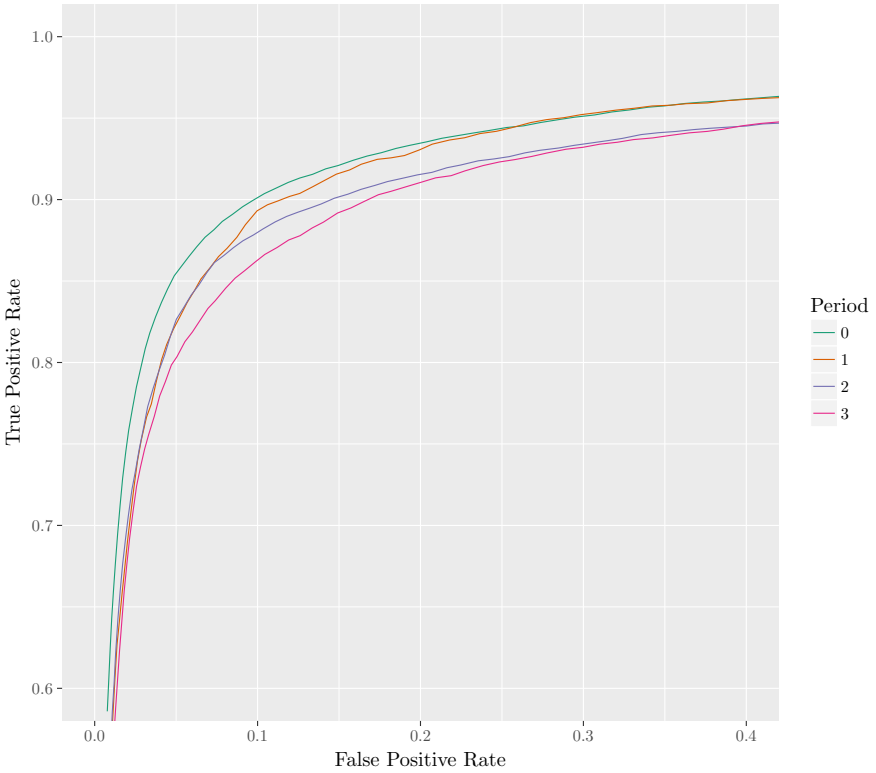

Figure 5.9: ROC when authenticating users using all words

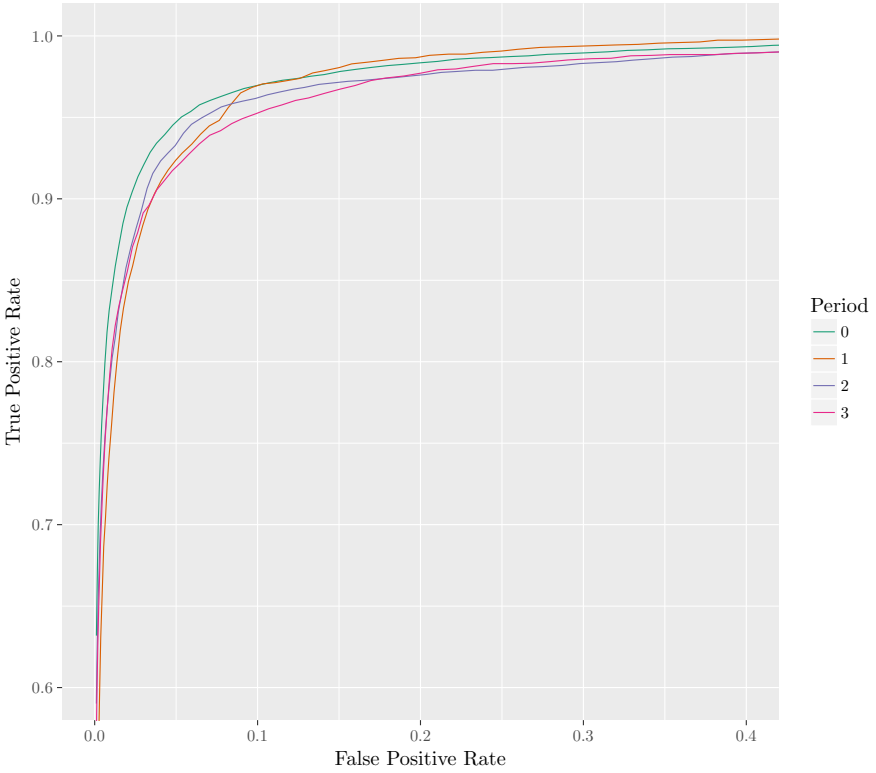

Figure 5.10: ROC when authenticating users using sessions with at least 50 words

To perform authentication, a threshold value from which to accept users into the system has been established. The current test has tried to find the value for the threshold were both the FRR and FAR values were equal. The best EER value has been that of 4.67% when using a large dataset and a minimum number of words found in the model. The threshold for this particular combination of parameters has been 16.40. It can be seen that this threshold value is not the best in all cases, though. As soon as these values vary the EER value also increases, rendering the system less accurate. In the worst case scenario, an EER of 12.77% has been found when using a small dataset using all available sessions.

It should be noted that, for this test, only the best 60 users have been used. If all 471 users available in the dataset had been used, the results would have been much worse. This has to be kept in mind all the time: with good information Keystroke Dynamics is a valid biometric technique, otherwise, the possibility of using it in multimodal schemes should be considered.

5.8 Test 7 – Dealing with age group and gender

This last experiment focuses on determining if the age group and the gender a user belongs to can help identify or authenticate them when using the proposed methods. In this sense, the question whether the age and gender of a user could affect the accuracy of the identification or authenticating system has risen many times. If relevant differences are found when a separation is performed or not, these could be useful in terms of model building and optimization, as well as for applications in gender recognition or age group classification.

The experiments in this section have been suggested after noticing that the available keystroke dataset has enough users from different age groups. At the same time, data from both genders is available, even if much more information comes from the female genre. Using this information, it is possible to build and test models separated by age group and gender.

The results obtained from these tests come from comparing testing sessions to groups of sizes 10, 15, and 20 users. The need to have such sizes comes from the fact that if larger groups are used, these become unbalanced. The idea has been to have groups as homogeneous as possible to obtain meaningful and comparable results. These different group sizes have already been evaluated in Test 5. In that case, though, no separation of any kind has been performed. Table 5.33 shows a summary of the previously obtained results. These can be used as reference to compare the results obtained in the current experiment.

When dealing with age group and gender, different approaches have been attempted:

Period	Words	Group size (PCIS)		
		10	15	20
$P0$	All	91.18	89.84	88.47
	50	99.17	99.09	98.82
$P1$	All	90.37	88.23	86.10
	50	98.37	97.86	97.23
$P2$	All	88.76	87.18	85.95
	50	98.23	98.01	97.76
$P3$	All	87.86	86.00	84.24
	50	97.97	97.42	96.93

Table 5.33: Results for group sizes 10, 15, and 20

- Determine if gender is determinant when identifying or authenticating users. This test tries to determine if the processes attempted vary when sessions are compared against models of the same gender as the owner of a session, or if there is no remarkable difference when genders are mixed. If a relevant difference was found, it could be useful in terms of gender recognition applications, for instance.

- The same is attempted categorizing users by age group. In this case, in order to have age groups of similar size and quality, three different groups have been formed: $(18, 34]$, $(34, 45]$, and $(45, 69]$. These three age groups have been labeled as: Young, Middle age, and Senior respectively. The objective of this test is to see whether by using separated groups for each age group accuracy improves, if EER when authenticating decreases, or if, on the other hand, mixed age groups perform better.

- Finally, the last test uses separated groups by age group and by gender to determine if mistakes users make help identify them. The methodology in this experiment is different than in the previous two, and no exhaustive comparison of models is carried out. Only the evaluation of the effect of mistakes incorporated in the models is performed taking age and gender into account. It should be noted that this test was originally performed before developing the Weighted mean of distances method, revised, and when results were still far from optimal. It was thought that dealing with such age and gender parameters the accuracy of the system would improve significantly.

For the following tests, in order to have a frame of reference with previous experiments, more specifically with results obtained in Test 5, the number of mean Sessions

across all 10 different MCCV runs, mean Incorrect and mean Correct identification values, as well as the ME with a 95% of confidence are also shown. These values are shown to evidence that once groups of users are small, the number of compared sessions also decreases, increasing the ME.

5.8.1 Number of independent tests performed

Like in previous experiments, the Chebyshev distance measurement, the Weighted mean of distances, revised method to identify the owner of a session, and the model building parameters that have given the best results have been chosen for this test. The particular parameters for these tests are the groups and their combinations (C). The 4 different periods (P), and again, an MCCV methodology with 10 repetitions, has also been used. For the cross-validation technique 70% of the sessions have been used for training and the remaining 30% have been used for testing.

For the two first experiments in this section, both options, using all available sessions, and only those with at least 50 words, are used (W). For the gender separation experiment, there are 2 different groups, Male and Female, and 4 possible combinations. The number of tests when performing identification has been: $4C \cdot 2W \cdot 4P \cdot 10 = 320$ tests. For the age group separation experiment, also when identifying, 3 different groups are considered, Young, Middle age, and Senior, with 9 combinations. The number of tests has been: $9C \cdot 2W \cdot 4P \cdot 10 = 720$ tests. For these two experiments, where authentication has also been evaluated the number of tests performed has to be doubled. The grand total is $2,080$ tests.

For the last experiment, the methodology has been different because not all combinations have been attempted. Only identification has been evaluated for each group with and without mistakes incorporated in the models. This means that there are $4W$ different possibilities: all words with and without mistakes in the models, and sessions with at least 50 words, again, with and without mistakes in the models. The number of groups has been $6G$. The total number of tests has been: $6G \cdot 4W \cdot 4P \cdot 10 = 960$ tests.

5.8.2 Gender separation

This first test has been performed using the best 60 users from each of the 4 considered periods. Users have been grouped by gender. From this point, all testing sessions from any user have been tested against models containing only male users and models containing only female individuals. Of course, the model of the real user being tested is always used otherwise identification would not be possible.

To ensure that results are comparable, and having observed that there are much more women than men in the keystroke dataset, a subset of data has been used. For

each period, a random set of 20 models, one for male users and one for female users, has been selected. This group size has already been evaluated in Section 5.6, with rather good results, but in that case no separation of gender has been carried out.

5.8.3 Results for the gender separation test

Tables 5.34, and 5.35 show the results of this first experiment when identifying and when authenticating users respectively. In bold, in both tables, the best results have been highlighted comparing the gender of the origin testing sessions.

Words	Gender		Period (PCIS)			
	Origin	Model	$P0$	$P1$	$P2$	$P3$
All	Women	Women	**88.19**	84.95	85.23	82.70
	Women	Men	88.17	**86.14**	**86.65**	**83.27**
	Men	Women	**89.03**	**85.95**	**85.59**	**86.03**
	Men	Men	88.31	84.88	85.35	85.03
50	Women	Women	**98.95**	96.95	97.51	**97.13**
	Women	Men	**98.95**	**97.74**	**98.61**	96.39
	Men	Women	**98.45**	96.83	**96.51**	97.37
	Men	Men	98.17	**97.35**	95.89	**97.93**

Table 5.34: Identifying users using gender separated models

Words	Gender		Period (EER)			
	Origin	Model	$P0$	$P1$	$P2$	$P3$
All	Women	Women	**9.20**	**10.42**	14.63	18.38
	Women	Men	10.28	11.93	**11.60**	**13.25**
	Men	Women	**9.00**	**11.57**	**11.05**	**9.42**
	Men	Men	11.05	13.45	13.62	11.55
50	Women	Women	5.20	**5.38**	**2.08**	**5.39**
	Women	Men	**5.04**	6.86	6.49	7.93
	Men	Women	**4.20**	**5.43**	**5.58**	5.49
	Men	Men	4.79	6.28	6.51	**5.15**

Table 5.35: Authenticating users using gender separated models

From these results, it can be observed that there is no straight recognizable pattern throughout all the values. Only when using all words and identifying users it can be though that comparing samples from an origin gender to a different model gender could

render slight better results. In all other cases this behavior is not observed so clearly, though. With these numbers in hand, no real conclusion can be stated about the effect of gender in identification and authentication schemes.

At the same time, the fact of using gender separated models, when compared to models that do no present this separation, does not always improve the results. When compared to results showed in Table 5.33 (using a group size of 20 users), some values are above the previous results and some are below, if only by a small difference. No straight pattern is identified for all periods, either.

Tables 5.36 and 5.37 show the mean number of Sessions, Incorrect and Correct values, as well as the ME in percentage. Since the group of women is the one with most information it is also normal to see that the error is smaller.

Period	Gender		Sessions	Incorrect	Correct	ME (%)
	Origin	Model				
$P0$	Women	Women	1267.50	68.40	1199.10	1.24
	Women	Men	1267.50	145.10	1122.40	1.75
	Men	Women	474.60	50.80	423.80	2.78
	Men	Men	474.40	54.10	420.30	2.86
$P1$	Women	Women	294.40	25.10	269.30	3.19
	Women	Men	294.40	39.50	254.90	3.89
	Men	Women	150.80	20.50	130.30	**5.47**
	Men	Men	150.80	17.10	133.70	**5.06**
$P2$	Women	Women	628.60	41.60	587.00	1.94
	Women	Men	628.60	80.80	547.80	2.62
	Men	Women	179.10	24.90	154.20	**5.07**
	Men	Men	179.10	25.30	153.80	**5.10**
$P3$	Women	Women	448.60	45.30	403.30	2.79
	Women	Men	448.60	72.20	376.40	3.40
	Men	Women	254.60	34.30	220.30	4.19
	Men	Men	254.60	30.20	224.40	3.97

Table 5.36: Error when all words are used

5.8.4 Age group separation

Like in the previous test, this one has also been performed using the best 60 users from each of the 4 considered periods. Users have been separated only by age group, though. All testing sessions from any user have been tested against models containing only models from the same age group and against models with users from the two other age

Period	Gender		Sessions	Incorrect	Correct	ME (%)
	Origin	Model				
$P0$	Women	Women	788.40	3.50	784.90	0.46
	Women	Men	787.40	8.30	779.10	0.71
	Men	Women	299.80	4.20	295.60	1.33
	Men	Men	301.20	5.50	295.70	1.51
$P1$	Women	Women	172.70	3.10	169.60	1.98
	Women	Men	172.50	3.90	168.60	2.22
	Men	Women	97.00	3.10	93.90	**3.50**
	Men	Men	96.60	1.90	94.70	2.77
$P2$	Women	Women	411.30	3.80	407.50	0.92
	Women	Men	429.20	6.00	423.20	1.11
	Men	Women	98.40	3.30	95.10	**3.56**
	Men	Men	104.10	4.30	99.80	**3.82**
$P3$	Women	Women	238.70	4.10	234.60	1.65
	Women	Men	239.70	8.60	231.10	2.35
	Men	Women	156.20	4.10	152.10	2.51
	Men	Men	156.70	2.60	154.10	2.00

Table 5.37: Error when at least 50 words are needed

groups. All possible combinations have been tested. For example, a session belonging to a user of the Young age group has been tested against the models of Young, Middle age, and Senior age groups separately.

To ensure that results are comparable and statistically relevant with the rather scarce information available, and having observed that there are much more users belonging to the Young age range group, a subset of the dataset has been used. For each considered period a random set of 15 models for each of the age groups has been used. This ensures that all age groups have a similar number of users.

5.8.5 Results for the age group separation test

Tables 5.38, and 5.39 show the results of this experiment when identifying and when authenticating users respectively. In bold, in both tables, the best results have been highlighted comparing the age group of the origin sessions.

Words	Age group		Period (PCIS)			
	Origin	Model	$P0$	$P1$	$P2$	$P3$
All	(18, 34]	(18, 34]	84.98	87.31	75.38	76.57
		(34, 45]	86.31	86.20	73.47	80.18
		(45, 69]	**87.66**	**88.62**	**77.08**	**81.15**
	(34, 45]	(18, 34]	**94.33**	91.03	93.29	87.93
		(34, 45]	94.12	91.86	93.44	89.54
		(45, 69]	93.77	**93.51**	**93.96**	**89.67**
	(45, 69]	(18, 34]	**89.09**	**87.80**	**86.65**	**88.04**
		(34, 45]	87.55	84.90	84.71	87.48
		(45, 69]	86.97	84.73	82.68	86.56
50	(18, 34]	(18, 34]	97.78	96.18	93.80	93.79
		(34, 45]	98.03	97.11	94.50	95.82
		(45, 69]	**98.43**	**98.38**	**95.31**	**97.03**
	(34, 45]	(18, 34]	99.45	97.97	98.57	96.01
		(34, 45]	99.23	97.58	98.53	96.78
		(45, 69]	**99.52**	**98.21**	**98.99**	**96.84**
	(45, 69]	(18, 34]	**99.13**	**98.10**	**98.40**	98.70
		(34, 45]	98.65	96.52	97.46	**99.08**
		(45, 69]	98.42	96.82	96.74	98.22

Table 5.38: Identifying users using age group separated models

These results show an interesting fact. In general, except for a couple of cases and even these are marginal, the best results are obtained when there is a large gap between age groups. These can be especially observed when comparing the Young age group against the Senior age group. In both cases, when either of them is the origin testing group, the results are always better than when compared to the same age groups or

Words	Age group		Period (EER)			
	Origin	Model	$P0$	$P1$	$P2$	$P3$
		(18, 34]	12.81	18.11	19.51	14.00
	(18, 34]	(34, 45]	11.82	11.23	**18.96**	16.92
		(45, 69]	**8.51**	**7.13**	20.35	**12.88**
All		(18, 34]	**6.33**	9.92	**7.39**	13.78
	(34, 45]	(34, 45]	9.62	9.10	8.40	**13.39**
		(45, 69]	8.81	**8.11**	8.56	13.66
		(18, 34]	**8.25**	**11.25**	12.62	12.66
	(45, 69]	(34, 45]	12.00	13.08	**11.82**	**10.59**
		(45, 69]	12.77	11.85	14.00	11.59
		(18, 34]	5.07	10.26	7.77	**5.84**
	(18, 34]	(34, 45]	**3.45**	6.45	7.08	10.42
		(45, 69]	4.51	**3.23**	**7.04**	6.55
50		(18, 34]	4.46	7.69	**4.37**	**7.66**
	(34, 45]	(34, 45]	5.07	4.69	6.46	7.77
		(45, 69]	**4.27**	**3.45**	5.47	7.69
		(18, 34]	**3.07**	**4.07**	**3.03**	4.62
	(45, 69]	(34, 45]	4.84	9.31	4.20	**3.81**
		(45, 69]	6.65	5.13	8.94	5.91

Table 5.39: Authenticating users using age group separated models

against the Middle age group. This suggests that, with age, the Keystroke Dynamics template of users may tend to adapt. Also, generally, any age group will obtain better results when not compared to the same age group. This is highly relevant and suggests that users in the same age group tend to type more alike.

The question that cannot be answered at this moment is if users that are currently in the Young age group will adapt their Keystroke Dynamics pattern over the years to a rhythm comparable to today's Senior age group. At the same time, and in direct relation to this, the question of how current users *learned* to type on computers and if this process affected their natural rhythm, could be analyzed.

Using age group separated models, when compared to models that do not have this separation, presents interesting differences depending on the age group being evaluated. When compared to results showed in Table 5.33 (using a group size of 15 users), the $(34, 45]$ age group, the Middle age group, is the one that shows the most interesting improvement in comparison to other age groups, especially for rich Periods like 0 and 2. Even if the improvement, at this stage, can be considered only marginal, the fact that other age groups present worse results is thought to be relevant. It could be argued that, from all three, this group has the most defining models. The one that suffers most is the Younger group with worse results than when no age group separation is

carried out. This is considered relevant and further studies of the particularities of the typing rhythms in each age group could be proposed.

Tables 5.40 and 5.41 show the mean number of Sessions, Incorrect and Correct values, as well as the ME in percentage.

5.8.6 Age group and gender separation analyzing mistakes

The last test combines both the age and gender groups and analyzes the effect of using the mistakes users make in the logical tree models. The goal of this test is to see, not only if the different groups, be it in gender or in age, can benefit from having the errors into the model, but also whether these groups present differences in terms of accuracy when errors are incorporated into the model.

It may be of interest to see, for example, whether younger users benefit from having errors in the models as opposed to senior users. If this was true, the model building phase could be adapted to take this fact into account, and build better models, adapted to age group and gender.

To ensure that results are comparable, again, a subset of data has been used. Unfortunately, as soon as both gender and age group are used in the same experiment, it is found that the number of users per group decreases alarmingly. For each considered period a randomly chosen set of 10 models has been used. This ensures that all age groups and genders have a similar and comparable number of users.

Only identification has been attempted for the current experiment. The results should give an idea of what the tendency is when age and gender are separated, and it should be perfectly possible to use the proposed methodology also when authenticating users.

This experiment is included only for completeness, showing a test that was performed, chronologically, before some of the session identification methods tested in Test 3 were developed. The initial idea was to find out if this particular behavioral feature (mistakes users make) affected users differently when age and gender was considered.

5.8.7 Results when separating by age group and gender

Table 5.42 shows the global PCIS results after separating users using age group and gender. At the same time, the possibility of using the mistakes as part of the models is also evaluated.

In Table 5.42, the best results when using all sessions or when using sessions with more than 50 words found in the models have been highlighted in bold. The values in the table show somewhat discouraging results, but this, on the other hand, is coherent

Period	Age group		Sessions	Incorrect	Correct	ME (%)
	Origin	Model				
		(18,34]	229.90	22.20	207.70	3.82
	(18,34]	(34,45]	229.70	30.30	199.40	4.38
		(45,69]	229.70	27.30	202.40	4.19
		(18,34]	633.30	34.90	598.40	1.78
$P0$	(34,45]	(34,45]	633.30	25.00	608.30	1.52
		(45,69]	633.30	38.40	594.90	1.86
		(18,34]	878.90	92.80	786.10	2.03
	(45,69]	(34,45]	878.90	105.90	773.00	2.15
		(45,69]	878.90	93.00	785.90	2.03
		(18,34]	98.60	7.60	91.00	5.26
	(18,34]	(34,45]	98.60	12.90	85.70	**6.66**
		(45,69]	98.60	10.60	88.00	**6.11**
		(18,34]	123.50	10.90	112.60	5.00
$P1$	(34,45]	(34,45]	123.50	9.90	113.60	4.79
		(45,69]	123.50	7.90	115.60	4.32
		(18,34]	223.10	26.30	196.80	4.23
	(45,69]	(34,45]	223.10	32.60	190.50	4.64
		(45,69]	223.10	23.00	200.10	3.99
		(18,34]	128.50	19.80	108.70	6.24
	(18,34]	(34,45]	128.50	31.50	97.00	**7.44**
		(45,69]	128.50	27.20	101.30	**7.06**
		(18,34]	385.60	25.20	360.40	2.47
$P2$	(34,45]	(34,45]	385.60	15.30	370.30	1.95
		(45,69]	385.60	22.70	362.90	2.35
		(18,34]	293.60	37.70	255.90	3.83
	(45,69]	(34,45]	293.60	43.20	250.40	4.05
		(45,69]	293.60	48.90	244.70	4.26
		(18,34]	166.70	21.30	145.40	5.07
	(18,34]	(34,45]	166.70	31.70	135.00	5.96
		(45,69]	166.70	30.10	136.60	5.84
		(18,34]	147.20	17.20	130.00	5.19
$P3$	(34,45]	(34,45]	147.20	13.80	133.40	4.71
		(45,69]	147.20	14.70	132.50	4.84
		(18,34]	389.30	44.80	344.50	3.17
	(45,69]	(34,45]	389.30	46.90	342.40	3.23
		(45,69]	389.30	44.80	344.50	3.17

Table 5.40: Error when all words are used

Period	Age group		Sessions	Incorrect	Correct	ME (%)
	Origin	Model				
		(18,34]	146.30	2.00	144.30	1.88
	(18,34]	(34,45]	147.20	2.90	144.30	2.25
		(45,69]	147.50	2.10	145.40	1.91
		(18,34]	459.50	2.30	457.20	0.65
$P0$	(34,45]	(34,45]	460.30	2.20	458.10	0.63
		(45,69]	460.60	2.20	458.40	0.63
		(18,34]	482.40	4.20	478.20	0.83
	(45,69]	(34,45]	484.50	6.60	477.90	1.03
		(45,69]	484.90	6.50	478.40	1.02
		(18,34]	61.90	1.20	60.70	3.43
	(18,34]	(34,45]	61.70	1.60	60.10	3.97
		(45,69]	61.30	0.40	60.90	2.02
		(18,34]	86.30	1.40	84.90	2.67
$P1$	(34,45]	(34,45]	86.20	1.90	84.30	3.10
		(45,69]	86.50	1.10	85.40	2.36
		(18,34]	122.10	2.10	120.00	2.31
	(45,69]	(34,45]	122.10	4.30	117.80	3.27
		(45,69]	121.70	2.10	119.60	2.31
		(18,34]	61.50	1.90	59.60	4.32
	(18,34]	(34,45]	65.40	3.60	61.80	**5.53**
		(45,69]	66.00	2.80	63.20	**4.86**
		(18,34]	290.60	3.50	287.10	1.25
$P2$	(34,45]	(34,45]	305.90	2.50	303.40	1.01
		(45,69]	307.10	3.10	304.00	1.12
		(18,34]	155.70	2.40	153.30	1.94
	(45,69]	(34,45]	165.60	4.20	161.40	2.39
		(45,69]	168.00	5.40	162.60	2.67
		(18,34]	89.60	3.10	86.50	3.78
	(18,34]	(34,45]	90.40	3.80	86.60	4.14
		(45,69]	90.50	2.70	87.80	3.51
		(18,34]	98.30	3.90	94.40	3.86
$P3$	(34,45]	(34,45]	99.00	2.40	96.60	3.03
		(45,69]	98.90	3.10	95.80	3.43
		(18,34]	207.20	2.40	204.80	1.46
	(45,69]	(34,45]	208.10	1.90	206.20	1.29
		(45,69]	208.70	3.00	205.70	1.61

Table 5.41: Error when at least 50 words are needed

Period	Gender	Age group	Words			
			All	All w. errors	50	50 w. errors
$P0$	Women	(18, 34]	**83.29**	82.24	**97.64**	97.50
		(34, 45]	**93.99**	92.85	**99.53**	99.50
		(45, 69]	**87.04**	86.38	98.96	**99.11**
	Men	(18, 34]	**92.19**	91.92	99.08	**99.38**
		(34, 45]	95.30	**95.50**	99.35	**99.42**
		(45, 69]	**90.12**	90.03	**98.47**	98.31
$P1$	Women	(18, 34]	**82.01**	79.51	95.61	**96.78**
		(34, 45]	**92.93**	90.50	**98.51**	98.33
		(45, 69]	**87.94**	86.97	**98.12**	97.06
	Men	(18, 34]	**89.58**	86.61	**98.55**	97.00
		(34, 45]	**97.14**	94.29	**100.00**	96.90
		(45, 69]	**86.10**	84.66	**97.45**	95.52
$P2$	Women	(18, 34]	**70.43**	65.79	**92.93**	85.47
		(34, 45]	**94.05**	93.07	98.72	**98.74**
		(45, 69]	**86.61**	84.42	**99.46**	99.36
	Men	(18, 34]	**89.59**	89.18	**99.58**	99.10
		(34, 45]	**92.12**	91.82	**98.09**	97.44
		(45, 69]	87.31	**87.83**	94.25	**95.04**
$P3$	Women	(18, 34]	**72.17**	70.83	**92.48**	92.15
		(34, 45]	**93.56**	91.80	**99.05**	97.76
		(45, 69]	**91.56**	90.81	**99.16**	98.80
	Men	(18, 34]	**87.55**	87.13	96.84	**97.34**
		(34, 45]	95.80	**95.82**	**99.41**	98.32
		(45, 69]	87.44	**87.48**	**99.00**	98.81

Table 5.42: Identifying users using age and gender separated models

with the results that have been obtained in Test 4 regarding behavioral features.

Beginning with Period 0, no real improvement is reflected in any case. Women almost never benefit from having errors into the model, while men do on some cases, especially on the Young and Middle age groups, but more research should be carried out to find out if this tendency is general because the differences are very small and the number of users in these groups has been also very small.

Period 2, also a good one in terms of number of events, also shows little improvement when errors are considered, and when compared to the values from Period 0, there is no consistency in the results.

Periods 1 and 3, being the small datasets show that this is even worse. In almost no case, having the errors incorporated in the model provides a benefit for any group, be it age group or gender, and when they do the difference is only marginal.

Figures 5.11, 5.12, 5.13, and 5.14 show bar graphs for the values shown in Table 5.42.

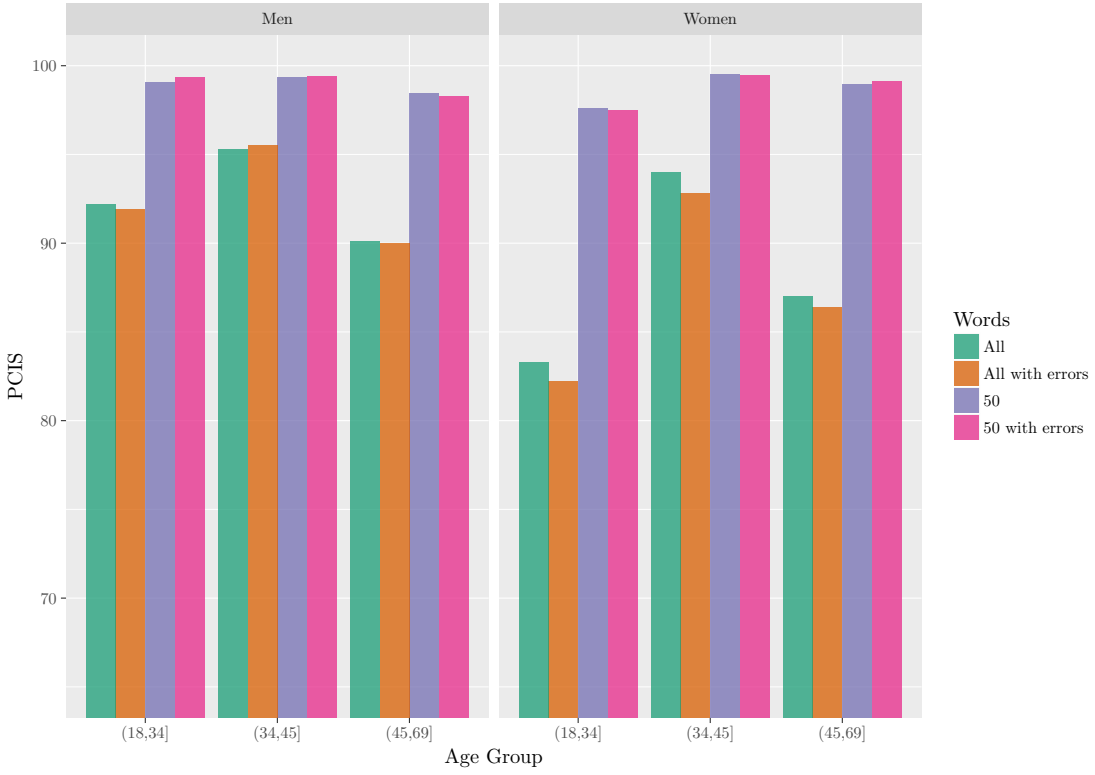

Figure 5.11: Period 0 ($P0$) – Identifying users using age and gender separated models

Even if the possibility of using non-standard word constructions in the models has been discouraged after seeing these results, something that catches the eye in Figures 5.13, and 5.14 is the difference between the results obtained in Younger women when using all sessions and when only using those with more than 50 words. This is, by far, the group that yields the worst results in identification in all cases, even also after discarding sessions of bad quality. Having much more samples of this age group and gender could help understand if this behavior is, in fact, normal.

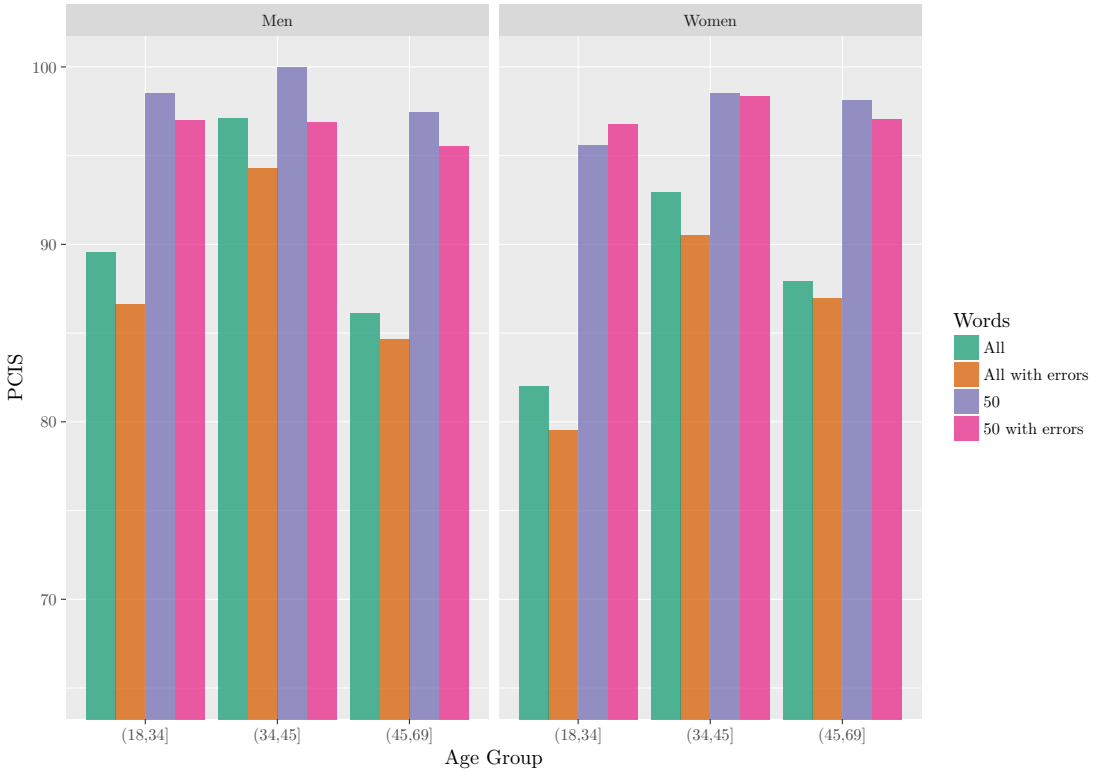

Figure 5.12: Period 1 ($P1$) – Identifying users using age and gender separated models

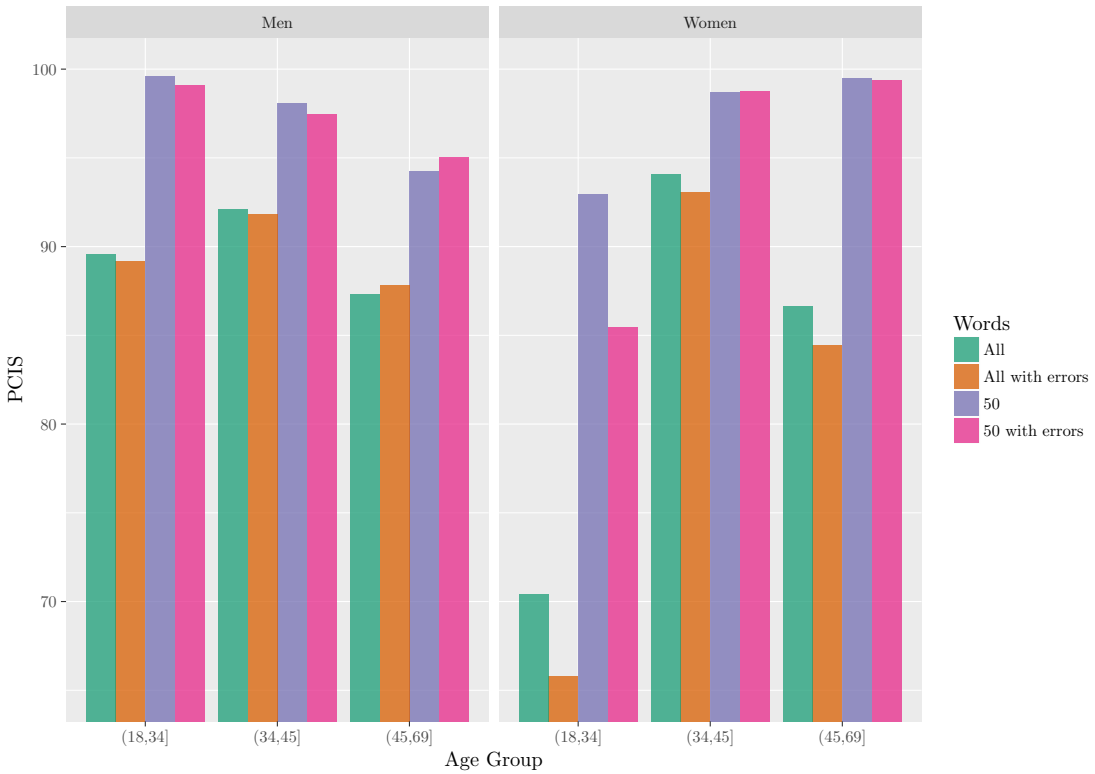

Figure 5.13: Period 2 ($P2$) – Identifying users using age and gender separated models

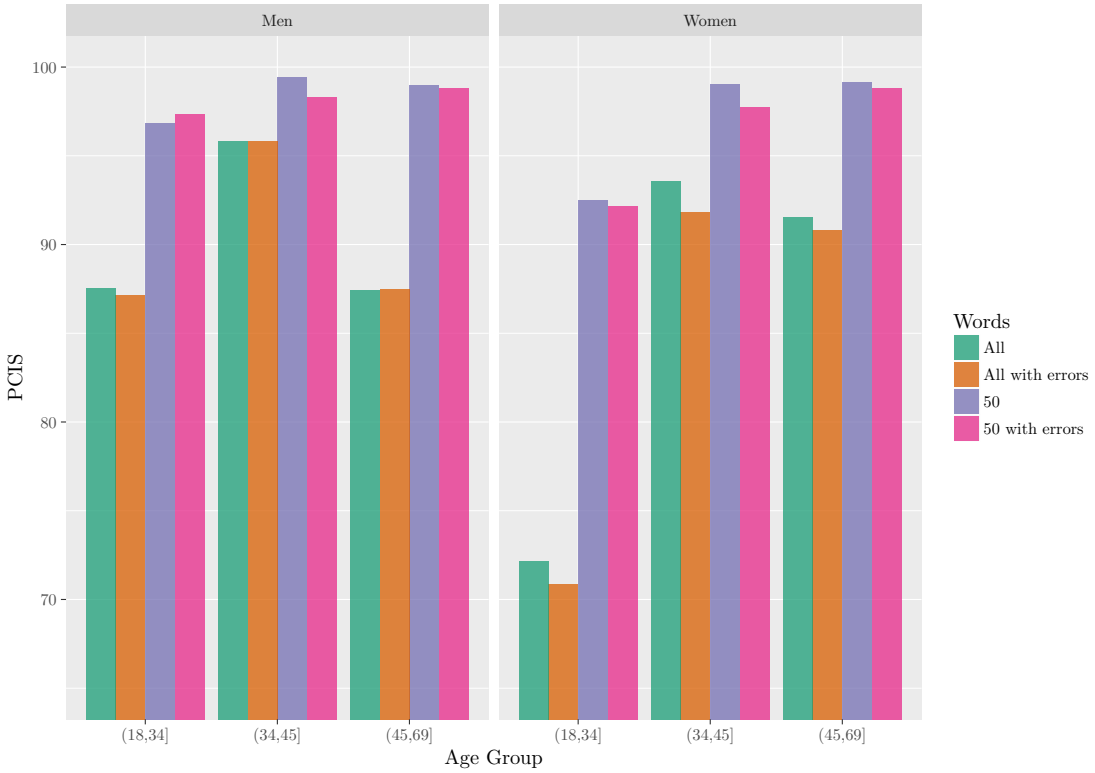

Figure 5.14: Period 3 ($P3$) – Identifying users using age and gender separated models

Also, after having observed that in the age group separation test, the Younger group had given the worst results, it could be argued that this was because of the influence of younger women in respect to other groups. It would be very interesting to go deep in this issue and find out why younger women present such irregular models, and if this behavior will be maintained throughout their lives or, on the other hand, their natural rhythm settles with age.

From the obtained results, a new methodology could be proposed to determine the minimum number of words necessary to be found in a tree model to consider the session valid. So far, in the experiments carried out in this research, a value of 50 words has been considered most favorable to have good levels of accuracy while not discarding too many sessions. Higher values may have given better results, while lower values, like the case that is accepting any session, deem lower accuracy values. The results from the age group and gender separation test show that some user groups present a much better accuracy than others when no minimum number of words is established. When enough samples have been collected (in period 0, for example), the accuracy of male subjects in the $(34, 45]$ group is above 95%. Females in the same age group are also the ones with the best results. It could be proposed to lower the number of words necessary to consider a session valid if such sessions come from users that are known to have a good rate of identification. This, especially when authenticating users, could be a benefit for the user. In such cases, users would not need to type so many sentences

for the supervisors to have a good level of certainty of their identity.

As with previous tests in this experiment, Tables 5.43, Tables 5.44, Tables 5.45, and 5.46 show the mean number of Sessions, Incorrect and Correct values, as well as the ME in percentage for each of the different cases: with or without errors being taken into account, and when all words and when 50 words were needed.

Period	Gender	Age group	Sessions	Incorrect	Correct	ME (%)
$P0$	Women	(18, 34]	138.10	22.00	116.10	6.10
		(34, 45]	589.30	34.50	554.80	1.90
		(45, 69]	540.10	67.40	472.70	2.79
	Men	(18, 34]	91.60	7.00	84.60	5.44
		(34, 45]	44.00	2.00	42.00	6.15
		(45, 69]	338.80	32.70	306.10	3.14
$P1$	Women	(18, 34]	48.10	8.10	40.00	10.58
		(34, 45]	116.50	8.10	108.40	4.62
		(45, 69]	129.80	15.10	114.70	5.52
	Men	(18, 34]	50.50	5.00	45.50	8.24
		(34, 45]	7.00	0.20	6.80	12.34
		(45, 69]	93.30	12.60	80.70	6.94
$P2$	Women	(18, 34]	86.50	23.40	63.10	9.36
		(34, 45]	350.60	20.30	330.30	2.44
		(45, 69]	191.50	24.60	166.90	4.74
	Men	(18, 34]	42.00	4.20	37.80	9.07
		(34, 45]	35.00	2.70	32.30	8.84
		(45, 69]	102.10	12.50	89.60	6.36
$P3$	Women	(18, 34]	102.30	27.30	75.00	8.57
		(34, 45]	124.20	7.80	116.40	4.27
		(45, 69]	222.10	17.90	204.20	3.58
	Men	(18, 34]	64.40	7.70	56.70	7.92
		(34, 45]	23.00	0.90	22.10	7.92
		(45, 69]	167.20	20.40	146.80	4.96

Table 5.43: Error when using all available words without mistakes

5.8.8 Test 7 summary

Do age and gender affect the identification or authentication of users? By the results shown in this test it seems that the most significant values come from the age group a user belongs to. It has been observed that the best results are always obtained when sessions are evaluated against models where there is a significant age difference. Also, it has been found that the Middle age group presents the most consistent models when compared to other groups. On the other hand, gender has been discarded as a factor affecting identification or authentication.

As per the use of the mistakes users make, as it had already been observed in Test 4, there is no need to incorporate them into the models. In most cases, there does not seem to be a pattern by which it can be stated that a particular group benefits from having them considered.

Period	Gender	Age group	Sessions	Incorrect	Correct	ME (%)
$P0$	Women	(18, 34]	142.40	23.90	118.50	6.14
		(34, 45]	599.80	41.50	558.30	2.03
		(45, 69]	548.70	71.40	477.30	2.82
	Men	(18, 34]	92.30	7.30	85.00	5.51
		(34, 45]	44.00	1.90	42.10	6.01
		(45, 69]	339.20	33.00	306.20	3.15
$P1$	Women	(18, 34]	49.20	9.50	39.70	11.03
		(34, 45]	117.00	11.00	106.00	5.29
		(45, 69]	132.00	16.50	115.50	5.64
	Men	(18, 34]	51.00	6.50	44.50	9.15
		(34, 45]	7.00	0.40	6.60	17.20
		(45, 69]	93.80	13.90	79.90	7.19
$P2$	Women	(18, 34]	88.70	27.80	60.90	9.65
		(34, 45]	355.40	24.00	331.40	2.61
		(45, 69]	196.90	29.10	167.80	4.96
	Men	(18, 34]	46.20	4.40	41.80	8.46
		(34, 45]	35.00	2.80	32.20	8.99
		(45, 69]	102.10	12.00	90.10	6.25
$P3$	Women	(18, 34]	107.70	29.50	78.20	8.42
		(34, 45]	125.50	10.00	115.50	4.74
		(45, 69]	223.00	19.60	203.40	3.72
	Men	(18, 34]	64.70	8.10	56.60	8.06
		(34, 45]	23.00	0.90	22.10	7.92
		(45, 69]	167.50	20.40	147.10	4.95

Table 5.44: Error when using all available words with mistakes in the model

Period	Gender	Age group	Sessions	Incorrect	Correct	ME (%)
$P0$	Women	(18, 34]	84.00	2.00	82.00	3.26
		(34, 45]	429.40	2.00	427.40	0.64
		(45, 69]	278.50	2.90	275.60	1.19
	Men	(18, 34]	62.90	0.60	62.30	2.40
		(34, 45]	32.50	0.20	32.30	2.69
		(45, 69]	207.90	3.20	204.70	1.67
$P1$	Women	(18, 34]	29.80	1.30	28.50	7.33
		(34, 45]	79.80	1.20	78.60	2.67
		(45, 69]	64.40	1.20	63.20	3.30
	Men	(18, 34]	33.00	0.50	32.50	4.17
		(34, 45]	6.40	0.00	6.40	–[1]
		(45, 69]	57.60	1.50	56.10	4.11
$P2$	Women	(18, 34]	40.60	2.70	37.90	7.66
		(34, 45]	280.20	3.60	276.60	1.32
		(45, 69]	109.60	0.60	109.00	1.38
	Men	(18, 34]	22.50	0.10	22.40	2.75
		(34, 45]	26.40	0.50	25.90	5.20
		(45, 69]	59.30	3.40	55.90	5.92
$P3$	Women	(18, 34]	49.10	3.70	45.40	7.38
		(34, 45]	81.90	0.80	81.10	2.13
		(45, 69]	110.20	0.90	109.30	1.68
	Men	(18, 34]	41.10	1.30	39.80	5.35
		(34, 45]	17.40	0.10	17.30	3.55
		(45, 69]	99.60	1.00	98.60	1.96

[1] When $\hat{p} = 1$ the ME cannot be obtained.

Table 5.45: Error when using sessions with at least 50 words without mistakes

Period	Gender	Age group	Sessions	Incorrect	Correct	ME (%)
$P0$	Women	(18, 34]	87.10	2.20	84.90	3.30
		(34, 45]	438.10	2.20	435.90	0.66
		(45, 69]	280.20	2.50	277.70	1.10
	Men	(18, 34]	63.50	0.40	63.10	1.95
		(34, 45]	34.30	0.20	34.10	2.55
		(45, 69]	214.00	3.60	210.40	1.72
$P1$	Women	(18, 34]	30.90	1.00	29.90	6.24
		(34, 45]	83.30	1.40	81.90	2.76
		(45, 69]	65.20	1.90	63.30	4.08
	Men	(18, 34]	33.50	1.00	32.50	5.76
		(34, 45]	6.40	0.20	6.20	13.48
		(45, 69]	58.10	2.70	55.40	5.41
$P2$	Women	(18, 34]	42.20	5.50	36.70	10.16
		(34, 45]	284.70	3.60	281.10	1.30
		(45, 69]	110.10	0.70	109.40	1.48
	Men	(18, 34]	24.20	0.20	24.00	3.61
		(34, 45]	26.90	0.70	26.20	6.02
		(45, 69]	61.30	3.00	58.30	5.40
$P3$	Women	(18, 34]	52.40	4.10	48.30	7.27
		(34, 45]	84.40	1.90	82.50	3.16
		(45, 69]	110.10	1.30	108.80	2.02
	Men	(18, 34]	41.30	1.10	40.20	4.91
		(34, 45]	17.40	0.30	17.10	6.12
		(45, 69]	101.10	1.20	99.90	2.11

Table 5.46: Error when using sessions with at least 50 words and mistakes in the model

As a side effect from the results obtained in this test, the fact that Younger women are the ones that are most incorrectly identified, should set the grounds for further testing regarding this particular group.

5.9 Summary

This chapter has detailed the experiments carried out and has presented the results of the proposed research. It has used an approach of increasing complexity, testing different parameters and keeping only the best for the following tests. The proposed methodology and models have been evaluated to determine the moment results were good enough to be used in production environments, comparing them against an *n-graph* frequency scheme that uses Absolute and Relative distances to determine the owner of the testing sessions.

Many features, both structural and behavioral have been evaluated. At the same time, different distance measurements and methods to determine the owner of a session have been analyzed. From these, the best results have been selected and a procedure to build and compare sessions against logical tree models has been established. A good ratio of computer resources – performance – accuracy has been obtained, improving over the base benchmark selected that used an *n-graph* methodology.

6 | Conclusions

The proposed research study focuses on the possibility of identifying or authenticating users using Keystroke Dynamics, contextual information, analyzing the largest letter sequence of a typed word against logical tree models, behavioral features, and features related to age group and gender. It proposes an alternative to the traditional *n-graph* frequency methodology. The samples submitted by the users have not been tailored in any way and the free text environment has been respected at all times. With the collected information during a period of three semesters, how words and timing intervals are related to context has been evaluated.

This chapter presents the conclusions that can be derived from the experiments that have been carried out and detailed in the previous chapter. The current chapter is organized taking into account the Objectives and Hypotheses defined in Chapter 3 and the different tests from Chapter 5. In Chapter 3, the following lines of research have been proposed:

1. Determine or establish how much information is needed to build a valid model.
2. Study which features better help identify users, either from model parameters or behavioral features.
3. Determine if the proposed methodology can also be used to authenticate users.
4. Determine if age group and gender are relevant and can be helpful to build better and optimized models.

The Objectives and Hypotheses associated to each of these lines of study have been evaluated in Chapter 5 by proposing a series of tests, each focused on a particular line of study and, at the same time, determined to prove a Hypothesis or achieve a goal determined by the Objectives. The performed tests have been the following:

- *n-graphs* test: This initial test has been performed to determine what is the best value that can be obtained with the available dataset when identifying users using a traditional *n-graph* methodology. The possibility of comparing the proposed methods to a method already proven valid by previous research has been deemed of utmost importance.

- Test 1 – Quality and size of the model: This test is related to the first line of study, and its goal is threesome: (1) determine whether by using the proposed model users can be identified; (2) find the parameters that better helped adjust the structure of the logical tree models; and, (3) prove that the size of the models is highly relevant.

- Test 2 – Parameters related to model searching: This test is related to the second line of study, and deals with the parameters used, when searching words in the logical tree model, that better help classify users. Its goal is to prove that different parameters have a different, and relevant, effect when identifying users.

- Test 3 – Methods to classify users: Again, this test is also related to the second line of research, and focused on obtaining comparable results to the *n-graph* methodology. By using different distance measurements and methods to identify users, the main goal is to find a procedure that yields good enough results to consider the proposed method valid and that its performance, at the same time, is better than the current alternative.

- Test 4 – Behavioral features: This test is related to the third line of study, and tries to determine if behavioral features, such as mistakes users make, word delimiters, and word and sentence frequencies, are relevant when trying to identify users. These features, even though not directly related to Keystroke Dynamics, can be used to weight distance measurements using a system of rewards and penalties.

- Test 5 – The effect of increasing the number of users a sample is compared to: This test is related, again, to the first line of research, and focuses on trying different values for the group size that the sessions will be compared to. This should help determine the accuracy of the system in different conditions.

- Test 6 – Authenticating users: This test is focused on achieving the proposed goal that stated that the proposed methods also allows for the users to be authenticated instead of just being identified.

- Test 7 – The effect of age group and gender: Finally, related to the last line of research, and focused on proving the effect of age and gender when identifying or authenticating users, this test separates users in different groups and studies the particularities of each to find patterns that can help build better models.

The main goal of the present chapter is to state the different conclusions that have been obtained from each of these experiments. The Objectives are evaluated first, and then, the discussion is centered on the different Hypotheses.

6.1 Conclusions on the proposed Objectives

This section focuses on the different proposed Objectives in Chapter 3. These objectives are evaluated individually in the following sections. Finally, the global Objective is also evaluated to determine how much of what was initially intended has been accomplished by this study.

6.1.1 On the validity of the model

The first objective was: *Determine if the proposed methodology of classifying samples based on contextual information is useful enough to identify users using a computer system.*

It is thought that this objective has been fully achieved. It has been proved by Tests 1, 2, 3, and 5, that using contextual information and a logical tree structure based on words and time intervals is a good enough model to identify users, always with a degree of error, using a computer system.

The most relevant results related to this objective are the following:

- The better the model is in terms of the number of events incorporated into the logical tree model, the better the results are. This suggests that the proposed methodology is highly dependent to the dataset size.
- No limits should be set to the number of words or to the maximum number of instances per word per user in the model. If there is an upper limit to these parameters, the research carried out has not yet been able to determine them. On the other hand, from a certain number of words it has been found that accuracy does not improve much. This should be taken into account to be able to build models limited in size that do not grow endlessly, consuming memory, storage and processing resources, in vain.
- The number of standard deviations to clean the model is crucial. In order to build robust models a value of at least 2 standard deviations is proposed. Higher values, though, do not improve the results in a relevant way.
- The best suited logical tree model is the Combined one. This result is of great interest because it proves the importance of the position of the letters in a word when using Keystroke Dynamics and contextual information, and that trying to have as much valuable information as possible is a must.
- The number of letters in a word are not equally interesting, relevant or optimal. This parameter is highly conditioned by the distance measurement and the

method used to identify users. It has been observed that some methods favor shorter words, while those methods that take advantage of the Depth feature from the logical tree model favor longer words.

- Results prove that higher hit rates (be it in number of words or graphs, as proved by the initial test), improves the results radically. In this research, a minimum value of 50 words has been proposed as most favorable to obtain good results and, at the same time, not discard too many sessions. Higher values can indeed improve the results but pose a dilemma on how many sessions should be discarded because of this parameter and how much users should write before being able to identify them reliably. Related to this conclusion, the age group and gender separation test has revealed that different groups present radical different accuracy values when sessions with any number of words are allowed to be tested. This could suggest the possibility of adapting this parameter depending on the age group and gender of the user.

- The best method to identify users has been the Weighted mean of distances, revised method. This method has been evaluated both using fusion schemes and using only mean and median values.

- The best distance measurement was the Chebyshev one most of the time, even if the Euclidean alternative did also provide good results.

- The number of models a session is compared to has a relevant effect in the global accuracy of the system. It is interesting to see that the size of the dataset becomes more important as soon as the size of the group also grows. Larger datasets maintain a better throughout accuracy even if the size of the group increases.

Also important are the following statements that show a series of problems with the proposed model:

- Small models do not take advantage of some parameters since not enough words or instances may be always available. This is highly related to the necessity of having large enough models for the accuracy to be good enough. At the same time, this problem goes hand in hand with hit rates. When using *n-graphs* methodologies, hit rate is higher than when searching whole words. If fewer words are found in the model, as expected, the accuracy decreases.

- A Forest of trees structure can be a better alternative than the single tree model. Unfortunately, the scalability problems it presents, comparable to those found when using the *n-graph* methodology using Relative and Absolute distances, deem this structure unfeasible with current available technology when large datasets are used.

- The discussion of where to set a threshold of the minimum number of words to be found in the models to consider an origin session valid, could depend on the information available, the desired accuracy, and the overall security established value. Setting this threshold value too high means that many sessions will be discarded for identification or authentication. On the other hand, setting it too low could mean that the accuracy could decrease radically.

- In the case where the number of events may be too low, a multi-structural strategy could be used, combining the results from both the *n-graphs* frequency and the logical tree models methodologies using weighted fusion methods.

6.1.2 On the underlying methodology

The second proposed objective was the following: *Identify a user using the largest sequence of letters of a word and the latencies associated with each keystroke.*

This objective is also considered to have been fully achieved. These are the most relevant results related to the proposed objective:

- The best results are achieved when no recursion is used. This is considered to be of high relevance because it confirms that contextual information, and how it is handled, is very important. This parameter not only improves the results when a certain number of words have been found in the model, but also reduces the number of searches that have to be performed of a word, or a sub-word, in the model, thus, increasing the performance vastly.

- Child times should not be discarded. The improvement is substantial, proving, again, how important contextual information is. In this sense, further experiments could be proposed to better determine what timing intervals should or should not be used when partial words are found in the model.

- One-letter words do not help identify users even if these are the most common when comparing a session and a model. Discarding these short words improves the results. On the other hand, two-letter words and up are interesting when identifying users, even if differently depending on the distance measurement and the chosen method.

The first two conclusions prove that having more information just for the sake of it is not always good, relevant, or even necessary. Fewer data, but of much better quality, helps reduce misleading variance. In both situations, what is being evaluated is whether the position of letters is relevant, and, in both cases, it is found that it is. More specifically, using the tail of a word that has not been found in the tree model

as if it was a new word has always been considered *cheating* because the tail is being searched in the model as if it was a whole new word. This goes against the conceived idea that position is relevant. On the second case, even if a word is partially found in the tree, using the same time intervals of similar words, that have the same root letters, proves an improvement in the results, than simply discarding these words. The possibility of inferring time intervals should be further researched to improve, even more, the presented results.

6.1.3 On the parameters to build and search the models

The third objective was: *Determine the model building and searching parameters that better help during the identification process.*

The results show that this objective has also been achieved. It is probable that not all the parameters that better discriminate users have been found, but the ones that have been identified do help in a relevant way when identifying users. These parameters have been identified, and described, in Tests 2, 3 and 4, being the following the most relevant conclusions:

- The better the model is in terms of the number of events incorporated, the better the results are. Again, as in the first objective, this is a conclusion that is observed throughout the results. The best results have always been obtained when more information was available in the tree. This volume of data should not be confused with what, in the previous section, has been described as quality data. Both statements can be true at the same time. Having more words in the tree improves the results. Then, when words are searched, keeping only the most relevant features is also paramount.

- These tests have also shown that the depth at which a word is found in the tree may be a relevant feature. Unfortunately, this parameter seems to depend on the distance measurement and the method being used to identify the owner of a session. When depth is not a parameter of the method to identify a user, it has been found out that lengths of $[2-5]$, $[3-7]$ provide the best results, discarding shorter and longer words. This suggests the idea that users have a natural rhythm specially during short bursts of information. On the other hand, when depth is a key parameter of the method to identify sessions it has been found that the best results are obtained when all words from two letters upward were used. More research could be focused on this matter.

- The minimum number of words has already been discussed in the previous objective. Again, when evaluating this objective, it has been found that the number of words found in the model conditions greatly the obtained results.

Apart from these conclusions, it is also considered highly relevant that the test performed on behavioral features pointed out that adding more information to the tree, of another nature, such as mistakes and other key combinations, thus segregating samples even more and causing a lower hit rate, punishes previously stored and valid samples, and eventually, the overall accuracy of the system. In this sense, keeping only samples of words in the tree is recommended. A further study could be carried out to determine if, from these samples, only those, for example, typed in lower case letters are better than those typed using modifier keys that can slow the natural rhythm a user has.

6.1.4 On age group and gender separation

Another research objective was: *Determine if gender or age group present particularities that can be useful to build better models.*

This objective has been evaluated in Test 7 with different results depending on the feature being analyzed. The results of this test reveal the following conclusions:

- The most significant and interesting result is obtained from the age group a user belongs to. The best results are obtained when sessions are compared against models where there is a significant age difference. This suggests that, with the available dataset, different groups of users present significant different rhythms. These differences allow for adapted models to be built so that accuracy can be increased. At the same time, the possibility of determining the age group a user belongs to through its typing rhythm could be suggested.
- Gender has been discarded as a feature affecting identification. No pattern has been found when only gender has been evaluated. Unfortunately, the idea of determining gender through the proposed methodology is not that obvious.
- The fact that younger women are the ones that are most incorrectly identified should set the grounds for further research regarding this particular group. The idea that younger students have irregular rhythms when typing on computer keyboards could suggest a study on behavioral features and younger users, much more affected by the generalized and ubiquitous use of mobile devices.

6.1.5 On authentication

Finding if the proposed methodology is also a valid authentication method has also been one of the objectives of this research: *Find out if the proposed methodology is also a good candidate to authenticate users instead of only identifying them.*

This objective has also been fully achieved and with very good results. The problem this methodology presents is that users have to type a minimum number of words for the accuracy to be good enough. This procedure may annoy users, especially when, after typing such number of words, a user may not be correctly authenticated and the process has to be started all over again. In the end, this method may not be feasible to implement, even if being a valid one, if user sentiment has to be taken into account. If a user belongs, though, to an age and gender group where identification accuracy is *normally* or *intrinsically* better, the number of words they should type could be revised and lowered, improving user satisfaction.

Some web services have implemented this solution, as has been previously commented. The idea of using it in multimodal schemes, or as a way to harden a previously entered password, could be proposed as a less annoying methodology. Another example of this procedure could be subject confirmation (rather than identification), where a user, before beginning a test could be asked to submit a number of sentences only to have an additional prove that could help avoid cheating. In such cases the importance of the biometric results could be weighted to lower values, not requiring such high levels of accuracy, as has been suggested by studies on password hardening.

As with other results previously outlined in this chapter, the bigger the dataset used, the easier it is to correctly authenticate a user. At the same time, the minimum number of words is what most determines the accuracy of the authenticating system.

6.1.6 On behavioral features

The last objective is based on behavioral features: *Determine if other behavioral features, such as mistakes users make, word or sentence frequency, or word delimiters, are also valid features to identify users or, on the other hand, these should be discarded.*

This has been evaluated in Test 4. The results for these tests were also quite explicit in terms of the validity of behavioral features:

- Word frequency scaling improves the results in some cases but only marginally and only when the hit rate is high due to a much larger number of words present in the model.
- The use of the Sentence scaling algorithm proposed in this study does not improve the results. It could be argued that additional and different methods or algorithms to evaluate the most common use of word combinations could be analyzed. Some of these could substantially improve the results. This is left as future work.
- The study of the mistakes users make, and the use of only the *space* key as a delimiter show an interesting fact. Both only accomplish giving worse results, but

it is interesting to see that as soon as more and more *special* keys are allowed in the model, the results worsen even more. A study where only a limited number of events are captured is proposed. This could be achieved, for example, limiting the capture to letters in the [a-z] range (lower case values), without taking into account modifier keys, numbers, function keys or other elements that may distract the users from their natural rhythm.

- When all modifiers are used at the same time, the results are the worse. Again, it seems verified that having specific quality data is much more important than having lots of random data.

In all cases it is considered that the main problem is the low hit rate and the process of messing with information in the tree. The behavioral features analyzed depend, in great measure, on the number of instances found in the models. A basic frequency test suggested that, compared to the number of times common words were found, these features brought little to the overall result, and at the same time, decreased the accuracy due to the fact that instances previously allocated to other words of the tree, had now a particular node that was seldom used, deeming the relevance of the previous word.

6.1.7 On the main objective

The main objective of the proposed research has been defined as: *Determine if the use of Keystroke Dynamics and models based on contextual information and behavioral features allows the possibility of identifying or authenticating users with a small margin of error.*

After having evaluated all previous sub-objectives it is considered that the main objective of the research has been accomplished. Again, it should be noted that this research has been performed on an uncontrolled environment where users have never been told what to write or how to write it. This should be considered as highly relevant when the presented results and conclusions are evaluated. Excellent results can be achieved disregarding the location users type from, their computer equipment or other environmental features. Extending these tests to the use of mobile devices and the possibility to build different models for different user environments could be also evaluated in further studies.

6.2 Conclusions on the proposed Hypotheses

In Chapter 3, the Hypotheses and sub-Hypotheses below have been proposed. This section discusses whether these have been proved true or not.

The first hypothesis stated: *The global size of the model and the number of samples are highly relevant when building quality models.* This hypothesis had also two sub-hypotheses related to it. The first was: *If more samples are collected from users, the template will be better and the chances of identifying them with a smaller error will improve.* The second was: *If a good number of samples per user are available, the proposed method will perform better than using n-graphs frequency models.*

From the results obtained, it has been clearly proved that these Hypotheses are true, with only a nuance regarding the second sub-hypothesis. It is true that users can be identified using Keystroke Dynamics and contextual information. At the same time, the size of the models has been tested throughout the research proving that bigger models, with a larger number of words, perform better. The comparison against a traditional *n-graph* methodology depends greatly on the dataset and the computer resources available. With resources available today and applying what has been tested to a real environment, the use of single models combining all sessions is much more computer-resource friendly than comparing a session to other sessions individually. With this in mind, it is also true that the Forest of trees method did produce somewhat better results when datasets were smaller. It should be pointed out that all these methodologies should be considered optimal in the long run. As has been seen, initially, when the number of training sessions is low, accuracy is also affected. On the other hand, once the training data becomes relevant, something that could easily happen when the proposed methods are implemented in learning environments where users are constantly asked to submit written assignments, the results improve vastly. With time, the proposed methodology in this study surpasses the results obtained using the *n-graph* methodology, not only in accuracy but also in performance and computer needs.

The second hypothesis proposed the following: *It is possible to identify a user on a computer, with a small margin of error, using Keystroke Dynamics, contextual information, and behavioral features.* This had also two sub-hypotheses. The first one said: *Not all model building parameters are equally relevant, some will be more suited to better identify users.* The second was: *Behavioral features such as mistakes users make, word and sentence repetition, and the use of particular key combinations can also be important features when identifying users.* The proposition of these sub-hypotheses may lead to a misinterpretation of what was really being tested in this research because the study of the building parameters has little relation to contextual information. The main hypothesis has been proved true: users have been identified with a small error using contextual information. This is considered paramount. The first sub-hypothesis has also been proved true: choosing optimal parameters, not only when building the proposed tree models but also when searching information in the tree has been considered essential to the research and the accuracy of the system. Even if this may

sound obvious, part of the efforts in this study have been centered on finding which are these parameters and how these affect the outcome of the experiments.

In the case of behavioral features, the sub-hypothesis has been proved false or non-relevant. Only in those cases where lots of information is available and hit rate of special key combinations is high, it can be said that either errors users make or other behavioral features such as word or sentence frequency, or the use of special key combinations, can provide somewhat better results. With the available dataset for this research it has been found that these features do not add relevant information to the models and are not encouraged to be used. The same experiments could be replicated elsewhere, on an environment with much more activity, to reevaluate these features and their effect.

The third hypothesis stated: *The proposed methodology can be valid to authenticate users instead of only identifying them.* Again, this hypothesis has been proved true. Using the proposed methodology centered on contextual information, it is possible to authenticate users with rather low error rates. The best EER value is comparable to results from previous research.

Finally, the last hypothesis stated: *The gender and the age group a user belongs to can be useful to build better models and improve accuracy.* This hypothesis has been proved true, if only partially. As the results have shown, gender does not really provide a means to improve classification. Even if it is true that younger women have been identified as the group with the most irregular models, women and men tend to have similar and regular models that, once separated, do not help in the identification process. On the other hand, age group has been found relevant and distinct enough to think of the possibility of optimizing models based on this feature. More specifically, the difference in rhythm is mostly seen between younger and senior models. Again, the possibility of identifying the age group a user belongs to becomes a reality when enough models of the different age groups are available.

6.3 Conclusions about performance

One of the key points during the execution of the proposed experiments to evaluate the Objectives and Hypotheses has been that of *performance*. When comparing many sessions against a large number of models, the execution time could be very high, and the computer resources needed could be very demanding.

For this research a cluster of 8 machines (with 16 GB of RAM, 8 processors each, and an additional 2 TB of shared disk space) running *slurm*[1] over a *glusterfs*[2] distributed

[1]slurm: `https://slurm.schedmd.com`. Last accessed: September 30, 2017
[2]glusterfs: `https://www.gluster.org`. Last accessed: September 30, 2017

file system has been used. This material is part of the computational cluster at the University of Andorra, and was generously made available free of cost.

Initially, the process of building models has been slower when using the proposed logical trees models. After setting those parameters that better helped organize the information in the tree the time to build the models has reduced considerably. Still, building an *n-graph* model was faster (up to 2 times faster). On the other hand, searching words in the tree models was much faster than comparing samples between sessions using the Relative and Absolute distance measurements. Once the number of sessions became rather large, a process that could be achieved in seconds using the logical tree methodology became totally unfeasible using either the *n-graph* alternative, or the Forest of trees logical structure. To reduce the impact of the size of available sessions two ideas have been proposed: use a subset of session to compare an origin session to (chosen randomly or by quality), and choose a minimum number of graphs or words to counter the possible effect where the accuracy is reduced when comparing a training session to a smaller number of sessions.

All in all, *performance* should be considered a key factor when choosing a methodology to identify or authenticate users using Keystroke Dynamics and contextual features. It is thought that, when little data is available, a Forest of trees structure or an *n-graph* methodology could be used, if only to increase hit rate. On the other hand, if a large dataset, in time, is made available, a single data storage structure that takes advantage of contextual information is recommended.

6.4 Applications

In the State of the Art chapter a series of common applications of Keystroke Dynamics have already been hinted at. The most common applications of this biometric technique focus on authentication, verification, and identification of individuals. How and why this is performed can also imply a different range of applications. Authentication is basically used when a user wants to be granted access to an application or system, or when sensible operations have to be performed, establishing a short-lived token that allows the user to do such operations for a limited period.

The basis of the proposed research has been, mostly, identification, but also authentication based on free text captured over rather long periods. It has been proved that good rates of identification can be achieved using contextual information related to timing intervals from the natural rhythms users have when typing on a keyboard. A clear application of the proposed research involves online learning environments. Nowadays, it is normal to see that high education studies are, more and more, offered using online environments that do not require for students to be present in *traditional*

face-to-face classes. This, as is the case of the Open University of Catalonia, involves the possibility of carrying out the exams also in a non-present way.

Having such online learning environments suggests the need of having a more robust way of knowing the true identity of users using virtual campuses, be it during normal classes but also when taking tests. The first application that comes from the proposed research is the possibility of implementing an application-wide capture module that collects samples and builds models for all the users in the virtual campus. This procedure should be carried out during long periods to ensure that the accuracy of the system is good enough.

As hinted throughout this document, when not enough samples are available to build reliable tree models, the possibility of using multi-model schemes could be attempted, combining, for example, *n-graphs* methodologies. Over time, though, these could be discarded in favor of using only contextual tree models.

Students and teachers alike use the virtual campus for a wide different range of applications. Examples of such applications could be: submitting and accessing course documentation, submitting questions to teachers, participating in debates, submitting tasks and papers for evaluation, chatting with other students, composing wikis, writing assignments in collaboration, among many others. When there is no real contact with the student, being sure that whoever sends the information is who they claim to be can be tricky. Keystroke Dynamics, and the use of contextual information proposed in this study, can be of help to ascertain the identity of such individuals.

When users access the virtual campus environment they have to be authenticated. Keystroke Dynamics can be used to harden passwords. When users are asked to enter their password, at the same time, they could be asked to write a short piece of text to be validated together with the traditional credentials. This could add another layer of confirmation that could be used to be sure that users are who they claim to be.

Once authenticated, the capture module could run transparently, updating the user model continuously. A drawback of this methodology is that when users have to submit information to the virtual campus, they can *prepare* it offline, using other editors than the ones available online. Once ready, they could simply copy and paste it into the forms available online. These behavior is not interesting to the capture module, because no typing information is ever captured. The possibility of having system-wide capturing applications could be suggested, always taking into account the privacy of the users.

When users compose messages for the teacher, to participate in debates in discussion forums, while composing wikis, when chatting or while writing online assignments, the logical tree models could be queried to continuously assert the identity of these students. What to do if the identification process fails is something that is left as a

policy to be determined. An interesting application would be to be able to determine who has written what on assignments authored by more than one student.

The direct application in online tests is obvious. Users could not only be monitored for the whole duration of the exam, but also when accessing the test, asking them to validate their identity before taking the test, as a form of additional authentication.

All these applications have been restricted to online learning environments, but such limitation should not be a requirement or be imposed. The widespread use of social networks applications, for example, where users submit a lot of content suggests the possibility of verifying that no user is supplanted, damaging a user's reputation. On the other hand, the same technique could be used to verify that a user really sent incriminating information in fraudulent activities.

When talking about online reputation, the same could be attempted in large scale blogging sites, where many users may have access to publishing written content. On such cases, the stealing of a password, via social engineering skills for example, could also lead to identity theft. Monitoring the typing rhythm of submissions authored on the blogging platform could also lead to identifying impostors.

Another possible application would be to have a centralized keystroke database to authenticate or identify users on the Internet. One of the problems Keystroke Dynamics may present is the number of samples available to build reliable models. This could be solved by having a centralized keystroke database. Many sites outsource user authentication to third-party companies like Facebook[3], Twitter[4], or GitHub[5] to avoid having a user local database that can be exploited, and at the same time, to avoid users having a large number of accounts spread throughout many different services. Using a single login/password combination users can access a plethora of sites and services. The same could be implemented using Keystroke Dynamics. Using this methodology, models would be much more robust, and processes of authentication and identification could be performed with greater accuracy across several applications.

Finally, even if the identity of an individual cannot be established, after having enforced privacy policies, still, the possibility of placing them in an age group could be attempted, as has been proved by the performed research. This could be useful, for instance, for advertising companies to better focus their campaigns.

[3]Facebook: `https://www.facebook.com`. Last accessed: September 30, 2017
[4]Twitter: `https://twitter.com`. Last accessed: September 30, 2017
[5]GitHub: `https://github.com`. Last accessed: September 30, 2017

6.5 Summary

This chapter has presented the conclusions of the work carried out in this Thesis. Most of the goals have been achieved and, at the same time, most hypotheses have been proved true while a minority have been proved false, non-relevant, or in need of more testing.

After the conclusions, a range of applications have also been outlined. These are just an example of possible implementations without the idea in mind of being exhaustive. Any kind of application or environment where users submit large quantities of text could be used as a basis for the proposed methodology to be applied.

No study is ever complete. Many details and ideas have been left as future work due to the impossibility of analyzing all possible combinations of parameters. The following chapter discusses some of these.

7 | Future Work

Many details, tests, features, combinations of parameters, and ideas have been left behind when performing the experiments in Chapter 5.

7.1 Proposed ideas left as future work

Detailed below are some ideas or concepts that have been left as future work and that could be further researched:

- At the beginning of the research, when the initial steps to build the logical tree models were being taken, a decision had to be made to determine where a word would end and a new one would begin. If a delimiter was found, the solution was easy. On the other hand, what would happen if a user simply stopped mid-word to think, or if they used the mouse to change position in the document to continue writing? In such cases, if no action was taken, the different words may not be detected. So, a value of $300ms$ was chosen empirically as the interval of silence to determine when a word had finished and a new one had begun. Different values were tried, from 100 to $1,000ms$. In all cases, different results were obtained without being certain which was the one that behaved better. It was discussed whether every user could have a different value from which to determine new words. Imagine a proficient user and compare their typing to that of an inexperienced writer. Each could have different values of word interval silence that could help build better models to identify them proficiently. It is left as future work to determine whether different values should be assigned to different users and, if not, which is the best global value for this parameter.

- In this study, five different distance measurements have been evaluated to determine the owner of a session. In most cases, the best distance measurement has been the Chebyshev one. It has been seen, too, that some methods favored other distances. The five distance measurements tried in this study are among those that are most popular in the literature. Something that is left as future work is to try other distance measurements, for example, those from a recent study [112].

- When different methods have been evaluated to determine the owner of a session, something that has never been totally addressed is which of the four features analyzed (Press–Release (PR), Press–Press (PP), Release–Press (RP) and Release–Release (RR)), and their possible combinations, is the best. It is true that a couple have been identified as those that *most of the time* have performed better or worse. In the end, though, when using the Weighted mean of distances, revised method, a sole combination of these four features has been used. If this is the best option is yet to be determined. In the case of this study, the chosen feature for this particular method has given good enough results. On the other hand, using these same combinations with other methods, has produced as good results as other combinations. Also, using all combinations when using fusion methods has been proved to be highly valuable. In the end, no feature, or feature combination outshines the rest all the time, so more research could be focused in this area.

- Related to the previous point, a modification of the Weighted mean of distances, revised method could be tried using only PR or RP values. These have been identified as those that behaved better and worse, respectively, in many tests. Choosing these to try the Weighted mean of distances, revised method could be interesting, mainly, in terms of performance. So far, determining the mean value of the four chosen features, for every word found in the models, and then finding the median value can be a highly demanding task when the number of available words is high. If this process is to be simplified, with the same results, it would be highly interesting in terms of performance.

- Related to the previous points, it could be argued that not all possible settings have been evaluated for all possible methods, distance measurements and parameters. A more exhaustive study could be carried out. It seems, though, that if any of these combinations is to be a serious improvement over the rest, this may be marginal taking into account the already obtained results. Of course, a completely different thing would be to find a combination that achieved much better results when all sessions, no matter the number of words found in the model, are accepted as valid sessions, or when little information is available. Throughout this study these have been found to be highly difficult to classify correctly. If any efforts should be put into improving the presented research it should be in this area.

- Some studies have suggested the idea that the typing rhythm is not constant over time. This idea has been taken as far as having models adapted, keeping only recent samples that better suit the natural rhythm at a given moment in time. The same methodology could be tried with the available samples. In the case of this study, all samples have been added into the model and only those outside a

number of standard deviations have been later removed. In no way the moment those samples have been captured, has been used to build the models. It could be tried to set interval dates and allow the models to be built only with information from such intervals. All throughout the results sections it has been seen that the quality of the data was paramount. Many times it has been stated that it was far better to have little data of quality than huge amounts of misleading information.

- Taking the idea of the previous point a step further, it could be studied if the time of day, or the day of the week a sample is submitted presents different characteristics that can help build better models for each user. It should be noted that as soon as new models are built with less and less information, the need for additional samples increases rapidly. If this is not an issue, then it would be suggested to evaluate all these possible scenarios.

- Also related to the particularities of the users' environment, the possibility of differentiating the device used, be it a desktop computer, a mobile device (smartphone or tablet), or laptops could be attempted. At the same time, the working place could also be evaluated as a relevant factor. It could be argued that users may have different timing templates if they are relaxed at home, rather than at the university or in stressful working environments. Previous research shows that users that are most consistent in their environment are the ones that are easier to identify and, at the same time, the ones that have tighter models. For this study, all data coming from all types of devices and settings has been captured, but not all has been used. It has been stated that only information coming from desktop and laptop computers has been used. This was done in such way because the information from mobile devices was rather scarce and was thought that it could render biased results. On the other hand, no distinction has been made on the origin of the data. It could have been easy to limit the data only to that generated in the labs at the university, or that coming from user's homes. In this case, it was thought that having as much information as possible would be useful. A possible line of future work could evaluate all these factors and determine if these affect the quality of the models and the accuracy of the whole system.

- Something interesting from the results is that as soon as additional information from the keys a user had used are added into the model to be used as new *words* the results tend to be worse. Examples of these are the mistakes users make and, also, other keys like navigation keys, or combinations such as CTRL+C, CTRL+Z... It would seem that as soon as other keys outside the letters are used the rhythm is broken. This could suggest whether other keys could also be left out of the models. For example, the number keys, accents, Fx keys ... In the

extreme case, any key not in the range of [a-z] could be discarded and the model could be evaluated.

- All in all, all previous points, apart from being focused on how to increase accuracy, not only when optimal conditions are met, but also when little information is available, deal with model optimization and scalability. When the level of information is very high, and many users are available, the computer resources needed to evaluate sessions may become prohibitive. At this point, any idea that can maintain accuracy and remove less interesting information from the model trees would be welcome. Once an optimal point is reached, the idea of trying a Forest of trees could be again considered.

- A recurrent problem with the accuracy obtained was the necessity of having a higher hit rate for some of the studied features. This was deemed obvious when features such as word frequency and sentence constructions were evaluated. With small datasets, it also becomes obvious that such features are difficult to use but, even more, evaluate. These features have been used to reward or punish distance measurements tying together behavioral particularities and Keystroke Dynamics. The work carried out in this study is pretty much incipient and has much to be explored, not only from the features that can be extracted, but also from the different formulas that can be used to reward or punish frequent elements.

- Related to the previous point and an idea that could be attempted to increase the reliability of the models, and also the hit rate, could be to further train models using especially crafted sentences or paragraphs for the users to copy type. These could be arranged to be performed at regular periods, as if it was a change of password. The data to be typed could be extracted from those words that *other* users had typed the most on their models. This way, all models could have a basic set of similar information from the words that are most frequently used. This could render less biased results when words are not found and discarded in those models where a user has never typed a particular word.

- Another idea could be to attempt to store only those words that better differentiate users. Figure 4.12 shows an interesting fact. The user owner of the session has most distances in the range of 0 to $50ms$. In contrast, other users have most distances in values from 0 to $200ms$. The idea would be to identify the words that fall between the range of distances between 50 and $200ms$ and try to establish a model from these words. These could be repeated for all models and users to create a density map of *interesting* words. Do these have a particular length? Some combination of letters that deem them more equal than others? Why

these words have a greater distance to zero could set the basis of an interesting experiment.

- One of the most interesting results from the age group and gender test is the fact that the results when either identifying or authenticating users is better if the compared age group is rather distant. A field of research that could be explored is precisely what makes younger and senior users different in their natural typing rhythm. Is this rhythm adapted over time? Do all users adapt their typing when they grow older? Will today's younger users have a similar rhythm when they are in the older age group as today's older users, or has this something to do with the particularities of our current time? Does this mean that all users, years from now, will have a similar typing rhythm and that this difference between age groups will no longer be relevant? Does it have to do with the relative young age of computers and how users from all ages have to learn to use them? Too many questions left to be answered in need of a proper research study.

- A very interesting study could be to replicate the tests and experiments performed in this research in other universities or other learning institutions. It is not uncommon to see that experiments are carried out in different parts of the planet to determine if the observed behaviors in one place are also observed in other social or cultural environments. In general, the greater the differences the more relevant the results are considered. In this sense, trying the proposed methods with different input languages, or other cultural and social settings would be of high interest.

7.2 Summary

This chapter has described some of the ideas proposed throughout this document that have been left as future work. Some of the proposed concepts are easier than others to implement or test. Nevertheless, the proposed research in a PhD Thesis has to have a finite goal and, eventually, be done with it. It is considered that a specific goal has been achieved with this research and that all proposed lines of future work should be considered as such: future work.

Bibliography

[1] Ahmed, Ahmed Awad E., Traore, Issa, and Ahmed, Almulhem. "Digital Fingerprinting Based on Keystroke Dynamics". In: *Second International Symposium on Human Aspects of Information Security & Assurance*. 2008, pp. 94–104 (cited on p. 36).

[2] Al Solami, Eesa. "An examination of keystroke dynamics for continuous user authentication". PhD thesis. Queensland University of Technology, 2012 (cited on pp. 19, 43).

[3] Al Solami, Eesa, Boyd, Colin, Clark, Andrew, and Islam, Asadul K. "Continuous Biometric Authentication: Can It Be More Practical?" In: *High Performance Computing and Communications (HPCC), 2010 12th IEEE International Conference on*. IEEE. 2010, pp. 647–652 (cited on p. 25).

[4] Alsultan, Arwa and Warwick, Kevin. "Keystroke Dynamics Authentication: A Survey of Free-text Methods". In: *International Journal of Computer Science Issues* 10.4 (2013) (cited on pp. 26, 30, 40, 47).

[5] Alsultan, Arwa and Warwick, Kevin. "User-Friendly Free-Text Keystroke Dynamics Authentication for Practical Applications". In: *Systems, Man, and Cybernetics, 2013 IEEE International Conference on*. IEEE. 2013, pp. 4658–4663 (cited on pp. 38, 47).

[6] Alsultan, Arwa, Warwick, Kevin, and Wei, Hong. "Free-text keystroke dynamics authentication for Arabic language". In: *IET Biometrics* (2016) (cited on pp. 15, 39, 47).

[7] Antal, Margit and Nemes, Győző. "Gender recognition from mobile biometric data". In: *2016 IEEE 11th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. IEEE. 2016, pp. 243–248 (cited on p. 19).

[8] Antal, Margit, Szabó, László Zsolt, and László, Izabella. "Keystroke Dynamics on Android Platform". In: *8th International Conference Interdisciplinarity in Engineering* (2014) (cited on pp. 26, 30, 42, 48).

[9] Antal, Margit, Szabó, László Zsolt, and László, Izabella. "Keystroke dynamics on android platform". In: *Procedia Technology* 19 (2015), pp. 820–826 (cited on p. 57).

[10] Araújo, Lívia C. F. et al. "User authentication through typing biometrics features". In: *IEEE Transactions on Signal Processing* 53.2 (2005), pp. 851–855 (cited on pp. 14, 25).

[11] Awad, Ahmed, Ahmed, Ahmed Awad E., Traore, Issa, and Almulhem, Ahmad. "Digital Fingerprinting Based on Keystroke Dynamics". In: *Second International Symposium on Human Aspects of Information Security & Assurance*. 2008 (cited on p. 47).

[12] Azevedo, Gabriel L. F. B. G., Cavalcanti, George D. C., and Carvalho Filho, Edson C. B. "An approach to feature selection for keystroke dynamics systems based on PSO and feature weighting". In: *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. IEEE. 2007, pp. 3577–3584 (cited on p. 31).

[13] Balagani, Kiran S., Phoha, Vir V., Ray, Asok, and Phoha, Shashi. "On the discriminability of keystroke feature vectors used in fixed text keystroke authentication". In: *Pattern Recognition Letters* 32.7 (2011), pp. 1070–1080 (cited on p. 11).

[14] Banerjee, Salil P. and Woodard, Damon L. "Biometric authentication and identification using keystroke dynamics: A survey". In: *Journal of Pattern Recognition Research* 7.1 (2012), pp. 116–139 (cited on p. 14).

[15] Bartlow, Nick and Cukic, Bojan. "Evaluating the reliability of credential hardening through keystroke dynamics". In: *Software Reliability Engineering, 2006. ISSRE'06. 17th International Symposium on*. IEEE. 2006, pp. 117–126 (cited on p. 16).

[16] Bartlow, Nick and Cukic, Bojan. *User Credential Hardening through Keystroke Dynamics*. Tech. rep. West Virginia University, 2014 (cited on p. 43).

[17] Bello, Luciano et al. "Collection and publication of a fixed text keystroke dynamics dataset". In: *XVI Congreso Argentino de Ciencias de la Computación*. 2010 (cited on p. 45).

[18] Bergadano, Francesco, Gunetti, Daniele, and Picardi, Claudia. "User authentication through keystroke dynamics". In: *ACM Transactions on Information and System Security* 5.4 (2002), pp. 367–397 (cited on pp. 27, 107).

[19] Bishop, Christopher M. *Pattern recognition*. Springer, 2006 (cited on p. 30).

[20] Bleha, Saleh Ali. "Recognition systems based on keystroke dynamics". PhD thesis. University of Missouri, Columbia, 1988 (cited on p. 40).

[21] Bleha, Saleh Ali, Slivinsky, Charles, and Hussien, Bassam. "Computer-access security systems using keystroke dynamics". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.12 (1990), pp. 1217–1222. ISSN: 0162-8828 (cited on pp. 15, 26, 40, 48).

[22] Bolle, Ruud M et al. *Guide to biometrics*. Springer Science & Business Media, 2004 (cited on pp. 20, 21).

[23] Bor, Daniel. *The ravenous brain: How the new science of consciousness explains our insatiable search for meaning*. Basic Books, 2012 (cited on p. 80).

[24] Bours, Patrick. "Continuous keystroke dynamics: A different perspective towards biometric evaluation". In: *Information Security Technical Report* 17.1 (2012), pp. 36–43 (cited on pp. 17, 19, 38, 43, 47).

[25] Bours, Patrick and Barghouthi, Hafez. "Continuous Authentication using Biometric Keystroke Dynamics". In: *Norsk informasjonssikkerhetskonferanse (NISK)* (2009) (cited on pp. 37, 38, 47).

[26] Brizan, David Guy et al. "Utilizing linguistically-enhanced keystroke dynamics to predict typist cognition and demographics". In: *International Journal of Human-Computer Studies* (2015) (cited on pp. 38, 47).

[27] Buch, Tarjani et al. "An enhanced keystroke biometric system and associated studies". In: *Proceedings of Student-Faculty Research Day, CSIS, Pace University* (2008) (cited on pp. 37, 47).

[28] Buriro, Attaullah, Crispo, Bruno, Del Frari, Filippo, and Wrona, Konrad. "Touchstroke: Smartphone User Authentication Based on Touch-Typing Biometrics". In: *New Trends in Image Analysis and Processing–ICIAP 2015 Workshops*. Springer. 2015, pp. 27–34 (cited on p. 42).

[29] Campisi, Patrizio, Maiorana, Emanuele, Lo Bosco, Maurizio, and Neri, Alessandro. "User authentication using keystroke dynamics for cellular phones". In: *IEEE Transactions on Signal Processing* 3.4 (2009), pp. 333–341 (cited on p. 12).

[30] Ceker, Hayreddin and Upadhyaya, Shambhu. "Enhanced Recognition of Keystroke Dynamics using Gaussian Mixture Models". In: *Military Communications Conference, MILCOM 2015* (2015) (cited on pp. 39, 47).

[31] CENELEC. *European Standard EN 50133-1: Alarm systems. Access control systems for use in security applications. Part 1: System requirements*. 2002 (cited on p. 21).

[32] Chantan, Charoon, Sinthupinyo, Sukree, and Rungkasiri, Tippakorn. "Improving Accuracy of Authentication Process via Short Free Text using Bayesian Network". In: *International Journal of Computer Science Issues* 9.3 (2012) (cited on pp. 38, 47).

[33] Cheng, Qi. "User Habitation in Keystroke Dynamics Based Authentication". MA thesis. Morgantown, West Virginia, 2007 (cited on p. 16).

[34] Clarke, Nathan L. and Furnell, Steven M. "Authenticating mobile phone users using keystroke analysis". In: *International Journal of Information Security* 6.1 (2007), pp. 1–14 (cited on p. 12).

[35] Clarke, Nathan L., Furnell, Steven M., Lines, Brian M., and Reynolds, Paul L. "Keystroke dynamics on a mobile handset: a feasibility study". In: *Information Management & Computer Security* 11.4 (2003), pp. 161–166 (cited on p. 12).

[36] Curtin, Mary et al. "Keystroke biometric recognition on long-text input: A feasibility study". In: *Proc. Int. MultiConf. Engineers & Computer Scientists (IMECS)* (2006) (cited on p. 68).

[37] Davoudi, Homa and Kabir, Ehsanollah. "Modification of the relative distance for free text keystroke authentication". In: *Telecommunications (IST), 2010 5th International Symposium on*. IEEE. 2010, pp. 547–551 (cited on p. 107).

[38] Davoudi, Homa and Kabir, Ehsanollah. "User Authentication Based on Free Text Keystroke Patterns". In: *Proceedings of the 3rd Joint Congress on Fuzzy and Intelligent Systems*. 2010 (cited on p. 107).

[39] De Ru, Willem G. and Eloff, Jan H. P. "Enhanced password authentication through fuzzy logic". In: *IEEE Expert* 12.6 (1997), pp. 38–45 (cited on p. 30).

[40] Dorca Josa, Aleix, Morán Moreno, Jose Antonio, and Santamaría Pérez, Eugènia. "Using Keystroke Dynamics and context features to assess authorship in online learning environments". In: *INTED2017 Proceedings*. 2017 (cited on p. 39).

[41] Dorca Josa, Aleix, Santamaría Pérez, Eugènia, and Morán Moreno, Jose Antonio. "Identificación de usuarios mediante dinámica de tecleo en entornos de entrada libre usando información de contexto". In: *XXXI Simposium Nacional de la Unión Científica Internacional de Radio (URSI, 2016)*. 2016 (cited on p. 39).

[42] Dowland, Paul S., Singh, Harjit, and Furnell, Steven M. "A preliminary investigation of user authentication using continuous keystroke analysis". In: *Proceedings of the International Conference on Information Security Management and Small Systems Security*. 2001, pp. 215–226 (cited on pp. 35, 47).

[43] Duc, Nguyen Minh and Minh, Bui Quang. "Your face is NOT your password Face Authentication ByPassing Lenovo–Asus–Toshiba". Ha Noi University of Technology – Viet Nam. 2009 (cited on p. 2).

[44] Dvorak, August, Merrick, Nellie L, Dealey, William L, and Ford, Gertrude C. "Typewriting behavior". In: *New York: American Book Company* 1.6 (1936) (cited on p. 12).

[45] Epp, Clayton. "Identifying emotional states through keystroke dynamics". PhD thesis. University of Saskatchewan, 2010 (cited on pp. 13, 14, 25, 43, 57).

[46] Furnell, Steven M. and Clarke, Nathan L. "Biometrics: no silver bullets". In: *Computer Fraud & Security* 2005.8 (2005), pp. 9–14 (cited on pp. 2, 14, 22).

[47] Gaines, R. Stockton, Lisowski, William, Press, S. James, and Shapiro, Norman. *Authentication by keystroke timing: Some preliminary results*. Tech. rep. DTIC Document, 1980 (cited on pp. 15, 26, 40, 48).

[48] Giot, Romain, Dorizzi, Bernadette, and Rosenberger, Christophe. "Analysis of template update strategies for keystroke dynamics". In: *Computational Intelligence in Biometrics and Identity Management (CIBIM), 2011 IEEE Workshop on*. IEEE. 2011, pp. 21–28 (cited on pp. 14, 25).

[49] Giot, Romain, El-Abed, Mohamad, and Rosenberger, Christophe. "Greyc keystroke: a benchmark for keystroke dynamics biometric systems". In: *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*. IEEE. 2009, pp. 1–6 (cited on p. 41).

[50] Giot, Romain, El-Abed, Mohamad, and Rosenberger, Christophe. "GREYC Keystroke: a Benchmark for Keystroke Dynamics Biometric Systems". In: *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. Washington, District of Columbia, USA: IEEE Computer Society, 2009 (cited on p. 45).

[51] Giot, Romain, El-Abed, Mohamad, and Rosenberger, Christophe. "Keystroke dynamics with low constraints svm based passphrase enrollment". In: *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*. IEEE. 2009, pp. 1–6 (cited on p. 48).

[52] Giot, Romain, Hemery, Baptiste, and Rosenberger, Christophe. "Low cost and usable multimodal biometric system based on keystroke dynamics and 2d face recognition". In: *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE. 2010, pp. 1128–1131 (cited on pp. 11, 32).

[53] Giot, Romain and Rosenberger, Christophe. "A new soft biometric approach for keystroke dynamics based on gender recognition". In: *International Journal of Information Technology and Management* 11.1 (2012), pp. 35–49 (cited on p. 19).

[54] Giuffrida, Cristiano, Majdanik, Kamil, Conti, Mauro, and Bos, Herbert. "I Sensed It Was You: Authenticating Mobile Users with Sensor-Enhanced Keystroke Dynamics". In: *Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2014, pp. 92–111 (cited on p. 32).

[55] Gunetti, Daniele and Picardi, Claudia. "Keystroke Analysis of Free Text". In: *ACM Transactions on Information and System Security* 8.3 (2005), pp. 312–347. ISSN: 1094-9224 (cited on pp. 15, 19, 27, 28, 35, 37, 45, 47, 58, 61, 106, 107).

[56] Gunetti, Daniele, Picardi, Claudia, and Ruffo, Giancarlo. "Dealing with different languages and old profiles in keystroke analysis of free text". In: *AI*IA 2005: Advances in Artificial Intelligence*. Springer, 2005, pp. 347–358 (cited on p. 15).

[57] Gunetti, Daniele, Picardi, Claudia, and Ruffo, Giancarlo. "Keystroke analysis of different languages: A case study". In: *Advances in Intelligent Data Analysis VI*. Springer, 2005, pp. 133–144 (cited on p. 15).

[58] Gunetti, Daniele and Ruffo, Giancarlo. "Intrusion detection through behavioral data". In: *Advances in Intelligent Data Analysis*. Springer, 1999, pp. 383–394 (cited on pp. 30, 35, 47).

[59] Hashiyada, Masaki. *DNA Biometrics*. Tech. rep. Tohoku University Graduate School of Medicine, 2013 (cited on p. 8).

[60] Hempstalk, Kathryn, Frank, Eibe, and Witten, Ian H. "One-class classification by combining density and class probability estimation". In: *Machine Learning and Knowledge Discovery in Databases*. Springer, 2008, pp. 505–519 (cited on pp. 37, 47).

[61] Hill, Robert B. *Apparatus and method for identifying individuals through their retinal vasculature patterns*. US Patent 4,109,237. Aug. 1978 (cited on p. 8).

[62] Hu, Jiankun, Gingrich, Don, and Sentosa, Andy. "A k-nearest neighbor approach for user authentication through biometric keystroke dynamics". In: *Communications, 2008. ICC'08. IEEE International Conference on*. IEEE. 2008, pp. 1556–1560 (cited on pp. 37, 47).

[63] Huopio, Simo. "Biometric Identification". In: *Authorization and Access Control in Open Network Environment* (1998) (cited on pp. 2, 7).

[64] Hussien, Bassam, McLaren, Robert, and Bleha, Saleh Ali. "An application of fuzzy algorithms in a computer access security system". In: *Pattern Recognition Letters* 9.1 (1989), pp. 39–43 (cited on pp. 26, 30).

[65] Hwang, Seong-seob, Cho, Sungzoon, and Park, Sunghoon. "Keystroke dynamics-based authentication for mobile devices". In: *Computers & Security* 28.1 (2009), pp. 85–93 (cited on p. 12).

[66] Jain, Anil K., Bolle, Ruud, and Pankanti, Sharath. *Biometrics: personal identification in networked society*. Vol. 479. Springer Science & Business Media, 2006 (cited on p. 6).

[67] Jain, Anil K., Dass, Sarat C, and Nandakumar, Karthik. "Can soft biometric traits assist user recognition?" In: *Defense and Security*. International Society for Optics and Photonics. 2004, pp. 561–572 (cited on p. 7).

[68] Jain, Anil K., Hong, Lin, and Pankanti, Sharath. "Biometric identification". In: *Communications of the ACM* 43.2 (2000), pp. 90–98 (cited on p. 8).

[69] Jain, Anil K., Nandakumar, Karthik, and Ross, Arun. "Score normalization in multimodal biometric systems". In: *Pattern recognition* 38.12 (2005), pp. 2270–2285 (cited on p. 31).

[70] Jain, Anil K., Ross, Arun, and Prabhakar, Salil. "An introduction to biometric recognition". In: *Circuits and Systems for Video Technology, IEEE Transactions on* 14.1 (2004), pp. 4–20 (cited on pp. 6, 7, 9, 18).

[71] Janakiraman, Rajkumar and Sim, Terence. "Keystroke dynamics in a general setting". In: *Advances in Biometrics*. Springer, 2007, pp. 584–593 (cited on pp. 25, 36, 47).

[72] Johansen, Uno Andre. "Keystroke dynamics on a device with touch screen". MA thesis. Gjøvik University College, 2012 (cited on p. 12).

[73] Joyce, Rick and Gupta, Gopal. "Identity authentication based on keystroke latencies". In: *Communications of the ACM* 33.2 (1990), pp. 168–176 (cited on pp. 15, 26, 40, 48).

[74] Kang, Pilsung and Cho, Sungzoon. "Keystroke dynamics-based user authentication using long and free text strings from various input devices". In: *Information Sciences* 308 (2015), pp. 72–93 (cited on p. 39).

[75] Kang, Pilsung, Hwang, Seong-seob, and Cho, Sungzoon. "Continual retraining of keystroke dynamics based authenticator". In: *Advances in Biometrics*. Springer, 2007, pp. 1203–1211 (cited on pp. 14, 25, 41, 48).

[76] Karnan, Marcus and Akila, M. "Identity authentication based on keystroke dynamics using genetic algorithm and particle swarm optimization". In: *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*. IEEE. 2009, pp. 203–207 (cited on p. 30).

[77] Karnan, Marcus and Akila, M. "Personal authentication based on keystroke dynamics using soft computing techniques". In: *Communication Software and Networks, 2010. ICCSN'10. Second International Conference on*. IEEE. 2010, pp. 334–338 (cited on p. 30).

[78] Karnan, Marcus, Akila, M., and Krishnaraj, N. "Biometric personal authentication using keystroke dynamics: A review". In: *Applied Soft Computing* 11.2 (2011), pp. 1565–1573 (cited on pp. 7, 48).

[79] Khanna, Preeti and Sasikumar, M. "Recognising emotions from keyboard stroke pattern". In: *International journal of Computer Applications* 11.9 (2010), pp. 1–5 (cited on pp. 14, 43).

[80] Kołakowska, Agata et al. "Automatic recognition of males and females among web browser users based on behavioural patterns of peripherals usage". In: *Internet Research* 26.5 (2016) (cited on p. 19).

[81] Lee, Jae-Wook, Choi, Sung-Soon, and Moon, Byung-Ro. "An evolutionary keystroke authentication based on ellipsoidal hypothesis space". In: *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. ACM. 2007, pp. 2090–2097 (cited on pp. 14, 25).

[82] Lee, Wei-Han and Lee, Ruby B. "Multi-sensor authentication to improve smartphone security". In: *Conference on Information Systems Security and Privacy*. 2015 (cited on pp. 42, 48).

[83] Leggett, John and Williams, Glen. "Verifying identity via keystroke characteristics". In: *International Journal of Man-Machine Studies* 28.1 (1988), pp. 67–76 (cited on pp. 15, 26).

[84] Leggett, John, Williams, Glen, Usnick, Mark, and Longnecker, Mike. "Dynamic identity verification via keystroke characteristics". In: *International Journal of Man-Machine Studies* 35.6 (1991), pp. 859–870 (cited on p. 3).

[85] Li, Yilin et al. "Study on the BeiHang keystroke dynamics database". In: *Biometrics (IJCB), 2011 International Joint Conference on*. IEEE. 2011, pp. 1–5 (cited on pp. 41, 45, 48).

[86] Loy, Chen Change, Lai, Weng Kin, and Lim, Chee Peng. "Development of a pressure-based typing biometrics user authentication system". In: *ASEAN Virtual Instrumentation Applications Contest Submission* (2005) (cited on pp. 13, 41, 48).

[87] Magalhães, Sérgio Tenreiro de, Revett, Kenneth, and Santos, Henrique M. D. *Keystroke dynamics: stepping forward in authentication*. Tech. rep. Universidade do Minho, 2006 (cited on p. 16).

[88] Maiorana, Emanuele, Campisi, Patrizio, González Carballo, Noelia, and Neri, Alessandro. "Keystroke dynamics authentication for mobile phones". In: *Proceedings of the 2011 ACM Symposium on Applied Computing*. ACM. 2011, pp. 21–26 (cited on p. 12).

[89] Maisuria, Leenesh Kumar, Ong, Cheng Soon, and Lai, Weng Kin. "A comparison of artificial neural networks and cluster analysis for typing biometrics authentication". In: *Neural Networks, 1999. IJCNN'99. International Joint Conference on*. Vol. 5. IEEE. 1999, pp. 3295–3299 (cited on p. 26).

[90] Marsters, John-David. "Keystroke dynamics as a biometric". PhD thesis. University of Southampton, 2009 (cited on p. 18).

[91] Matsubara, Yoshitomo, Samura, Toshiharu, and Nishimura, Haruhiko. "Keyboard Dependency of Personal Identification Performance by Keystroke Dynamics in Free Text Typing". In: *Journal of Information Security* 6.03 (2015), p. 229 (cited on pp. 39, 47).

[92] Messerman, Arik, Mustafic, Tarik, Camtepe, Seyit Ahmet, and Albayrak, Sahin. "Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics". In: *Biometrics (IJCB), 2011 International Joint Conference on*. IEEE. 2011, pp. 1–8 (cited on pp. 15, 37, 47, 68, 107).

[93] Monrose, Fabian, Reiter, Michael K., and Wetzel, Susanne. "Password hardening based on keystroke dynamics". In: *International Journal of Information Security* 1.2 (2002), pp. 69–83 (cited on pp. 33, 43).

[94] Monrose, Fabian and Rubin, Aviel D. "Authentication via keystroke dynamics". In: *Proceedings of the 4th ACM conference on Computer and communications security*. ACM. 1997, pp. 48–56 (cited on pp. 35, 47, 57).

[95] Monrose, Fabian and Rubin, Aviel D. "Keystroke dynamics as a biometric for authentication". In: *Future Generation computer systems* 16.4 (2000), pp. 351–359 (cited on pp. 2, 33).

[96] Montalvão Filho, Jugurta R. and Freire, Eduardo O. "On the equalization of keystroke timing histograms". In: *Pattern Recognition Letters* 27.13 (2006), pp. 1440–1446 (cited on p. 47).

[97] Morales, Aythami, Fiérrez Aguilar, Julian, Vera-Rodriguez, Ruben, and Ortega-Garcia, Javier. "Autenticación Web de Estudiantes Mediante Reconocimiento Biométrico". In: *III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad*. 2016 (cited on pp. 39, 47).

[98] Morales, Aythami et al. "KBOC: Keystroke biometrics ongoing competition". In: *Biometrics: Theory, Applications, and Systems (BTAS), 2016 IEEE 8th International Conference on*. IEEE. 2016, pp. 1–6 (cited on p. 45).

[99] Napier, Renee et al. "Keyboard user verification: toward an accurate, efficient, and ecologically valid algorithm". In: *International Journal of Human-Computer Studies* 43.2 (1995), pp. 213–222 (cited on p. 26).

[100] Nonaka, Hidetoshi and Kurihara, Masahito. "Sensing Pressure for Authentication System Using Keystroke Dynamics". In: *International Journal of Computer, Control, Quantum and Information Engineering*. Citeseer. 2004 (cited on pp. 13, 40, 48).

[101] Obaidat, Mohammad S. and Macchiarolo, David T. "An online neural network system for computer access security". In: *IEEE Transactions on Industrial Electronics* 40.2 (1993), pp. 235–242 (cited on p. 40).

[102] Obaidat, Mohammad S. and Sadoun, Balqies. "Verification of computer users using keystroke dynamics". In: *IEEE Transactions on Systems, Man and Cybernetics* 27.2 (1997), pp. 261–269 (cited on pp. 40, 48).

[103] Pavithra, M and Sathya, KB Sri. "Continuous User Authentication Using Keystroke Dynamics". In: *International Journal of Computer Science and Information Technologies* (2016) (cited on p. 11).

[104] Pedernera, Gissel Zamonsky et al. "Revisiting clustering methods to their application on keystroke dynamics for intruder classification". In: *IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications (BIOMS)*. IEEE. 2010, pp. 36–40 (cited on p. 26).

[105] Pereyda, Joshua Thomas. "Keystroke Timing Attacks in a Free-Text Environment". PhD thesis. University of Idaho, 2014 (cited on pp. 13, 43).

[106] Rahman, Khandaker A., Balagani, Kiran S., and Phoha, Vir V. "Making impostor pass rates meaningless: A case of snoop-forge-replay attack on continuous cyber-behavioral verification with keystrokes". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*. IEEE. 2011, pp. 31–38 (cited on pp. 13, 38, 41, 47).

[107] Rahman, Khandaker A., Balagani, Kiran S., and Phoha, Vir V. "Snoop-forge-replay attacks on continuous verification with keystrokes". In: *IEEE Transactions on Information Forensics and Security* 8.3 (2013), pp. 528–541 (cited on pp. 13, 43).

[108] Ramzi, Saifan, Salem, Asma, Zaidan, Dema, and Swidan, Andraws. "A Survey of behavioral authentication using keystroke dynamics: Touch screens and mobile devices". In: *Journal of Social Sciences (COES & RJ-JSS)* 5.1 (2016), pp. 29–41 (cited on p. 42).

[109] Revett, Kenneth and Khan, Aurangzeb. "Enhancing login security using keystroke hardening and keyboard gridding". In: *Virtual Multi Conference on Computer Science and Information Systems* (2005) (cited on pp. 16, 43).

[110] Roli, Fabio, Didaci, Luca, and Marcialis, Gian Luca. "Adaptive biometric systems that can improve with use". In: *Advances in Biometrics*. Springer, 2008, pp. 447–471 (cited on pp. 14, 25).

[111] Roth, Joseph, Liu, Xiaoming, and Metaxas, Dimitris. "On continuous user authentication via typing behavior". In: *Image Processing, IEEE Transactions on* 23.10 (2014), pp. 4611–4624 (cited on p. 11).

[112] Roy, Soumen, Roy, Utpal, and Sinha, DD. "Performance Evaluation of Various Distance-based Data-Mining Classifiers on Typing Patterns for User Authentication/Identification". In: *International Journal of Innovative Research and Development* 5.2 (2016) (cited on pp. 29, 57, 192).

[113] Saini, Baljit Singh, Kaur, Navdeep, and Bhatia, Kamaljit Singh. "Keystroke Dynamics for Mobile Phones: A Survey". In: *Indian Journal of Science and Technology* 9.6 (2016) (cited on p. 42).

[114] Samura, Toshiharu and Nishimura, Haruhiko. "Keystroke timing analysis for individual identification in Japanese free text typing". In: *ICCAS-SICE, 2009*. IEEE. 2009, pp. 3166–3170 (cited on pp. 37, 47).

[115] Sheng, Yong, Phoha, Vir V., and Rovnyak, Steven M. "A parallel decision tree-based method for user authentication based on keystroke patterns". In: *IEEE Transactions on Systems, Man and Cybernetics* 35.4 (2005), pp. 826–833 (cited on pp. 41, 48).

[116] Shimshon, Tomer, Moskovitch, Robert, Rokach, Lior, and Elovici, Yuval. "Clustering di-graphs for continuously verifying users according to their typing patterns". In: *Electrical and Electronics Engineers in Israel (IEEEI), 2010 IEEE 26th Convention of*. IEEE. 2010, pp. 000445–000449 (cited on p. 15).

[117] Sim, Terence and Janakiraman, Rajkumar. "Are digraphs good for free-text keystroke dynamics?" In: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE. 2007, pp. 1–6 (cited on pp. 15, 36).

[118] Sim, Terence, Zhang, Sheng, Janakiraman, Rajkumar, and Kumar, Sandeep. "Continuous verification using multimodal biometrics". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.4 (2007), pp. 687–700 (cited on p. 17).

[119] Song, Dawn Xiaodong, Wagner, David, and Tian, Xuqing. "Timing Analysis of Keystrokes and Timing Attacks on SSH". In: *USENIX Security Symposium*. Vol. 2001. 2001 (cited on pp. 13, 43).

[120] Srivastava, Prakash Chandra et al. "Fingerprints, Iris and DNA Features based Multimodal Systems: A Review". In: *International Journal of Information Technology and Computer Science (IJITCS)* 5.2 (2013), p. 88 (cited on pp. 8, 11).

[121] Stewart, John C., Monaco, John V., Cha, Sung-Hyuk, and Tappert, Charles C. "An investigation of keystroke and stylometry traits for authenticating online test takers". In: *Biometrics (IJCB), 2011 International Joint Conference on*. IEEE. 2011, pp. 1–7 (cited on pp. 43, 47).

[122] Teh, Pin Shen, Teoh, Andrew Beng Jin, and Yue, Shigang. "A survey of keystroke dynamics biometrics". In: *The Scientific World Journal* 2013 (2013) (cited on pp. 14, 15, 19, 26, 48).

[123] Teh, Pin Shen, Yue, Shigang, and Teoh, Andrew Beng Jin. "Feature fusion approach on keystroke dynamics efficiency enhancement". In: *International Journal of Cyber-Security and Digital Forensics* 1.1 (2012), pp. 20–31 (cited on p. 31).

[124] Teh, Pin Shen, Teoh, Andrew Beng Jin, Tee, Connie, and Ong, Thian Song. "A multiple layer fusion approach on keystroke dynamics". In: *Pattern Analysis and Applications* 14.1 (2011), pp. 23–36 (cited on p. 31).

[125] Teh, Pin Shen, Zhang, Ning, Teoh, Andrew Beng Jin, and Chen, Ke. "A survey on touch dynamics authentication in mobile devices". In: *Computers & Security* 59 (2016), pp. 210–235 (cited on p. 42).

[126] Teh, Pin Shen, Teoh, Andrew Beng Jin, Tee, Connie, and Ong, Thian Song. "Keystroke dynamics in password authentication enhancement". In: *Expert Systems with Applications* 37.12 (2010), pp. 8618–8627 (cited on pp. 11, 31).

[127] Teh, Pin Shen, Teoh, Andrew Beng Jin, Ong, Thian Song, and Neo, Han Foon. "Statistical fusion approach on keystroke dynamics". In: *Signal-Image Technologies and Internet-Based System, 2007. SITIS'07. Third International IEEE Conference on*. IEEE. 2007, pp. 918–923 (cited on p. 32).

[128] Teh, Pin Shen et al. "TDAS: A Touch Dynamics based Multi-Factor Authentication Solution for Mobile Devices". In: *International Journal of Pervasive Computing and Communications* 12.1 (2016) (cited on p. 42).

[129] Tey, Chee Meng, Gupta, Payas, and Gao, Debin. "I can be you: Questioning the use of keystroke dynamics as biometrics". In: *The 20th Annual Network & Distributed System Security Symposium*. 2013 (cited on pp. 41, 43, 48).

[130] Tran, Dat, Ma, Wanli, Chetty, Girija, and Sharma, Dharmendra. "Fuzzy and markov models for keystroke biometrics authentication". In: *Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization*. Citeseer. 2007, pp. 89–94 (cited on p. 26).

[131] Umphress, David and Williams, Glen. "Identity verification through keyboard characteristics". In: *International Journal of Man-Machine Studies* 23.3 (1985), pp. 263–273 (cited on pp. 15, 26, 40, 48).

[132] Venugopalan, Shreyas, Juefei-Xu, Felix, Cowley, Benjamin, and Savvides, Marios. "Electromyograph and Keystroke Dynamics for Spoof-Resistant Biometric Authentication". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2015, pp. 109–118 (cited on pp. 42, 48).

[133] Villani, Mary et al. "Keystroke biometric recognition studies on long-text input under ideal and application-oriented conditions". In: *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*. IEEE. 2006, pp. 39–39 (cited on pp. 16, 36, 37, 47, 60, 61).

[134] Wang, Yang, Du, Guang-Yu, and Sun, Fu-Xiong. "A model for user authentication based on manner of keystroke and principal component analysis". In: *Machine Learning and Cybernetics, 2006 International Conference on*. IEEE. 2006, pp. 2788–2792 (cited on p. 23).

[135] Wayman, James, Jain, Anil K., Maltoni, Davide, and Maio, Dario. "An introduction to biometric authentication systems". In: *Biometric Systems*. Springer, 2005, pp. 1–20 (cited on p. 21).

[136] Yampolskiy, Roman V. and Govindaraju, Venu. "Behavioural biometrics: a survey and classification". In: *International Journal of Biometrics* 1.1 (2008), pp. 81–113 (cited on p. 13).

[137] Yu, Enzhe and Cho, Sungzoon. "Novelty detection approach for keystroke dynamics identity verification". In: *Intelligent Data Engineering and Automated Learning*. Springer, 2003, pp. 1016–1023 (cited on pp. 40, 48).

[138] Zahid, Saira, Shahzad, Muhammad, Khayam, Syed Ali, and Farooq, Muddassar. "Keystroke-based user identification on smart phones". In: *Recent Advances in Intrusion Detection*. Springer. 2009, pp. 224–243 (cited on p. 30).

[139] Zhao, Ying. "Learning user keystroke patterns for authentication". In: *Proceedings of World Academy of Science, Engineering and Technology*. Vol. 14. Citeseer. 2006, pp. 65–70 (cited on p. 30).

[140] Zhong, Yu, Deng, Yunbin, and Jain, Anil K. "Keystroke dynamics for user authentication". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE. 2012, pp. 117–123 (cited on p. 12).

A | Attended courses and Milestones

While following the full-time dedication itinerary of the Doctoral Program at the University of Andorra, the following courses have been attended:

- Metodologia de recerca (*Research methodology*)
- Tecnologies de la informació i de la comunicació (*Information and communication technologies*)
- Ètica en la recerca (*Research ethics*)
- Propietat intel·lectual, patents i altra legislació aplicable a la recerca (*Intellectual property, patents and other legislation applicable to research*)
- Tècniques quantitatives i qualitatives de valoració (*Quantitative and qualitative research methods*)
- Comunicació científica I (*Scientific communication I*)
- Anglès especialitzat IV (*English applied to research IV*)

At the same time, the following milestones have been achieved:

- Research Project: Presented on June 2015 and defended on July 2015
- Progress Report: Presented on June 2016 and defended on October 2016

B | Contributions

The following contributions have been submitted, accepted and presented at the following congresses:

Dorca Josa, Aleix, Morán Moreno, Jose Antonio and Santamaría Pérez, Eugènia. "Identificación de usuarios mediante dinámica de tecleo en entornos de entrada libre usando información de contexto." In: *XXXI Simposium Nacional de la Unión Científica Internacional de Radio.* 2016

Accepted June 3rd, 2016 and presented September 7th, 2016, in Madrid.

Dorca Josa, Aleix, Morán Moreno, Jose Antonio and Santamaría Pérez, Eugènia. "Using Keystroke Dynamics and context features to assess authorship in online learning environments." In: *11th annual International Technology, Education and Development Conference*, 2017

Accepted December 26th, 2016 and presented March 6th, 2017, in Valencia.

Identificación de usuarios mediante dinámica de tecleo en entornos de entrada libre usando información de contexto

Aleix Dorca Josa[(1)], Eugènia Santamaría Pérez[(2)], Jose Antonio Morán Moreno[(2)]
adorca@uda.ad, esantamaria@uoc.edu, jmoranm@uoc.edu
[(1)]Dpto. de Sistemas Informáticos. Universitat d'Andorra.
Pl. Germandat 7, AD600 Sant Julià de Lòria, Andorra
[(2)]Estudios de Informática, Multimedia y Telecomunicación. Universitat Oberta de Catalunya.
Rambla de Poblenou 156, 08018 Barcelona

***Resumen*—User identification using biometric techniques has been a proven method to complement, or substitute, other methods like passwords or tokens when these have not been robust enough. In this article a study is detailed where keystroke dynamics have been used in conjunction with context information of the written words. User samples have been gathered on a free and uncontrolled environment. With this information a tree model has been built that has allowed the search of whole or partial words and the obtaining of distances measures. User identification has been performed on four groups of ten users each. The result of using this technique not only shows that user identification is possible but also that context information is an important feature to take into account.**

I. Introducción

La identificación de usuarios es uno de los objetivos de la biometría. En entornos de aprendizaje virtual a menudo se manifiesta la duda sobre la autoría de los trabajos de los estudiantes. La autenticación en el Campus Virtual no garantizan que los documentos enviados a evaluar los haya elaborado el usuario en cuestión. La dinámica de tecleo puede ayudar a paliar este problema.

La dinámica de tecleo se estudia desde finales de los años setenta. El campo de estudio se divide, básicamente, en dos ramas: la *autenticación* y la *verificación continua*. A la vez, el análisis del modo en el que los usuarios teclean también se ha llevado a cabo usando, principalmente, dos métodos de entrada: el *texto fijo*, y el *texto libre*.

La metodología típica consiste en crear un modelo de las características de los usuarios. Contra este modelo se pueden comparar nuevas muestras para obtener, con cierto nivel de error, la validez de las mismas. Los estándares que regulan cómo de robusta debe ser una técnica biométrica son estrictos: valores por encima del 1 % en el número de falsas alarmas no son aceptables. Todavía más estricto es el valor de aceptaciones falsas, que no debería superar el 0.001 % [1].

A diferencia de la mayoría de estudios en este campo, se pretende estudiar y analizar la influencia de características de contexto a la hora de identificar a los usuarios. La gran parte de investigación sobre este tema se centra en parejas u otras combinaciones de letras sin tener en cuenta en qué parte de la palabra estas aparecen. En este artícuo se discute si, por ejemplo, la dinámica del usuario es la misma cuando escribe ES, ESTE, SURESTE o ÁRBOLES. La combinación de letras E-S presente en todas las palabras anteriores, en general, se consideraría un *digraph* y se agruparían todas las ocurrencias en una estructura de datos común en la que no se tendría en cuenta el lugar concreto en el que esta combinación ha aparecido en las palabras. El contexto ha sido una característica poco estudiada si bien ha sido citada recientemente como posible línea de trabajo [2], [3].

El resto de este artículo se estructura de la siguiente manera: la sección II muestra el estado del arte; la sección III explica cómo se han recogido las muestras; la sección IV explica cómo se han tratado las muestras para construir el modelo de los usuarios y cómo se han verificado nuevas muestras; la sección V detalla los resultados obtenidos y sus implicaciones en el campo de la biometría de tecleo; la sección VI discute los resultados y, finalmente, la sección VII propone posibles líneas futuras para seguir con la investigación aquí expuesta.

II. Estado del arte

Este estudio se centra en la rama del *texto libre*. En uno de los primeros artículos que se trabajó con texto libre se obtuvieron resultados del 23 % de acierto [4]. Estos resultados no eran muy prometedores.

Uno de los trabajos más citados es el de los investigadores D. Gunetti y C. Picardi en el que se llevó a cabo un detallado estudio de la posibilidad de identificar y autenticar usuarios usando distancias relativas y absolutas entre dos muestras. El método calculaba el nivel de desorden de dos muestras usando combinaciones de 2, 3 y hasta 4 letras. Los resultados del estudio estaban alrededor del 0.005 % FAR y del 5 % FRR [5]. Uno de los problemas que podía presentar el método propuesto era el coste de calcular los valores de las distancias relativas, algo que otros estudios han intentado paliar usando distintas técnicas.

La influencia del uso de diferentes teclados también ha sido estudiada [6]. Este estudio es de vital importancia para evaluar los resultados obtenidos en el estudio que aquí se presenta. La conclusión fue que la identificación era mucho mejor si el usuario usaba solo un único teclado (99.8 % de acierto). En este estudio no se ha podido garantizar esta característica dada la necesidad de estudiar un entorno real en el que el usuario puede utilizar distintos dispositivos.

Una característica que comparten la mayoría de trabajos es el estudio de *digraphs* (o *trigraphs*) a la hora de construir los modelos. Un estudio pone en duda que esta característica

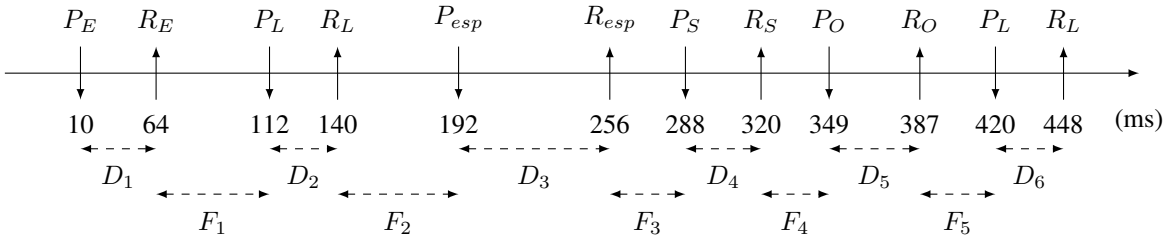

Fig. 1. Análisis de las palabras: *EL SOL*

sea la más adecuada cuando se trata con texto libre ya que no se cree que aporte suficiente información [3]. El estudio concluye que el uso de palabras enteras puede dar iguales o mejores resultados que la contrapartida de usar combinaciones cortas. Este estudio profundiza sobre elementos clave de esta técnica.

La metodología usada en este artículo presenta similitud con la de Messerman et al. [7]. En ese caso se usaron muestras de *n-graphs* para construir el modelo, de nuevo sin tener en cuenta el contexto de dónde aparecían las combinaciones de letras. Un resultado interesante de su estudio muestra que contra cuantos más modelos se comparaba una muestra, más problemas aparecían a la hora de identificarla correctamente. Lo mismo se puede decir sobre el estudio de M. Curtin et. al. Con una metodología similar, también vieron que los resultados eran peores a medida que el número de modelos con los que comparar aumentaba [8]. Las principales diferencias con estos estudios recaen en cómo se construye el modelo y en cómo se escogen las características que definen a los usuarios.

También se puede comparar este estudio con el de Brizan et al. [9]. En el estudio trataron de identificar la demografía de les usuarios analizados con un 82.2 % de acierto cuando las muestras contenían más de 50 palabras. También se adentraron en el estudio de variables de contexto para tratar de establecer la tarea cognitiva que realizaba un usuario.

Una iniciativa reciente interesante en el campo de la dinámica de tecleo es la *Keystroke Biometrics Ongoing Competition* (KBOC)[1], en la que se parte de una base de muestras común y los participantes tratan de obtener el mayor grado de identificación posible. En la misma línea, el articulo [10] presenta los resultados de enfrentar diferentes metodologías para identificar una serie de usuarios cuando solo tecleaban con una sola mano. El ganador usó técnicas de aprendizaje máquina como *Redes neuronales* y *SVM*.

Esta técnica se ha usado también en entornos multimodales en conjunción con, por ejemplo, el reconocimiento mediante vídeo o voz para incrementar el ratio de acierto [11], [12].

III. Recogida de datos

La información relativa a la dinámica de tecleo se ha recogido durante el período de un semestre a partir de los mensajes en los foros de discusión del Campus Virtual de la Universidad de Andorra. En este artículo se hace referencia a cada una de estas entradas como una *sesión*.

Para la recogida de datos se implementó un *snippet* en lenguaje *javascript* y *jQuery* que se añadió al código del entorno virtual *Moodle*. El código enviaba, para cada evento de teclado, un identificador de usuario, el identificador de la sesión, el código de la tecla, el tipo de evento (*keyup* o *keydown*) la marca de tiempo en milisegundos y otros metadatos del cliente. El primer tratamiento fue descartar toda aquella información que no fuera suficiente para ser tratada, como por ejemplo, todos aquellos usuarios que no tenían un número de eventos mínimo, aun cuando hubiesen contribuido con un gran número de sesiones. Al final se tomó en consideración la información recopilada de 40 usuarios, 1502 sesiones y cerca de 500.000 pulsaciones. Para este estudio solo se ha considerado la información generada desde un ordenador de escritorio.

El perfil de los usuarios ha resultado ser muy heterogéneo, característica importante en este ámbito de investigación, incluyendo estudiantes y profesores de todo tipo de estudios. El rango de edad comprende desde los 19 hasta los 65 años.

IV. Metodología

Esta sección detalla cómo se han tratado los datos recogidos para construir el modelo de un usuario y cómo, posteriormente, nuevas sesiones se han verificado contra este modelo.

A. Análisis de intervalos

El análisis de una sesión consiste en tratar los eventos *keydown* y *keyup* (*Press* y *Release*) y calcular el intervalo de tiempo entre eventos sucesivos. Esto permite obtener la información relativa a los intervalos *Press-Release* (PR) y *Release-Press* (RP) para cada tecla pulsada.

La detección de palabras se ha realizado teniendo en cuenta dos características: delimitadores como por ejemplo el *espacio*, la *coma* o el *punto*, o a partir de un intervalo máximo de silencio RP. Se ha establecido este valor en 300 ms empíricamente, pero se podría discutir si se trata de un valor que debería ser establecido para cada usuario.

En la Fig. 1 se muestra un ejemplo de los intervalos comentados y la separación entre las palabras: EL SOL. En este caso la primera palabra EL consiste de los intervalos PR: $D_1 = 54$ y $D_2 = 28$ y del intervalo RP: $F_1 = 48$. Al detectarse un cambio de palabra los intervalos F_2, D_3 y F_3 correspondientes al *espacio* se descartan. La segunda palabra SOL consiste de los intervalos PR: $D_4 = 32$, $D_5 = 38$ y $D_6 = 28$ y de los intervalos RP: $F_4 = 29$ y $F_5 = 33$. Cualquier palabra detectada de N letras tendrá N intervalos PR y $N - 1$ intervalos RP.

B. Creación del modelo de árbol

Los datos se han organizado en un modelo de árbol como el que se muestra en la Fig. 2. En este ejemplo se muestran las palabras: EL SOL ES OCRE EN EL ESTE.

[1]https://sites.google.com/site/btas16kboc/home (6 de junio de 2016)

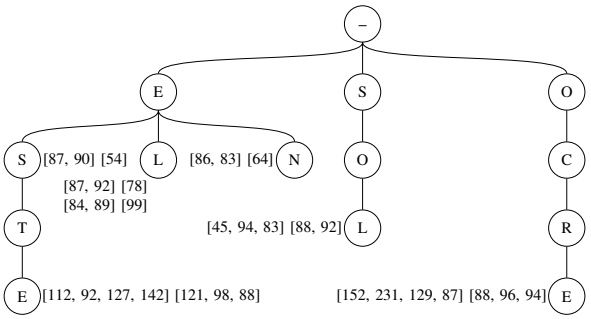

Fig. 2. Modelo de árbol propuesto

Los nodos del árbol *pueden* contener información de intervalos PR y RP (primera y segunda lista en el nodo, respectivamente). Esta información se almacena siempre en el nodo correspondiente a la última letra de la palabra. Si una palabra aparece más de una vez habrá dos listas de intervalos PR y RP, y así sucesivamente (p. ej., EL). Del mismo modo si una palabra ya se encuentra parcialmente en el árbol será normal encontrar una lista de intervalos en nodos intermedios (p. ej., ES y ESTE).

Se puede apreciar que, aunque se pierda información de combinaciones de letras repetidas, se gana la información relativa a la posición de cualquier combinación de letras sin necesidad de disponer de estructuras de datos con *n-graphs*. De ser necesario, esta información se podría calcular sin ninguna dificultad, *a posteriori*, a partir de la información presente en el modelo de árbol.

Después de crear el modelo de árbol se ha procedido a la limpieza del mismo, eliminando todas aquellas muestras que estuviesen fuera de un rango de 3 desviaciones típicas del nodo al que perteneciesen. Todos estos proceso, tanto de creación como de limpieza de muestras, se realizan de forma automática sin intervención humana.

C. Evaluación de una sesión

Una vez construido el modelo de árbol se pueden obtener las distancias de todas las palabras de una nueva sesión al modelo. Cuando se busca una palabra en el modelo se cae en una de estas situaciones:

- La palabra no está en el modelo y se descarta.
- La palabra está completamente en el modelo. Se puede calcular inmediatamente la distancia.
- La palabra buscada se ha encontrado parcialmente pero, en el modelo, el nodo de la última letra de la palabra buscada no contiene información de intervalos. Se puede optar por descartar la palabra buscada o obtener los intervalos a partir de la información de las hojas del árbol a partir del último nodo accedido.
- La palabra buscada se ha encontrado parcialmente pero en el modelo la palabra era más corta. En este caso podemos obtener los intervalos del árbol solo hasta un punto de la palabra buscada.

En este último caso, ¿qué ocurre con la parte de la palabra no tratada? Hay tres opciones según el grado de recursividad que se desee obtener:

- Buscar la subpalabra no tratada en el árbol como si se tratase de una nueva palabra. Si esta no se encuentra, eliminar la primera letra e intentarlo de nuevo hasta agotar todas las letras. Este es el grado de recursividad más exhaustivo. En la Tabla I se indica como R0.
- Volver a buscar la subpalabra no tratada en el árbol como si se tratase de una nueva palabra. Si no se encuentra descartar la subpalabra. En este caso se trata de una recursividad parcial ya que solo se intenta buscar la subpalabra una vez. En la Tabla I se indica como R1.
- Finalmente, simplemente descartar la subpalabra. En este caso no hay recursividad de ningún tipo. En la Tabla I se indica como R2.

La distancia entre las palabras de una nueva sesión y el modelo se ha calculado usando la distancia euclidiana. En la sección de resultados se muestran los cálculos obtenidos usando el intervalo *Release-Press*, que es el que ha dado una mejor tasa de identificación.

Los parámetros que se han estudiado para ver el efecto del contexto han sido los siguientes: la longitud de la palabra y el nivel de recursividad de las subpalabras.

D. Generalización

Para evaluar el método propuesto se dividieron los 40 usuarios en cuatro grupos según el número de eventos. Así, el primer grupo tenía los 10 usuarios que habían entrado más información a lo largo del semestre. Se entendía que, *a priori*, con esta información se podrían generar los modelos más *ricos*. Del mismo modo, el resto de grupos contenían también 10 usuarios, con modelos cada vez menos completos.

El proceso seguido ha sido el propio de un estudio de minería de datos. Se ha creado el modelo de árbol para todos los usuarios. Cada sesión se ha comparado contra todos los modelos del grupo sin que ésta fuese usada en el modelo del usuario propietario. Este proceso se ha repetido para todas las sesiones y para todos los usuarios. Finalmente, se ha encontrado el porcentaje de sesiones que, con distancia mínima al modelo, se han identificado correctamente como pertenecientes al usuario propietario.

V. Resultados

La Tabla I muestra el porcentaje de acierto del método descrito en la sección anterior. El mejor resultado ha sido un 83,99 % de sesiones bien identificadas.

Este resultado se ha obtenido a partir de los modelos del grupo A, una longitud de palabra entre 2 y 5 letras y usando recursividad exhaustiva a la hora de buscar subpalabras en el modelo.

VI. Conclusiones

El porcentaje de acierto está lejos de los estándares que se aplican a las técnicas biométricas [1]. Este hecho es aún más visible cuando los modelos son pobres donde apenas se supera el 50 % de acierto. En general, este problema ha sido una de las constantes en la investigación relativa a la dinámica de tecleo y, a la vez, una de sus mayores críticas. Además, los resultados tienden a ser peores cuando se trabaja en entornos de entrada libre y, peor aún, cuando estos entornos son sin ningún tipo de supervisión. De todos modos la investigación en el campo del biometría de tecleo usando el contexto es aún

TABLA I
PORCENTAJE DE SESIONES IDENTIFICADAS CORRECTAMENTE

Long. palabras		Ilimitada	> 2	$[2-5]$	$[3-7]$
Grupo[1]	Método[2]				
A	R0	78,83	78,18	**83,99**	79,76
	R1	72,14	73,66	80,00	76,60
	R2	76,92	74,35	81,71	76,08
B	R0	79,14	84,27	79,43	82,20
	R1	78,86	84,27	79,71	82,20
	R2	80,52	80,42	79,37	78,93
C	R0	62,35	67,90	74,07	72,84
	R1	61,73	67,28	74,69	72,22
	R2	57,41	58,02	68,52	63,58
D	R0	45,95	50,82	50,27	58,47
	R1	45,41	49,73	50,27	57,38
	R2	40,00	50,55	50,27	55,00

[1] Eventos: A: $\approx 250K$; B: $\approx 125K$; C: $\approx 50K$; D: $\approx 35K$
[2] Recursividad: R0: exhaustiva; R1: parcial; R2: ninguna

incipiente y da pie a profundizar mucho en las características de los usuarios y en su trato para conseguir mejores resultados. En este sentido, algunas propuestas para seguir con el camino aquí propuesto se recogen en la siguiente sección. A menudo se ha planteado combinar la dinámica de tecleo con otras técnicas biométricas para aumentar el rendimiento general de los sistemas de autenticación.

El tamaño del modelo es proporcional a la tasa de acierto. Del mismo modo, se puede apreciar que, dada la gran diferencia entre los grupos A ($\approx 250K$ eventos) y B ($\approx 125K$), la mejora no es siempre presente a partir de un cierto número de eventos. Este dato puede sugerir la existencia de sobreinformación o ruido en el modelo que va en contra de una identificación correcta.

La longitud de la palabra en el modelo es extremadamente importante. Como ya se apuntaba en [3] se verifica que la utilización de combinaciones de letras *muy* cortas no es suficiente. Del mismo modo, el uso de palabras largas tampoco ayuda en el proceso de identificación. Se aprecia que una media de longitud de palabra entre 2 y 7 letras (según el caso) da los mejores resultados.

La recursividad en el momento de buscar subpalabras indica que lo mejor es utilizar toda la información disponible hasta agotar las letras de una palabra. De todos modos, la diferencia es sutil, sobretodo teniendo en cuenta la diferencia de tamaño en la información que se trata según el nivel escogido. No se debería dejar de estudiar este parámetro en otras circunstancias para seguir evaluando su comportamiento.

En el momento de calcular los resultados expuestos, se ha observado que existen un gran número de sesiones con poca información (de una a diez palabras) que, a menudo, se identifican mal. Esto da que pensar en la necesidad de un trato especial para aquellas sesiones que no aporten suficiente información de calidad. Se estudiará si el nivel de recursividad y la calidad de una sesión están relacionados.

En este estudio cada uno de los grupos era de 10 usuarios. Quedaría por verificar (como apuntaban Messerman et al. en su estudio [7]) que si la comparación se hace con grupos con más o menos usuarios se consiguen mejores o peores resultados, respectivamente.

VII. TRABAJO FUTURO

De las conclusiones expuestas en la sección anterior se pueden intuir toda una serie de líneas para su posterior investigación. Se detallan a continuación algunos de los posibles caminos para seguir con la investigación aquí propuesta:

- Analizar a partir de qué momento tenemos demasiada información en el modelo y cómo adaptarla, o tratarla, para que aporte resultados óptimos.
- Seguir buscando características de contexto que incrementen la efectividad de la identificación.
- Evaluar si la recursividad es un parámetro que, en otras condiciones, mejora el resultado global.
- Visto que no todas las longitudes de palabra son igual de influyentes en el resultado, se podría investigar a fondo cual es el valor ideal. Esto puede implicar descartar información para obtener modelos más concretos y ricos.
- Verificar si, incrementando el número mínimo de palabras encontradas en el modelo, se consiguen mejores resultados.
- Se podría evaluar si los parámetros estudiados son válidos para todos los usuarios por igual o, por contra, aplicar diferentes parámetros a diferentes usuarios incrementa los resultados globales.

REFERENCIAS

[1] CENELEC, *European Standard EN 50133-1: Alarm systems. Access control systems for use in security applications. Part 1: System requirements*, 2002.

[2] P. Bours, "Continuous keystroke dynamics: A different perspective towards biometric evaluation," *Information Security Technical Report*, vol. 17, no. 1, pp. 36–43, 2012.

[3] T. Sim and R. Janakiraman, "Are digraphs good for free-text keystroke dynamics?," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–6, IEEE, 2007.

[4] F. Monrose and A. D. Rubin, "Authentication via keystroke dynamics," in *Proceedings of the 4th ACM conference on Computer and communications security*, pp. 48–56, ACM, 1997.

[5] D. Gunetti and C. Picardi, "Keystroke analysis of free text," *ACM Transactions on Information and System Security*, vol. 8, no. 3, pp. 312–347, 2005.

[6] M. Villani, C. C. Tappert, G. Ngo, J. Simone, H. S. Fort, and S.-H. Cha, "Keystroke biometric recognition studies on long-text input under ideal and application-oriented conditions," in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pp. 39–39, IEEE, 2006.

[7] A. Messerman, T. Mustafic, S. A. Camtepe, and S. Albayrak, "Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics," in *Biometrics (IJCB), 2011 International Joint Conference on*, pp. 1–8, IEEE, 2011.

[8] M. Curtin, C. C. Tappert, M. Villani, G. Ngo, J. Simone, H. S. Fort, and S.-H. Cha, "Keystroke biometric recognition on long-text input: A feasibility study," *Proc. Int. MultiConf. Engineers & Computer Scientists (IMECS)*, 2006.

[9] D. G. Brizan, A. Goodkind, P. Koch, K. Balagani, V. V. Phoha, and A. Rosenberg, "Utilizing linguistically-enhanced keystroke dynamics to predict typist cognition and demographics," *International Journal of Human-Computer Studies*, 2015.

[10] J. V. Monaco, G. Perez, C. C. Tappert, P. Bours, S. Mondal, S. Rajkumar, A. Morales, J. Fierrez, and J. Ortega-Garcia, "One-handed keystroke biometric identification competition," in *Biometrics (ICB), 2015 International Conference on*, pp. 58–64, IEEE, 2015.

[11] R. Giot, B. Hemery, and C. Rosenberger, "Low cost and usable multimodal biometric system based on keystroke dynamics and 2d face recognition," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp. 1128–1131, IEEE, 2010.

[12] J. Roth, X. Liu, and D. Metaxas, "On continuous user authentication via typing behavior," *Image Processing, IEEE Transactions on*, vol. 23, no. 10, pp. 4611–4624, 2014.

Using Keystroke Dynamics and context features to assess authorship in online learning environments

Aleix Dorca Josa[1], Jose Antonio Morán Moreno[2], Eugènia Santamaría Pérez[2]

[1]Universitat d'Andorra
[2]Universitat Oberta de Catalunya

Abstract

Using off-the-shelf keyboards and the possibility of measuring the particular rhythm a user has when typing on a computer system has led to the possibility of identifying these users. Over the years, obtaining good Authentication, Identification and Verification methods has been the main focus of Keystroke Dynamics.

The objective of the proposed research is to determine how well the identity of users can be established when they use online resources like e-learning environments when context features are evaluated. This research was performed on a real-life environment using a free text methodology. The proposed method focuses on the hypothesis that the position of a particular combination of letters in a word is of high importance. The template of the user is built using the latency between successive keystrokes, and the context of the written words, that is, taking into account where a particular letter stroke has taken place. Other contextual features have also been studied to determine the ones that better help ascertain the identity of a user.

The results of the proposed research should help determine if using Keystroke Dynamics and the proposed method is enough to identify users from the content they create with a good enough level of certainty. From this moment, it could be used as a method to ensure that a user is not supplanted by another, in authentication schemes, or to help determine the authorship of different parts of a document written by more than one user.

Keywords: Keystroke Dynamics, context, free text, assessment, e-learning environments

1 Introduction

Identifying users is one of the main objectives when using biometric techniques. In online learning environments, the doubt whether an assignment has been written by the user who submitted it may sometimes appear. Having the users authenticated in the online platform or even in their desktop environment is no guarantee that the submitted papers were authored by these users. Keystroke Dynamics can be of help in asserting their identity using the time intervals between keystrokes and the context features related to the written words.

Keystroke Dynamics has been studied since the late seventies. This field of study has been divided into different branches, being authentication, identification and verification the most relevant. At the same time, the study of how users type on the keyboard has been carried out using two main approaches: fixed text and free text. A typical fixed text example would be that of a password, something known to the users that they always type in the same manner. Opposed to this is the idea of the free text methodology in which users can type anything they want without restrictions in length or content.

The typical methodology consists in creating a template of the features that best describe a user when typing on a keyboard. Against this model, new samples can be compared to verify their validity, always with a certain level of error. Efforts should be put into minimizing this error. The European standards for control-access systems specifies a false-alarm rate of less than 1%, with a miss rate of no more than 0.001% [1].

This study uses the context data of the written words to identify the users as opposed to other well-known techniques like, for example, n-graph frequency. This method discusses whether the rhythm of a particular user is the same when they type, for example: IS, IRIS, THESIS or DISAPPEAR. The combination of letters I-S would normally be considered a digraph and would be grouped in a common data structure without considering if it had appeared at the beginning, in the middle, or at the end of the

word. This particular feature has not been thoroughly studied before even though it has been proposed as a possible line of work [2, 3].

2 Background

As previously stated, this study focuses on free text. This research field has been far less studied than the fixed text alternative. In one of the very first articles that dealt with free text, the results were not very promising with only a 23% of positive identification [4]. The wide range of different environments (sometimes highly tailored and controlled), user and sample count, classification methods and other factors makes it very difficult to establish a standard to be compared to and even more so when less studied features, like context data, are studied [5]. Some public Keystroke databases have been made available but mostly for fixed text environments [6].

One of the most cited works is that by D. Gunetti and C. Picardi [7]. The authors calculated both Relative and Absolute distances between newly collected samples and previously stored templates, and combined these to obtain their results. These were around 0.005% FAR and 5% FRR. One of the problems with their method was that the resources needed to obtain the degree of disorder of a sample vector could be very demanding. Other studies have tried to deal with this scalability problem [8] or, at the same time, improve their results by slightly modifying their method [9].

The influence of different keyboards is something that has also been studied [10]. The study M. Villani et al. carried out is of high relevance in order to evaluate the results presented in this study. User identification was more precise if the user always used the same keyboard or input device (99.8% identification rate). In this study, users were able not only to submit information using any device but also from any location so results can be affected by this lack of consistency.

Another study criticizes the methodology based on n-graphs suggesting that this data structure does not provide enough information about the way a user types [11]. The study suggests that whole words could give equal or better results than just using short n-graphs. The present study will answer the question whether length matters by analyzing different word lengths.

The methodology used in this paper shares similitudes with the work of Messerman et al. [12] and M. Curtin et. al [13]. They used n-graphs samples to build the models. An interesting result of their research was the fact that if a new sample was compared to an increasing number of models the chances of correctly identifying the user diminished at a speeding rate.

Brizan et al. [14] published a very interesting article. In their study, they tried to identify the demographics of the users studied with 82.2% accuracy when samples were at least 50 words long. This is in consonance with what was found in the research presented in this paper. The authors also studied other features related to context to try to establish the cognitive task a user was performing.

Also, close to the methodology that will be proposed in this study is the work of Morales et al. [15]. They studied 64 students using different distance measurements, digraphs and trigraphs obtained an accuracy over 90% when identifying users in online learning environments. They did not use context features, though.

The research presented in this paper is the continuation of the work started in [16].The most relevant results were that with a small dataset word length was of importance. This is something that will be evaluated again in this paper using a different set of samples (more interesting in terms of size and robustness).

It is worth noting that this biometric technique has been used in multimodal schemes including, but not limited to, face recognition and speech recognition to improve the global identification rate [17, 18].

3 Methodology

3.1 Samples collection

The samples for this study were collected over a period of two semesters (a whole year) from the messages sent to the forums at the Virtual Campus of the University of Andorra. In this paper, each of these messages is referred to as a Session (S).

A snippet of code combining PHP, jQuery, Javascript and AJAX was developed and added to the base code of the Forum module of the *Moodle* Learning Content Management System (LCMS). The time intervals for every pressed key were collected and securely sent to a remote server where they were stored in a database for later analysis. For every key event this was the gathered information: a user and a

session identifier, the key event code, the type of event (either Keydown or Keyup), the timestamp of the moment the event had been recorded and other minor metadata regarding the user's device and location.

A total of 60 users were used for this study. These were selected among the ones that had sent the most number of events to the LCMS. Close to 4.000 sessions were evaluated. It is worth noting that the information was collected only from desktop computers. Unfortunately, there was not enough information to perform the tests with events sent from mobile devices.

The profile of the selected users was highly heterogeneous, a characteristic that has been highly regarded in this kind of studies. Samples from students and faculty alike, from all kinds of studies offered at the University of Andorra, were collected. Their age ranged from 18 to 65.

3.2 Interval analysis

The study of a Session (S) consists in analyzing the different Keydown (KD) and Keyup (KU) events in order to find the time intervals between them. This allows the possibility of finding the information of the Press–Release (also known as *dwell time* or PR) and Release–Press (also known as *fly time* or RP) intervals for every pressed key.

The process of detecting words was done taking two features into account: known delimiters (i.e. the space key, the comma key, the period key...), and a maximum time interval of silence (300 ms).

Figure 1 shows an example of the time intervals for the words: THE SUN. The first word (THE) is formed by the following PR intervals: $D_1 = 54$, $D_2 = 28$ and $D_3 = 18$. The RP intervals are: $F_1 = 25$ and $F_2 = 5$. When a word separator is detected (a space key event in this case) the intervals of that event are discarded (F_3, D_4 and F_4). The second word (SUN) is formed by the following PR intervals: $D_5 = 32$, $D_6 = 38$ and $D_7 = 28$ and of the following RP intervals: $F_5 = 29$ and $F_6 = 33$. From this information other features like Press–Press (PP) or Release–Release (RR) intervals can also be easily obtained.

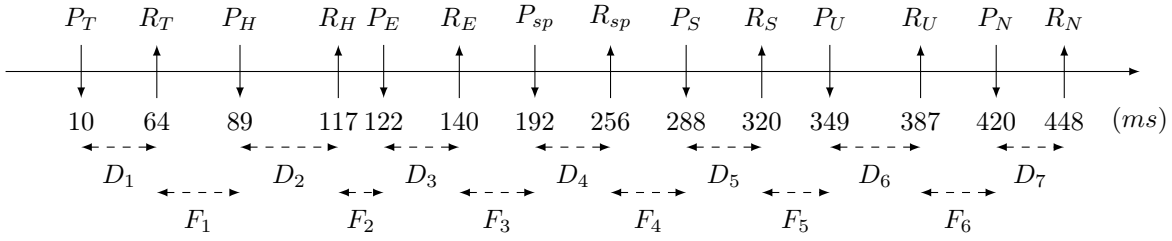

Figure 1: Timing intervals for the words: THE SUN

3.3 The tree model

Detected words were stored in a logical tree structure like the one shown in Figure 2. In this example, the following words have been added to the tree: A, T, W, ALL, ALBERT, THE, THERE, THIS, WORD and WIT. Each of the nodes containing a letter can have PR and RP timing intervals (first and second list, respectively). Single letter words do not have RP values. Only PR intervals can be obtained. Since this research used four features (namely PR, RP, PP, and RR) one letter words were discarded. At the same time these seemed to add little valuable information [16].

In the tree model, a node can have PR and RP timing intervals or not depending on whether the user has ever typed that particular whole word. The timing information is always stored in the node corresponding to the last letter of the word. If a word is detected more than once there will be a different PR and RP list for each instance found (i.e. ALL in the figure). If a word is a sub-word of an already stored word, there will be PR and RP timing information in a non-leaf node (i.e. THE – THERE in the figure).

This tree model stores the information from the beginning to the end of each word. It is thus called a *straight tree* model. This means, for example, that for the word THIS the first node would contain the letter T, its first child node would contain the H, then the I and the leaf node, at depth 4, would finally have the S. The timing intervals would be stored on the S node.

Another model that has been used in this research is an *inverted tree* model. This model is built following the same methodology but from the end to the beginning of words. Using the previous example, for the word THIS the first node would contain the letter S, the first child would be the I, then the H and finally, the leaf node with the timing information, also at depth 4, would contain the letter T. It was

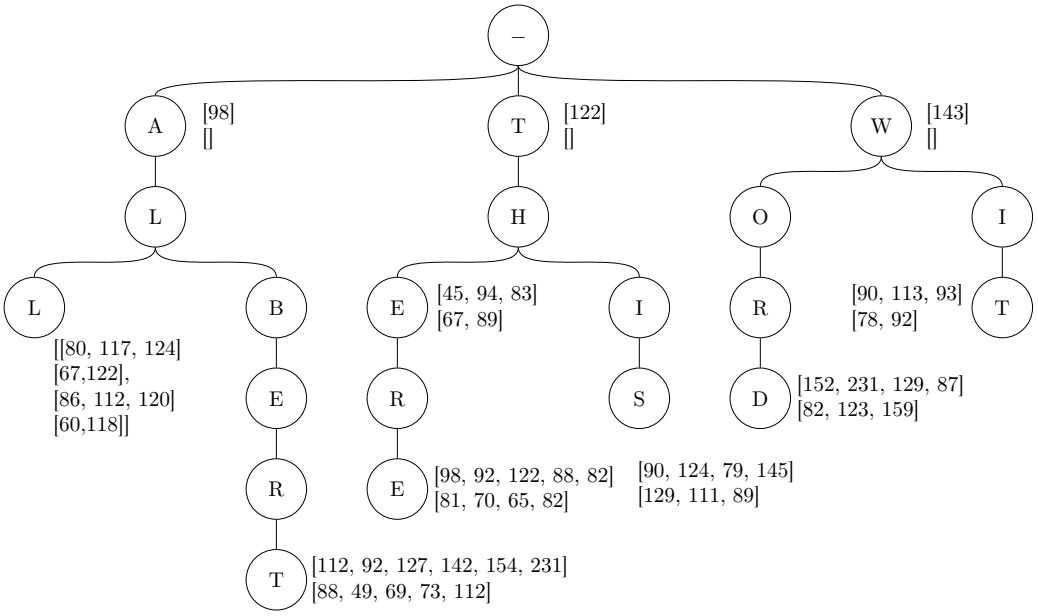

Figure 2: Straight tree model

observed that when comparing new sessions against the *straight tree* model many words would be found only up to a certain depth because the user had previously typed a different word with the same root letters. It would be normal to discard information from the end of these partially found words. The idea of using an inverted tree model was to make sure that most of the context data available would be used. The combined tree model that was used to generate the results used the data from both, *straight* and *inverted*, trees.

It is also worth noting that both tree models were cleaned of word instances outside three standard deviations to avoid having excessive noise, as it had also been done in [16].

3.4 Session evaluation

Once the tree model had been built it was possible to compare new sessions against it and try to establish the author of a given session. The process consisted in searching every word of the new session in the tree model and calculate the distance between the origin word and the word found on the model.

For this study the *Chebyshev* distance measurement was used. Other distance measurements were also evaluated but this was chosen because it was the one that behaved better. To obtain the distance between a word and a model an origin vector and a target vector were needed. The origin vector was the list of interval times from the word being searched and the target vector was the one obtained from the information stored in the tree model. If a word in the tree model had more than one instance the mean vector of all recorded instances would be used.

When searching words in the tree model one of the following situations would be encountered:

- The word was not found in the model. It would simply be discarded.
- The full word was found in the model and the last letter was that of a leaf node. The distance would be immediate to obtain.
- The word was partially found but the node in which the last letter of the origin word was found did not have timing information because this was the first time the user had typed this particular whole word. Partial timings from the leaves from the node of the last letter could be determined and used to find the distance between these partial sub-words.
- The word was partially found in the model but there were still letters from the origin word left to be found. Previously, the user had only entered shorter words with the same root letters. In this case, only the timings of the partial sub-word found would be used. The partial origin sub-word not found in the tree model still contained data, though. How this data was to be used, using recursion, is one of the studied parameters in this research. Three different options have been studied:

– Search the partial sub-word again in the model as if it were a new word. If not found, loose the first letter and repeat the process until all letters have been used or a sub-word is found. This method uses the highest level of recursion and is also the most exhaustive. This method is identified by $R0$.

– Search the partial sub-word again as if it were a new word and discard it if not found. Only partial recursion is used. This method is identified by $R1$.

– Discard the sub-word. No recursion is used. This method is identified by $R2$.

3.5 Studied parameters related to context

The following parameters have been studied in order to see their effect when evaluating context data:

- Length of words: this parameter analyzes whether all word lengths in the tree model are equally relevant. This is of interest, not only in terms of performance and model optimization, but also in order to determine if users have a natural tendency to be more consistent in their typing for a limited number of keystrokes. In this study the following values were tried: unlimited number of letters (≥ 2); greater than 2 (> 2); between 2 and 5 ($[2-5]$); and between 3 and 7 ($[3-7]$). One letter words were discarded.

- Recursion when searching partial sub-words. The effect of using the different types of recursion previously described in Section 3.4 when searching partial sub-words is analyzed with this parameter.

- Number of words found when searching the model. A recurrent problem appeared when the number of words in a session was too low. It could well happen that a user had only accessed the forum to contribute with a few words. Also, having abnormally small models could lead to incorrect identification because the user's template did not have enough information. This parameter tries to mitigate this problem by establishing a minimum number of words either in the session being analyzed or in the model. In this study a threshold value of 50 words found was established based on the results of other studies and on incremental tests performed on the available data.

3.6 Determining the owner of a session

The *Chebyshev* distance between two Vectors $\vec{X}$ and $\vec{Y}$ is defined by the following equation:

$$D_{CH}(\vec{X},\vec{Y}) = max_{i=1}^{n}|X_i - Y_i| \tag{1}$$

Each Session S has W words. Each Word W_i is a vector of values $\vec{X}$. This vector may include a combination of the *dwell times* and/or the *fly times* from the recorded timing intervals depending on the feature F that is analyzed. F can be one of the following: PR (Press–Release), RP (Release–Press), PP (Press–Press), and RR (Release–Release).

The Word W_i searched in the Model M belonging to User U produces another vector $\vec{Y}$. From these two $\vec{X}$ and $\vec{Y}$ vectors the distance D_{CH} can be determined:

$$\forall W_i \in S, D_i(W_i, M_U) = D_{CH}(\vec{X_i}, \vec{Y_i}) \tag{2}$$

From these distances two values are then calculated: the Mean md and the Weighted Mean wmd for all Features. The md and the wmd values make use of the Depth d at which each W_i is found. The Weighted Mean value is obtained using the following weights: all values up to 100 have a weight of 15; values between 100 and 200 have a weight of 5; and values between 200 and 500 have a weight of 1. Values over 500 are discarded. These weights were obtained empirically.

$$\forall F_j \in [PR, RP, PP, RR], md(W_i) = Mean(D_i(W_i, M_U)_{F_j})/d \tag{3}$$

$$\forall F_j \in [PR, RP, PP, RR], wmd(W_i) = WeigthedMean(D_i(W_i, M_U)_{F_j})/d \tag{4}$$

At this point, there is an $md(W_i)$ and a $wmd(W_i)$ value for every Word W_i searched in the model M. The final global distance gd between a Session S and the Model M is composed of four values (gd_m, gd_{med}, gd_{wm}, gd_{wmed}) calculated using the following method:

$$\forall md(W_i) \in S, gd_m = Mean(md(W_i)), gd_{med} = Median(md(W_i)) \quad (5)$$
$$\forall wmd(W_i) \in S, gd_{wm} = Mean(wmd(W_i)), gd_{wmed} = Median(wmd(W_i)) \quad (6)$$

As an example of the proposed method, Table 1 shows a results table after having calculated the *Chebyshev* distance measurement between the words of an origin session and the user's tree model. Five different users are shown in this example (column *Test*). In this example, each user has had four Words compared (*here*, *sun*, *there*, and *moon*) between the origin session and the tree model. The distance values for the four features used are shown (*PP*, *RP*, *PP*, and *RR*). The column *Real* identifies the real owner of the session. The *Depth* column shows the number of letters that were found in the tree model. If the origin word had only been found partially this value would show the depth at which the last letter had been found. Finally, columns *md* and *wmd* show the calculated Mean and Weighted Mean values for each word.

For the first row of user 3207 the Mean value would be: $(69 + 144 + 176 + 99)/4 = 122$. Similarly, the Weighted Mean value would be: $(69 \cdot 15 + 144 \cdot 5 + 176 \cdot 5 + 99 \cdot 15)/40 = 103$. These two values would be then divided by the depth at which the last letter of the word was found: $md = 122/4 = 30.50$ and $wmd = 103/4 = 25.75$.

Word	Feature				Depth	User		*md*	*wmd*
	PR	RP	PP	RR		Test	Real		
here	69	144	176	99	4	3207	192	30.50	25.75
sun	67	19	48	21	3	3207	192	12.92	12.92
there	56	135	145	93	5	3207	192	21.45	18.18
moon	88	33	66	30	4	3207	192	13.56	13.56
here	84	200	163	124	4	37	192	35.69	30.79
sun	71	16	58	74	3	37	192	18.25	18.25
there	72	187	145	110	5	37	192	25.70	21.93
moon	66	25	70	60	4	37	192	13.81	13.81
here	23	11	16	20	4	192	192	4.38	4.38
sun	15	15	14	23	3	192	192	5.58	5.58
there	34	20	13	18	5	192	192	4.25	4.25
moon	20	30	15	28	4	192	192	5.81	5.81
here	71	13	43	59	4	56	192	11.63	11.63
sun	48	31	24	17	3	56	192	10.00	10.00
there	80	22	55	48	5	56	192	10.25	10.25
moon	56	40	40	25	4	56	192	10.06	10.06
here	60	120	155	140	4	78	192	29.69	24.79
sun	30	15	10	45	3	78	192	8.33	8.33
there	52	112	163	132	5	78	192	22.95	18.77
moon	33	5	3	38	4	78	192	4.94	4.94

Table 1: Distances after comparing a session against 5 different models

From each of these *md* and *wmd* values and for each user U the final four values gd_m, gd_{med}, gd_{wm}, gd_{wmed} are then calculated. Table 2 shows this final values for the proposed example. Again, as an example, for user 3207, $gd_m = (30.50 + 12.92 + 21.45 + 13.56)/4 = 19.61$

3.7 Fusion using a voting method

In Table 2, the Votes column shows the total number where each of the *gd* values was a minimum when compared to each other user. It was observed that when evaluating sessions using these four *gd* values, there would be some incorrectly identified sessions but most of the time these errors would not be reported by the four *gd* values at the same time. It was decided to use a fusion method to try to improve the global rate of identification by using a voting scheme. A session would be determined as owned by a particular

User		gd_m	gd_{med}	gd_{wm}	gd_{wmed}	Votes
Test	Real					
3207	192	19.61	17.60	17.51	15.87	0
37	192	23.36	21.20	21.98	20.09	0
192	192	5.01	5.01	4.98	4.98	4
56	192	10.48	10.48	10.16	10.16	0
78	192	16.48	14.21	15.64	13.55	0

Table 2: Final values for the proposed method

user by selecting the one that had the majority of minimum gd values. In the example in Table 2, user 192 obtained 4 votes and thus it is determined as the owner of the session.

3.8 Generalization

Thirty different randomly chosen test sets of 40 users from the available pool of 60 users were used to test the proposed method. The partition of sessions to test and build the models was 30/70%.

The process to evaluate the sessions was that of a typical data mining study. Each session would be compared to all models. This process was repeated for every session of every user. The percentage of correctly identified sessions would be then determined. For the best result the mean FAR and FRR values are also shown as well as the Wilson confidence interval at 95%.

Just as a comparison to a methodology not using context data the experiment that had given the best results in this study was repeated using only trigraphs.

4 Results

The results presented in this section show the effect of the analyzed parameters related to context (length of word, recursion method, and minimum word count found per session). Table 3 also shows the mean value of the percentage of correctly identified sessions when each of the gd values and the Voting system were used.

The best value in Table 3 is a percentage of **98.74%** correctly identified sessions with a Wilson binomial confidence interval, at 95%, of [0.77, 3.52]. With a mean value of 377 sessions compared against the models, the FRR was 0.0126 and the FAR was 0.0002.

This result was obtained using all word lengths. Throughout the table, it can be seen that discarding larger word lengths does not improve the results. On the other hand, if optimization and computer performance is of great concern, the difference in the number of correctly identified sessions when using all word lengths and when only using the $[2-5]$ interval, for example, is marginal.

No doubt the most important parameter is the minimum number of words found in the model. When this is established to 50 words the results improve vastly. As a disadvantage of setting this parameter less sessions are being evaluated.

As per the recursion parameter it is interesting to see that when there is no inferior limit regarding word count, using all available information tends to be somewhat better, at the cost of having to evaluate more than twice the information. On the contrary, when sessions are of better quality and 50 words are mandatory, this behavior is inverted, something that proves the importance of contextual information. It is worth noting that using no recursion improves the performance not only of the correctly identified sessions but also of the computation speed. It seems that having a large number of events is not always the best solution to build a concise and rich model.

As a comparison to previously studied methods the test was repeated against templates built using only trigraphs, without considering context features or recursion methods. The quantity of available information using this method was much higher (up to a double) than the data available for the context and recursion tests. Using this method, though, the effectiveness of the system decreased to an 84%. The proposed method benefits from the fact that having less information but of much better quality greatly improves the results.

Word count		No inferior limit				> 50			
Word length		≥ 2	> 2	$[2-5]$	$[3-7]$	≥ 2	> 2	$[2-5]$	$[3-7]$
Method	Recursion[1]								
gd_m	R0	84.95	84.78	81.88	83.07	96.65	96.56	95.10	95.36
	R1	84.86	84.73	81.78	83.06	96.67	96.49	95.11	95.30
	R2	83.98	83.43	80.26	81.59	97.48	96.60	96.00	95.57
gd_{med}	R0	85.81	84.97	82.79	83.45	97.78	96.95	96.84	96.11
	R1	85.70	84.93	82.73	83.38	97.70	96.90	96.85	96.10
	R2	84.78	83.73	81.31	82.07	98.28	96.93	97.18	96.22
gd_{wm}	R0	87.24	87.35	84.80	86.12	97.75	97.97	96.99	97.18
	R1	87.13	87.21	84.67	86.01	97.74	97.93	96.94	97.13
	R2	86.14	86.17	83.23	84.87	98.10	98.10	97.47	97.42
gd_{wmed}	R0	86.20	85.72	83.17	84.33	97.79	97.33	96.96	96.55
	R1	86.08	85.69	83.08	84.29	97.83	97.30	96.98	96.55
	R2	85.02	84.68	81.87	83.12	98.18	97.71	97.41	96.97
Voting	R0	88.81	88.29	86.61	87.18	98.43	98.32	97.88	97.71
	R1	88.72	88.23	86.53	87.14	98.43	98.29	97.90	97.67
	R2	87.90	87.29	85.22	86.08	**98.74**	98.44	98.18	97.95

[1] $R0$: Exhaustive recursion; $R1$: Partial recursion; $R2$: No recursion

Table 3: Results by features and methods

5 Conclusions

The aim of this study was to find out if using Keystroke Dynamics and context data, as opposed to other well-known techniques, was an effective method when trying to identify users. A new data structure, based on logical trees of words, has been proposed. From the results obtained the following conclusions can be derived:

- The most important outcome is the validity of context data as an identification feature. It has been proved, using a highly hostile and real-life environment, that using only simple statistical techniques offers a very good rate of accuracy, comparable, if not better, to previous studies in similar harsh environments.
- The results obtained when using combined tree models proves that context is a very important feature. This result is highly relevant in order to perform future research based on contextual information.
- The best word length result was to use all available word lengths.
- The best recursion method is not using any recursion but only when sessions and models are of a certain quality. This is of paramount importance and it confirms the importance of the position of the letters and that not all information in a word should be treated equally. It is better to have less information but of better quality than loads of bad information.
- When there is a minimum number of words found in the model, as opposed to accepting any sized session to be compared against the models, the results are far better. This is in concordance with what other studies have also stated.
- The fusion method based on the proposed voting scheme always improves the results when compared to partial gd values. From these, the Weighted Mean and the Median statistic tend to be the ones that perform better.

6 Future work

Some lines of future work can also be put forward here. Below are some ideas to continue with the research line started in this study:

- Study if other factors such as age, gender, time of day of submission... are relevant when it comes to identifying users. Since users from all kinds of ages are available, and other metadata is also available, segmentation could be tried.
- Study other distances measurements and evaluate if there are significant differences when choosing one over another.
- Search for other features, methods, and strategies to increase the percentage of correctly identified sessions without having to sacrifice poor or shorter sessions.
- To improve the performance of the system, and seeing that in most cases choosing a parameter over another gives little improvement on the results, some restrictions could be set when building the tree model. For example: limit the length of words and/or avoid recursion when searching. In this study the optimized tests could be up to 5 times faster taking these considerations into matter.
- It could be analyzed if the studied parameters are valid for all users in the same way or if some users are more susceptible to some parameters.

References

[1] CENELEC. *European Standard EN 50133-1: Alarm systems. Access control systems for use in security applications. Part 1: System requirements.* 2002.

[2] Bours, Patrick. "Continuous keystroke dynamics: A different perspective towards biometric evaluation". In: *Information Security Technical Report* 17.1 (2012), pp. 36–43.

[3] Sim, Terence, Zhang, Sheng, Janakiraman, Rajkumar, and Kumar, Sandeep. "Continuous verification using multimodal biometrics". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29.4 (2007), pp. 687–700.

[4] Monrose, Fabian and Rubin, Aviel D. "Authentication via keystroke dynamics". In: *Proceedings of the 4th ACM conference on Computer and communications security.* ACM. 1997, pp. 48–56.

[5] Alsultan, Arwa and Warwick, Kevin. "Keystroke Dynamics Authentication: A Survey of Free-text Methods". In: *International Journal of Computer Science Issues* 10.4 (2013).

[6] Giot, Romain, Dorizzi, Bernadette, and Rosenberger, Christophe. "A review on the public benchmark databases for static keystroke dynamics". In: *Computers & Security* 55 (2015), pp. 46–61.

[7] Gunetti, Daniele and Picardi, Claudia. "Keystroke Analysis of Free Text". In: *ACM Transactions on Information and System Security* 8.3 (2005), pp. 312–347. ISSN: 1094-9224.

[8] Hu, Jiankun, Gingrich, Don, and Sentosa, Andy. "A k-nearest neighbor approach for user authentication through biometric keystroke dynamics". In: *Communications, 2008. ICC'08. IEEE International Conference on.* IEEE. 2008, pp. 1556–1560.

[9] Davoudi, Homa and Kabir, Ehsanollah. "A new distance measure for free text keystroke authentication". In: *Computer Conference, 2009. CSICC 2009. 14th International CSI.* IEEE. 2009, pp. 570–575.

[10] Villani, Mary et al. "Keystroke biometric recognition studies on long-text input under ideal and application-oriented conditions". In: *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on.* IEEE. 2006, pp. 39–39.

[11] Sim, Terence and Janakiraman, Rajkumar. "Are digraphs good for free-text keystroke dynamics?" In: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on.* IEEE. 2007, pp. 1–6.

[12] Messerman, Arik, Mustafic, Tarik, Camtepe, Seyit Ahmet, and Albayrak, Sahin. "Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics". In: *Biometrics (IJCB), 2011 International Joint Conference on.* IEEE. 2011, pp. 1–8.

[13] Curtin, Mary et al. “Keystroke biometric recognition on long-text input: A feasibility study”. In: *Proc. Int. MultiConf. Engineers & Computer Scientists (IMECS)* (2006).

[14] Brizan, David Guy et al. “Utilizing linguistically-enhanced keystroke dynamics to predict typist cognition and demographics”. In: *International Journal of Human-Computer Studies* (2015).

[15] Morales, Aythami, Fierrez, Julian, Vera-Rodriguez, Ruben, and Ortega-Garcia, Javier. “Autenticación Web de Estudiantes Mediante Reconocimiento Biométrico”. In: *III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad*. 2016.

[16] Dorca Josa, Aleix, Santamaría Pérez, Eugènia, and Morán Moreno, Jose Antonio. “Identificación de usuarios mediante dinámica de tecleo en entornos de entrada libre usando información de contexto”. In: *XXXI Simposium Nacional de la Unión Científica Internacional de Radio (URSI, 2016)*. 2016.

[17] Giot, Romain, Hemery, Baptiste, and Rosenberger, Christophe. “Low cost and usable multimodal biometric system based on keystroke dynamics and 2d face recognition”. In: *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE. 2010, pp. 1128–1131.

[18] Montalvao Filho, Jugurta R. and Freire, Eduardo O. “Multimodal biometric fusion-joint typist (keystroke) and speaker verification”. In: *Telecommunications symposium, 2006 international*. IEEE. 2006, pp. 609–614.

C | Samples collector code

The code for the **PHP**/**Javascript**/**jQuery**/**AJAX** code that collects samples and timing intervals from the user's way of typing is provided below. This code is executed on the client's side. Comments inline should help understand what is being done in each section. The code was stored in the `lib` folder of the *Moodle* application and it was included in the file `moodle/mod/forum/post.php` with the following instruction:

```
1 require_once($CFG->libdir.'/phdrec.php');
```

Listing C.1: Code to include the gatherer into the Forum module

moodle/lib/phdrec.php:

```
1  <?php
2  echo "<script type=\"text/javascript\">";
3  echo "var phd_id = ".$USER->id;
4
5  echo <<<END
6
7  // userAgent check to determine if the device is a mobile device.
8  function isMobile() {
9    // Code omitted because of length.
10   return isMobile;
11 }
12
13 // Get a Random Id for the current session
14 function phd_getRandomId(N) {
15   var S4 = function() { return (((1+Math.random())*0x10000)|0).
       toString(16).substring(1); };
16   return (S4()+S4()+S4()+S4());
17 }
18
19 // Add the function passed as a parameter to the window.onload
     variable without overwriting it.
20 function addLoadEvent(func) {
21   var oldonload = window.onload;
22   if (typeof window.onload != 'function') {
23     window.onload = func;
24   } else {
25     window.onload = function() {
```

```
26        if (oldonload) {
27          oldonload();
28        }
29        func();
30      }
31    }
32  }
33
34  // Variables that store the data that will be sent to the server.
35  var phd_incrementalData = [];
36  var phd_session = phd_getRandomId(16);
37
38  // Creates a Javascript object that will store the relevant
      information of every keyboard event.
39  function phd_theData(usrid, session, event) {
40    if (isMobile()) {
41      var date = new Date();
42      this.timeStamp = date.getTime();
43    } else {
44      this.timeStamp = event.timeStamp;
45    }
46    this.usrid = usrid;
47    this.session = session;
48    this.keyCode = event.keyCode;
49    this.type = event.type;
50    this.altKey = event.altKey;
51    this.ctrlKey = event.ctrlKey;
52    this.metaKey = event.metaKey;
53    this.shiftKey = event.shiftKey;
54  }
55
56  // For every keyboard event a new phd_theData object is created and
      pushed into the variable phd_incrementalData.
57  function record(event) {
58    phd_incrementalData.push(new phd_theData(phd_id, phd_session, event
        ));
59  }
60
61  // Stringify the information stored in the variable
      phd_incrementalData and send it to the remote server. This is done
       using jQuery and an AJAX post method.
62  function sendData() {
63    var phd_partialData = phd_incrementalData.slice();
64    phd_incrementalData.length = 0;
65    if (phd_partialData.length > 0) {
66      require(['jquery'], function($) {
67        $.ajax({
```

```
68          url: "https://remote.server.ad/phd_record.php",
69          type: "POST",
70          data: JSON.stringify(phd_partialData),
71          contentType: "application/json; charset=UTF-8",
72        });
73      });
74    }
75  }
76
77  // Initial configuration:
78  // - Send collected data to the remote server every 5 seconds
79  // - Configure the callback functions for the KeyDown and KeyUp
        events
80  addLoadEvent(function() {
81    setInterval(sendData, 5000);
82    require(['jquery'], function($) {
83      $(document).on('keydown', function(event) { record(event); });
84      $(document).on('keyup', function(event) { record(event); });
85    });
86  });
87  END;
88
89  echo "</script>";
90  ?>
```

Listing C.2: Keystroke collector (client side)

Below is the code that was executed on the server side. This code stored the received samples into the persistent layer, that is, a MySQL database. It was also developed in PHP. This code is, indeed, very simple and little effort was put into the security of it or in the sanitation of the received data. This was not part of the objectives, and it is something that should be improved if ever implemented into a production environment.

phdrec.php:

```
1  <?
2  function connectToDB() {
3    $servername = "localhost";
4    $username = "username";
5    $password = "password";
6    $dbname = "keystrokedb";
7
8    try {
9      $conn = new PDO("mysql:host=$servername;dbname=$dbname",
           $username, $password);
10     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
11   } catch(PDOException $e) {
12     syslog (7, "PhD Exception: Connect error: " . $e->getMessage());
```

```
13    }
14
15    return $conn;
16  }
17
18  function saveToDB($data, $conn) {
19    if ($_SERVER['REMOTE_ADDR']) {
20      $headers = apache_request_headers();
21      $client_ip = $headers["X-Forwarded-For"];
22    }
23
24    if ($client_ip != "") $ip = $client_ip; else $ip = $_SERVER['
        REMOTE_ADDR'];
25
26    try {
27      foreach($data as $row) {
28        $stmt = $conn->prepare("INSERT INTO sessions (session, usrid,
            ip, agent, lang) VALUES (:session, :usrid, :ip, :agent, :
            lang) on duplicate key update session=session");
29        $stmt->bindParam(':session', $row['session']);
30        $stmt->bindParam(':usrid', $row['usrid']);
31        $stmt->bindParam(':ip', $ip);
32        $stmt->bindParam(':agent', $_SERVER['HTTP_USER_AGENT']);
33        $stmt->bindParam(':lang', $_SERVER['HTTP_ACCEPT_LANGUAGE']);
34        $stmt->execute();
35
36        $stmt = $conn->prepare("INSERT INTO ks (session, keycode,
            altkey, ctrlkey, metakey, shiftkey, timestamp, type,
            date_created) VALUES (:session, :keycode, :altkey, :ctrlkey,
             :metakey, :shiftkey, :timestamp, :type, NOW())");
37        $stmt->bindParam(':session', $row['session']);
38        $stmt->bindParam(':keycode', $row['keyCode']);
39        $stmt->bindParam(':altkey', $row['altKey']);
40        $stmt->bindParam(':ctrlkey', $row['ctrlKey']);
41        $stmt->bindParam(':metakey', $row['metaKey']);
42        $stmt->bindParam(':shiftkey', $row['shiftKey']);
43        $stmt->bindParam(':timestamp', $row['timeStamp']);
44        $stmt->bindParam(':type', $row['type']);
45        $stmt->execute();
46      }
47    } catch(PDOException $e) {
48      syslog (7, "PhD Exception: " . $e->getMessage());
49    }
50  }
51
52  function closeDB($conn) {
53    $conn = null;
```

```
54 }
55
56 if($_SERVER['REQUEST_METHOD'] == "OPTIONS") {
57   header("Access-Control-Allow-Origin:*");
58   header("Access-Control-Allow-Headers:Content-Type");
59   header("Access-Control-Allow-Methods: POST");
60 }
61
62 if($_SERVER['REQUEST_METHOD'] == "POST") {
63   header("Content-Type: application/json");
64   header("Access-Control-Allow-Origin:*");
65   header("Access-Control-Allow-Headers:Content-Type");
66
67   $postData = json_decode(file_get_contents('php://input'), true);
68   syslog(7, count($postData)." events detected");
69
70   if (count($postData) > 0) {
71     $db = connectToDB();
72     saveToDB($postData, $db);
73     closeDB($db);
74   }
75 }
76 ?>
```

Listing C.3: Keystroke collector (server side)

D | analyzer.py application options menu

Below is the output of the `analyzer.py` Python application help menu. This includes all modes of operation as well as all the parameters that the program accepts. The output of this application (specified with the `-o` or the `--output` options) should then be fed to an R script to obtain the final classification results.

```
$ ./analyzer.py -h
[analyzer] info: checking parameters
usage: analyzer.py [-h] [-a | -r | -u USER | -s SESSION] [-b USER_B] [-d]
                [-e {0,1,2}] [-f FIRST_USER] [-g] [-m MODE] [-o OUTPUT]
                [-t TOTAL_USERS] [-v] [-p PERIOD] [--delay DELAY] [--dm DM]
                [--dists] [--do_not_clean] [--forests] [--forest_mode {0,1}]
                [--max_word_instances_in_tree MAX_WORD_INSTANCES_IN_TREE]
                [--max_words_in_tree MAX_WORDS_IN_TREE]
                [--max_words_in_model MAX_WORDS_IN_MODEL]
                [--min_depth MIN_DEPTH] [--min_found_words MIN_FOUND_WORDS]
                [--min_gp_graphs MIN_GP_GRAPHS]
                [--max_gp_sessions MAX_GP_SESSIONS] [--min_events MIN_EVENTS]
                [--gp] [--num_graphs NUM_GRAPHS] [--show_trees] [--show_words]
                [--stds STDS] [--gp_graphs GP_GRAPHS] [--use_code8]
                [--use_only_space] [--use_valid_sessions]
                [--discard_child_times] [--gender {H,D}]
                [--age_range AGE_RANGE] [--normalize_code8_histogram]
                [--frequency_scale] [--word_scale] [--ban BAN]
                [--perc_test PERC_TEST] [--seed SEED] [--suspects SUSPECTS]

optional arguments:
  -h, --help            show this help message and exit
  -a, --all             tests for ALL (best) users (configure with
                        --total_users and --first_user)
  -r, --random          tests for RANDOM users (configure with --total_users)
  -u USER, --user USER  tests with this USER
  -s SESSION, --session SESSION
                        tests for this SESSION
  -b USER_B, --user_b USER_B
                        USER_B to compare model to
  -d, --debug           enable debug (LOTS of messages)
  -e {0,1,2}, --exhaust {0,1,2}
                        exhaustive search mode (default: 2): 0 - search to the
                        end of the word, 1 - search sub-words until NOT FOUND
                        status, discard the rest, 2 - search only until first
                        match
  -f FIRST_USER, --first_user FIRST_USER
                        TOTAL_USERS starting at FIRST_USER, based on number of
                        events (default: 0)
  -g, --graphs          execute the graphs tests
  -m MODE, --mode MODE  check MODE (default: 0)
```

```
42  -o OUTPUT, --output OUTPUT
43                        OUTPUT file
44  -t TOTAL_USERS, --total_users TOTAL_USERS
45                        number of TOTAL_USERS to use to build the models
46                        (default: 20)
47  -v, --verbose         enable verbosity
48  -p PERIOD, --period PERIOD
49                        period users to test (0 = ALL, 1 = 1st sem. 2015-2016,
50                        2 = 2nd sem. 2015-2016, 3 = 1st sem. 2016-2017
51                        (default: 0)
52  --delay DELAY         max delay time inter-words (default: 300)
53  --dm DM               distance measurements to obtain (comma separated;
54                        valid values: E (Euclidean), M (Manhattan), K
55                        (Chebyshev), C (Canberra), W (Wordgraph); default:
56                        ALL)
57  --dists               find r dist when working with tree models
58  --do_not_clean        do NOT clean outliers
59  --forests             build the forest model
60  --forest_mode {0,1}   mode to build the forest model (default: 0): 0 - max
61                        words (configure with --max_words_in_tree), 1 - a tree
62                        for each session
63  --max_word_instances_in_tree MAX_WORD_INSTANCES_IN_TREE
64                        maximum number of instances in tree for each word
65                        (default: -1)
66  --max_words_in_tree MAX_WORDS_IN_TREE
67                        maximum number of words in each tree for the forest
68                        model (default: 150)
69  --max_words_in_model MAX_WORDS_IN_MODEL
70                        maximum number of words for the tree model (default:
71                        -1)
72  --min_depth MIN_DEPTH
73                        words found on the model should have this number of
74                        MIN_DEPTH letters (default: 2)
75  --min_found_words MIN_FOUND_WORDS
76                        MIN_FOUND_WORDS in tree to consider a session valid
77                        (default: 50)
78  --min_gp_graphs MIN_GP_GRAPHS
79                        MIN_GP_GRAPHS to consider a session valid (default:
80                        100.0)
81  --max_gp_sessions MAX_GP_SESSIONS
82                        MAX_GP_SESSIONS to compare a session (default: 15)
83  --min_events MIN_EVENTS
84                        MIN EVENTS when choosing a random session (default:
85                        1000)
86  --gp                  use Gunetti-Picardi method
87  --num_graphs NUM_GRAPHS
88                        NUM_GRAPHS to build the graphs model (default: 2)
89  --show_trees          show model trees
90  --show_words          show found words by parser
91  --stds STDS           number of standard deviations STDS to supress an
92                        outlier (default: 3)
93  --gp_graphs GP_GRAPHS
94                        Number of graphs to test GP method (all combinations).
95                        Should be entered as NN,MM,PP (default: '2')
96  --use_code8           use code 8 (backspace) as part of the model for error
97                        studying (default: NO)
98  --use_only_space      use space as the only word delimiter (default: NO)
99  --use_valid_sessions  use only valid sessions to build and test models
100                       (default: NO)
```

```
101   --discard_child_times
102                         do not find the mean of child leafs if no intervals
103                         found in current node (default: NO)
104   --gender {H,D}        Limit users to this gender (only for top users)
105                         (default: None)
106   --age_range AGE_RANGE
107                         Limit users to this age group (only for top users).
108                         Should be entered as LL-HH (default: None)
109   --normalize_code8_histogram
110                         returns the histogram in percentages instead of
111                         absolute values (default: NO)
112   --frequency_scale     modifies distances based on number of occurences
113                         (default: NO)
114   --word_scale          modifies distances based on word sequences (default:
115                         NO)
116   --ban BAN             Ban these users from being tested U1,U2,... (default:
117                         None)
118   --perc_test PERC_TEST
119                         Percentage of testing sessions (default: 10%)
120   --seed SEED           Seed to feed the random sessions chooser (default:
121                         None)
122   --suspects SUSPECTS   Number of usual suspects to compare to (default: 4)
```

A typical execution command with the following options: *leave-one-out* mode (`-a -m 3`); use a group of 10 users (`--total_users 10`); starting with the 20$^{\text{th}}$ user with most events (`--first_user 20`); using the *backspace* as part of the tree model (`--use_code8`); discarding child times when a partial word is found in a non-leaf node (`--discard_child_times`); and storing the results in a CSV file called results.csv (`-o results.csv`) would look like this:

```
1 $ ./analyzer.py -a -m 3 -o results.csv --first_user 20 --total_users 10 --use_code8
      --discard_child_times
```

E | MySQL database schema

All the collected keystrokes were stored in a MySQL relational database. In the initial steps of the research, this database had only one table called *ks*. This TABLE stored all the information regarding collected keystrokes but none regarding user information. It is worth noting that for each event, either a *keydown* or a *keyup*, the information regarding *agent*, *session* data, *ip*, *language*... was stored repeatedly in every row. As soon as the number of recorded events increased it was obvious that this approach was far from optimal.

At this point, two new TABLES were added to the schema, the *sessions* TABLE and the *users* TABLE. The *sessions* TABLE collected, in one place, all the information regarding a single session. This included the information regarding the user's browser and origin information. The *users* TABLE was added the moment it was thought that age and gender features could be of use.

Also, when the number of events had surpassed the million, it was a hard task for the Database Management System (DBMS) to rank the top users that had submitted the most number of events. This had to be done every time a test was performed so it was thought that, since this data did not change for a given date, helper TABLES with ranked users could be useful. The users_top_*date* TABLES were thus also added.

At the same time, a number of VIEWS were also created to be able to have a deeper insight of how data was evolving through time. Some of these include the different browser agents from sessions, the most used devices, the events and sessions count per user...

The following sections describe each of these TABLES in more detail.

E.1 TABLES and VIEWS in the *keystrokes* database

Table E.1 shows the list of all TABLES and VIEWS in the *keystrokes* database. The following sections go into the details of the most relevant elements.

Tables_in_keystrokes	Table_type
agents	VIEW
ks	BASETABLE
latest_100_events	VIEW
sessions	BASETABLE
user_code8_percentage	VIEW
user_code8_total	VIEW
user_devices_top	VIEW
user_event_count	VIEW
user_ip_agent	VIEW
user_latest_100_events	VIEW
user_session_keydown_count	VIEW
user_sessions	VIEW
user_sessions_count	VIEW
user_top_all_date	BASETABLE
users	BASETABLE

Table E.1: MySQL *keystrokes* database tables and views

E.2 TABLES description

For each of the BASETABLE elements in the *keystrokes* database these are the CREATE TABLE commands and the resulting TABLES.

E.2.1 MySQL *ks* TABLE

The *ks* TABLE stores the information related to the keystrokes detected in the user's browser when they type messages in Forum modules of the Virtual Campus at the University of Andorra. The description of each of the fields is normally self-explanatory, but they are described for completeness nonetheless.

Description

- id: Unique id for each of the events. Since all tables are normally recommended having a unique id this was added, but it was completely useless to this study.
- session: A unique session identifier. Used as the relation field with the session id field in the *sessions* TABLE.
- key code: For every detected event, the key code as sent by the user's browser. See Appendix F for a list of the most common ones.
- type: Either *keyup* or *keydown*.

- altkey, ctrlkey, metakey, shiftkey: Flags that helped identify when a combination of keystrokes was being performed. These were set to 1 respectively when a modifier key was pressed.
- timestamp: The moment the keystroke had been recorded by the user's browser. Each browser used a different reference to store this information. This field was used to detect words and build the different models.
- date_created: The moment the event was stored into the database. The *timestamp* field and this field had, by no means, to be identical or even related. This field was used to filter events in time to perform the tests.

CREATE TABLE command

```
CREATE TABLE `ks` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `session` varchar(50) NOT NULL,
  `keycode` varchar(10) NOT NULL,
  `type` varchar(10) NOT NULL,
  `altkey` tinyint(1) NOT NULL DEFAULT '0',
  `ctrlkey` tinyint(1) NOT NULL DEFAULT '0',
  `metakey` tinyint(1) NOT NULL DEFAULT '0',
  `shiftkey` tinyint(1) NOT NULL DEFAULT '0',
  `timestamp` varchar(50) NOT NULL,
  `date_created` datetime NOT NULL,
  PRIMARY KEY (`id`),
  KEY `session` (`session`),
  KEY `type` (`type`),
  KEY `keycode` (`keycode`),
  KEY `keycode-type` (`keycode`,`type`),
  CONSTRAINT `ks_ibfk_1` FOREIGN KEY (`session`)
  REFERENCES `sessions` (`session`)
  ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=7479591 DEFAULT CHARSET=utf8
```

Listing E.1: CREATE TABLE for the *ks* TABLE

Resulting TABLE

Table E.2 show the resulting TABLE structure.

E.2.2 MySQL *sessions* TABLE

The *sessions* TABLE stores the information related to each of the different sessions. Every time a new message was written in the Forum modules a new row would be

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
session	varchar(50)	NO	MUL	NULL	
keycode	varchar(10)	NO	MUL	NULL	
type	varchar(10)	NO	MUL	NULL	
altkey	tinyint(1)	NO		0	
ctrlkey	tinyint(1)	NO		0	
metakey	tinyint(1)	NO		0	
shiftkey	tinyint(1)	NO		0	
timestamp	varchar(50)	NO		NULL	
date_created	datetime	NO		NULL	

Table E.2: MySQL *ks* TABLE

added. Since every keystroke event sent by the user's browser had this information, only the first time it was detected it was added to the *sessions* TABLE. Subsequent repeated information was discarded. This was, by no means, optimal as it used some additional bandwidth to send useless information. Further developments should have this in mind to improve performance and scalability.

Description

- session: Unique id for every different session. Used to perform joins with the *ks* TABLE.
- usrid: A unique id for the user sending the events related to this particular session.
- ip: The ip address from the user's location. This was used to determine if a particular user typed from very different locations.
- agent: A string with information about the user's browser and operating system versions.
- lang: The user's browser default language. It was thought that this field would help when dealing with sessions in different languages. In the end, since almost all sessions were written in Catalan, this field became redundant and useless.

CREATE TABLE command

```
CREATE TABLE `sessions` (
  `session` varchar(50) NOT NULL,
  `usrid` varchar(10) NOT NULL,
```

```
  `ip` varchar(15) NOT NULL,
  `agent` varchar(250) NOT NULL,
  `lang` varchar(50) NOT NULL,
  PRIMARY KEY (`session`),
  KEY `usrid` (`usrid`),
  KEY `agent` (`agent`),
  KEY `ip` (`ip`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

Listing E.2: CREATE TABLE for the *sessions* TABLE

Resulting TABLE

Table E.3 show the resulting TABLE structure.

Field	Type	Null	Key	Default	Extra
session	varchar(50)	NO	PRI	NULL	
usrid	varchar(10)	NO	MUL	NULL	
ip	varchar(15)	NO	MUL	NULL	
agent	varchar(250)	NO	MUL	NULL	
lang	varchar(50)	NO		NULL	

Table E.3: MySQL *sessions* TABLE

E.2.3 MySQL *users* TABLE

The *users* TABLE stores the information related to each of the different users. Initially, this TABLE was not going to be used. Only when age and gender were studied it was though necessary. It is worth noting that the information in this TABLE was not automatically generated from the recorded events. Once all the information from a period was available, the different user ids from the *sessions* TABLE were collected and completed with the information from the users TABLE from the university academic information. The age was established at the end of the recording process. Since the recording process lasted for more than a year this information may not reflect the real age of every user at the moment of each session recorded.

Description

- usrid: A unique id identifying the users. Used as key to perform joins with the *sessions* TABLE.
- username: The university's user username. Useless in this study even if it helped have an idea of which *known* users sent the most number of events.

- age: The user's age at the end of the gathering process.
- gender: The user's gender: Male, Female or Others.

CREATE TABLE command

```
CREATE TABLE `users` (
  `usrid` varchar(10) NOT NULL,
  `username` varchar(50) NOT NULL,
  `age` int(11) DEFAULT NULL,
  `gender` varchar(1) NOT NULL,
  PRIMARY KEY (`usrid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

Listing E.3: CREATE TABLE for the *users* TABLE

Resulting TABLE

Table E.4 show the resulting TABLE structure.

Field	Type	Null	Key	Default	Extra
usrid	varchar(10)	NO	PRI	NULL	
username	varchar(50)	NO		NULL	
age	int(11)	YES		NULL	
gender	varchar(1)	NO		NULL	

Table E.4: MySQL *users* TABLE

E.2.4 Helper TABLES

Additionally, different BASETABLES where created from the *ks*, *sessions* and *users* TABLES to get the top ranked users based on the number of events. These were the generic SQL commands used to create them:

```
CREATE TABLE `user_top_date` (
  `usrid` varchar(10) NOT NULL,
  `username` varchar(50) NOT NULL,
  `age` int(11) DEFAULT NULL,
  `gender` varchar(1) NOT NULL,
  `count` int(11) DEFAULT NULL,
  PRIMARY KEY (`usrid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO user_top_date
SELECT sessions.usrid, users.username, users.age, users.gender,
```

```
COUNT(*) cnt
FROM sessions
INNER JOIN users ON sessions.usrid = users.usrid
INNER JOIN ks ON ks.session = sessions.session
WHERE date_created < DATE
GROUP BY sessions.usrid
```

Listing E.4: Create command to rank users by date and number of events

F | Javascript Key Codes

The following table shows a list of common Key – Key Code values.

Key	Key Code	Key	Key Code	Key	Key Code
backspace	8	e	69	numpad 8	104
tab	9	f	70	numpad 9	105
enter	13	g	71	multiply	106
shift	16	h	72	add	107
ctrl	17	i	73	subtract	109
alt	18	j	74	decimal point	110
pause/break	19	k	75	divide	111
caps lock	20	l	76	f1	112
escape	27	m	77	f2	113
page up	33	n	78	f3	114
page down	34	o	79	f4	115
end	35	p	80	f5	116
home	36	q	81	f6	117
left arrow	37	r	82	f7	118
up arrow	38	s	83	f8	119
right arrow	39	t	84	f9	120
down arrow	40	u	85	f10	121
insert	45	v	86	f11	122
delete	46	w	87	f12	123
0	48	x	88	num lock	144
1	49	y	89	scroll lock	145
2	50	z	90	semi-colon	186
3	51	left window key	91	equal sign	187
4	52	right window key	92	comma	188
5	53	select key	93	dash	189
6	54	numpad 0	96	period	190
7	55	numpad 1	97	forward slash	191
8	56	numpad 2	98	grave accent	192
9	57	numpad 3	99	open bracket	219
a	65	numpad 4	100	back slash	220
b	66	numpad 5	101	close bracket	221
c	67	numpad 6	102	single quote	222
d	68	numpad 7	103		

Table F.1: Common Javascript Key – Key Code values

G | Other references

Listed in this Appendix are all the other references that, in one way or another, have been read, consulted, or studied during the research.

[1] Acevedo, Daniel and Hernández, Glemarys. *Identificación de Usuarios Basado en el Reconocimiento de Patrones de Tecleo*. Tech. rep. Universidad Central de Venezuela, 2000.

[2] Adam, Manuel Rodenes, Chismol, Ramón, and Serna, Martín Darío Arango. "Un enfoque sistemático para realizar la tesis doctoral". In: *Psicothema* 12.Suplemento (2000), pp. 474–478.

[3] Adeoye, Olufemi Sunday. "Evaluating the performance of two-factor authentication solution in the banking sector". In: *International Journal of Computer Science Issues* 9.2 (2012), pp. 457–462.

[4] Aguirre Anaya, Eleazar. "Modelo de autenticación de usuarios por cadencia de tecleo". PhD thesis. Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacan, 2009.

[5] Ahmad, Abd Manan and Abdullah, Nik Nailah. "User authentication via neural network". In: *Artificial Intelligence: Methodology, Systems, and Applications*. Springer, 2000, pp. 310–320.

[6] Ahmed, Ahmed Awad E. and Traore, Issa. "A new biometric technology based on mouse dynamics". In: *Dependable and Secure Computing, IEEE Transactions on* 4.3 (2007), pp. 165–179.

[7] Ahmed, Ahmed Awad E. and Traore, Issa. "Anomaly intrusion detection based on biometrics". In: *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*. IEEE. 2005, pp. 452–453.

[8] Ahmed, Ahmed Awad E. and Traore, Issa. *Detecting Computer Intrusions Using Behavioral Biometrics*. Tech. rep. University of Victoria, 2005.

[9] Ahmed, Ejaz, Clark, Andrew, and Mohay, George. "A novel sliding window based change detection algorithm for asymmetric traffic". In: *Network and Parallel Computing, 2008. NPC 2008. IFIP International Conference on*. IEEE. 2008, pp. 168–175.

[10] Al Solami, Eesa, Boyd, Colin, Ahmed, Irfan, and Nayak, Richi. "User-independent threshold for continuous user authentication with keystroke dynamics". In: *The Seventh International Conference on Internet Monitoring and Protection*. 2012.

[11] Al Solami, Eesa, Boyd, Colin, Clark, Andrew, and Ahmed, Irfan. "User-representative feature selection for keystroke dynamics". In: *Network and System Security (NSS), 2011 5th International Conference on*. IEEE. 2011, pp. 229–233.

[12] Ali, Md Liakat et al. "Keystroke Biometric Authentication on Short Numeric Input on Mobile Devices". In: *Proceedings of Student-Faculty Research Day, CSIS, Pace University* (2016).

[13] Ali, Md Liakat, Monaco, John V., Tappert, Charles C., and Qiu, Meikang. "Keystroke Biometric Systems for User Authentication". In: *Journal of Signal Processing Systems* (2016), pp. 1–16.

[14] Andersen, Aleksander Sveløkken. "Biometric authentication and identification using keystroke dynamics with alert levels". MA thesis. Oslo University College, 2007.

[15] Andersen, Alex and Hagen, Simen. *Using Alert Levels to enhance Keystroke Dynamic Authentication*. Tech. rep. Oslo University College, 2007.

[16] Antal, Margit and Nemes, Lehel. "The MOBIKEY Keystroke Dynamics Password Database: Benchmark Results". In: *Software Engineering Perspectives and Application in Intelligent Systems*. Springer, 2016, pp. 35–46.

[17] Araújo, Lívia C. F. et al. "Typing Biometrics User Authentication based on Fuzzy Logic". In: *IEEE Latin America Transactions* 2.1 (2004), pp. 69–74.

[18] Avramidis, Loukas. "Keystroke Dynamic Authentication as a Service". MA thesis. Harokopio University, 2014.

[19] Azevedo, Gabriel L. F. B. G., Cavalcanti, George D. C., and Carvalho Filho, Edson C. B. "Hybrid solution for the feature selection in personal identification problems through keystroke dynamics". In: *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*. IEEE. 2007, pp. 1947–1952.

[20] Azzini, Antonia and Marrara, Stefania. "Impostor users discovery using a multimodal biometric continuous authentication fuzzy system". In: *Knowledge-Based Intelligent Information and Engineering Systems*. Springer. 2008, pp. 371–378.

[21] Azzini, Antonia, Marrara, Stefania, Sassi, Roberto, and Scotti, Fabio. "A fuzzy approach to multimodal biometric continuous authentication". In: *Fuzzy Optimization and Decision Making* 7.3 (2008), pp. 243–256.

[22] Bakelman, Ned, Monaco, John V., Cha, Sung-Hyuk, and Tappert, Charles C. "Continual keystroke biometric authentication on short bursts of keyboard input". In: *Proceedings of Student-Faculty Research Day, CSIS, Pace University* (2012).

[23] Barghouthi, Hafez. "Keystroke Dynamics. How typing characteristics differ from one application to another". MA thesis. Gjøvik University College, 2009.

[24] Bartlow, Nick. "Username and password verification through keystroke dynamics". PhD thesis. West Virginia University, 2005.

[25] Bartolacci, Gary et al. "Long-Text Keystroke Biometric Applications over the Internet". In: *International Conference on Machine Learning: Models, Technologies & Applications*. Citeseer, 2005, pp. 119–126.

[26] Bartolacci, Mary Curtin et al. "Applying Keystroke Biometrics for User Verification and Identification". In: *Proceedings of the MCSCE*. 2005.

[27] Baynath, Purvashi, Soyjaudah, KM Sunjiv, and Khan, Maleika Heenaye-Mamode. "Improving Security Of Keystroke Dynamics By Increasing The Distance Between Keys". In: *Proceddings of the 3rd World Congress on Computer Applications and Information Systems* (2016).

[28] Bazrafshan, Fazel, Javanbakht, Ahmad, and Mojallali, Hamed. "Keystroke identification with a genetic fuzzy classifier". In: *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*. Vol. 4. IEEE. 2010, p. 136.

[29] Bechtel, Jason, Serpen, Gursel, and Brown, Marcus. "Passphrase authentication based on typing style through an ART 2 neural network". In: *International Journal of Computational Intelligence and Applications* 2.02 (2002), pp. 131–152.

[30] Bergadano, Francesco, Gunetti, Daniele, and Picardi, Claudia. "Identity verification through dynamic keystroke analysis". In: *Intelligent Data Analysis* 7.5 (2003), pp. 469–496.

[31] Bertacchini, Maximiliano, Benitez, Carlos, and Fierens, Pablo I. "User clustering based on keystroke dynamics". In: *XVI Congreso Argentino de Ciencias de la Computación*. 2010.

[32] Bhattacharyya, Debnath, Ranjan, Rahul, Alisherov, Farkhod, and Choi, Minkyu. "Biometric authentication: A review". In: *International Journal of u- and e-Service, Science and Technology* (2009).

[33] Bleha, Saleh Ali and Gillespie, Dave. "Computer user identification using the mean and the median as features". In: *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*. Vol. 5. IEEE. 1998, pp. 4379–4381.

[34] Bleha, Saleh Ali and Obaidat, Mohammad S. "Computer users verification using the perceptron algorithm". In: *IEEE Transactions on Systems, Man and Cybernetics* 23.3 (1993), pp. 900–902.

[35] Bleha, Saleh Ali and Obaidat, Mohammad S. "Dimensionality reduction and feature extraction applications in identifying computer users". In: *IEEE Transactions on Systems, Man and Cybernetics* 21.2 (1991), pp. 452–456.

[36] Boechat, Gláucya C., Ferreira, Jeneffer C., and Carvalho Filho, Edson C. B. "Authentication personal". In: *Intelligent and Advanced Systems, 2007. ICIAS 2007. International Conference on.* IEEE. 2007, pp. 254–256.

[37] Boechat, Gláucya C, Ferreira, Jeneffer C, and Carvalho Filho, Edson C. B. "Using the keystrokes dynamic for systems of personal security". In: *Transactions on engineering, computing and technology* 18.1 (2006), pp. 200–205.

[38] Boyd, Colin, Clark, Andrew, and Khandoker, Asadul Islam. *Continuous biometric authentication: can it be more practical?* Tech. rep. Queensland University of Technology, 2010.

[39] Brown, Marcus E. and Rogers, Samuel Joe. *Method and apparatus for verification of a computer user's identification, based on keystroke characteristics.* US Patent 5,557,686. 1996.

[40] Brown, Marcus E. and Rogers, Samuel Joe. "User identification via keystroke characteristics of typed names using neural networks". In: *International Journal of Man-Machine Studies* 39.6 (1993), pp. 999–1014.

[41] Buchoux, Arnaud and Clarke, Nathan L. "Deployment of keystroke analysis on a smartphone". In: *Australian Information Security Management Conference.* 2008, p. 48.

[42] Buciu, Ioan and Gacsadi, Alexandru. "Biometrics Systems and Technologies: A survey". In: *International Journal of Computers Communications & Control* 11.3 (2016), pp. 315–330.

[43] Capuano, Nicola, Marsella, Marco, Miranda, Sergio, and Salerno, Saverio. *User authentication with neural networks.* Tech. rep. University of Salerno, 1999.

[44] Cavalcanti, George D. C., Pinheiro, Eggo H. F., and Carvalho Filho, Edson C. B. "Um sistema de verificaçao de identidade pessoal através de dinâmica da digitaçao". In: *Congresso da Sociedade Brasileira de Computação* (2005).

[45] Chang, Weide. "Improving hidden Markov models with a similarity histogram for typing pattern biometrics". In: *Information Reuse and Integration, Conf, 2005. IRI-2005 IEEE International Conference on.* IEEE. 2005, pp. 487–493.

[46] Chang, Woojin. "Keystroke biometric system using wavelets". In: *Advances in Biometrics*. Springer, 2005, pp. 647–653.

[47] Chang, Woojin. "Reliable keystroke biometric system based on a small number of keystroke samples". In: *Emerging Trends in Information and Communication Security*. Springer, 2006, pp. 312–320.

[48] Changshui, Zhang and Yanhua, Sun. "AR model for keystroker verification". In: *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*. Vol. 4. IEEE. 2000, pp. 2887–2890.

[49] Chen, Jun et al. "Personalized Keystroke Dynamics for Self-Powered Human–Machine Interfacing". In: *ACS Nano* 9.1 (2015), pp. 105–116.

[50] Cho, Sungzoon and Hwang, Seongseob. "Artificial rhythms and cues for keystroke dynamics based authentication". In: *Advances in Biometrics*. Springer, 2005, pp. 626–632.

[51] Cho, Sungzoon, Han, Chigeun, Han, Dae Hee, and Kim, Hyung-Il. "Web-based keystroke dynamics identity verification using neural network". In: *Journal of organizational computing and electronic commerce* 10.4 (2000), pp. 295–307.

[52] Cho, Tai-Hoon. "Pattern classification methods for keystroke analysis". In: *SICE-ICASE, 2006. International Joint Conference*. IEEE. 2006, pp. 3812–3815.

[53] Choraś, Michał and Mroczkowski, Piotr. "Keystroke dynamics for biometrics identification". In: *Adaptive and Natural Computing Algorithms*. Springer, 2007, pp. 424–431.

[54] Choraś, Michał and Mroczkowski, Piotr. "Recognizing individual typing patterns". In: *Pattern Recognition and Image Analysis*. Springer, 2007, pp. 323–330.

[55] Chudá, Daniela and Durfina, Michal. "Multifactor authentication based on keystroke dynamics". In: *Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing*. ACM. 2009, p. 89.

[56] Coltell, Oscar, Badía, José María, and Torres, Guillermo. "Biometric identification system based on keyboard filtering". In: *Security Technology, 1999. Proceedings. IEEE 33rd Annual 1999 International Carnahan Conference on*. IEEE. 1999, pp. 203–209.

[57] Costa, Carlos Roberto do N. et al. "Autenticação Biométrica via Dinâmica da Digitação em Teclados Numéricos". In: *XXII Simpósio Brasileiro de Telecomunicacões*. 2005.

[58] Crawford, Heather. "Keystroke dynamics: Characteristics and opportunities". In: *Privacy, Security and Trust (PST), 2010 Eighth Annual International Conference on*. IEEE. 2010, pp. 205–212.

[59] Dass, Sarat C, Zhu, Yongfang, and Jain, Anil K. "Validating a biometric authentication system: Sample size requirements". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.12 (2006), pp. 1902–1319.

[60] Davoudi, Homa and Kabir, Ehsanollah. "A new distance measure for free text keystroke authentication". In: *Computer Conference, 2009. CSICC 2009. 14th International CSI*. IEEE. 2009, pp. 570–575.

[61] Denning, Dorothy E. "An Intrusion-Detection Model". In: *IEEE Transactions on Software Engineering* 2 (1987), pp. 222–232.

[62] Douhou, Salima and Magnus, Jan R. "The reliability of user authentication through keystroke dynamics". In: *Statistica Neerlandica* 63.4 (2009), pp. 432–449.

[63] Dowland, Paul S. and Furnell, Steven M. "A long-term trial of keystroke profiling using digraph, trigraph and keyword latencies". In: *Security and protection in information processing systems*. Springer, 2004, pp. 275–289.

[64] Eltahir, Wasil Elsadig, Salami, M. J. E., Ismail, Ahmad Faris, and Lai, Weng Kin. "Dynamic keystroke analysis using AR model". In: *Industrial Technology, 2004. IEEE ICIT'04. 2004 IEEE International Conference on*. Vol. 3. IEEE. 2004, pp. 1555–1560.

[65] Epp, Clayton, Lippold, Michael, and Mandryk, Regan L. "Identifying Emotional States using Keystroke Dynamics". In: *Conference on Human Factors in Computing Systems*. 2011.

[66] Fawcett, Tom and Provost, Foster. "Activity monitoring: Noticing interesting changes in behavior". In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 1999, pp. 53–62.

[67] Fiérrez Aguilar, Julian, Ortega Garcia, Javier, Garcia Romero, Daniel, and Gonzalez Rodriguez, Joaquin. "A comparative evaluation of fusion strategies for multimodal biometric verification". In: *Audio and Video based Biometric Person Authentication*. Springer. 2003, pp. 830–837.

[68] Fiérrez Aguilar, Julian et al. "BiosecurID: a multimodal biometric database". In: *Pattern Analysis and Applications* 13.2 (2010), pp. 235–246.

[69] Flior, Eric and Kowalski, Kazimierz. "Continuous biometric user authentication in online examinations". In: *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*. IEEE. 2010, pp. 488–492.

[70] Foo Kune, Denis and Kim, Yongdae. "Timing attacks on pin input devices". In: *Proceedings of the 17th ACM conference on Computer and communications security*. ACM. 2010, pp. 678–680.

[71] Furnell, Steven M., Sanders, Peter W., and Stockel, Colin T. "The use of keystroke analysis for continuous user identity verification and supervision". In: *Proceedings of MEDIACOMM 95, International Conference on Multimedia Communication*. 1995, pp. 189–193.

[72] Furnell, Steven M., Morrissey, Joseph P., Sanders, Peter W., and Stockel, Colin T. "Applications of keystroke analysis for improved login security and continuous user authentication". In: *Information systems security*. Chapman & Hall, Ltd. 1996, pp. 283–294.

[73] Furnell, Steven M., Dowland, Paul S., Illingworth, H. M., and Reynolds, Paul L. "Authentication and supervision: A survey of user attitudes". In: *Computers & Security* 19.6 (2000), pp. 529–539.

[74] Gagbla, George Kofi. "Applying keystroke dynamics for personal authentication". MA thesis. Department of Telecommunication and Signal Processing, Blekinge Insitute of Technology, 2005.

[75] Giot, Romain, Dorizzi, Bernadette, and Rosenberger, Christophe. "A review on the public benchmark databases for static keystroke dynamics". In: *Computers & Security* 55 (2015), pp. 46–61.

[76] Giot, Romain, El-Abed, Mohamad, and Rosenberger, Christophe. "Keystroke dynamics authentication for collaborative systems". In: *Collaborative Technologies and Systems, 2009. CTS'09. International Symposium on*. IEEE. 2009, pp. 172–179.

[77] Giot, Romain, El-Abed, Mohamad, and Rosenberger, Christophe. "Web-based benchmark for keystroke dynamics biometric systems: A statistical analysis". In: *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012 Eighth International Conference on*. IEEE. 2012, pp. 11–15.

[78] Giot, Romain, Ninassi, Alexandre, El-Abed, Mohamad, and Rosenberger, Christophe. "Analysis of the acquisition process for keystroke dynamics". In: *Biometrics Special Interest Group (BIOSIG), 2012 BIOSIG-Proceedings of the International Conference of the*. IEEE. 2012, pp. 1–6.

[79] Giot, Romain, El-Abed, Mohamad, Hemery, Baptiste, and Rosenberger, Christophe. "Unconstrained keystroke dynamics authentication with shared secret". In: *Computers & Security* 30.6 (2011), pp. 427–445.

[80] Giroux, Shallen, Wachowiak-Smolikova, Renata, and Wachowiak, Mark Paul. "Keystroke-based authentication by key press intervals as a complementary behavioral biometric". In: *Systems, Man, and Cybernetics, 2009. SMC 2009. IEEE International Conference on*. IEEE. 2009, pp. 80–85.

[81] Guven, Aykut and Sogukpinar, Ibrahim. "Understanding users' keystroke patterns for computer access security". In: *Computers & Security* 22.8 (2003), pp. 695–706.

[82] Haider, Sajjad, Abbas, Ahmed, and Zaidi, Abbas K. "A multi-technique approach for user identification through keystroke dynamics". In: *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*. Vol. 2. IEEE. 2000, pp. 1336–1341.

[83] Harun, N., Woo, W. L., and Dlay, S. S. "Performance of keystroke biometrics authentication system using artificial neural network (ANN) and distance classifier method". In: *Computer and Communication Engineering (ICCCE), 2010 International Conference on*. IEEE. 2010, pp. 1–6.

[84] Hempstalk, Kathryn. "Continuous typist verification using machine learning". PhD thesis. The University of Waikato, 2009.

[85] Hernández, José Guadalupe Aguilar and Pérez, Luis Adrían Lizama. *Autenticación de usuarios a través de Biometría de Tecleo*. Tech. rep. Universidad Juárez Autónoma de Tabasco., 2007.

[86] Ho, Jiacang and Kang, Dae-Ki. "Sequence Alignment with Dynamic Divisor Generation for Keystroke Dynamics Based User Authentication". In: *Journal of Sensors* (2015).

[87] Hocquet, Sylvain, Ramel, Jean-Yves, and Cardot, Hubert. "Fusion of methods for keystroke dynamic authentication". In: *Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on*. IEEE. 2005, pp. 224–229.

[88] Hocquet, Sylvain, Ramel, Jean-Yves, and Cardot, Hubert. "User classification for keystroke dynamics authentication". In: *Advances in biometrics*. Springer, 2007, pp. 531–539.

[89] Hosseinzadeh, Danoush and Krishnan, Sridhar. "Gaussian mixture modeling of keystroke patterns for biometric applications". In: *IEEE Transactions on Systems, Man and Cybernetics* 38.6 (2008), pp. 816–826.

[90] Hosseinzadeh, Danoush, Krishnan, Sridhar, and Khademi, April. "Keystroke identification based on Gaussian mixture models". In: *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing*. Vol. 3. IEEE. 2006.

[91] Hsu, Chih-Wei and Lin, Chih-Jen. "A comparison of methods for multiclass support vector machines". In: *Neural Networks, IEEE Transactions on* 13.2 (2002), pp. 415–425.

[92] Huang, Xuan, Lund, Geoffrey, and Sapeluk, Andrew. "Development of a typing behaviour recognition mechanism on android". In: *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*. IEEE. 2012, pp. 1342–1347.

[93] Hussain, Abdulameer Khalaf and Al-Hassan, Mustafa Nouman. "Multifactor Strong Authentication Method Using Keystroke Dynamics". In: *International Journal of Science and Applied Information Technology*. Vol. 2. 2013, pp. 31–34.

[94] Hwang, Seong-seob, Lee, Hyoung-joo, and Cho, Sungzoon. "Improving authentication accuracy using artificial rhythms and cues for keystroke dynamics-based authentication". In: *Expert Systems with Applications* 36.7 (2009), pp. 10649–10656.

[95] Ilonen, Jarmo. "Keystroke dynamics". In: *Advanced Topics in Information Processing* (2003), pp. 03–04.

[96] Jagadeesan, Harini and Hsiao, Michael S. "A novel approach to design of user re-authentication systems". In: *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*. IEEE. 2009, pp. 1–6.

[97] Jain, Anil K., Mao, Jianchang, and Mohiuddin, K. M. "Artificial neural networks: A tutorial". In: *Computer* 29.3 (1996), pp. 31–44.

[98] Jiang, Cheng-Huang, Shieh, Shiuhpyng, and Liu, Jen-Chien. "Keystroke statistical learning model for web authentication". In: *Proceedings of the 2nd ACM symposium on Information, computer and communications security*. ACM. 2007, pp. 359–361.

[99] Jin, Zhe, Teoh, Andrew Beng Jin, Ong, Thian Song, and Tee, Connie. "Typing dynamics biometric authentication through fuzzy logic". In: *Information Technology, 2008. ITSim 2008. International Symposium on*. Vol. 3. IEEE. 2008, pp. 1–6.

[100] Jonge, Edwin de and Loo, Mark van der. *An introduction to data cleaning with R*. Tech. rep. Statistics Netherlands, 2013.

[101] Kaneko, Yoshihiro, Kinpara, Yuji, and Shiomi, Yuta. "A hamming distance-like filtering in keystroke dynamics". In: *Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on*. IEEE. 2011, pp. 93–95.

[102] Kang, Pilsung and Cho, Sungzoon. "A hybrid novelty score and its use in keystroke dynamics-based user authentication". In: *Pattern recognition* 42.11 (2009), pp. 3115–3127.

[103] Kang, Pilsung et al. "Improvement of keystroke data quality through artificial rhythms and cues". In: *Computers & Security* 27.1 (2008), pp. 3–11.

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

[105] Killourhy, Kevin S. and Maxion, Roy A. "Comparing anomaly-detection algorithms for keystroke dynamics". In: *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on*. IEEE. 2009, pp. 125–134.

[106] Killourhy, Kevin S. and Maxion, Roy A. "The effect of clock resolution on keystroke dynamics". In: *Recent Advances in Intrusion Detection*. Springer. 2008, pp. 331–350.

[107] Killourhy, Kevin S. and Maxion, Roy A. "Why did my detector do that?!" In: *Recent Advances in Intrusion Detection*. Springer. 2010, pp. 256–276.

[108] Krishna, D. Ramya and Koteswaramma, D. "KeyStroke Dynamics-Dangling Issues of Providing Authentication by Recognising User Input". In: *Control Theory and Informatics* 4.1 (2014), pp. 16–18.

[109] Kuan, Hung-i. *Evaluation of a biometric keystroke typing dynamics computer security system*. Tech. rep. DTIC Document, 1992.

[110] Kumar, Sandeep, Sim, Terence, Janakiraman, Rajkumar, and Zhang, Sheng. "Using continuous biometric verification to protect interactive login sessions". In: *Computer Security Applications Conference, 21st Annual*. IEEE. 2005, p. 10.

[111] Kwang, Geraldine, Yap, Roland H. C., Sim, Terence, and Ramnath, Rajiv. "An usability study of continuous biometrics authentication". In: *Advances in Biometrics*. Springer, 2009, pp. 828–837.

[112] Lane, Terran and Brodley, Carla E. "An application of machine learning to anomaly detection". In: *Proceedings of the 20th National Information Systems Security Conference*. Vol. 377. Baltimore, USA. 1997, pp. 366–380.

[113] Lau, Edmond, Liu, Xia, Xiao, Chen, and Yu, Xiao. "Enhanced user authentication through keystroke biometrics". In: *Massachusetts Institute of Technology* 9 (2004).

[114] Lee, Hyoung-joo and Cho, Sungzoon. "Retraining a keystroke dynamics-based authenticator with impostor patterns". In: *Computers & Security* 26.4 (2007), pp. 300–310.

[115] Lee, Wenke et al. "Real time data mining-based intrusion detection". In: *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings*. Vol. 1. IEEE. 2001, pp. 89–100.

[116] Li, Jamy. *Keystroke Analysis: A Novel Type of Authentication*. Tech. rep. ESC300, 2003.

[117] Li, Lingjun, Zhao, Xinxin, and Xue, Guoliang. "Unobservable Re-authentication for Smartphones". In: *NDSS*. 2013.

[118] Lin, Daw-Tung. "Computer-access authentication with neural network based keystroke identity verification". In: *Neural Networks, 1997. International Conference on*. Vol. 1. IEEE. 1997, pp. 174–178.

[119] Loy, Chen Change, Lai, Weng Kin, and Lim, Chee Peng. "Keystroke patterns classification using the ARTMAP-FD neural network". In: *Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007. Third International Conference on*. Vol. 1. IEEE. 2007, pp. 61–64.

[120] Lozano Rivas, William Antonio. "Determinación del número mínimo de observaciones en investigación, obviando las estimaciones de la varianza de datos". In: *Revista de Didáctica Ambiental* 10 (2011), pp. 54–61.

[121] Magalhães, Sérgio Tenreiro de, Revett, Kenneth, and Santos, Henrique M. D. "Password secured sites-stepping forward with keystroke dynamics". In: *Next Generation Web Services Practices, 2005. NWeSP 2005. International Conference on*. IEEE. 2005, p. 6.

[122] Mahar, Doug et al. "Optimizing digraph-latency based biometric typist verification systems: inter and intra typist differences in digraph latency distributions". In: *International Journal of Human-Computer Studies* 43.4 (1995), pp. 579–592.

[123] Mandujano, Salvador and Soto, Rogelio. "Deterring password sharing: User authentication via fuzzy c-means clustering applied to keystroke biometric data". In: *Computer Science, 2004. ENC 2004. Proceedings of the Fifth Mexican International Conference in*. IEEE. 2004, pp. 181–187.

[124] Mansfield, Anthony J and Wayman, James L. *Best practices in testing and reporting performance of biometric devices*. Centre for Mathematics and Scientific Computing, 2002.

[125] Maxion, Roy A. and Killourhy, Kevin S. "Keystroke biometrics with number-pad input". In: *Dependable Systems & Networks (DSN), 2010 IEEE/IFIP International Conference on*. IEEE. 2010, pp. 201–210.

[126] Mayer, DG and Butler, DG. "Statistical validation". In: *Ecological modelling* 68.1-2 (1993), pp. 21–32.

[127] McHugh, John. "Intrusion and intrusion detection". In: *International Journal of Information Security* 1.1 (2001), pp. 14–35.

[128] Meszaros, Attila, Banko, Zoltan, and Czuni, Laszlo. "Strengthening passwords by keystroke dynamics". In: *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2007. IDAACS 2007. 4th IEEE Workshop on*. IEEE. 2007, pp. 574–577.

[129] Mhenni, Abir, Rosenberger, Christophe, Cherrier, Estelle, and Amara, Najoua Essoukri Ben. "Keystroke Template Update with Adapted Thresholds". In: *International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. 2016.

[130] Miller, Benjamin. "Vital signs of identity". In: *IEEE Spectrum* 31.2 (1994), pp. 22–30.

[131] Miluzzo, Emiliano, Varshavsky, Alexander, Balakrishnan, Suhrid, and Choudhury, Romit Roy. "Tapprints: your finger taps have fingerprints". In: *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM. 2012, pp. 323–336.

[132] Modi, Shimon and Elliott, Stephen J. "Keystroke dynamics verification using a spontaneously generated password". In: *Carnahan Conferences Security Technology, Proceedings 2006 40th Annual IEEE International*. IEEE. 2006, pp. 116–121.

[133] Monaco, John V., Ali, Md Liakat, and Tappert, Charles C. "Spoofing key-press latencies with a generative keystroke dynamics model". In: *Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International, Conference on* (2015).

[134] Monaco, John V. et al. "One-handed keystroke biometric identification competition". In: *Biometrics (ICB), 2015 International Conference on*. IEEE. 2015, pp. 58–64.

[135] Montalvao Filho, Jugurta R., Almeida, Carlos Augusto S., and Freire, Eduardo O. "Equalization of keystroke timing histograms for improved identification performance". In: *Telecommunications Symposium, 2006 International*. IEEE. 2006, pp. 560–565.

[136] Montalvao Filho, Jugurta R. and Freire, Eduardo O. "Multimodal biometric fusion-joint typist (keystroke) and speaker verification". In: *Telecommunications symposium, 2006 international*. IEEE. 2006, pp. 609–614.

[137] Morales, Aythami, Fiérrez Aguilar, Julian, and Ortega Garcia, Javier. "Towards predicting good users for biometric recognition based on keystroke dynamics". In: *Computer Vision-ECCV 2014 Workshops*. Springer. 2014, pp. 711–724.

[138] Moskovitch, Robert et al. "Identity theft, computers and behavioral biometrics". In: *Intelligence and Security Informatics, 2009. ISI'09. IEEE International Conference on*. IEEE. 2009, pp. 155–160.

[139] Nagaraju, Bhagirathi. "Trigraph latency as a method to infer fatigue during typing". PhD thesis. Osmania University, 2006.

[140] Nascimento Costa, Carlos Roberto do. "Autenticação Biométrica via Teclado Numérico Baseada na Dinâmica da Digitação: Experimentos e Resultados". PhD thesis. Universidade Estadual de Campinas, 2006.

[141] Niinuma, Koichiro and Jain, Anil K. "Continuous user authentication using temporal information". In: *SPIE Defense, Security, and Sensing*. Vol. 7667. International Society for Optics and Photonics. 2010.

[142] Niinuma, Koichiro, Park, Unsang, and Jain, Anil K. "Soft biometric traits for continuous user authentication". In: *Information Forensics and Security, IEEE Transactions on* 5.4 (2010), pp. 771–780.

[143] Nisenson, Mordechai, Yariv, Ido, El-Yaniv, Ran, and Meir, Ron. "Towards behaviometric security systems: Learning to identify a typist". In: *Knowledge Discovery in Databases: PKDD 2003*. Springer, 2003, pp. 363–374.

[144] Obaidat, Mohammad S. "A methodology for improving computer access security". In: *Computers & Security* 12.7 (1993), pp. 657–662.

[145] Obaidat, Mohammad S. and Macchiarolo, David T. "A multilayer neural network system for computer access security". In: *IEEE Transactions on Systems, Man and Cybernetics* 24.5 (1994), pp. 806–813.

[146] Obaidat, Mohammad S. and Sadoun, Balqies. "A simulation evaluation study of neural network techniques to computer user identification". In: *Information Sciences* 102.1 (1997), pp. 239–258.

[147] Obaidat, Mohammad S. and Sadoun, Balqies. "Keystroke dynamics based authentication". In: *Biometrics*. Springer, 1996, pp. 213–229.

[148] Oel, Peter, Schmidt, Paul, and Schmitt, Alfred. "Time prediction of mouse-based cursor movements". In: *Proceedings of Joint AFIHM-BCS Conference on Human-Computer Interaction IHM-HCI*. Vol. 2. 2001, pp. 37–40.

[149] O'Gorman, Lawrence. "Comparing passwords, tokens, and biometrics for user authentication". In: *Proceedings of the IEEE* 91.12 (2003), pp. 2021–2040.

[150] Omote, Kazumasa and Okamoto, Eiji. "User identification system based on biometrics for keystroke". In: *Information and Communication Security*. Springer, 1999, pp. 216–229.

[151] Özdemir, Musa Kazim. "A framework for authentication of medical reports based on keystroke dynamics". PhD thesis. Middle East Technical University, 2010.

[152] Park, Sunghoon, Park, Jooseoung, and Cho, Sungzoon. "User authentication based on keystroke analysis of long free texts with a reduced number of features". In: *Communication Systems, Networks and Applications (ICCSNA), 2010 Second International Conference on*. Vol. 1. IEEE. 2010, pp. 433–435.

[153] Pavaday, Narainsamy and Soyjaudah, K. M. S. "Performance of the K Nearest Neighbor in Keyboard Dynamic Authentication". In: *Proceedings of the 2007 Computer Science and IT Education Conference*. 2007, pp. 599–604.

[154] Peacock, Alen, Ke, Xian, and Wilkerson, Matthew. "Typing Patterns: A Key to User Identification". In: *IEEE Security & Privacy* 2.5 (2004), pp. 40–47.

[155] Pérez, Luis Adrián Lizama and Aguilar, José Guadalupe. *Keystroke Dynamics Applied to Authentication of Network Users*. Tech. rep. Universidad Juárez Autónoma de Tabasco, 2003.

[156] Pisani, Paulo Henrique, Giot, Romain, De Carvalho, André CPLF, and Lorena, Ana Carolina. "Enhanced template update: Application to keystroke dynamics". In: *Computers & Security* 60 (2016), pp. 134–153.

[157] Pusara, Maja. "An examination of user behavior for user re-authentication". PhD thesis. Purdue University, 2007.

[158] Pusara, Maja and Brodley, Carla E. "User re-authentication via mouse movements". In: *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*. ACM. 2004, pp. 1–8.

[159] Raghu, D., Jacob, CH. Raja, and Bhavani, Y. V. K. D. "Neural Network Based Authentication and Verification for Web Based Key Stroke Dynamics". In: *Dimension* 2 (2011), p. 1.

[160] Revett, Kenneth. "A bioinformatics based approach to behavioural biometrics". In: *Frontiers in the Convergence of Bioscience and Information Technologies, 2007. FBIT 2007*. IEEE. 2007, pp. 665–670.

[161] Revett, Kenneth, Magalhães, Sérgio Tenreiro de, and Santos, Henrique M. D. "Data mining a keystroke dynamics based biometrics database using rough sets". In: *Portuguese Conference on Artificial intelligence, 2005 (EPIA)*. Ieee. 2005, pp. 188–191.

[162] Revett, Kenneth, Magalhães, Sérgio Tenreiro de, and Santos, Henrique M. D. *Developing a keystroke dynamics based agent using rough sets*. Tech. rep. University of Technology of Compiègne, University of Westminster, 2005.

[163] Revett, Kenneth, Magalhães, Sérgio Tenreiro de, and Santos, Henrique M. D. "On the use of rough sets for user authentication via keystroke dynamics". In: *Progress in Artificial Intelligence*. Springer, 2007, pp. 145–159.

[164] Revett, Kenneth et al. "A machine learning approach to keystroke dynamics based user authentication". In: *International Journal of Electronic Security and Digital Forensics* 1.1 (2007), pp. 55–70.

[165] Revett, Kenneth, Jahankhani, Hamid, Magalhães, Sérgio Tenreiro de, and Santos, Henrique M. D. "A survey of user authentication based on mouse dynamics". In: *Global E-Security*. Springer, 2008, pp. 210–219.

[166] Riha, Zdenek and Matyás, Václav. "Toward reliable user authentication through biometrics". In: *IEEE Security & Privacy* 1.3 (2003), pp. 45–49.

[167] Robinson, John A., Liang, V. W., Chambers, J. A. Michael, and Mackenzie, Christine L. "Computer user verification using login string keystroke dynamics". In: *IEEE Transactions on Systems, Man and Cybernetics* 28.2 (1998), pp. 236–241.

[168] Rodrigues, Ricardo Nagel et al. "Biometric access control through numerical keyboards based on keystroke dynamics". In: *Advances in biometrics*. Springer, 2005, pp. 640–646.

[169] Rundhaug, Fred Erlend N. "Keystroke dynamics - Can attackers learn someone's typing characteristics". MA thesis. Gjøvik University College, 2007.

[170] Ryan, Shea. "Mobile keystroke dynamics: assessment and implementation". PhD thesis. California State University, Northridge, 2015.

[171] Rybnik, Mariusz, Panasiuk, Piotr, and Saeed, Khalid. "User authentication with keystroke dynamics using fixed text". In: *Biometrics and Kansei Engineering, 2009. ICBAKE 2009. International Conference on*. IEEE. 2009, pp. 70–75.

[172] Rybnik, Mariusz, Tabedzki, Marek, and Saeed, Khalid. "A keystroke dynamics based system for user identification". In: *Computer Information Systems and Industrial Management Applications, 2008. CISIM'08. 7th*. IEEE. 2008, pp. 225–230.

[173] Saevanee, Hataichanok and Bhatarakosol, Pattarasinee. "User authentication using combination of behavioral biometrics over the touchpad acting like touch screen of mobile device". In: *Computer and Electrical Engineering, 2008. ICCEE 2008. International Conference on*. IEEE. 2008, pp. 82–86.

[174] Saggio, Giovanni, Costantini, Giovanni, and Todisco, Massimiliano. "Cumulative and Ratio Time Evaluations in Keystroke Dynamics To Improve the Password Security Mechanism". In: *Journal of Computer and Information Technology* 2.1 (2011), pp. 2–11.

[175] Samura, Toshiharu and Nishimura, Haruhiko. "Keystroke timing analysis for personal authentication in Japanese long text input". In: *SICE Annual Conference (SICE), 2011. Proceedings of*. IEEE. 2011, pp. 2121–2126.

[176] Sang, Yingpeng, Shen, Hong, and Fan, Pingzhi. "Novel impostors detection in keystroke dynamics by support vector machine". In: *Parallel and distributed computing: applications and technologies*. Springer, 2005, pp. 666–669.

[177] Schonlau, Matthias et al. "Computer intrusion: Detecting masquerades". In: *Statistical science* (2001), pp. 58–74.

[178] Schuckers, Michael E. "Test Sample and Size". In: *Encyclopedia of Biometrics* (2009), pp. 1328–1332.

[179] Schuckers, Michael E, Hawley, Anne, Livingstone, Katie, and Mramba, Nona. "A comparison of statistical methods for evaluating matching performance of a biometric identification device: a preliminary report". In: *Defense and Security*. International Society for Optics and Photonics. 2004, pp. 144–155.

[180] Shahzad, Muhammad, Zahid, Saira, and Farooq, Muddassar. "A hybrid GA-PSO fuzzy system for user identification on smart phones". In: *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*. ACM. 2009, pp. 1617–1624.

[181] Shanmugapriya, D. and Padmavathi, Ganapathi. "A survey of biometric keystroke dynamics: Approaches, security and challenges". In: *arXiv preprint arXiv:0910.0817* (2009).

[182] Shanmugavalli, V., Chandrasekar, V., and Krishna Sankar, P. *Biometric Authentication Based on Keystroke Dynamics for Realistic User*. Tech. rep. K S R Institute for Engineering and Technology, Tiruchengode, 2014.

[183] Sharif, Muhammad, Faiz, Tariq, and Raza, Mudassar. "Time signatures-an implementation of Keystroke and click patterns for practical and secure authentication". In: *Digital Information Management, 2008. ICDIM 2008. Third International Conference on*. IEEE. 2008, pp. 559–562.

[184] Shimshon, Tomer, Moskovitch, Robert, Rokach, Lior, and Elovici, Yuval. "Continuous verification using keystroke dynamics". In: *Computational Intelligence and Security (CIS), 2010 International Conference on*. IEEE. 2010, pp. 411–415.

[185] Shorrock, Steven Richard. "A new approach to securing passwords using a probabilistic neural network based on biometric keystroke dynamics". PhD thesis. University of Newcastle upon Tyne, 2003.

[186] Silva, Leandro. "Behavioural Biometrics in the World Wide Web". MA thesis. Universidade de Coimbra, 2014.

[187] Singh, Saurabh and Arya, K. V. "Key classification: a new approach in free text keystroke authentication system". In: *Circuits, Communications and System (PACCS), 2011 Third Pacific-Asia Conference on*. IEEE. 2011, pp. 1–5.

[188] Sinthupinyo, Sukree, Roadrungwasinkul, Warut, and Chantan, Charoon. "User recognition via keystroke latencies using SOM and backpropagation neural network". In: *ICCAS-SICE, 2009*. IEEE. 2009, pp. 3160–3165.

[189] Song, Dawn Xiaodong, Venable, Peter, and Perrig, Adrian. *User recognition by keystroke latency pattern analysis*. Tech. rep. Carnegie Mellon University, 1997.

[190] Stamatatos, Efstathios. "A survey of modern authorship attribution methods". In: *Journal of the American Society for information Science and Technology* 60.3 (2009), pp. 538–556.

[191] Stanciu, Valeriu-Daniel, Spolaor, Riccardo, Conti, Mauro, and Giuffrida, Cristiano. "On the Effectiveness of Sensor-enhanced Keystroke Dynamics Against Statistical Attacks". In: *Proceedings of the Sixth ACM on Conference on Data and Application Security and Privacy*. ACM. 2016, pp. 105–112.

[192] Stefan, Deian, Shu, Xiaokui, and Yao, Danfeng. "Robustness of keystroke-dynamics based biometrics against synthetic forgeries". In: *Computers & Security* 31.1 (2012), pp. 109–121.

[193] Stefan, Deian and Yao, Danfeng. "Keystroke-dynamics authentication against synthetic forgeries". In: *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010 6th International Conference on*. IEEE. 2010, pp. 1–8.

[194] Tapiador, Marino and Sigüenza, Juan Alberto. *Fuzzy keystroke biometrics on web security*. Tech. rep. Universidad Autónoma de Madrid, 2000.

[195] Tappert, Charles C., Villani, Mary, and Cha, Sung-Hyuk. "Keystroke biometric identification and authentication on long-text input". In: *Behavioral Biometrics for Human Identification: Intelligent Applications* (2009).

[196] Tappert, Charles C, Cha, Sung-Hyuk, Villani, Mary, and Zack, Robert S. "A keystroke biometric system for long-text input". In: *International Journal of Information Security and Privacy* 4.1 (2010), pp. 32–60.

[197] Team, R Core. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2015. URL: `https://www.R-project.org/`.

[198] Tseng, Chun-wei, Liu, Feng-jung, and Lin, Ting-yi. "Design and implementation of a RFID-based authentication system by using keystroke dynamics". In: *Systems, Man, and Cybernetics, 2010 IEEE International Conference on*. IEEE. 2010, pp. 3926–3929.

[199] Uludag, Umut and Jain, Anil K. "Attacks on biometric systems: a case study in fingerprints". In: *Electronic Imaging 2004*. International Society for Optics and Photonics. 2004, pp. 622–633.

[200] Vural, Esra, Huang, Jiaju, Hou, Daqing, and Schuckers, Stephanie. "Shared research dataset to support development of keystroke authentication". In: *Biometrics (IJCB), 2014 IEEE International Joint Conference on*. IEEE. 2014, pp. 1–8.

[201] Vuyyuru, Sampath K. et al. "Computer user authentication using hidden markov model through keystroke dynamics". In: *Transactions on Information and System Security* (2006).

[202] Weiss, Adam et al. "Mouse movements biometric identification: A feasibility study". In: *Proc. Student/Faculty Research Day CSIS, Pace University, White Plains, NY* (2007).

[203] Wespi, Andreas, Dacier, Marc, and Debar, Hervé. "Intrusion detection using variable-length audit trail patterns". In: *Recent Advances in Intrusion Detection*. Springer. 2000, pp. 110–129.

[204] Wong, Fadhli Wong Mohd Hasan et al. "Enhanced user authentication through typing biometrics with artificial neural networks and k-nearest neighbor algorithm". In: *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on*. Vol. 2. IEEE. 2001, pp. 911–915.

[205] Xi, Kai, Tang, Yan, and Hu, Jiankun. "Correlation keystroke verification scheme for user access control in cloud computing environment". In: *The Computer Journal* (2011), pp. 1632–1644.

[206] Ye, Nong. "A markov chain model of temporal behavior for anomaly detection". In: *Proceedings of the 2000 IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*. Vol. 166. West Point, NY. 2000, p. 169.

[207] Yu, Enzhe and Cho, Sungzoon. "GA-SVM wrapper approach for feature subset selection in keystroke dynamics identity verification". In: *Neural Networks, 2003. Proceedings of the International Joint Conference on*. Vol. 3. IEEE. 2003, pp. 2253–2257.

[208] Yu, Enzhe and Cho, Sungzoon. "Keystroke dynamics identity verification, its problems and practical solutions". In: *Computers & Security* 23.5 (2004), pp. 428–440.

[209] Zack, Robert S., Tappert, Charles C., and Cha, Sung-Hyuk. "Performance of a long-text-input keystroke biometric authentication system using an improved k-nearest-neighbor classification method". In: *Biometrics: Theory, Applications, and Systems (BTAS), 2010 Fourth IEEE International Conference on*. IEEE. 2010, pp. 1–6.

[210] Zhang, Kehuan and Wang, XiaoFeng. *Peeping tom in the neighborhood: keystroke eavesdropping on multi-user systems*. Tech. rep. Indiana University, 2009, p. 23.

[211] Zhang, Ying, Chang, Guiran, Liu, Lin, and Jia, Jie. "Authenticating User's Keystroke Based on Statistical Models". In: *Genetic and Evolutionary Computing (ICGEC), 2010 Fourth International Conference on*. IEEE. 2010, pp. 578–581.

[212] Zhou, Charles. "A Study of Keystroke Dynamics as a Practical Form of Authentication". MA thesis. Pomona College, Claremont, California, 2008.

Memento Mori

Principat d'Andorra, 2017