



EÖTVÖS LORÁND UNIVERSITY
FACULTY OF INFORMATICS

FRAUD DETECTION BASED ON KEYSTROKE DYNAMICS

DR. KRISZTIAN BUZA
ASSISTANT PROFESSOR AT ELTE

ZAKARYA FAROU
COMPUTER SCIENCE

BUDAPEST, 2019.

Acknowledgement

First and foremost, I would like to praise God for bestowing me with never ending patience, grace and will to pursue this work.

I would like to express my gratitude to my supervisor Dr Krisztian Buza, for all the support, help, and guidance as well as for his valuable advice and encouragement in producing my dissertation. It was a great pleasure to work with him. I would like to thank him warmly and to express my sincere gratitude to him.

I would also like to acknowledge Dr Brahim Farou my brother and my leader who is an assistant professor of the faculty of informatics at Guelma University in Algeria. His guidance and motivation helped me a lot the process of researching and writing this thesis.

My sincere thanks to the Eötvös Loránd University for giving me this chance to study at such a great university. Thank you for all members of ELTE University, faculty of informatics and data science and engineering department.

I would like also to express my very profound gratitude to my mom houria, my brothers: brahim, dhia eddine and houssem eddine, my girlfriend nesrine and to all my friends for providing me with unfailing support and continuous encouragement throughout my years of study.

Last but not least. I would like to dedicate this thesis to the memory of my father Dr Lhadi Farou, who unfortunately didn't stay in this world long enough to see his son become a doctor. I miss him every day, but I am glad to know he saw my progress during my whole life so far, he offered me all the support that I needed to make it possible, as well as plenty of friendly encouragement, I love you habibi.

Contents

Chapter 1: Introduction

1. Motivation.....	1
2. Organization of the thesis	2

Chapter 2: Background

1. Biometrics	3
1.1. Introduction.....	3
1.2. Biometric authentication.....	4
1.3. Applications of biometrics.....	6
1.4. Advantages and disadvantages of biometrics	6
1.5. Common biometric techniques and forms	7
1.6. Keystroke dynamics over other techniques	10
2. Keystroke dynamics.....	11
2.1. Introduction to keystroke dynamics.....	11
2.2. Keystroke dynamics origins	12
2.3. Keystroke dynamics System.....	13
2.4. Biometric evaluation.....	16
2.5. Classification techniques	17
2.6. Keystroke dynamics applications	19
3. Anomaly detection	20
3.1. Introduction to anomaly detection	20
3.2. Types of anomalies	21
3.3. Learning methodologies in anomaly detection.....	23
3.4. Anomaly detection techniques.....	26
3.5. Fraud-anomaly detection relationship	28
4. Time series analysis	29
4.1. Definition of time series	30
4.2. Time series data mining tasks.....	30
4.3. Time series similarity measures.....	31
5. Related works.....	33
5.1. Implementations	33
5.2. Local authentication.....	34

5.3. Web authentication	34
-------------------------------	----

Chapter 3: Web-based K.S.D person verification system mechanism

1. Methodology	36
2. Client-server communication.....	38
3. Classification process.....	41
4. Measuring the similarity of keystroke time series	42

Chapter 4: The implementation of TyPaVeS

1. Programming environmental	44
1.1. WAMP platform	44
1.2. User interface.....	45
2. Client side functions	45
3. Server side functions.....	46
3.1. Pre-processing of raw typing patterns	47
3.2. Extraction of features.....	48
3.3. Classification of candidate user typing patterns	49
4. Database conception	50
5. Web-site usability and design	50

Chapter 5: Evaluation and experiments

1. Evaluation of TyPaVeS	54
2. Review of DTW distance.....	54
3. Lower bounding the DTW distance.....	55
4. Experiment.....	57
4.1. Person authentication dataset.....	57
4.2. Methodology.....	58
4.3. Experimental results	58

Chapter 6: Summary

1. Conclusion	62
2. Food for thought	63
3. Future work.....	63

Bibliography

List of Figures

2.1. Taxonomy by Technology Type	4
2.2. Diagram illustrating the process of Enrollment and Authentication according to [4]	5
2.3. Fingerprint scanning system according to [1]	7
2.4. Facial features used in face recognition systems [2]	8
2.5. Iris [5]	8
2.6. Hand geometry features according to [9].....	9
2.7. Example of signature shape [3].....	9
2.8. A general timeline on the overview of keystroke research work evolution [28]	12
2.9. Latencies between keystrokes when writing the word “password” by three different people. [30]	13
2.10. Keystroke features	14
2.11. Some of the most important features of a keystroke sensitive password.	14
2.12. Typical biometric methodology.....	15
2.13. Accuracy [27].....	16
2.14. Relationship between FAR, FRR, and EER. [32]	17
2.15. A simple example of anomalies in a 2-dimensional data set. [61]	21
2.16. Illustration of a time series concerning logged sensor data. [62]	21
2.17. Sequential anomaly in a time series of logged sensor data. [62].....	22
2.18. Example of a contextual anomaly (marked in red) in a time series[62]	23
2.19. Supervised machine learning process	24
2.20. Unsupervised machine learning process (clustering)	25
2.21. Problem solving using anomaly detection techniques.....	28
2.22. Time series of the age of death of 42 successive kings of England. [65]	30
2.23. A possible taxonomy of time series similarity measures.....	33
3.1. Overview of verification process based on our proposed TyPaVeS.....	37
3.2. Overview of enrollment process based on our proposed TyPaVeS	37
3.3. Client-server communication	38
3.4. Detailed client-server communication	39
3.5. Password equality checking by using a hash function	40
3.6. Classification process between pairwise of users typing prototypes	41
3.7. Similarity measure between pairwise of time Series	43
4.1. Example of user typing pattern raw data.....	47
4.2. Eliminating same successive events.....	48
4.3. Database structure	50

LIST OF FIGURES

4.4. Registration form along with some greeting	51
4.5. Login form	51
4.6. Person typing verification form	52
4.7. Genuine/user typing prototypes	53
4.8. Typing pattern algorithm results	53
4.9. Classifier decision	53
5.1. Optimal warping path of a pairwise time series [97]	55
5.2. Sakoe-Chiba band and Itakura parallelogram [99]	56
5.3. Testing data classification process	58
5.4. The classification accuracies for all warping window sizes and Between KS Duration ...	59
5.5. The classification accuracies for all warping window sizes and KS Duration	59
5.6. The classification accuracies for all warping window sizes and Merge Duration.....	60
5.7. The classification accuracies for all warping window sizes. All accuracies peak at very small window sizes [98].....	60

List of Tables

2.1. Most important advantages and disadvantages of biometrics.....	7
2.2. Keystroke dynamics vs biometric techniques	11
5.1. The warping window size that yields maximum classification accuracy for each dataset, using DTW with Sakoe-Chiba Band [98]	61

List of abbreviations

CAS	Continuous Authentication System
CSeS	Cyber Security Scientist
CBAS	Continuous Biometric Authentication Systems
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
ROI	Return On Investment
INSPASS	INS Passenger Accelerated Service System
DNA	DeoxyriboNucleic Acid
KSD	Key Stroke Dynamics
KSDS	Key Stroke Dynamics System
FAR	False Acceptance Rate
FRR	False Rejection Rate
ERR	Equal Error Rate
CCN	Credit Card Number
SVM	Support Vector Machine
DH-11	Basic temeperature/humidity sensor
AD	Anomaly Detection
DB	Database
DTW	Dynamic Time Warping
1-NN	1-Nearest-Neighbor
LCSS	Longest Common Sub Sequence
ED	Edit Distance

LIST OF ABBREVIATIONS

EED	Extended Edit Distance
ERP	Edit distance with Real Penalty
BEAD	Assignment Management System
TyPaVeS	Typing Pattern Verification System
CSC	Client-Server Communication
AuS	Authentication Server
AppS	Application Server
MVC	Model-View-Controller
WAMP	Windows, Apache, MySQL, and PHP
RDBMS	relational database management system
SQL	Structured Query Language
PHP	Hypertext Pre-Processor
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
KS	Key Stroke

Chapter 1

Introduction

1. Motivation

Since the establishment of computer systems, many activities of everyday life have been simplified. Currently, individuals can easily process information using software and computer networks. Given its evolution, this system characterizes the majority of large companies whatever the sector of activity.

Today, computers are replacing papers, calculators, radios, mailboxes, and so on. In fact the goal of computer scientists is to develop this technology in all areas of life of the population and automate the processing of information. IT uses many tools to transmit information.

The exchanged data may be confidential. Furthermore, it must be guaranteed that the data is not modified while it is being transferred. It's an imperative. Hence the importance of establishing a perfectly secure communication channel. In order to build such a secure channel we use IT security that must offer availability, confidentiality and a full integrity of the system, let us assume that each system have an admin panel and each admin have his own and confidential key (unique username and password) that allows him to access the system properly, there is a high possibility that an imposter tries to attack the network, system and the user/admin levels even if it has a high level of firewalls. According to the literature, most information security research in the recent years are focusing with system and network-level attacks.

However there is a lack of research on attacks in the user level which leads to allow impostors to focus more in this small window overlooked by the cyber security scientist (CSeS), while CSeS tries to build a pure authentication layer to the system using some tricky asymmetric encryption with a high level of digital signature and a pretty hash coding to store passwords in a way that none could decrypt them, the impostor can just take over from a valid person either at the start of a computer session or during the session itself. Most current computer systems authorize the user at the start of the session and do not detect any anomaly regarding the identity of the user (it doesn't know if the current user is still the initial authorized user or he has been substituted by an impostor), so an impostor in this case can access the system and do whatever he wants. Therefore, a system that continuously checks the identity of the user throughout the session is necessary. Such system already exists and it's denoted as *continuous authentication system* (CAS).

Most of the CAS's uses biometrics. These *continuous biometric authentication systems* (CBAS) are supplied by user traits and characteristics. There are two major forms of biometrics: those based

on physiological attributes and those based on behavioral characteristics. The physiological type includes biometrics based on stable body traits, such face, iris, fingerprint and the hand, and are considered to be more robust and secure. However, they are also considered to be more intrusive, expensive and require regular equipment replacement [105]. On the other hand, behavioral biometrics include learned movements such as handwritten signatures, keyboard dynamics (typing), mouse movements, gait and speech. Collecting of these biometrics is less obtrusive and they do not require extra hardware [106].

Recently, keystroke dynamics has gained popularity as one of the main sources of behavioral biometrics for providing continuous user authentication. Keystroke dynamics is appealing for many reasons [107]:

- It is less obtrusive, since users will be typing on the computer keyboard anyway.
- It does not require extra hardware.
- Keystroke dynamics exist and are available after the authentication step at the start of the computer session.

Analyzing how the data is typed instead of its content has proved to be very useful in distinguishing between users and certainly can be used as a biometric authentication to make a difference between a legitimate user from impostor who may take over from the valid user by finding features and unique characteristics that define a user safely.

The goal of this thesis is to develop, implement and experimentally evaluate an approach that can be used to decide if a legitimate user is accessing the system or not. The detection should be based on the user's keystroke dynamics.

2. Organization of the thesis

The thesis is organized as follows:

- Chapter 2 provides an overview about biometric technologies, keystroke dynamics, anomaly detection, time series analysis and ends with a brief review of some related works.
- Chapter 3 gives a detailed explanation of our proposed model in term of conception, how to check user identity and an overview of the algorithms used for both client-server communication and user classification process.
- Chapter 4 gives a detailed implementation of the system that we used in the experiment, database structure, preprocessing, extraction of features and classification of typing patterns, and finish with a presentation of the web-based prototype.
- Chapter 5 shows a special study case over the DTW warping window size and the evaluation of the proposed model.
- Chapter 6 presents the conclusion of the work and provides the possible future works.

Chapter 2

Background

In this chapter biometric technologies, keystroke dynamics, anomaly detection and time series analysis are introduced while research conducted in the field of keystroke dynamics is briefly reviewed along with some related works.

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 [12].

1.1.Introduction

Around mid-19th century, body measurements such as biometrics has been used in criminal identification and law enforcement, it has a very few area of applications at that time due to the lack of the technology, nowadays biometric techniques have been used more and more as a means to recognize users in common daily applications [13]. The word biometric is a combination of two words, bio as in biological; and metric, as in measurement. That is to say, biometrics are biological measurements. But the technical meaning of biometrics refers to metrics related to human characteristics that can be used as a form of identification, that's the biometrics authentication [1, 2, 8, 9, 31].

There are lots of biological measurements that represent human characteristics (traits) that can be used as a biometric identifier. These traits fall into one of these two categories [14]:

- **Physiological traits** are biological or chemical characteristics that either the person has acquired during his life or he inherited them (innate characteristics). Examples of these are: the iris, the DNA, the hand palm, the ear, or the face geometry, among others.
- **Behavioral traits** these are either trained or acquired over time. Examples of these could be: signature dynamics, voice print, gait, gestures and keystroke dynamics.

The following figure represents the taxonomy by technology type of biometrics [20]:

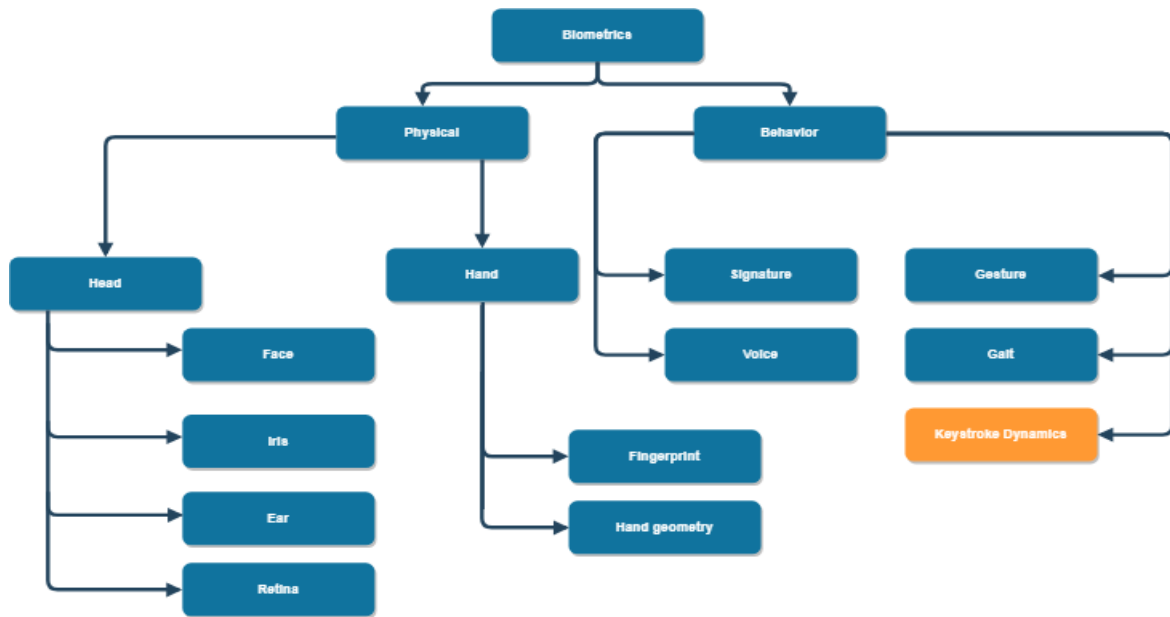


Fig 2.1: Taxonomy by technology Type

There exist another classification of the biometrics such soft/hard biometrics where soft biometrics traits are usually associated to behavioral traits that we mentioned above and doesn't take into account the distinction and permanence factors to sufficiently differentiate any two individuals [15], but hard biometrics are considered more efficient in term of distinctiveness and permanence.

In order to be considered as a valid biometric, it should satisfy as much as possible out of the following requirements [13]:

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

1.2. Biometric authentication

Biometrics are widely used in information systems, especially in the authentication process of the users admins or any other legitimate person in order to use a system, individuals must register their form of identity with the system by means of capturing raw biometric that can be used later on in the system. That was the enrollment process and it's composed of three distinct phases [4]:

- **Capture a physical or behavioral sample (raw biometric data)** is captured by the system during the initial enrollment phase.
- **Process/Feature extraction** a unique feature that represent and distinguish between individuals is extracted from the captured data (raw biometrics), this unique feature is denoted as *biometric template*.
- **Comparison** being stored in a suitable medium such as databases, text files or hard drives, the processed templates are compared with new collected samples during authentication process by using some special methodologies depending on the context.

Once enrolment process is done, the system can authenticate individuals by processing the stored templates. Authentication is the process whereby a fresh raw data is captured by the person who is authenticating with the system and compared to the registered (enrolled) processed templates. Bear in mind that there exists two forms of Authentication [5]:

- **Identification** performs the process of identifying an individual from their biometric features. It asks the question "*Who are you?*"
- **Verification** involves matching the captured biometric sample against the enrolled template that is stored and requires the user to assert a specific claim of identity. It asks the question "*Are you who you pretend to be?*"

The success of a system in performing verification is measured using the metrics below [6]. Successful systems will have high true positive (TP) and true negative (TN) values, a poor system will have high false positive (FP) and false negative (FN) values. Each metric have a detailed definition in this chapter (**Section 2.4**).

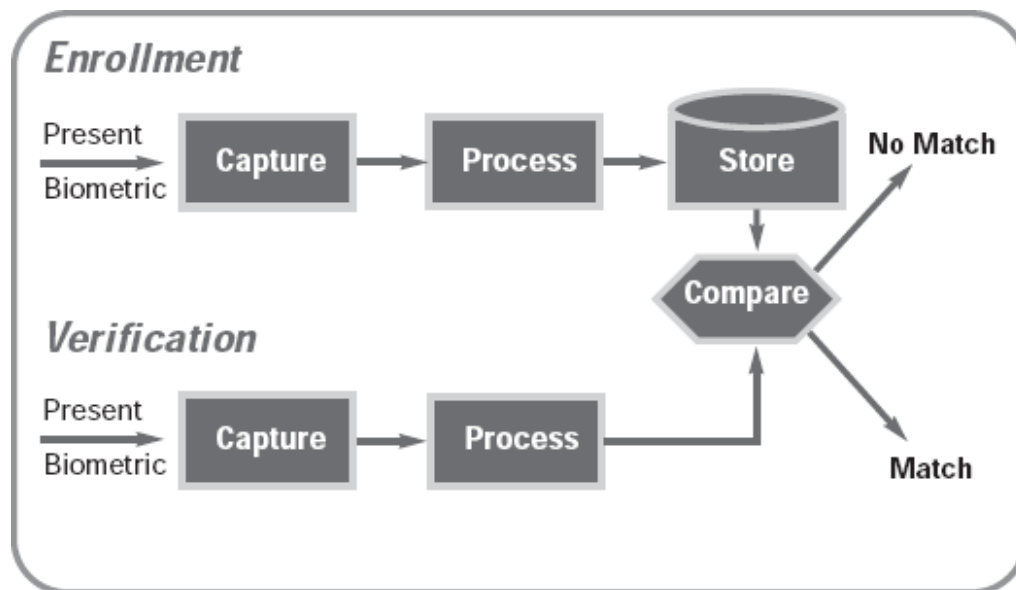


Fig 2.2: Diagram illustrating the process of enrollment and authentication according to [4]

1.3.Applications of biometrics

Most of the biometric applications [11] are related to security and are used extensively for military purposes and other government purposes. The applications in the public domain that are available to common people include:

- **Prison visitor systems**, where visitors to inmates are subject to verification procedures in order that identities may not be swapped during the visit - a familiar occurrence among prisons worldwide.
- **Driver's licenses**, whereby drivers are expected to have multiple licenses or swapped licenses among themselves when crossing state lines or national borders.
- **Canteen administration**, particularly on campus where subsidized meals are available to bona fide students, a system that was being heavily abused in some areas.
- **Benefit payment systems in America**, several states have saved significant amounts of money by implementing biometric verification procedures. The numbers of individuals claiming benefit has also dropped dramatically in the process, validating the systems as an effective deterrent against multiple claims.
- **Border control**, a notable example for this is the INSPASS trial in America where travelers were issued with a card enabling them to use the strategically based biometric terminals and bypass long immigration queues. There are other pilot systems operating elsewhere in this respect.
- **Voting systems**, where eligible politicians are required to verify their identity during a voting process. This is intended to stop 'proxy' voting where the vote may not go as expected.
- **Junior school areas**, where problems are experienced with children being either molested or kidnapped.
- In addition there are numerous applications in gold and diamond mines, bullion warehouses and bank vaults as well as the more commonplace physical access control applications in industry.

1.4.Advantages and disadvantages of biometrics

As any system biometrics have advantages and disadvantages, the following table is a recapitulation of the most important dis/advantages of biometrics [10]:

Advantages
+ Increase security by providing a convenient and low-cost additional tier of security
+ Reduce fraud by employing hard-to-forge technologies and materials
+ Eliminate problems caused by lost ID's / forgotten passcodes by using physiological attributes

<ul style="list-style-type: none"> + Reduce password administration costs + Replace hard-to-remember passwords which may be shared or observed. + Make it possible, automatically, to know WHO did WHAT, WHERE and WHEN! + Offer significant cost savings or increasing ROI in areas such as loss prevention or time & attendance
Disadvantages
<ul style="list-style-type: none"> - Fingerprint of those people working in chemical industries is often affected. - With age, when the person has flu or throat infection the voice of a person differs. - For people affected with diabetes, the eyes get affected resulting in differences. - In certain cases, biometrics may be an expensive security solution.

Table 2.1: Most important advantages and disadvantages of biometrics.

1.5. Common biometric techniques and forms

There are plenty of biometric techniques, in this section we will try to mention the most known ones but keep in mind that current and new techniques are researched and studied constantly. The most interested corporations are government, military and security industries. Below are some of the most common techniques and their main defining characteristics (adapted from [16-20]):

- **Fingerprint scanning** as each person on Earth has a unique set of fingerprints (even twins doesn't have the same set of fingerprints), it is one of the best biometrics, which is widely used in smartphones, airports and the new generation of computers. This technique has been used for centuries and its validity has been well-established. A fingerprint scanner system has two major tasks, obtain the raw image of a given fingerprint and determine whether the captured pattern matches with the processed templates (pre-scanned images).

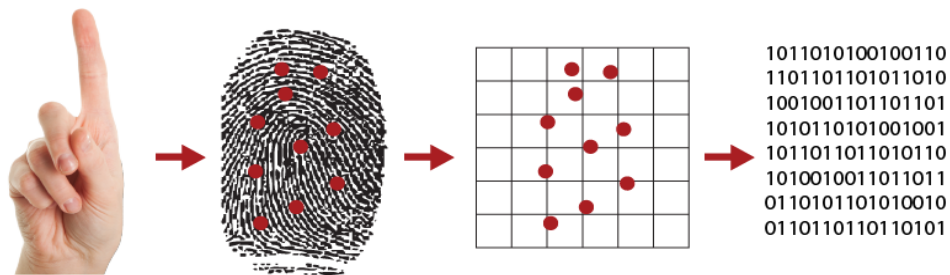


Fig 2.3: Fingerprint scanning system according to [1]

The red points (2nd image stating from the left of **Fig 2.3**) are a specific characteristics unique to every person's fingerprint. The fingerprint scanner filter and save an encrypted biometric key from the raw images. No image of a fingerprint is ever saved, only a series of numbers (a binary code) which is used for verification purposes [1]. It is virtually impossible to reverse the encrypted key and reconstruct a raw fingerprint image from it, so no one can forge your fingerprints.

- **Face recognition** used as access control in security systems, this technique focuses on recognizing the global positioning and shape of the eyes, eyebrows, nose, lips, ear, forehead and chin of the face of an individual. It is able to identify and verify a person from an image or a video. The face recognition system works by selecting facial features dispersed in the face of the person, those points or facial features are the most important points that can make differences between persons by simply calculating the distance between the points. Many studies affirmed that face recognition systems have a lower accuracy compared to iris and fingerprint recognition systems but still widely used due to its contactless and non-invasive process [7].

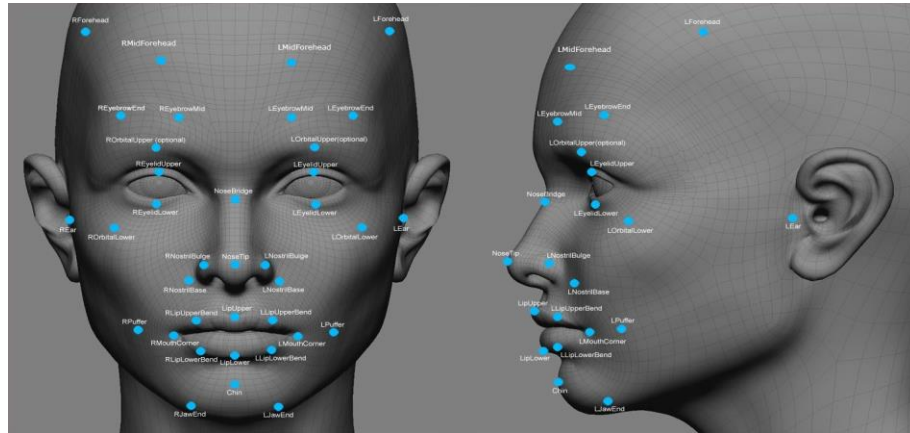


Fig 2.4: Facial features used in face recognition systems [2]

- **Iris scan** is the annular region of the eye bounded by the pupil and the sclera (white of the eye). It has a very high potential performance while matching between the iris and the processed data but it needs a very precise location of the person during the identification process, the user should be around 5-15 cm maximum from the camera that has the duty to check the identity of the identifiers.

The camera captures the center and the edge of the pupil, the edge of the iris, the eyelids and eyelashes, transform them into codes and then process and match them with the stored data.

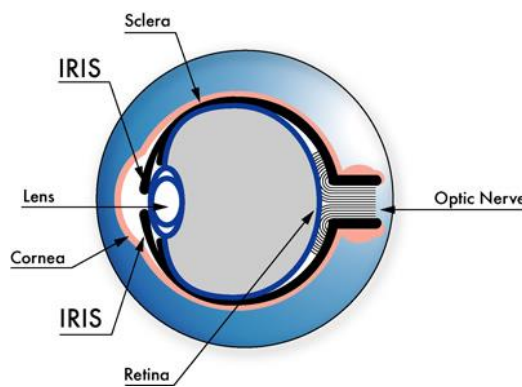


Fig 2.5: Iris [8]

- **Hand geometry** is based on the palm and fingers structure including their width and length and even the thickness of the palm area. Although these measurements are not very distinctive among people, hand geometry can be very useful for person authentication. It is widely accepted and the verification includes simple data processing.

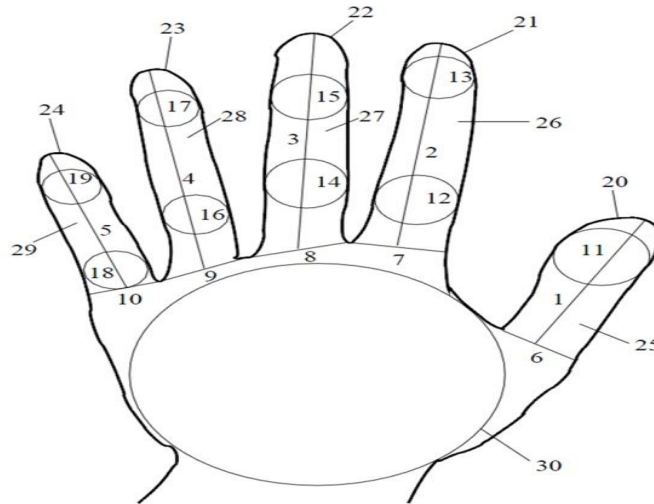


Fig 2.6: Hand geometry features according to [9]

- **Signature recognition** each person has his own handwriting style, it could be either captured by using a digital screen or by scanning the classical signature on a paper. Person signature may change over the time (simply using a new pen may change it) that's why the probability of having identical signatures is very low. 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. One of its other problematic is that any person can try to copy the way the signature is wrote. In fact many studies in criminology cited that there exist thousands of criminals that falsify papers by copying how the signature is written, yet some banks still use it but with extra biometric factors.

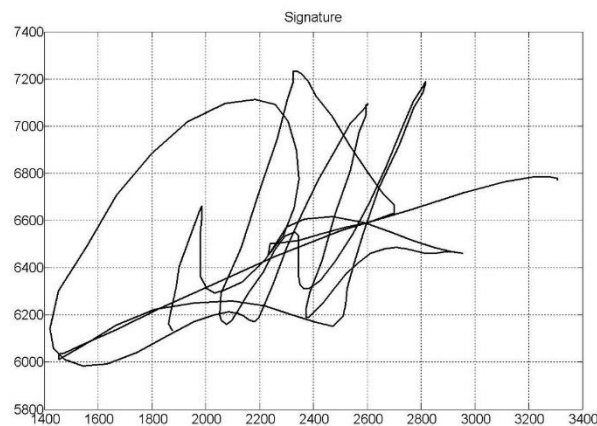


Fig 2.7: Example of signature shape [3]

- **DNA samples** each person has a DNA, a double helix structure made up of molecules, and even if individuals have a highly similar DNA, there still some partial portions that differs from a person to another expect for identical twins, and that portion can be used as a biometric technique which is a very common personal identification technique in criminology domain.
- **Speech / Voice** the unique patterns in our voices can be used as individual characteristics of human, analyzed and compared to recorded samples that represents voiceprints if it matches then it confirms your ID. This is already used to access some online banking services and automated customer service phone lines.

There exists other techniques but they are not really present in the user's daily life such as gait analysis that takes into account the way a person walks or the gestures, in fact not only the face, eyes, hand geometry and fingerprints that are considered as behavior biometrics, but there is other unique feature in the physical human gestures, some smart devices uses smile recognition, facial emotions or even hand movement in a special pattern in order to control and unlock devices or security checking systems.

1.6. Keystroke dynamics over other techniques

In the context where biometrics are fundamental in this study, that is the identification and authentication of users in an online system for fraud detection as a main application and domain of activity, keystroke dynamics is a suitable approach and could have good results regarding the economic costs, the transparency and an easy usability from the point of view of end users.

As the aim of this project is to be able to identify users and reduce the probability that an illegitimate user is accepted by the system, we should provide transparency to the end users (it means that even if the internal behavior of the system changes due to updates or fixing bugs the interface that the user interacts with should remain same).

Many literature surveys have argued that implementing a keystroke dynamics system is cheap, simple and trustworthy more than he looks like. Even though any biometric can change over time, typing patterns have smaller time scale for changes [108], and that's why it's considered to be as a none behavioral change. Based on the requirements of the biometric authentication systems criteria, the following table will definitely say why keystroke dynamics is a good approach for biometric systems:

Factors \ Biometrics	Highly accurate	Low total cost	Non-invasive	User-friendly	No behavioral change	No special sensor or device	Simple to deploy
Keystroke dynamics	X	X	X	X	X	X	X
Facial recognition			X	X			

Hand geometry	X			X			
Signature recognition	X		X	X			
Iris recognition	X						
Fingerprint	X						
Voiceprint		X	X	X			
DNA samples	X				X		

Table 2.2: Keystroke dynamics vs biometric techniques.

Authenticate a person by the way characters are typed in a keyboard is a good approach not only to capture person's unique rhythm of typing but to give systems and application a very high level of security, according to what I have seen from previous papers and articles, I can say that the advantages of using KSD are:

- KDS is the only soft-only biometric, it's because that it doesn't need any external devices or special sensors.
- It's non-invasive, user-friendly and very simple to deploy and manage in the point of view of a developer or a maintainer.
- Very easy to integrate with the existing systems and processes
- Highly accurate and have a good cost-performance ratio
- Another interesting and positive point related to research, is the possibility of accessing public keystroke databases [58-60].

2. Keystroke dynamics

During the last 40 years, computers have played a very important part of our lives, using them in daily activities has increase in parallel with the development of new technologies , as most of the companies and implemented systems must be secured, information and system security becomes all the most essential in the last decade. In fact each day computers are being used to access emails, banking transaction and even manages a huge amount of sensitive data. Thus, it is obvious that a compromised computer or account could cause a lot of damage, and not only to its owner but for all people that interact with the system. That is where the keystroke dynamics step in.

Researches have shown that each person has certain unique features which can be calculated through his keyboard typing rhythm [27]. The delay between each keystroke or the duration that he holds a certain key pressed while typing can distinguish him between other users. It is a behavioral biometric that can be used in many ways such as authentication, identification or even emotion detection, similarly to hand writing.

2.1.Introduction to keystroke dynamics

Keystroke dynamics (KSD) [21] is a behavioral biometric that uses the way of typing (called typing pattern) of a person on a keyboard , it can be either a password, username , credit card number or any free text field and allows the authentication of individuals through their unique

typing pattern. One of its advantages, the no necessity of adding any external devices other than the keyboard. This biometric modality also allows continuous authentication through time. [22, 23]

User authentication with KSD is generally done in real time (i.e., online) in a real world system. Scientists working on keystroke dynamics usually analyze the performance of their system by working in an offline context and using samples previously collected by other researchers, and stored in a benchmark dataset. A complete list of available keystroke dynamics datasets has been made in [24, 25]. As it can be seen, most of datasets have a limited amount of individuals and very restricted number of samples for each user. The collection of such datasets is very time consuming, this is the main reason why there is not more very large datasets like for the face modality as for example [26].

2.2. Keystroke dynamics origins

The use of KSD as a method for identifying individuals is not new, indeed the roots of KSD goes back to the telegraph era, individuals created at that time a unique patterns that can distinguish and identify each other by the way they tapped out Morse codes. This identification method, known as the “fist of the sender”, was also valuable as a verification/identification method during World War II. Figure 1 depicts a timeline that shows how the technology has evolved. [28]

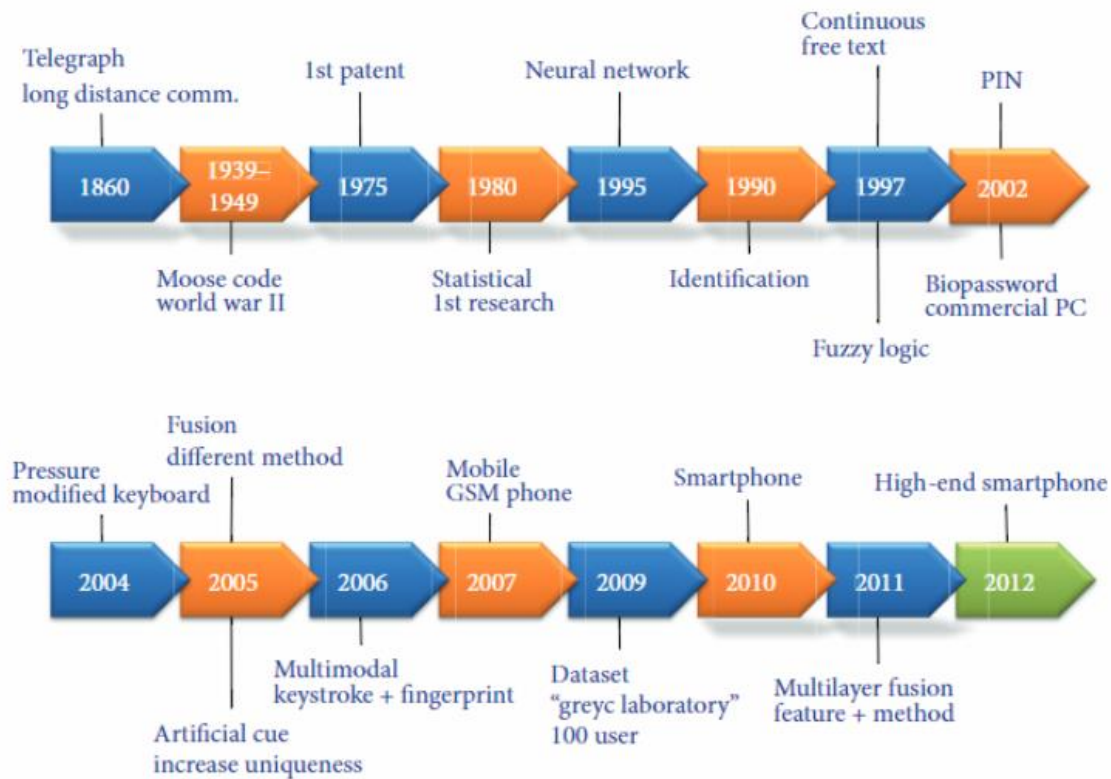


Fig 2.8: A general timeline on the overview of keystroke research work evolution. [28]

2.3. Keystroke dynamics system

Before going into detail about the application developed, the technology of KSD needs to be understood appropriately. In this section, we will talk about the main principal of KSD, how it is analyzed and what are the operational phases of KSD.

As we already have mentioned in the introduction, KSD is the technology of gathering a person's typing rhythm, transforming it into useful information, and using it in some way. The main principle behind this technology is that for each person there exist a unique features which can be calculated by his keyboard typing rhythm [29]. On one hand, those features are used to identify the real identity of a given user (person), on the other hand they distinguish between individuals, similarly to handwriting or the signature. It is a behavioral biometric that can be used in many ways such as to detect emotions, but most importantly, to authenticate or even identify a person.

In order to understand some of the features that a person's typing rhythm consists of, we display the delay between each keystroke and the duration of holding a certain pressed key by the following figure: [30]

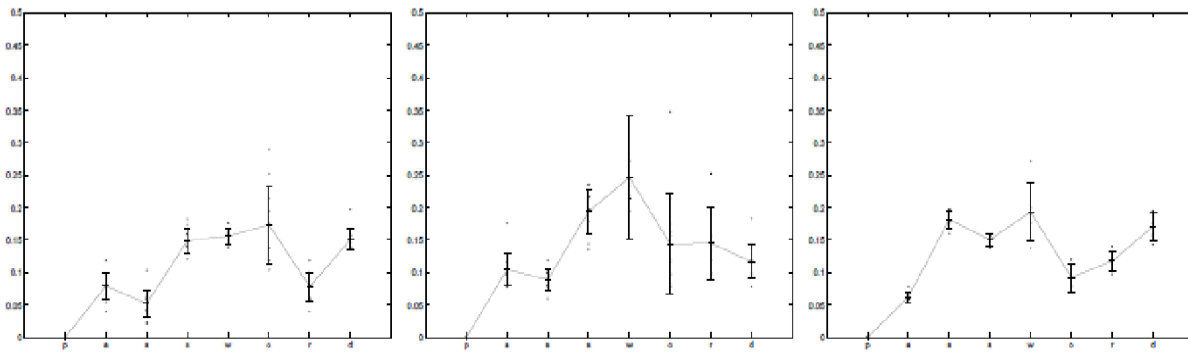


Fig 2.9: Latencies between keystrokes when writing the word “password” by three different people. [30]

Keystroke dynamics, just like all biometric systems, consist of three operational phases, the collection of the data, training it by extracting the features and finally the classification process, some people takes into account extra phases such as matching, decision and re-training and in fact they are very crucial phases, but let us regroup them inside the classification phase.

▪ Phase1: Data Acquisition

Data acquisition or raw data collection is the first phase of keystroke dynamics system (KSDS), it's a fundamental stage whereby raw keystroke data are collected via various input devices (keyboard, special purpose num-pad, cellular phone or a smart phone). It lasts long enough to gather the required amount of samples needed to efficiently train a pattern recognition model.

More specifically, in KSD data collection refers to the process of saving raw keystroke data which are processed and stored as reference template for future usages.

▪ **Phase 2: Feature extraction**

Feature extraction refers to keystroke events timings, such as the press and release time of a user. As the data in its raw format can't be very useful, transforming it into durations between key events is a must to ensure or to increase the quality of the data. The figure bellow shows two basic features used in keystroke dynamics.

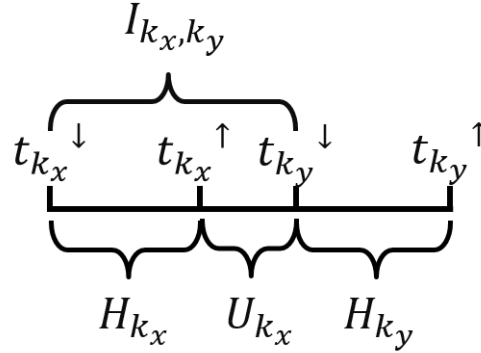


Fig 1.10: Keystroke features.

Pressing and releasing a keystroke pair (k_x, k_y) , results in 4 timings:

- Key-down time for $t_{k_x}^{\downarrow}$ and key-down time for $t_{k_y}^{\downarrow}$
- Key-up time for $t_{k_x}^{\uparrow}$ and key-up time for $t_{k_y}^{\uparrow}$

Four features are derived:

- Inter-stroke timing: $I_{k_x, k_y} = t_{k_y}^{\downarrow} - t_{k_x}^{\downarrow}$
- Holding time of k_x : $H_{k_x} = t_{k_x}^{\uparrow} - t_{k_x}^{\downarrow}$
- Holding time of k_y : $H_{k_y} = t_{k_y}^{\uparrow} - t_{k_y}^{\downarrow}$
- Up-down timing: $U_{k_x} = t_{k_y}^{\downarrow} - t_{k_x}^{\uparrow}$

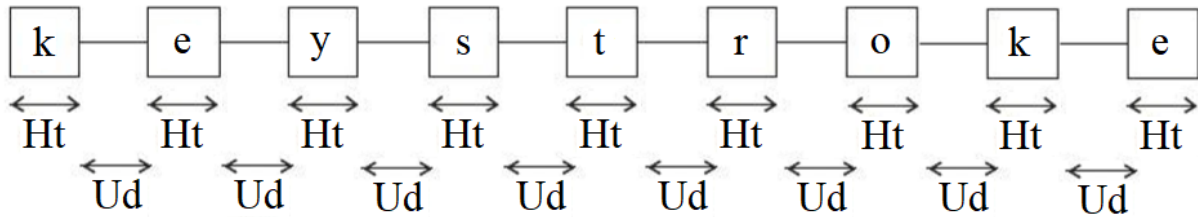


Fig 2.11: Some of the most important features of a keystroke sensitive password.

Other features might include the duration of the release of two keystrokes (up-up) etc.

According to some research papers, the three most important features that we should focus on while collecting the data are:

- **Type of event:** key-down (press down) or key-up (release).
- **Time:** Time in milliseconds (having a very small metric is preferred) when the key is pressed down or released.
- **Value:** what key is pressed.

▪ Phase 3: Classification

During the classification phase the newly inserted test data is compared with the stored data, indeed if we take the verification process that we have seen in the section of *biometric authentication*, a user will enter his password and from it the most important features will be extracted based on his captured typing pattern and compare it with the already stored samples, the classification process uses an anomaly score, if it is lower than the threshold calculated in the training phase, it gets accepted by the model. If not it is rejected, along with its user. The decision made by the model represents the internal matching-decision phase embedded inside the classification stage.

The “update” variable that is seen on the figure above refers to the ability of an algorithm to improve itself. An update mechanism for example, would replace a newly acquired sample which was accepted with the oldest in the training data, and retrain the model to create an updated pattern. That was the re-training phase which is another internal stage performed inside the classification process, indeed due to the variability of user typing pattern, it is therefore necessary to constantly renew the stored reference template to reflect the ongoing changes

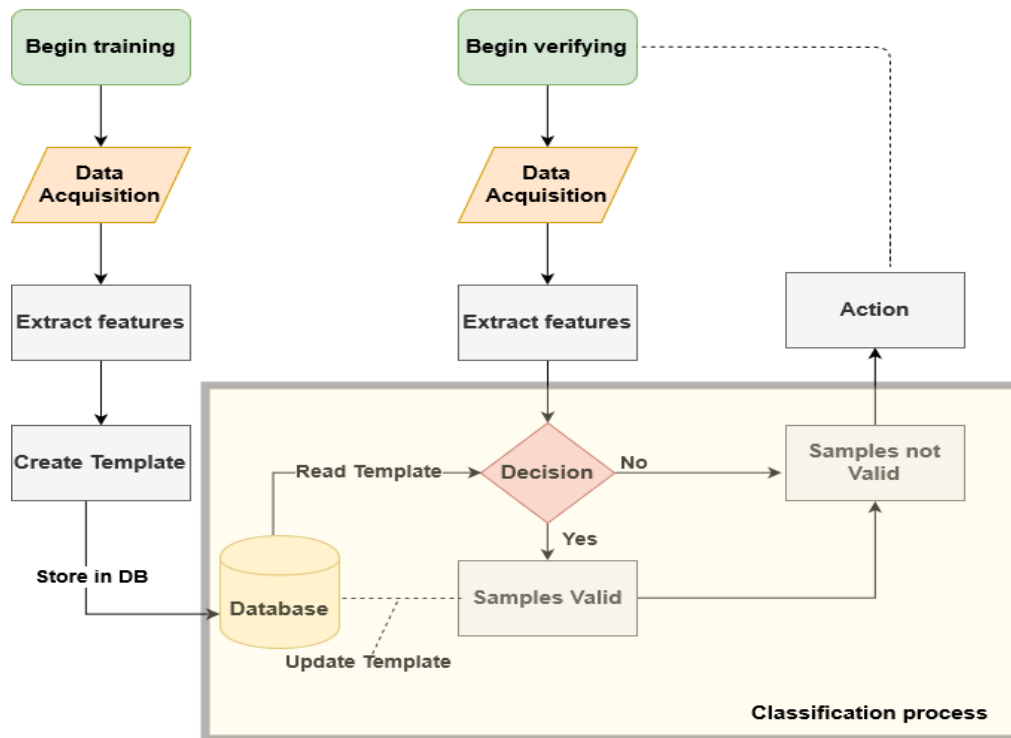


Fig 2.12: Typical biometric methodology.

2.4. Biometric evaluation

Evaluating the performances of a biometric system can be done by different measures, for the classification problems we usually use classification performance metrics such as accuracy. But we must take into account that there always exists statistical errors in recognition patterns, thus for any algorithm used in keystroke dynamic, that's why we must have a look at errors rates that help us to define the quality of the application of a keystroke dynamic system and its effectiveness in distinguishing people from each other.

We are able to calculate the accuracy and the efficiency of biometric system by the following metrics (types of errors in KSD):

▪ Accuracy

The accuracy of a given system has been measured using the percentage of effectiveness (how often a classifier gives the correct prediction). This value is defined as the fraction of the proportion of correctly identified (predicted) elements compared to the totality of predicted elements. Formally accuracy is defined as:

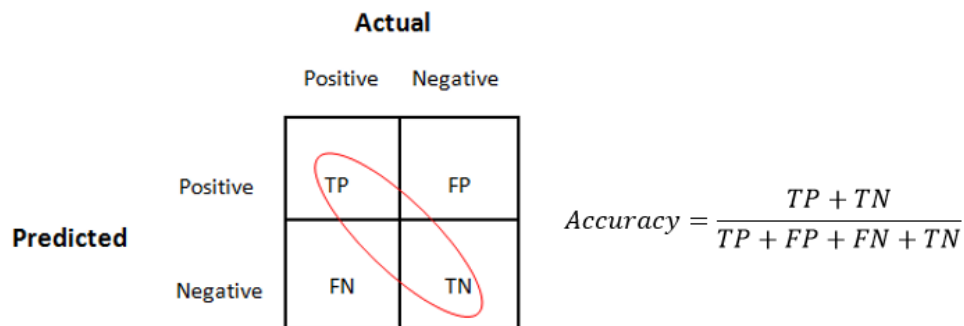


Fig 2.13: Accuracy [27]

- **True Positives (TP)** are the cases when the actual value was true and the predicted is also true.
- **True Negatives (TN)** are the cases when the value class was false and the predicted is also false.
- **False Positives (FP)** are the cases when the actual value was false and the predicted is true.
- **False Negatives (FN)** are the cases when the actual value of the data point was true and the predicted is false.

In most studies regarding biometrics the False Acceptance Rate (FAR) and False Rejection Rate (FRR) rates have been used extensively [49]. 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 probability for none legitimate users that are allowed to access the system, according to the literature it is always recommend to have a FAR value as low as possible.

$$FAR = \frac{FP}{FP + TN}$$

- **FRR** measures the probability for legitimate users that have not been given access to the system. Having a very low value guarantee that genuine users are able to access the system.

$$FRR = \frac{FN}{TP + FN}$$

- **EER** the Equal Error Rate is determined as the value where the FAR and FRR values are equal. The lower the EER value the better the classification is.

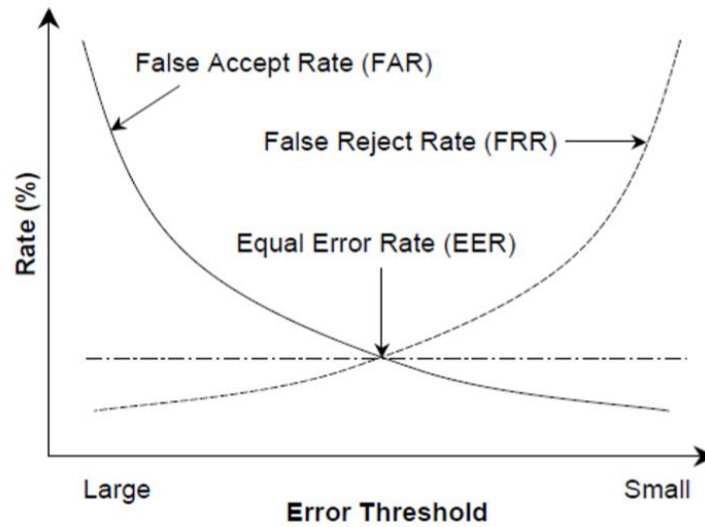


Fig 2.14: Relationship between FAR, FRR, and EER. [32]

2.5. Classification Techniques

Several techniques that concerns the way to classify the typing pattern of a user while writing on a keyboard exist nowadays. These include, among many others, statistical, distance-based or machine learning techniques. In this section we will try to overview most of the widely known techniques used during research and KSD biometric systems developments.

▪ Statistical techniques

The statistical methods used for KSD concerned the mean, the median and the standard deviation and gave excellent results According to [33-37]. Nowadays, these techniques are still widely discussed, improved and implemented. Methods related to statistics and probabilities are also used to classify keystrokes. Clustering methods, like k-means and fuzzy c-means, have also been used. These, of course, are not the only techniques that have been used in the field of statistics. Many other approaches have been performed and ended with a good results and had an appropriate accuracy to the biometric systems.

▪ Distance-based techniques

In this section we will try to cover the most important distance-based techniques according to the literature, the following detectors described below are used to analyze a sequence of digits that represents a credit card number (CCN) timing. [39]

We decided to use the CCN to feed our proposed model (keystroke dynamics-based identification system) for two major reasons, verifying users identity by checking how they types the CCN and in order to stay around fraud detection application domain, keep in mind that each detector have a stored genuine patterns which represents a legitimate user and based on those stored patterns, a user typing behavior model is built in, and of course each new pattern will be assigned an anomaly score during to testing phase. [39]

a. Euclidean distance

This classic anomaly-detection algorithm [38] models each CCN as a point in p-dimensional space, p is the number of features in the timing vectors. The training data can be represented as a cloud of points, in this case the anomaly score is based on how much a given test vector is closer to the center of the cloud. Specifically, in the training phase, the mean vector of the set of timing vectors is calculated. In the test phase, the anomaly score is calculated as the squared Euclidean distance between the test vector and the mean vector. The Euclidean distance is defined as:

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

b. Manhattan distance

This classic anomaly-detection algorithm [38] is close enough to the Euclidean detector except that the distance measure is not Euclidean distance, but Manhattan named also city-block distance. The only difference is how the anomaly score is calculated, indeed the anomaly score for Manhattan detector is calculated as the Manhattan distance between the mean vector and the test vector. The Manhattan distance is defined as:

$$d_{Manhattan}(X, Y) = \sum_{i=1}^n |X_i - Y_i|$$

c. Mahalanobis distance

This is another classical anomaly-detection algorithm [38], it have the same fundamentals of the Euclidean and Manhattan detectors but the distance measure is more complex. Mahalanobis distance can be viewed as an extension of Euclidean distance to account for correlations between features. In the training phase, the covariance matrix of the timing vectors are calculated also with the mean vector. In the test phase, the anomaly score is calculated as the

Mahalanobis distance between the mean vector and the test vector (i.e., if X is the mean vector, Y is the test vector, and S is the covariance matrix, the Mahalanobis distance is defined as:

$$d_{Mahalanobis}(X, Y) = \sqrt{(X - Y)^T S^{-1} (X - Y)}$$

2.6. Keystroke dynamics applications

Applied in many domain of applications, the keystroke is considered to be a very challenging research area, this section is dedicated to the most common applications where the use of KSD can be relevant.

- **Authentication**

Users resources can be protected while accessing a system, indeed adding a KSDS to the authentication process of users in addition of the classical login-password system (i.e. C2 security level) will guarantee to have a more secure system that reduces the margin for an unauthorized user, because if that user have login ID and password combination, he will have a complete access to the computer system in a transparent manner

- **Emotion detection**

According to some research papers, some studies related to emotion detection have proven that using KSD, it is possible to detect some emotions [52, 53].

- **Password complexity**

Robust authentication is actually possible. In fact encryption system combines a message sent by a user with his private key to create a short digital signature on the message which make the message very hard to decrypt and sometimes physically impossible due to the complexity of the algorithm used during the authentication, the same case can be done to the users that have access to any system by adding a KSD signature to the password to protect highly sensible resources. Examples where this has been studied and applied can be found in [54-56].

- **Student ID examination control**

Checking the identity of students by studying those typing patterns during e-exams (such as the BEAD system and canvas used at our university) is possible by using KSD, it could be used to detect abrupt changes on the template of a user and conclude that another user is taking the exam. Apart from [50], another article that comments on the use of KSD to control remote users when doing exams is found in [51].

- **Employee monitoring systems**

Usually in companies specially in call centers, the employees shares the same offices and works by shift times, it means that they switch places between each other, so to check when the employees switched places, if or not they arrived at time, constantly monitoring them inputs can determine at which moment the user has been supplanted by another [50, 57].

- **Identification**

Giving another try for users that has not been allowed to access the system during the authentication process by providing an interface that checks the identity of the user, like a forgot password link, the identification process is matching user typing pattern against stored templates in the database (see **Fig. 2.12**)

- **Verification**

Another advantage of using KSD is to have the possibility to make a real time biometric system, it means that the system will continuously have a look at users while using a computer system by constantly checking their way of typing against a template.

- **Online fraud detection**

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 KSD when surfing the internet to prevent crime.

3. Anomaly detection

In this chapter (**Section 2.5** titled classification techniques) we have considered some techniques used to classify users typing patterns such as statistical methods (based on the mean, the median and the standard deviation), but also about anomaly detection techniques based on similarity/distance measures (Euclidean, Manhattan and Mahalanobis distances), unsupervised learning methods (K-means clustering, Fuzzy logic and neural networks) and supervised learning techniques (SVM one-class and one nearest neighbor), each from the above cited techniques was described as an anomaly detection technique used as a detector that helps to classify users patterns (legitimate user / impostor) based on the anomaly score. Therefore in this section will go deeper into anomaly detection and discuss about the fundamental concepts of anomaly detection.

3.1.Introduction to anomaly detection

Any pattern or value which is out of the normal behavior is considered as an outlier, an anomaly. The non-conformity of some values with the values that we are waiting for are often referred to as anomalies, outliers, discordant observations or exceptions and it vary from an application domain to another one. The most used terms in research papers regarding the context of anomaly detection are anomalies and outliers. Anomaly detection finds extensive use in a wide variety of applications such as fraud detection for credit cards, insurance or health care, intrusion detection for cyber-security, fault detection in safety critical systems, and military surveillance for enemy activities.

Due to its importance in many domain applications, many techniques have been developed starting from the late 19th century, some of them are dedicated to a specific application domain while others are more generic.

Formally, anomalies are patterns in data that do not conform to a well-defined notion of normal behavior [61], so if we have a set of data points and we want to classify them based on some similarities between the points (for example regroup points that are very close to each other into

the same class), here the role of anomaly detection is to detect data points in data that does not fit well with the rest of the data. **Fig. 2.15** illustrates anomalies in a simple 2-dimensional data set.

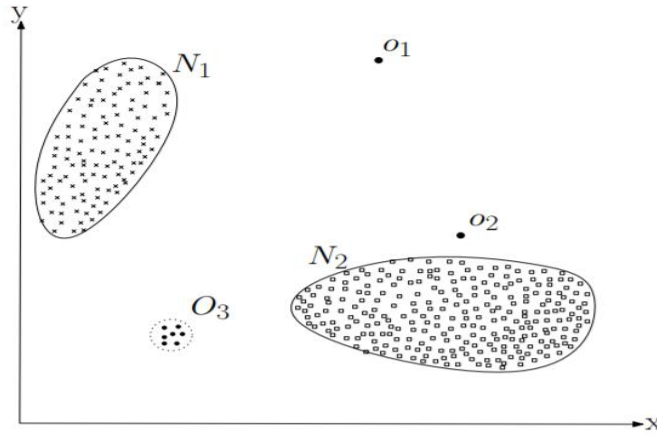


Fig 2.15: A simple example of anomalies in a 2-dimensional data set. [61]

The data has two normal regions, N_1 and N_2 , most of observations belongs to one of these two regions but not all of them. Indeed, there are points far away from the regions and doesn't belong to anyone of them, e.g., points o_1 and o_2 , and points in region O_3 , are anomalies.

3.2.Types of anomalies

According to the literature, we can distinguish three types of anomalies which are point anomalies, sequential anomalies and contextual anomalies and will give a brief description and an example for each type in the following sub-sections given below:

- **Point anomalies**

If a single point deviates from the considered normal pattern it is referred to as a point anomaly. This is the simplest form of an anomaly and is the most researched form [61].

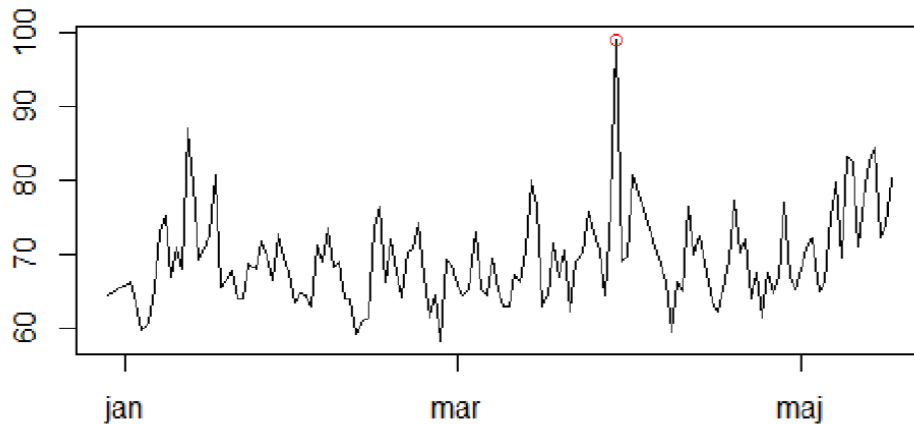


Fig 2.16: Illustration of a time series concerning logged sensor data. [62]

As you can see in this example of a point anomaly, scanning the values shows that there is a peak around the first days of April where the value suddenly changed from a very low value (65) to a very high value (almost 100). This illustrates a point anomaly marked with the red color in the figure.

▪ Sequential anomalies

If a sequence or collection of points is anomalous with respect to the rest of the data, but not the points themselves, it is referred to as a sequential or collective anomaly [63]. Since this thesis deals with anomalies in time series we will refer to this type of anomaly as a sequential anomaly but instead of using a sub-sequence where an anomaly may appear we will consider the entire sequence (time series recorded data).

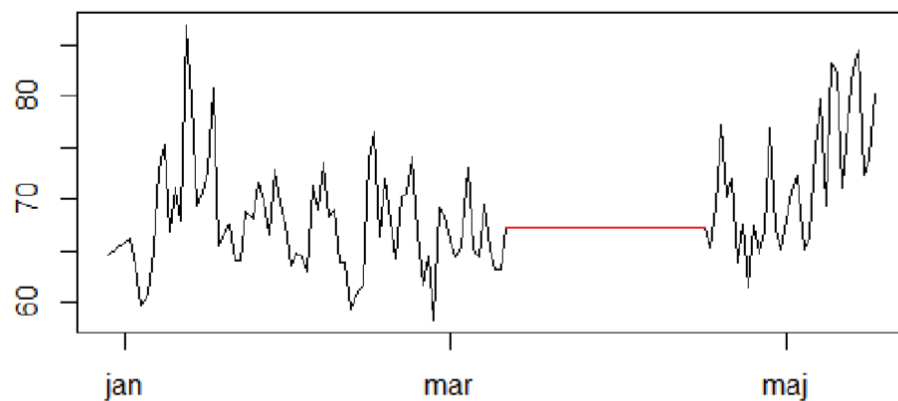


Fig 2.17: Sequential anomaly in a time series of logged sensor data. [62]

The picture shows the recorded values for a DH-11 sensor (temperature sensor) during more than six sequential months, the values are in Fahrenheit, during the recording session we had a failing in recording values, you can see that there is a red line in the figure and the value didn't change for more than one month which is abnormal, it should at least vary a little bit even if the temperature is quite regular, here is the sequential anomaly. Indeed these values are not considered anomalous themselves, but the sequence of them is.

▪ Contextual anomalies

If a point or a sequence of points are considered as an anomaly with respect to its local neighborhood, but not otherwise, it is referred to as a contextual anomaly [61].

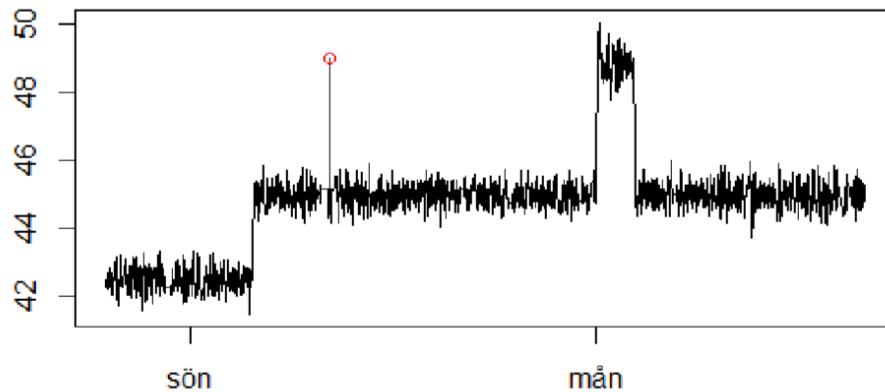


Fig 2.18: Example of a contextual anomaly (marked in red) in a time series. [62]

The abnormality is context specific. This type of anomaly is common in time-series data. *For example we are in the middle of the month and usually we spend \$20 on food every day, but then we have a peak where it shouldn't, it can be buying a new car, healing some diseases or even having an expected important dinner with important persons, in this case it's considered as a contextual anomaly, the difference between contextual and point anomaly is that in point anomaly, the abnormality is where we have a single instance of data which is anomalous because it's too far off from the rest. While contextual anomaly appears if a data instance is anomalous in a specific context, but not otherwise, it means that we can expect it at a certain season or period of the year but not otherwise.*

3.3.Learning methodologies in anomaly detection

Giving the priority to a specific anomaly detection techniques over other depends on what types of data are we dealing with, usually the data are in two disjoint forms some of them are labelled, other are unlabeled. Labelled data have labels, classes or categories associated with each data point which gives information if the instance is normal or abnormal but for unlabeled data instances there is no such information. We have talked before that about anomaly detection techniques that can be used to solve KSD problem what we will do in the training phase and how we can extract the anomaly score but we didn't mention what methodologies are we able to use because we didn't take into account the type of data that we will deal with it. In fact according to the type of data we can have three distinct learning approaches for the training phase:

- **Supervised learning**

When applying supervised learning the system is fed with labelled data on which the algorithm defines what is normal or not. In other words, we have both input and output variables and we try to train our algorithm (extract the mapping function from the input to the output), after that we can classify or predict any new data by applying the built model.

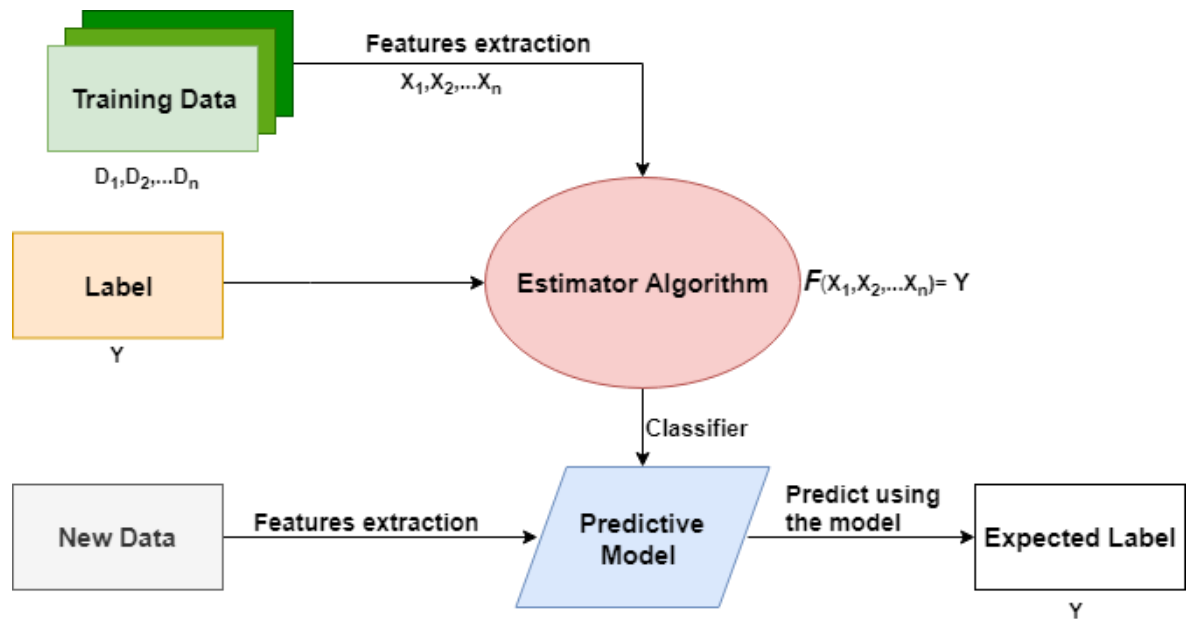


Fig 2.19: Supervised machine learning process.

There exists two types of supervised learning, firstly the regression, where the output value Y should be a continuous value (the value can be a real number or a continuous value but not a category or class). The second one is classification where we regroup items into categories or classes, its goal is to predict the target class that a given data belongs to, the output should be in discrete terms that is it should be either yes(1) or no(0). In some cases the options may increase to more than two. I will use the classification as a supervised learning during my thesis work, because we want to classify the typing patterns of users either by yes/no that a typing patterns belongs to a legitimate user or not.

▪ Unsupervised learning

In unsupervised learning, the input data is unlabeled and the system tries to learn structure from that data automatically, without any human guidance. Anomaly detection, such as flagging unusual credit card transactions to prevent fraud, is an example of unsupervised learning. [110]

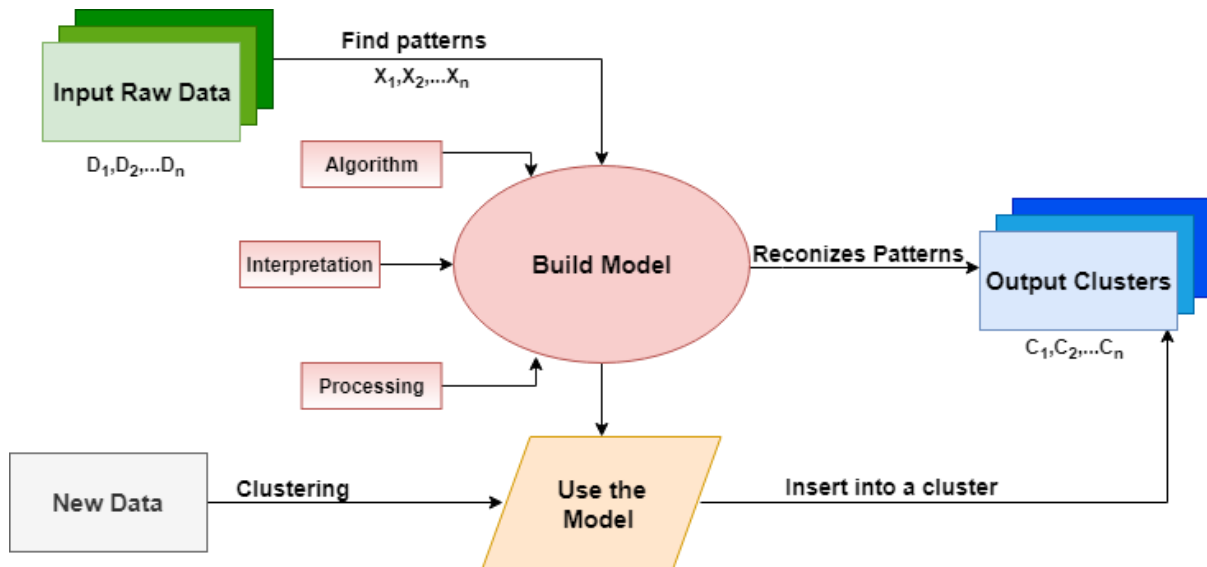


Fig 2.20: Unsupervised machine learning process (clustering).

As a simple example of unsupervised learning is regrouping students based on their grades, we can easily divide them into three groups based on the grade for example (from 2.0 up to 2.99 as acceptable, from 3.00 to 3.99 as intermediate and from 4.00 and up as outstanding).

In this case it's a very simple example that we could even use supervised learning, so to make it a little bit complicated let us have grades of all subjects for each students, in this way we must process and interpret the grades based on the average and coefficients and then extract the output clusters, now if an incoming students come and we want to put him in the appropriate cluster, all what we have to do is to use the build model and insert him into an already existing cluster.

▪ Semi-supervised learning

Semi-supervised learning is often a combination of the first two approaches. That is, the system trains on partially labeled input data—usually a lot of unlabeled data and a little bit of labeled data. Facial recognition in photo services from Facebook and Google are real-world applications of this approach. [110]

Other learning techniques exist (e.g. reinforcement learning [64]), however, as they are not directly relevant for my work, they are not considered in this overview.

In conclusion, the learning technique that can be used depends on the available data [63]. If labelled data is available then supervised learning is the most suitable. If labelled data is missing then we can use unsupervised learning as an alternative and learn more about our data so that we can extract a useful information from it based on the input values, otherwise using the novel approach that doesn't require models but an experience instead of it then reinforcement learning [64] is the solution.

3.4. Anomaly detection techniques

Before discussing the different techniques used in anomaly detection, it is important to describe the modes under which those techniques can operate. The supervised mode operates under the assumption that both normal and anomalous instances are labeled. The anomaly detection technique then trains a model using the labeled data to setup a normal class and an anomalous one. New instances are then tested with that predictive model, and are assigned to one of those classes. When only normal instances can be labeled, semi-supervised techniques are more efficient. This is more applicable than the supervised as the critical requirement of having labeled anomalous instances, which is hard to satisfy, is not necessary.

When neither normal nor anomalous classes can be labeled, the unsupervised mode is the only one to use. Knowing that this is the case of most real-world datasets, it is understandable that unsupervised mode is the most widely used and the most popular among the three modes. The following are the most known anomaly detection techniques:

a. Nearest-neighbor

This detector was described by Cho et al. [40]. During the training phase, the detector saves the list of training vectors, and calculates the covariance matrix. And while the test phase, the detector calculates the distance between each of the training vectors and the test vector. The anomaly score is calculated as the distance from the test vector to the nearest training vector. We will use this detector during our experiment described in the next chapter.

b. Fuzzy-logic

This detector was described by Haider et al. [41]. It uses a fuzzy-logic inference procedure. The main idea of this approach is that ranges of typing times are assigned to fuzzy. The sets are called fuzzy because elements can partially belong to a set. During the training phase, the detector determines how strongly each feature belongs to each set, and each feature is matched with the set in which its membership is strongest while during the test phase, each timing feature is checked to see if it belongs to the same set as the training data. The anomaly score is calculated as the average lack of membership across all test vector timing features.

c. SVM (one-class)

This detector was described by Yu and Cho [42]. It represents an algorithm called a support-vector machine (SVM) that projects two classes of data into a high dimensional space and finds a linear separator between the two classes. A “one-class” SVM variant was developed for anomaly detection and instead of projecting two classes of data into a high dimensional space it projects the data from a single class and finds a separator between the projection and the origin (0, 0).

During the training phase, the detector builds a one-class SVM using the training vectors while in the test phase, the test vector is projected into the same high-dimensional space and the signed distance from the linear separator is calculated. The anomaly score is calculated as this distance, with the sign inverted, so that positive scores are separated from the data.

d. K-means

This detector was described by Kang et al. [43]. It uses the k-means clustering algorithm to identify clusters in the training vectors, and determines whether the test vector is close to any of the clusters according to its distance from the centroid. In the training phase, the detector simply runs the k-means algorithm on the training data (with $k=2$), two clusters one for normal and the other one for abnormal. The algorithm produces two centroids such that each training vector should be close to at least one of the three centroids. In the test phase, the anomaly score is calculated as the Euclidean distance between the test vector and the nearest of these centroids.

▪ Other techniques

Distance based techniques (described in **Section 2.5**) can also be considered as an anomaly detection technique along with the neural-network detector, which was described by Choet al. [44], who called it an “auto-associative, multilayer perceptron.” The author specified that structure of his detector was designed specially to be used as an anomaly predictor [45].

There is another technique which is not really specific to KSD or even biometrics but have been extremely used in many studies and even in our specific area, this technique is called fusion. As its name refers to, it’s a combination of techniques in order to achieve a better overall results. If we consider what we are going to do is nothing but just to fuse nearest neighbor algorithms as a classifier or anomaly detector with a more general form of the Euclidian distance (DTW - we will discuss it in the next chapter).

Combining anomaly detectors to determine a new value to accept or reject a new sample is possible but we should be aware of what kind of methodology we should use to combine the results, some researcher use a voting method while other tries to sum or get the max value. There are many other different possibilities, though. Some of these have been studied in relation to Keystroke Dynamics in [46-48].

Below are some examples of such fusion techniques:

- Average rule : $S_f = \frac{S_1 + S_2}{2}$
- Product rule: $S_f = \frac{S_1 \cdot S_2}{2}$
- Weighted sum rule: $S_f = W_1 S_1 + W_2 S_2$
- 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: $valid = \begin{cases} 1 & S_1 > thr, S_2 > thr \\ 0 & otherwise \end{cases}$

By combining results from anomaly detectors we can improve the overall classification, for that favoring the best results of each classifier by using fusion techniques listed above is the key.

Indeed 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 by raising up both advantages and minimizing both weaknesses.

3.5. Fraud-anomaly detection relationship

In this chapter (**Section 2.6**) we mentioned that one of the KSD application domain could be online fraud detection, in online systems we can always capture information that concerns users and use them to either improve the user experience or to identify them while login or accessing the system.

The following figure will show that for each domain application of anomaly detection (AD) such as fraud detection we have to use a specific AD technique depending on the problem characteristics

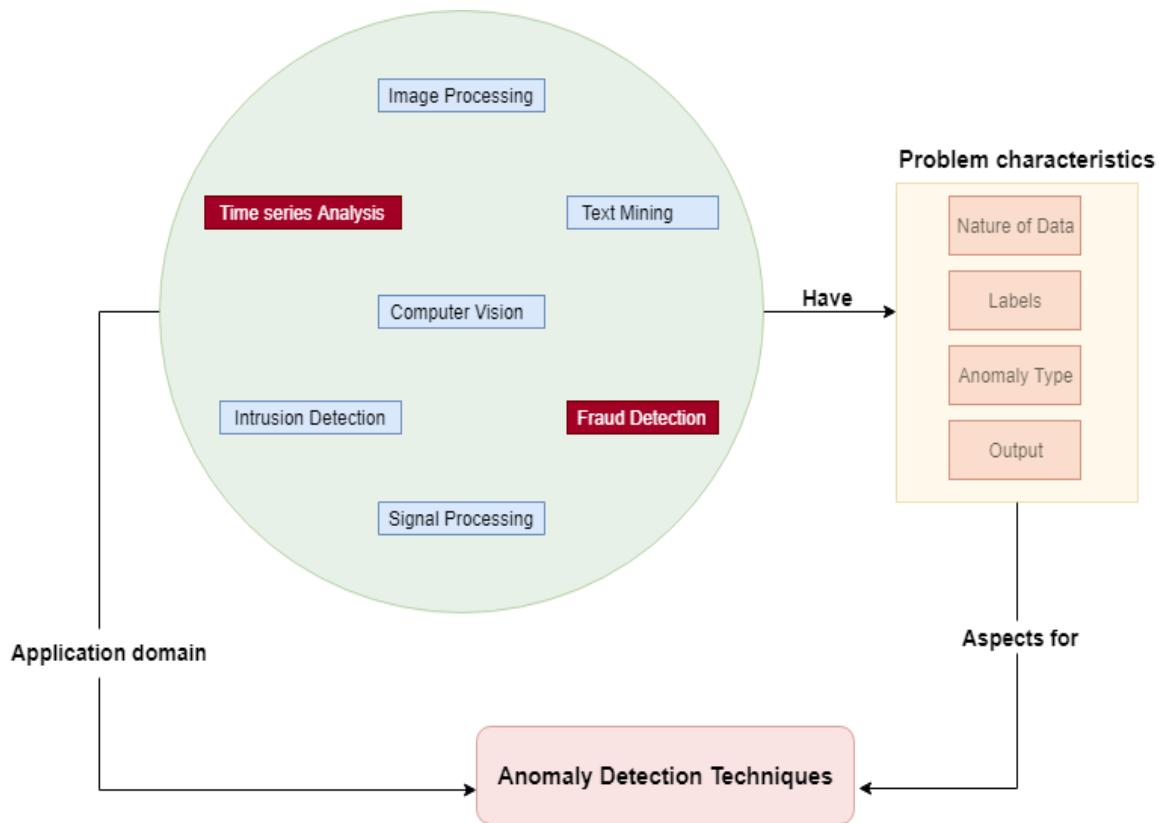


Fig 2.21: Problem solving using anomaly detection techniques.

From **Fig. 2.21** we can conclude that fraud detection belongs to the application domains that can use AD techniques to solve a problem related to a specific area, based on the characteristics of the problem or AD aspects we can have a better vision on which AD technique we can use to solve that problem.

- **Nature of input data**

The input data could be univariate (single variable), multivariate (multiple variables) and the nature of attributes could be binary, categorical, continuous or hybrid, if we take our project as an example, the input data is a multivariate as we have the event which can be of three types (key-down, key-up or key-press), the time in milliseconds that shows at what time exactly the event happened and we also may use the key-code of each typed character to check whether two text are similar or not, concerning the attributes nature they are categorical, continuous and continuous respectively.

Till now we didn't mention how our data is graphically represented, to give a hint check the second red rectangle on the **Fig 2.21**, you may notice it, it's called times series analysis even if it's totally an independent application domain, we will use times series analysis notion to study our data and apply the AD technique which is a person identification classification of the users patterns based on the graphical representation of that data which is nothing but just a time series.

We will talk about more details concerning times series on the next section.

- **Labels**

To make it easy labels are just used to say if an input is normal or abnormal, in our future model the labels represent the persons with their CCN, we will explain in details in next chapter.

- **Anomaly type**

During our study about anomaly detection we have extracted three distinct type of anomalies, point anomalies, contextual anomalies and sequential anomalies and as we already mentioned, this thesis deals with anomalies in time series that's why we will choose sequential anomaly as a type.

- **Output of anomaly detection**

After the execution of any algorithm we get results as an output, here it's the same after using AD techniques we will get an output which is represented by an anomaly score, depending on the score test instances are given a normal (legitimate user) or anomaly (fake or impostor user) label. This is especially true of classification-based approaches.

4. Time series analysis

We are almost ready for the thesis project implementation, all what we need to know is how using the graphical representation of the data will be beneficial for the classification task of users typing patterns. This section gives a small introduction to time series where we will get to know only the notions needed for my thesis work, we will not focus on specific techniques for time series analysis, but we will have a look at what kind of tasks we can perform when we deal with time series data, we will also see the existing time series distance measures and introduce which distance measure I will use in the context of person identification based on keystroke dynamics.

4.1. Definition of time series

“The time series are an ordered sequence of values of a variable most of the times sampled or can have an attribute at an increasing spaced time intervals”, according to the literature there is another definition where they args that time series is nothing but just a statistical modeling of time-ordered data observations.

A time series could be univariate where we refers to a time series that consists of a single observation recorded while the time is moving, or a multivariate time series which is used when one wants to model and explain the interactions and co-movements among a group of time series variables.

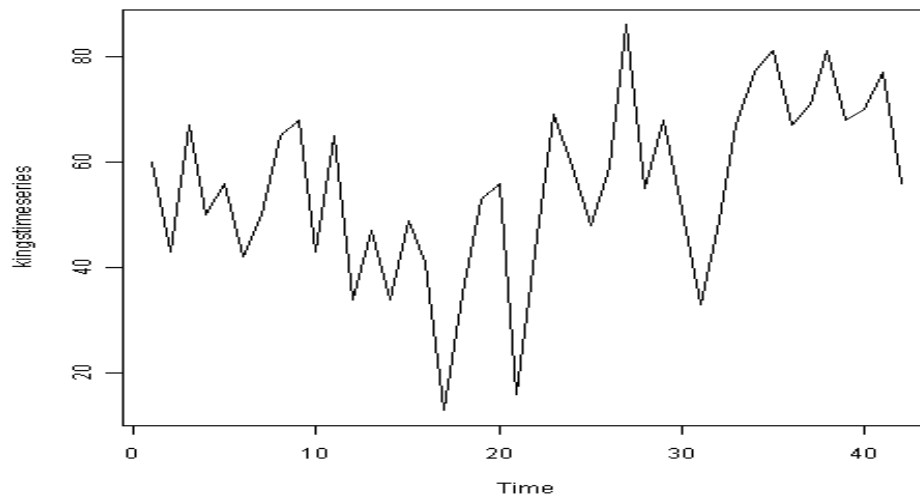


Fig 2.22: Time series of the age of death of 42 successive kings of England. [65]

4.2. Time series data mining tasks

Although statisticians have worked with time series for more than a century, many of their techniques hold little utility for researchers working with massive time series databases (for reasons discussed below). Below are the major task considered by the time series data mining community. [66]

- **Indexing (query by content):**

Given a query time series Q , and some similarity/dissimilarity measure $\text{Dist}(Q, C)$, find the most similar time series in database DB (*Chakrabarti et al., 2002; Faloutsos et al., 1994; Kahveci and Singh, 2001; Popivanov et al., 2002*).

- **Clustering:**

Find natural groupings of the time series in database DB under some similarity/dissimilarity measure $\text{Dist}(Q, C)$ (*Aach and Church, 2001; Debregeas and Hebrail, 1998; Kalpakis et al., 2001; Keogh and Pazzani, 1998*).

- **Classification**

Given an unlabeled time series Q , assign it to one of two or more predefined classes (*Geurts, 2001; Keogh and Pazzani, 1998*).

- **Prediction (forecasting):**

Given a time series Q containing n data points, predict the value at time $n + 1$. [66]

- **Summarization**

Given a time series Q containing n data points where n is an extremely large number, create a (possibly graphic) approximation of Q which retains its essential features but fits on a single page, computer screen, etc. (*Indyk et al., 2000; Wijk and Selow, 1999*).

- **Anomaly detection**

Given a time series Q , assumed to be normal, and an unannotated time series R , find all sections of R which contain anomalies or “surprising/interesting/unexpected” occurrences (*Guralnik and Srivastava, 1999; Keogh et al., 2002; Shahabi et al., 2000*).

- **Segmentation**

- Given a time series Q containing n data points, construct a model Q' , from K piecewise segments ($K \ll n$), such that Q' closely approximates Q (*Keogh and Pazzani, 1998*).
- Given a time series Q , partition it into K internally homogenous sections (also known as change detection (*Guralnik and Srivastava, 1999*)).

Many approaches related to classification, prediction, summarization, and anomaly detection use a distance measure in an implicit way while others such as indexing and clustering make an explicit use of a distance measure. We will therefore take the time to consider time series similarity in detail. [66]

4.3. Time series similarity measures

Many time series learning and data mining tasks need to use a similarity measure in order to quantify the degree of the dissimilarity or similarity between time series. Ding et. al [67] suggested that using different similarity measures is good to have a better view over the relationship between two time series but also claimed that similarity measures do not only differ in the way they calculate the similarity but also that while using them we are able to extract different aspects of similarity if they are applied to the same problem. That's why it's very important to select the appropriate measure that best captures the relevant information to better solve the addressed problem. Furthermore, the most desired and well known properties used as selection factors for similarity measure are: [67]

- Robustness to noise, outlier, temporal and spatial distortions.
- Yielding low computational complexity.
- Not implying user parameters.

In recent years, and due to the growing interest in using time series that have particular characteristics compared with traditionally used data, a large number of similarity measures were proposed. Most prominent ones include:

- **Euclidean distance**

A simple and effective distance that implies no user parameter, but is not robust against noise and different forms of distortions.

- **Dynamic Time Warping (DTW)**

DTW is more robust against temporal distortions but is computationally expensive (faster variants, such as [71], were proposed). In my thesis work i will use it as a distance measure along with the anomaly classifier (1-NN) in order to calculate the anomaly score based on the distance between two time series that represents two typing patterns related to one or many users. [68, 70, 72, 74]

- **Longest Common Sub Sequence (LCSS)**

LCSS is robust against noise and outlier but implies to set a threshold in order to assess the similarity (this parameter should be set with care since it defines the similarity between data). [69, 78, 79, 75, 82]

- **Threshold Query Execution for Large Sets of Time Series (TQuEST)**

TQuEST measures the similarity after coding time series, but provides good results only over some specific data sets. [72]

- **Spatial Assembling Distance (SpADe)**

SpADe based on feature extraction, this measure is robust against the temporal and spatial distortions, noise and outlier but is difficult to scale up. [73]

- **Other distances**

Some distances such as edit distance with real penalty (ERP) [76], edit distance on real sequence (EDR) [80] and extended edit distance (EED) [77] extend the edit distance (ED) in order to deal with different natures of applications. Many other distances and similarity measures (see [81] for a well-structured review on time series similarity measures) have been recently proposed in the literature and deals with time series learning problems.

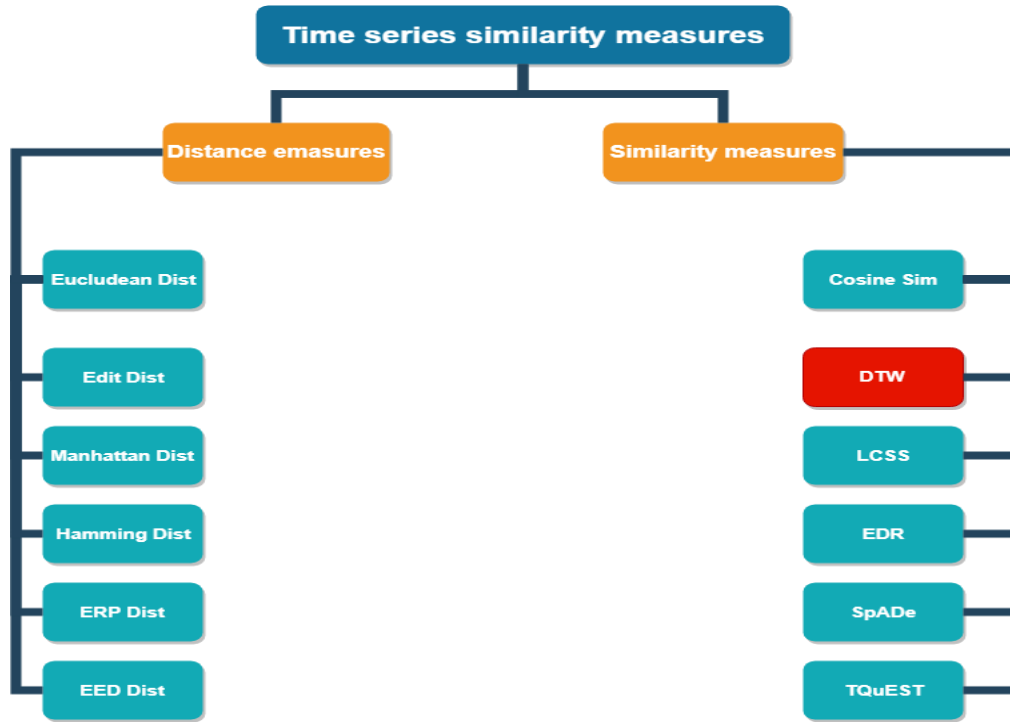


Fig 2.23: A possible taxonomy of time series similarity measures.

In time series classification, the combination of 1-Nearest-Neighbor (1NN) classifier with Dynamic Time Warping (DTW) distance has been shown to achieve high accuracy. However, if naively implemented, the 1NN-DTW approach is computationally demanding, because 1NN classification usually involves a great number of quadratic DTW calculations. We will discuss later on how to speed up the classification process and whether or not if we can improve the accuracy of 1NN-DTW approach as we already agreed to use DTW (marked in red color in **Fig. 2.23**) as a time series distance measure simply because it is considered as the most accurate measure for time series across a huge variety of domains.

5. Related works

Person identification (a.k.a. user authentication) is probably one of the most important applications of keystroke dynamics. A review of some local and web application developed either for research or commercial use will be held.

5.1. Implementations

Authentication with keystroke dynamics can be done by training some algorithm with the manner and rhythm in which an individual types characters on a keyboard or keypad, we name it the typing pattern of a person and omit all samples that do not meet certain anomaly criteria. In this section, some of the most famous implementations are reviewed and separated into categories depending on their functionality, local or web, and the scope of their development, academic or commercial. An academic approach recommend that the research has been done by some kind of institute or

university, while a commercial product on the other hand suggests an industry or company implementation with profit as their main objective.

5.2. Local authentication

Local keystroke dynamic authentication can be used in desktop application that doesn't require a connection to the internet, the program is installed locally in a computer, a station or a vehicle, etc. Those local programs have their own data storage and the computations are running in the background of the shown interface of the program after the user tries to login to a specific system.

▪ Academic approach

Authentication through keystroke dynamics has mostly been achieved with the help of pattern recognition systems, with the most common of them listed below.

- Statistical models [33-38].
- Neural networks [44].
- Fuzzy logic [41]
- Support-vector machines [42]

▪ Commercial products

Due of being closed sources, there is not much knowledge about the commercial products of keystroke dynamic authentication. Below is a list referring to some known products, with some of the information known about their implementations.

- **TypeWATCH**, released by Watchful software [85], free text typing patterns software.
- **Intensity analytics** [86], uses statistical weights and measures
- **BioTracker**, released by Pluriloc [87], also tracks mouse movements.
- **KeyTrac** [88], analyzes any text input in the background.

5.3. Web authentication

Web keystroke dynamic authentication refers to the authentication process by using websites or web-apps, data storage and processing can be either on a remote server that can be reached only while using internet connection/mobile data or locally through an installed program as the local keystroke dynamic authentication do.

▪ Academic approach

Sadly, there have not been as many researches in web authentication as in local authentication. Some of them will be listed below, depending on their pattern recognition approach.

- Statistical models [33-38]
- Neural networks [44]
- **Commercial products**

Just like in local authentication, there is limited knowledge on the mechanics of commercial products and patents. Below is a list referring to some of them, with some of the information provided about their mechanisms.

- **Trustable passwords**, released by iMagic Software [89], is used for both web authentication and large-scale enterprise authentication.
- **bioChec** [90], uses keystroke dynamics for ubiquitous web-based login.
- **behavioSec** [91], includes keystroke, mouse and environment dynamics

Chapter 3

Web-based K.S.D person verification system mechanism

In this chapter, the developed web-based prototypical system that uses keystroke dynamics for verifying the user identities is introduced, we named it *TyPaVeS (Typing Pattern Verification System)*, along with the reasons behind its creation and a the methodology that we propose.

1. Methodology

The main objective of TyPaVeS is the verification of users identity by checking whether they are really who they pretends to be for that, we need a client-server communication. As we are dealing with fraud detection (especially for credit card transactions) let us imagine that our methodology could be applied to a web shop for example.

Imagine the following scenario, i am a new user of a given web shop, i create my new profile with my payment details (credit card number, name, year of expiration... etc.), and save my username and password (add it to remember in a browser), then i let my laptop open and after few hours i get a notification that my transaction is fulfilled while I didn't order anything. In this case anyone who have access to my account either by hacking it or by simply using my stored details in my laptop can steal my money this is where TyPaVeS come to the rescue. Indeed with TyPaVeS even if you have all the users details your chance to pretend that you are that person is very low.

TyPaVeS is a system and one of its features is the extraction of user typing patterns by using a script, and based on user typing patterns we will be able either authenticate or verify user's identity.

Whenever a login is attempted, after the user get verified by the authentication server, a request is send to the web-service to determine if the signed in user is a legitimate user or not by using keystroke dynamic verification process.

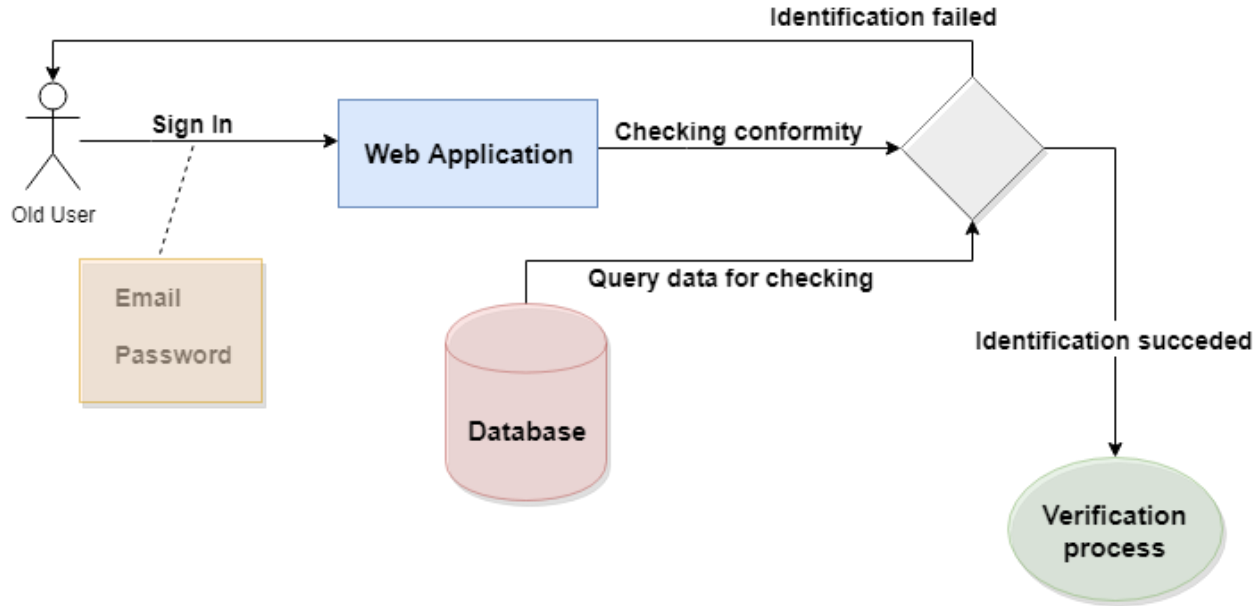


Fig 3.1: Overview of verification process based on our proposed TyPaVeS.

If the user doesn't have an account yet he must sign up to the system, while the new user is creating his new account by typing his details, not only the content of the fields will be registered but also them typing patterns (the timings of keys pressed is saved with its type of event and character code itself).

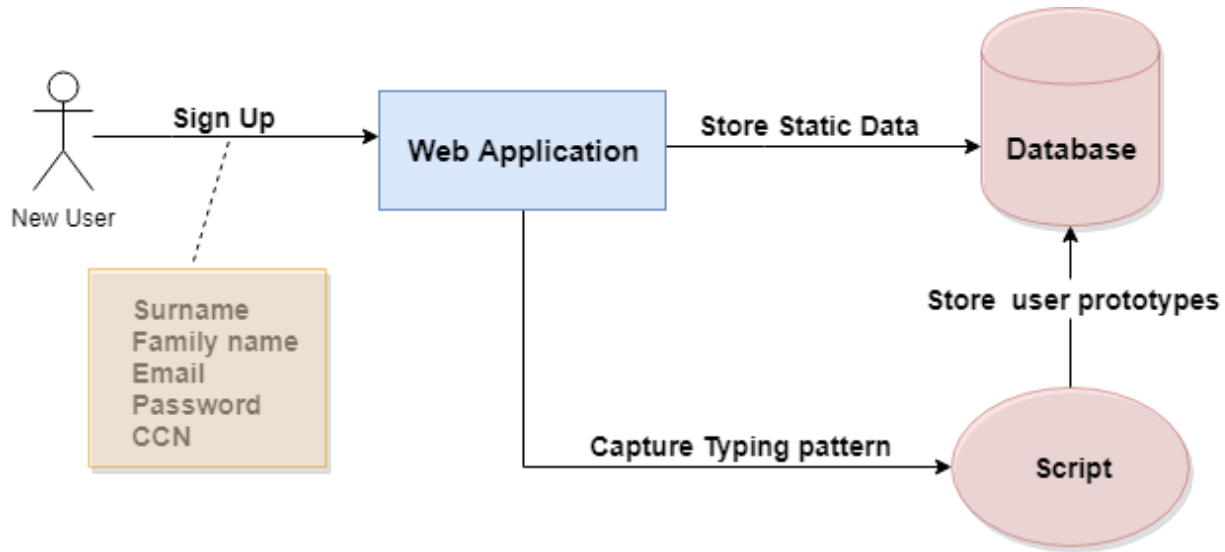


Fig 3.2: Overview of enrollment process based on our proposed TyPaVeS.

The values of the typing pattern in them raw version have no use at their current form, which is why the server after a set a calculations generates the down-down (also called between KS or Inter-stroke timing) and down-up (also called KS duration or up-down timing) durations. Two patterns (user prototypes) are created from the metrics above, they will be used during the classification mechanism which is embedded in the verification process that we have seen (**Fig. 3.1** above), and

each keystroke sample received is compared to those patterns. We will have more details in the next chapter.

2. Client-server communication

A client-server communication (CSC) is defined as “*The process of establishing a connection between a client and a server*”. As any web based application the CSC is a must, in our proposed model and to make it simple we have a CSC embedded on the basic known CSC. In fact while a user try to login to a given system, a request is sent to the authentication server and after checking the user details, the server have the choice to validate/ reject user request. That was the simple CSC widely used. As the communication between the script that capture the typing patterns and the web-server (application server) that decides whether a given user is legitimate or not is an internal process, a part of the mechanism and also based on the client-server model, we will use CSC also for that purpose. In this case the web-site that runs the script will acts as the client while the web-service as the server as shown below.

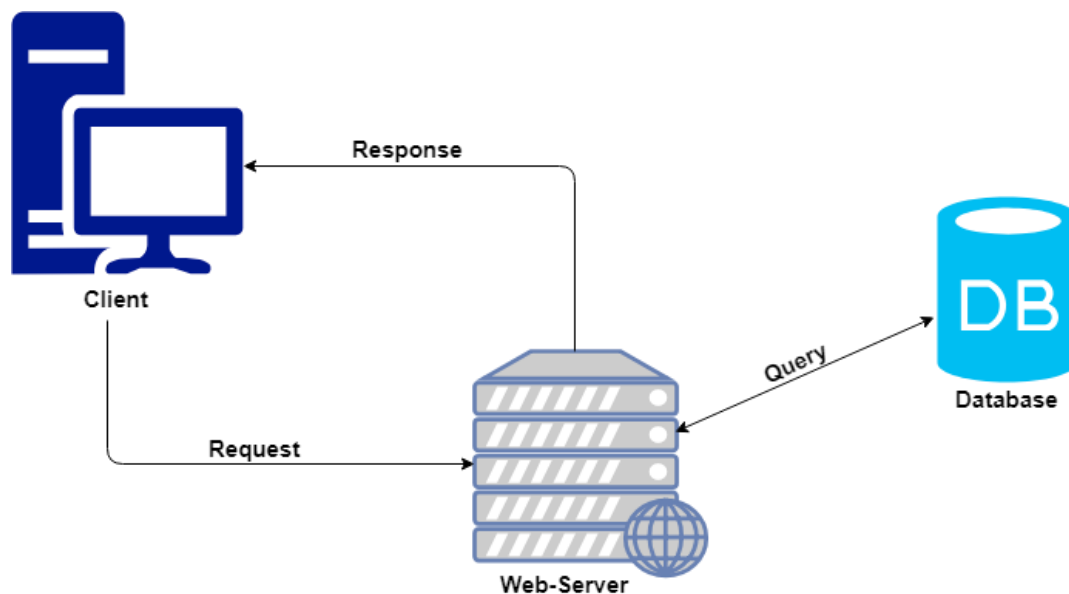


Fig 3.3: Client-server Communication.

There are two basic requests that are prompted in order for the keystroke dynamic verification to work. The first one determines if the user and password are legitimate, by contacting the web-site’s authentication server (i.e. C2 security level). If the user gets verified, then the client perform some tasks on the web-app (if we consider a shop web app then the client select items) and while the payment procedure a request which contain the keystroke timings of the user CCN is send to a web-service which will handle them accordingly to the anomaly score calculated during the verification process. The following figure shows each communication in a chronological order:

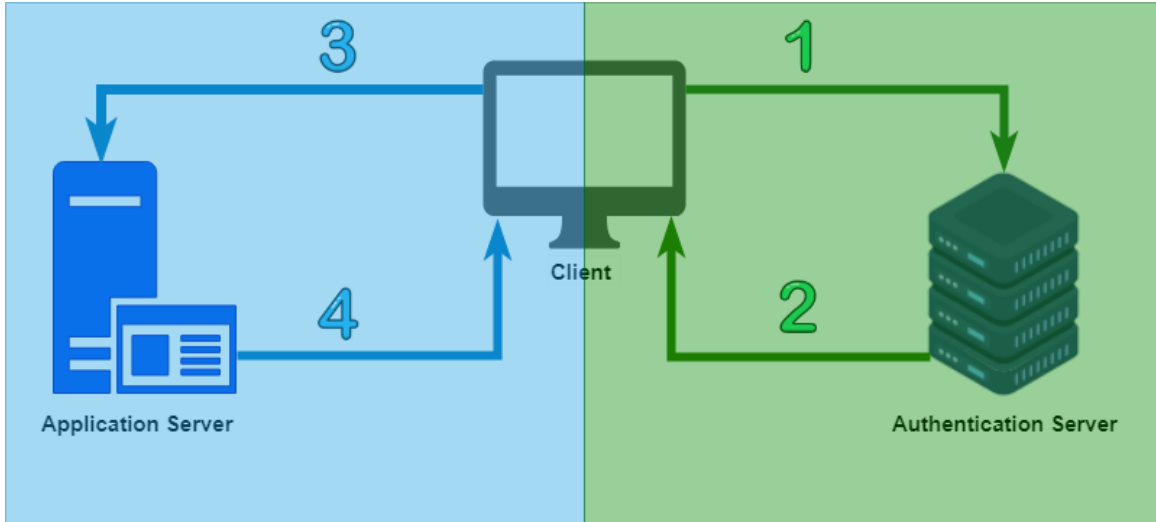


Fig 3.4: Detailed client-server communication.

The chronological order differs as the user may be a subject of two cases:

- **New user**

While a new user attempt to access the web-app, his personal data which will be subject of future authentication processes will be sent to the authentication server in the meantime the script that started working just after the loading of the web page content starts to capture user's personal data typing patterns and sends them to the application server when the user submit his request (we can say that both step 1 and 3 provided in the figure shown in the top of the page are executed in parallel), that was the client and the script (which acts as a client also) requests sent to the servers.

Concerning the servers, they will start the processing of the received requests, the authentication server (AuS) will check the validity of the user inputs (by using a simple lexical analyzer that will checks only the format of the entered text into the fields) after that, the AuS will register user's data into the database as a static data (see **Fig. 3.2**) in a secured way (hashing the password) and finally he sends the authentication response to the client (step 2 in **Fig. 3.4**) and redirect him to the login page. In parallel the application server (AppS) has started the preprocessing of the typing patterns by extracting only useful features (KS duration also called down-up duration and between KS known as down-down duration) , those features will represent the prototypes of the user typing patterns (see **Fig. 3.2**).

In **Fig. 3.2** we mentioned that the script will store user prototypes into the database which not shown explicitly in **Fig. 3.4**, in fact after the end of typing patterns processing made by the AppS, he will send a response to the script containing the prototypes (step 4 in **Fig. 3.4**), the script send a request to the AuS to store the prototypes and then Aus store them into the database and sends back an ACK (Acknowledgement). Therefore there is an interconnection between the servers (transitive relation).

▪ Enrolled user

An already registered user tries to use the web shop by simply typing his username and password at the login page, at this stage the client (user) sends an Authentication request to web-site server (step 1). The AuS will handle it by matching between the provided data by the client and what there already exists in the database, the provided username should already exists into the database otherwise the authentication will be denied directly, concerning the password a hash function will check the equality between the provided password and the stored password (as shown in **Fig. 3.5** below) and then a session will be given to the client (step 2) where the prototypes (let us call the genuine prototypes) of the user are temporary stored in the client mailbox (hidden from the user and accessible only for verification process of the application server). It consists of some variables that are included inside client script.

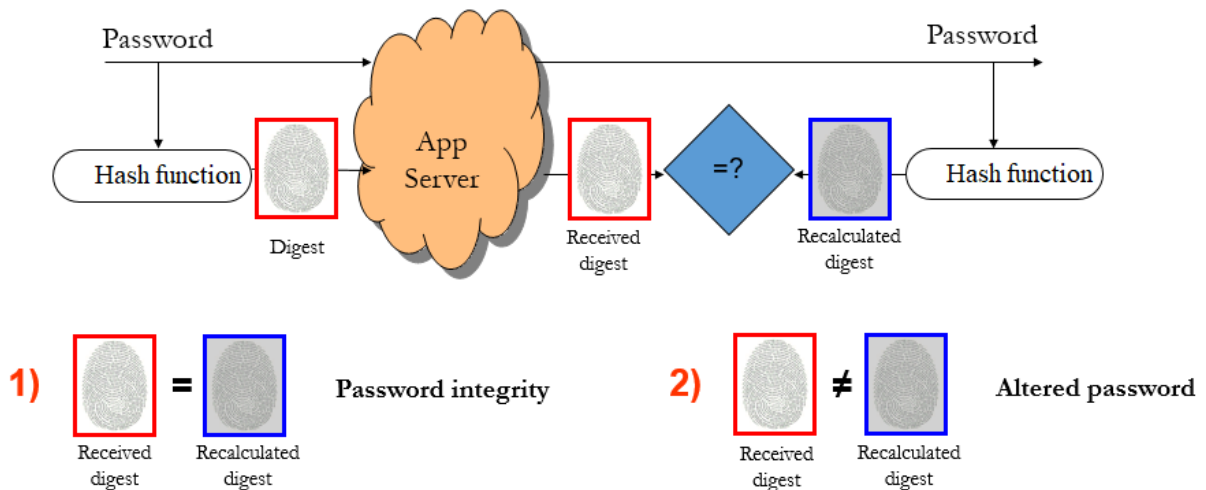


Fig 3.5: Password equality checking by using a hash function.

Just to explain briefly, while a new user sign up we register the hash code (Digest) of his password and not the password itself. During the identification process the typed password is translated to a digest by using a hash function, and then it checks the equality between the digests as shown above. Hash functions are useful because they are irreversible functions.

Now that a user has the access to the web shop and after adding some items to his cart, it's time to perform the payment (step 3). At this stage the client is asked by the AppS to type again his payment details such as credit card number, in the meantime AppS sends a request to the script in order to start capturing payment typing patterns (step 4), after the end of that process the raw data is sent back to AppS, where again the extraction of KS duration and Between KS are needed to get a potentially genuine user prototypes, AppS have now both prototypes all seems to be ready to start the verification process also called classification process.

3. Classification process

In order to distinguish between legitimate and illegitimate users we will use a classification algorithm (1-nearest neighbor) based on a distance model which will be the general form of Euclidian distance, the so called Dynamic Time Warping (DTW). The classification uses 1NN-DTW as a classifier (fusion between both techniques), usually with 1NN we compare the newly inserted typing pattern with many patterns of the training set, each one belonging to various classes, but in our case each new pattern (user pattern) is compared only with a single pattern (genuine prototype) that belong to the training set and we do not have many classes but a single class for each genuine user and those classes contains a unique typing prototypes (each user is represented by two prototypes) It calculate the anomaly score after checking the similarity/distance between stored prototypes of the legitimate user and the prototypes that belongs to the candidate user (the user that pretends to be a legitimate user), a threshold is then given based on some training phase on which its value is statistically fixed (we decided to put a constant threshold value based on our own training of the model) to serve as the point at which future users will be accepted or rejected. If the total anomaly score of the distance between both prototypes for the candidate user and genuine user prototypes are greater than the threshold, the sample and thus the user, will be denied access. The distances between the prototypes are represented as anomaly scores. The way to check user identity is explained below.

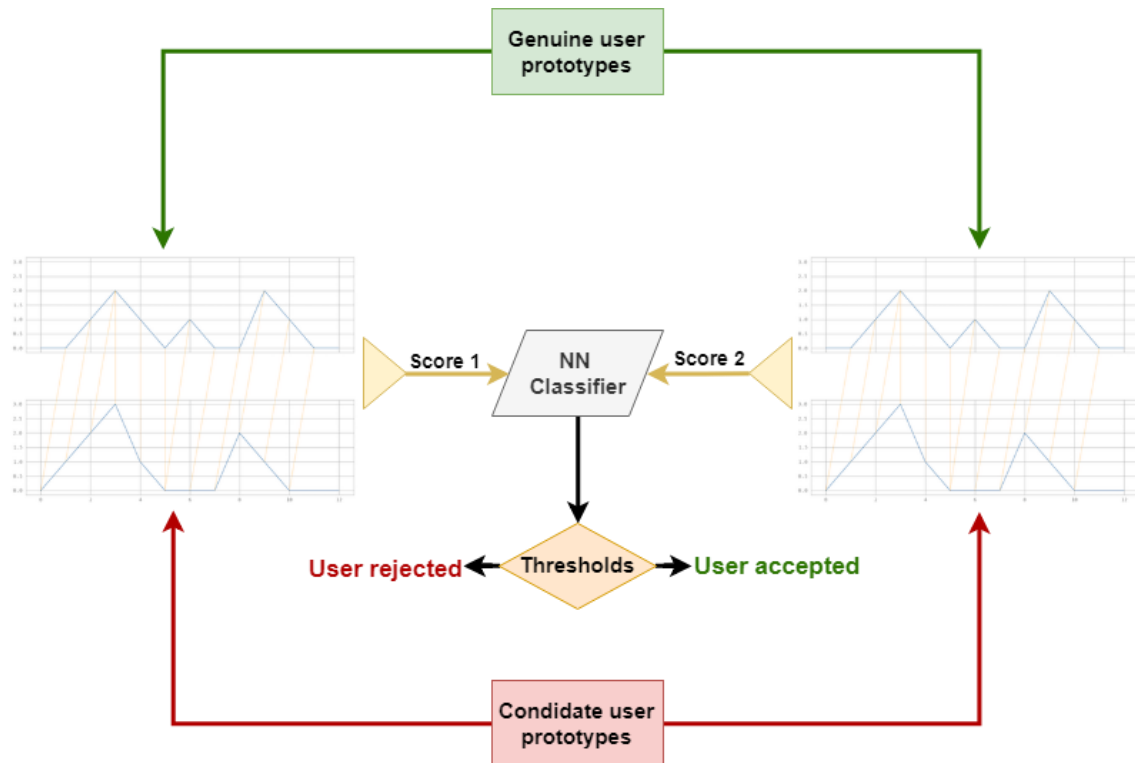


Fig 3.6: Classification process between pairwise of users typing prototypes.

The two signals for both users are represented as a keystroke time series (explained in next section) for both prototypes (between KS and KS duration), the classifier NN uses the anomaly scores

which are the output of the DTW algorithm and based on the thresholds he classifies the user as a legitimate or illegitimate.

4. Measuring the similarity of keystroke time series

According to the paper named “*Keyboard Usage Authentication Using Time Series Analysis*”, keystroke time series \mathbf{K}_{ts} are defined as the following:

▪ **Definition**

\mathbf{K}_{ts} is an ordered discrete sequence of points P ; $\mathbf{K}_{ts} = [P_1, P_2, \dots, P_i, \dots, P_M]$ where $M \in \mathbb{N}$ is the length of series and P_i is a tuple corresponding pairs of multi-dimensional features. [83]

A point tuple P_i in \mathbf{K}_{ts} consists of two instances $\langle t, k \rangle$ where t is the indexing sequence of time stamp (KN) in which keys are pressed and k is a set of timing attributes and descriptive features including: flight time (F^t) we used the down-down notion for it during our thesis (between ks), key-hold (KH^t) which correspond to the down-up (ks duration) and key code (K_{code}). So each p_i can be formally written as $p_i = \langle t_i, k_i \rangle$ where: [83]

- $\forall p_i \in \mathbf{K}_{ts}: p_i \leftarrow \langle t_i, k_i \rangle$
- $\forall t_i \wedge k_i \in p_i: t_i = KN; k_i = \{F_i^t, KH_i^t, K_{code_i}\}$

In our thesis work we used two prototypes, it means that for both flight time and the key-hold durations we have a separate keystroke time series and not a multi-dimensional features embedded on a single keystroke time series. [83]

The similarity can be computed between two or more series. Given two keystroke time series $\mathbf{K}_{ts1} = \{P_1, P_2, \dots, P_i, \dots, P_M\}$ and $\mathbf{K}_{ts2} = \{P_1, P_2, \dots, P_i, \dots, P_N\}$, where M and N are the length of \mathbf{K}_{ts1} and \mathbf{K}_{ts2} respectively, the simplest way to define similarity S (that we will consider as the anomaly score) is by directly computing the Euclidean distance between each points. However, this requires both time series to be of the same length $M = N$, where this is not necessarily the case at all time [84]. That’s why the similarity should be performed between sequences that have varied lengths (when $M \neq N$). To this end, DTW is the best choice that allows for non-linearity matching of two-time series with different lengths [84].

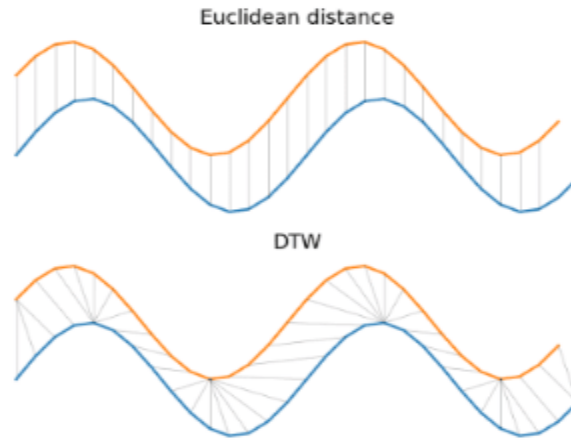


Fig 3.7: Similarity measure between pairwise of time Series

DTW distance is calculated by filling an N by M matrix, called DTW cost matrix (implementation available in next chapter), where N and M refer to the length of time series \mathbf{K}_{ts1} and \mathbf{K}_{ts2} , respectively. The cost matrix allow us to extract the warping path in addition of the score of the similarity measure between the two keystroke time series.

Chapter 4

The implementation of TyPaVeS

In the following chapter, the implementation of the TyPaVeS application will be thoroughly analyzed, in order to properly understand its usage and functionality. A few run-time screen shots will be presented along with the application's website interface.

1. Programming environmental

As any web application, TyPaVeS requires some frontend and backend components, we will not use a framework for this project but we will use the MVC pattern (model-view-controller), the following list is about the most important environmental that we have used during the development of TyPaVeS and for what we used them:

1.1.WAMP platform

"WAMP Stands for "Windows, Apache, MySQL, and PHP." WAMP is a variation of LAMP for Windows systems and is often installed as a software bundle (Apache, MySQL, and PHP). Often used for web development and internal testing, but may also be used to serve live websites". [94]

- **Apache server**

"Apache HTTP Server is used in order to run the web server within Windows. By running a local Apache web server on a Windows machine, a web developer can test webpages in a web browser without publishing them live on the Internet". [94]

- **MySQL database**

"MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms. Although it can be used in a wide range of applications, MySQL is most often associated with web applications" [95]. We chose it for its simplicity, easy coding and fast queries, all user personal data, typing patterns and user prototypes will be stored inside a MySQL database. Indeed it's used to store all data needed in order for the mechanism to run.

- **PHP**

"Stands for "Hypertext Preprocessor. PHP is an HTML-embedded Web scripting language. This means PHP code can be inserted into the HTML of a Web page. When a PHP page is

accessed, the PHP code is read by the server the page resides on. The output from the PHP functions on the page are typically returned as HTML code, which can be read by the browser. Because the PHP code is transformed into HTML before the page is loaded” [96]. PHP is very performant when we are dealing with database queries that’s why we will use it not only for registration and identification process but also to compute the DTW distance between samples and to decide whether a user is a legitimate or an illegitimate user.

1.2. User interface

Improving user experience is always recommended when we design and develop a website, for having a good website our web application front end content consists of:

- **HTML**

“HTML is a HyperText Markup Language file format used for describing the structure of Web pages. The HTML code consists of tags surrounded by angle brackets. The HTML tags can be used to define headings, paragraphs, lists, links, quotes, and interactive forms. It can also be used to embed JavaScript, and CSS codes”. [92]

- **CSS**

“CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. His separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments”. [92]

- **JS**

“Java Script is the most commonly used dynamic programming language for web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed ”[93]. In our project a java script that we will introduce its implementation in the next section is used to capture all user typing patterns for any given field.

- **JQuery**

“jQuery is a fast, small, and feature-rich JavaScript library”. We will only use it as a validator for user input data (check the format of the text entered in the field while sign-in o sign-up processes)

2. Client side functions

While a user load the website home page, the script that capture the typing pattern is loaded automatically. The collection of the keystrokes starts directly when the credit card number field is focused by the new user, while he is typing the CCN digits that has been auto generated by the script application (in order to avoid a real credit card numbers and use only random numbers for demonstration purposes) keyboard events are recorded and a typing session is created, the typing session contains the following information about the keyboard events:

- Type of the keyboard event (key-down/key-up/key-press)

- event.keyCode (the *keyCode* field of the corresponding JavaScript event)
- event.which (the *which* field of the corresponding JavaScript event)
- event.charCode (the *charCode* field of the corresponding JavaScript event)
- event.shiftKey (the *shiftKey* field of the corresponding JavaScript event)
- The value returned by JavaScript's Date.getTime() function, i.e., the number of milliseconds since the 1st of January 1970.

When the user hits the register button, all keystroke data is processed and transformed into an SQL query along with all user's personal data in order to be sent to the server. Before this happens, the username and password have to be verified. A JQuery call is made by the web-site's authentication server, containing the user credentials.

If the credentials get accepted (it means that there is no such an already registered user with the same username), the next JQuery starts the communication with the database in order to insert the data by sending a query which will be handled by the server side. After that the user will be redirected to the login page after the operation has a succesful execution.

Now that the user is registered, it's time to login to the system. The registered user is already in the login page, all what he have to do is to type his username (email address) and password and click login. By clicking the login button the data is sent to the authentication sever when there will be a password matching and checking (**Fig 3.5**).

After a successful login, the user is now ready to give the hand to other users and dare them to type the CCN the same way he did (we could run the typing pattern checking during the login which will be the use of authentication typing pattern system but we are focusing only in the verification process that can be applied online to avoid banking fraud. Our proposed model can be used to verify user identity before validating a transaction). At this stage the same script application that we have been using during sign up process is called again to extract candidate user keystrokes, but this time they will be directly sent to the application server and not stored into a database.

3. Server side functions

After receiving the first request that contains user's data as a POST request, the authentication server uses a GET request and stores the data by a SQL query. For this to happen, the database has to be accessed, which is quite easy using the PHP. All data concerning the keystrokes, the users, prototypes and everything needed by TyPaVeS is stored in the database. So, a query is send to the database with the username, password, email and CCN along with the keystroke details. Initially prototypes are set to empty as there is no prototypes available yet after that an Ack (response) is sent back to the client side. In the meantime (in parallel with the authentication server) the application server is working on extracting the keystroke features (user prototypes). In case the user doesn't exist, a new entry is made and returned. A few moments later, a next request is received concerning the keystroke prototypes. As the user id is attached to the prototypes, a query is sent to the database in order to find the specified user ID in the existing samples already registered in the database and update the values by the calculated prototypes.

Before any actions take place, the sample keystrokes contained in the request and sent to the sever application need to be transformed into the features that we will use for classification later on. Right now, there are two timings for each keystroke (key-down and key-up events), but in order to make any calculations, durations between pairs have to be computed into down-down, down-up metrics. The first condition that needs to be met is that the CCN contained in the request and the ones saved in the database should be of the same length. If not, the sample is rejected straight away. Most commonly though, they are of the same length, which allows the mechanism to continue the operation.

3.1.Pre-processing of raw typing patterns

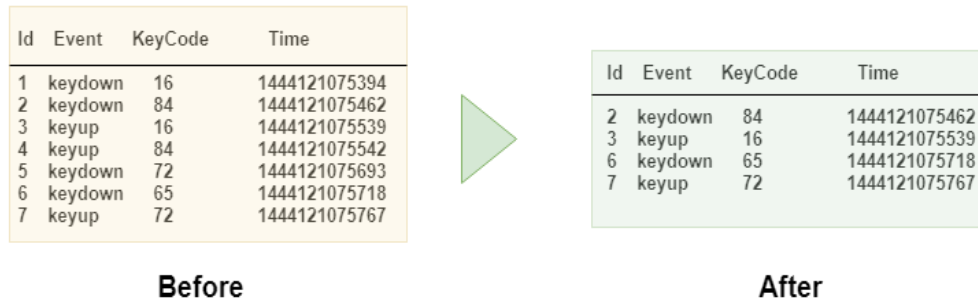
In order to extract down-down and down-up durations, the raw data must be pre proceeded before that.

```
|TYPING PATTERN 1
keyup 9 9 0 false 1444121074805
keydown 16 16 0 true 1444121075394
keydown 84 84 0 true 1444121075462
keypress 0 84 84 true 1444121075462
keyup 16 16 0 false 1444121075539
keyup 84 84 0 false 1444121075542
keydown 72 72 0 false 1444121075693
keypress 0 104 104 false 1444121075693
keydown 65 65 0 false 1444121075718
keypress 0 97 97 false 1444121075719
keyup 72 72 0 false 1444121075767
keyup 65 65 0 false 1444121075809
keydown 84 84 0 false 1444121075873
keypress 0 116 116 false 1444121075874
keyup 84 84 0 false 1444121075938
keydown 16 16 0 true 1444121076082
keydown 49 49 0 true 1444121076607
keypress 0 39 39 true 1444121076607
keyup 49 49 0 true 1444121076670
keyup 16 16 0 false 1444121076700
```

Fig 4.1: Example of user typing pattern raw data.

Below we have the pre-processing steps by the chronological order (can be done in any order):

1. Omit event.which, event.charCode and event.shiftKey from the data.
2. Ignore all key-press events as we can get the key code from the key-down event and also because we didn't take it into account in our feature extractions.
3. Sometimes the first event of the typing pattern is a key-up event (like we have in **Fig. 4.1**) which is caused by clicking the shift key to change the cursor between registration form fields, so we have to remove it.
4. For simplicity we will remove sequential duplicated events (successive key-down events or successive key-up events) and for that we keep the latest key-down and the oldest key-up

**Fig 4.2:** Eliminating same successive events.

3.2.Extraction of features

In this section we will give the algorithm of extracting down-down (between KS) and down-up (KS duration) metrics.

Algorithm 1: KS Duration prototype

Input: typingP : List []
Output: ksDuration: List []
 counter = 0;
 ksDuration = [];
while Not end of list **do**
 ksDuration[counter] = typingP[counter + 1] - typingP[counter];
 counter += 2
return ksDuration

For the KS duration we create an array of event timer, it means that for each pair (key-down/key-up a value is calculated), we have to increment by 2 otherwise in the next round of the while loop we will calculate the up-down duration which is not our goal. Then we just simply return the array.

Algorithm 2: Between KS prototype

Input: typingP : List []
Output: ksDuration: List []
 counter1 = 0;
 typing = [];
 betweenKS = [];
while Not end of list **do**
 if typingP[counter1] == 'keydown' **then**
 typing[] = typingP[counter1];
 counter1++;
 counter2 = 0;
while Not end of list **do**
 betweenKS[counter2] = typing[counter2 + 1] - typing[counter2];
 counter2 += 1
return betweenKS

For the between KS duration another step is required, first of all we must omit all key-up events and then simply calculating the duration between each key-down/key-down pair and after each step we increment by 1. And finally return the array that represents the second prototype.

Now that both prototypes are ready, they are sent back to the authentication sever along with the used id in order to be stored in the database.

3.3.Classification of candidate user typing patterns

Once a candidate user types the CCN of the registered user and click the check button. His typing pattern is sent and received by the application server where candidate user prototypes will be extracted. The same algorithms (KS duration and between KS) will be used to get user features. The prototypes are ready, the server loads registered user prototypes by sending an SQL request to the authentication server and starts the classification process (**Fig 3.6**).

During the classification process each pair of prototype is compared to another pair of prototypes by using the DTW distance which will calculate the distance between KS duration prototypes and Between KS prototypes based on the following algorithm:

Algorithm 3: DTW Distance

```

Input: kts1: array [1..n]
Input: kts2: array [1..m]
Output: score: double
i,j = 1;
costMatrix = [0..n][0..m];
while  $i \leq n$  do
  while  $j \leq m$  do
    costMatrix[i][j] = infinity
costMatrix[0][0] = 0;
i,j = 1;
while  $i \leq n$  do
  while  $j \leq m$  do
    cost = dist(kts1[i],kts[j]);
    costMatrix[i][j] = cost + minimum(costMatrix[i-1, j ],
                                     costMatrix[i , j-1],
                                     costMatrix[i-1, j-1])
return costMatrix[n][m]
```

The anomaly scores (DTW distance) are calculated, now it's time to classify the candidate user (**Fig 3.6**), for that the classifier NN will check whether the candidate user belongs to the same set, cluster or even a region (**Fig 2.16**), for each anomaly score there exists a fixed threshold, it will be used to determine user identity by using the simple algorithm described below.

Algorithm 4: NN Classifier

```
Input: score1, score2: double  
Input: threshold1, threshold2: int  
Output: result: boolean  
result= false;  
if ( $score1 \leq threshold1$ )  $\wedge$  ( $score2 \leq threshold2$ ) then  
  result= true;  
return result
```

If the output of the algorithm is true, it means that the user is accepted otherwise he is rejected (**Fig 3.6**). The results are then sent back to the client in the form of booleans, which concludes the keystroke verification procedure.

4. Database conception

The database is a pretty important part of the whole service functionality, since it holds all needed information for the keystroke verification process to take place. The database structure is presented in the figure below.

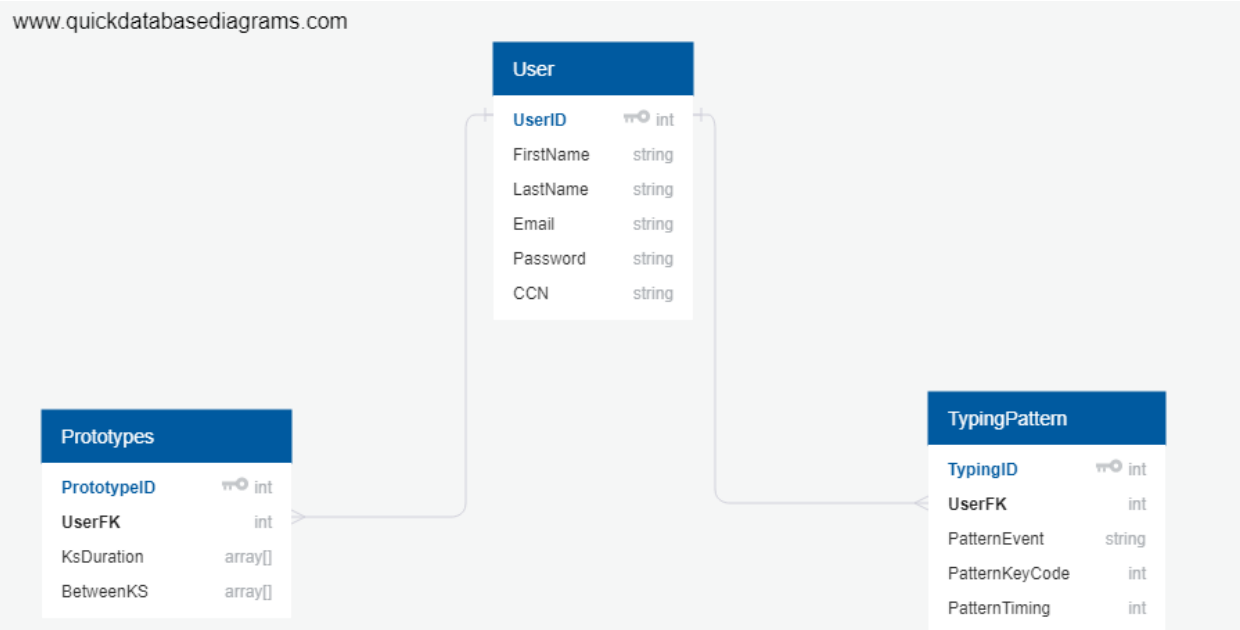


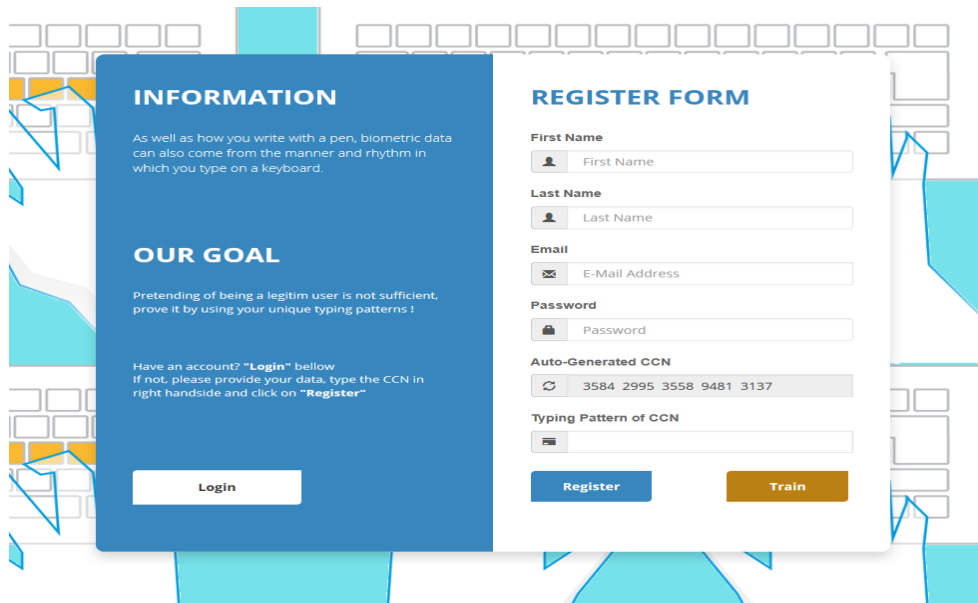
Fig 4.3: Database structure.

5. Web-site usability and design

We developed a simple web-site that contains the functionalities that we had mentioned about TyPaVeS. As a prototype, the web-site will allow users to try and checkout how keystroke dynamics verification system works in real-time.

The website includes the following content:

- **A home page**

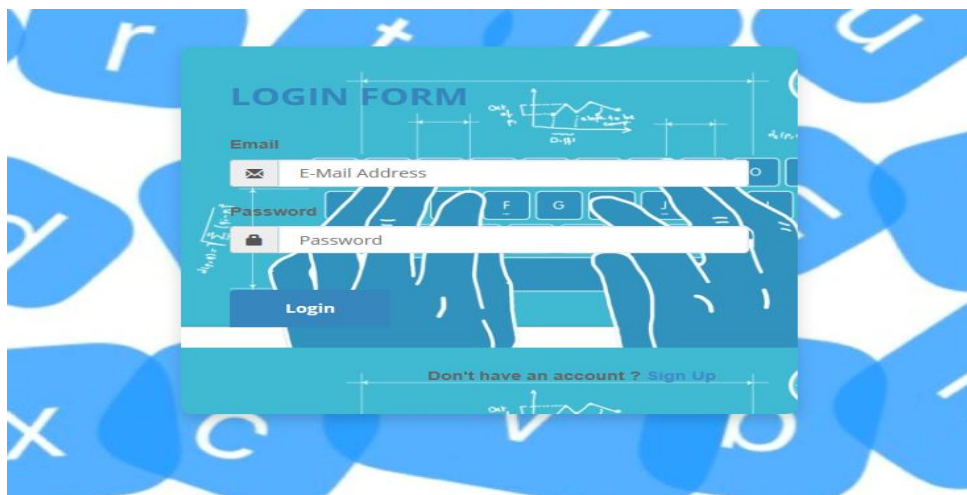


The image shows a registration form titled "REGISTER FORM" on a blue background. On the left, there is a section titled "INFORMATION" with text explaining that biometric data can come from typing patterns. Below this is a section titled "OUR GOAL" with text stating that pretending to be a legitimate user is not sufficient and that users should prove their identity by using their unique typing patterns. At the bottom of this section is a "Login" button. On the right, the "REGISTER FORM" contains fields for "First Name", "Last Name", "Email", and "Password". Below these fields is an "Auto-Generated CCN" field showing the number "3584 2995 3558 9481 3137" and a "Typing Pattern of CCN" field. At the bottom of the form are "Register" and "Train" buttons.

Fig 4.4: Registration form along with some greeting.

A new user have to type his data in order to register himself to the authentication server, for our demonstration purposes, we will use a random generated CCN instead of a real credit card numbers, as typing a randomly generated will not give us a unique typing pattern that really represent the user, he can repeat typing the CCN by clicking on the “Train” button until he gets familiar with it and then he can submit it.

- **A login page**



The image shows a login form titled "LOGIN FORM" on a blue background. It contains fields for "Email" (with a sub-label "E-Mail Address") and "Password". Below these fields is a "Login" button. At the bottom of the form is a link that says "Don't have an account ? Sign Up". The background of the form features a stylized keyboard and some mathematical symbols.

Fig 4.4: Login form

All what the user have to do is to type his username (email) and password, if the user is already registered, the access to the challenge page is guaranteed otherwise the user is denied.

- **Challenge page**

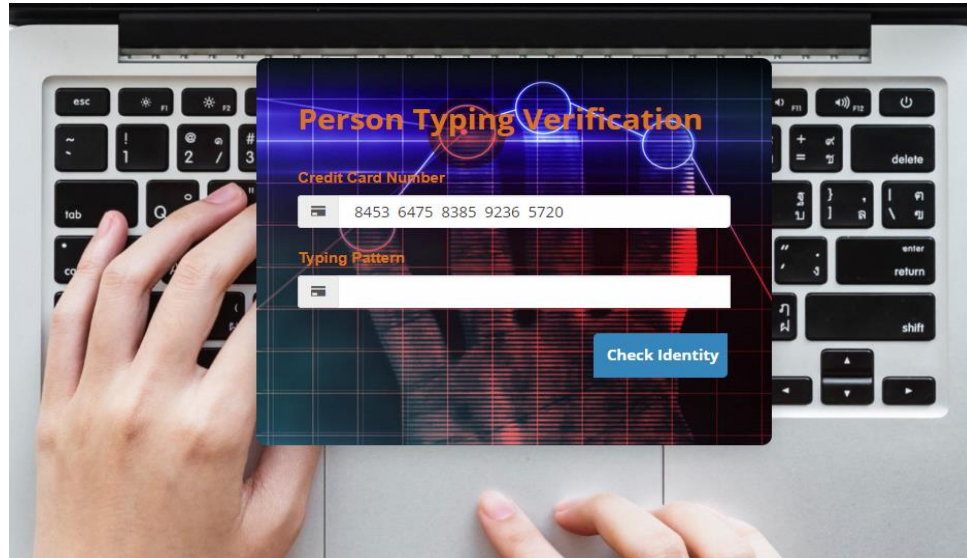


Fig 4.5: Person typing verification form

On this page a pretender (could be either the guanine user or a candidate user) will type the CCN shown and click “Check Identity”, this is a typical scenario after all the orders are done and we want to pay we enter our CCN, this is where our proposed model will take place.

- **Result page**

The result page consists of three parts, user and genuine typing patterns of the CCN, the output of the DTW algorithm that consists of the DTW distance (green if under threshold and red otherwise) along with the cost matrix for both durations (between KS and KS durations), the red small squares represent the warping path of the DTW distance and finally the decision made by the system (accepted user with the green happy smile or rejected user with a red unhappy smile).

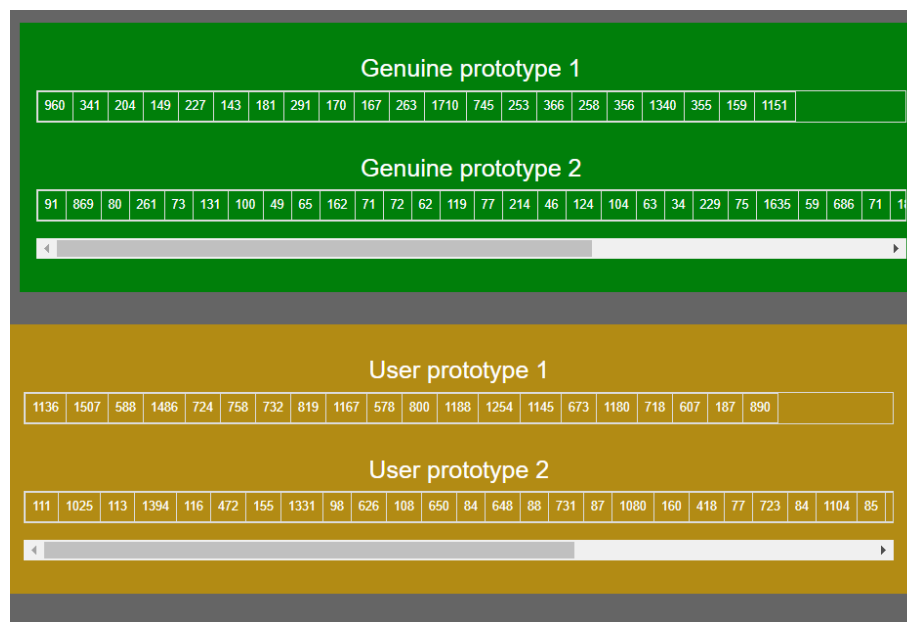


Fig 4.6: Genuine/user typing prototypes.

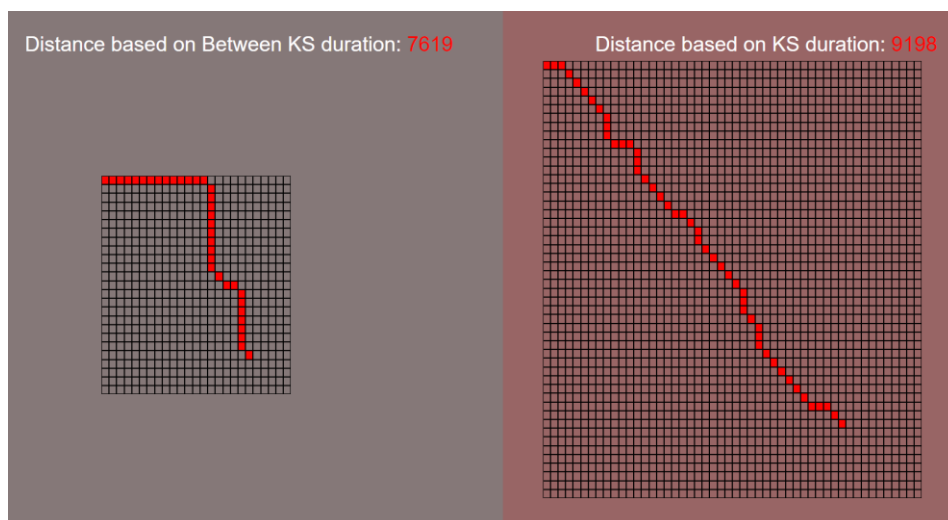


Fig 4.7: Typing pattern algorithm results.

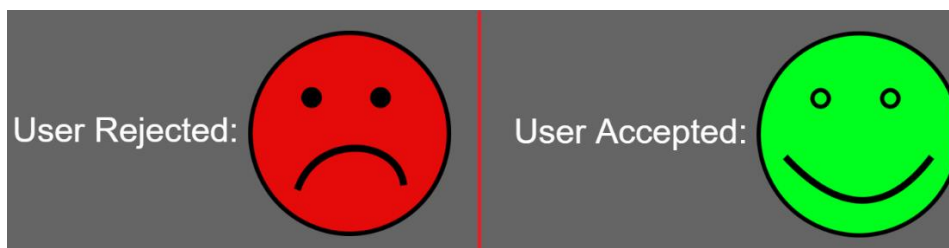


Fig 4.8: Classifier decision.

Chapter 5

Evaluation and Experiments

In order to evaluate the DTW classifier which is the principal subject of my thesis work, an experiment was made which is explained in the chapter along with its results.

1. Evaluation of TyPaVeS

When we are dealing with a restricted amount of data, using TyPaVeS will produce a good results, the schema of the model is good but not perfect and can be improved further. As you may noticed while the user attempt to create a new account, he type a randomly generated 20 digits number which is taken as a pseudo credit card number (as we wanted to avoid that users type them real CCN number). Re-typing a random 20 digits that you never have used before as an entire sequence for the first time cannot really define your unique typing rhythm of the CCN. That's why adding a training phase could be benefit to have an accurate CCN typing pattern that could really represents user unique features.

TyPaVeS is a typing verification system, we can extend it more in such a way that he could perform both authentication and verification processes, in this case we can store username and password typing patterns and then extract them features the same way we did with the CCN and then run our model while the use tries to login into his account.

The most important part that we should really evaluate is the DTW distance between a pair of keystroke typing patterns. Indeed the calculation of such a distance have a high cost in term of memory and time complexity. That's why we will review the DTW distance.

2. Review of DTW distance

Suppose we have two time series, a sequence A of length n, and a sequence B of length m, where

$$A = a_1, a_2, \dots, a_i, \dots a_n$$

$$B = b_1, b_2, \dots, b_j, \dots b_m$$

To calculate the distance or align these two sequences using DTW, we first construct an $n \times m$ matrix where the (i^{th}, j^{th}) element of the matrix corresponds to the squared distance, $d(a_i, b_j) = (a_i - b_j)^2$ which is the alignment between points a_i and b_j [98]. To find the best match between

these two sequences, we retrieve a path through the matrix that minimizes the total cumulative distance between them (as illustrated in **Fig 5.1**). In particular, the optimal path is the path that minimizes the warping cost:

$$DTW(A, B) = \min \left\{ \sqrt{\sum_{k=1}^K w_k} \right\}$$

This warping path can be found using dynamic programming to evaluate the following recurrence:

$$\delta(i, j) = d(a_i, b_j) + \min \begin{cases} \delta(i-1, j-1) \\ \delta(i-1, j) \\ \delta(i, j-1) \end{cases}$$

Where $d(a_i, b_j)$ is the distance found in the current cell, and $\delta(i, j)$ is the cumulative distance of $d(a_i, b_j)$ and the minimum cumulative distances from the three adjacent cells (match/ insertion/ deletion).

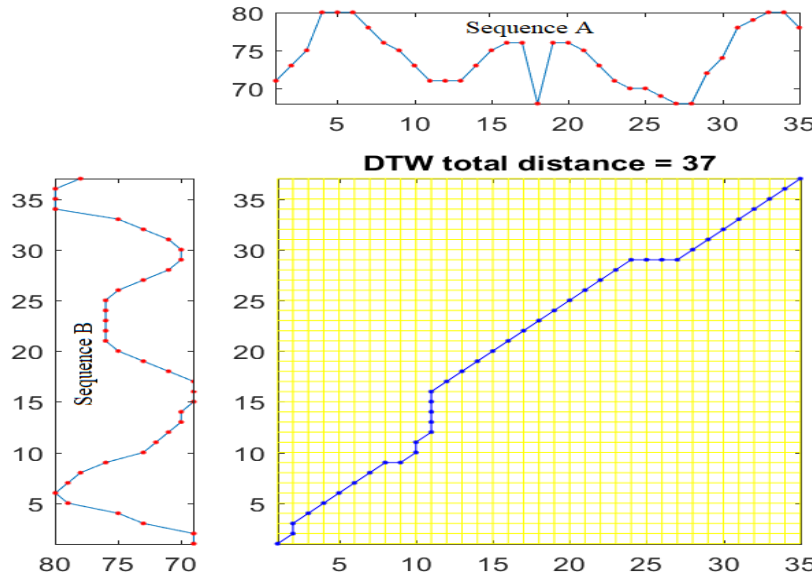


Fig 5.1: Optimal warping path of a pairwise time series. [97]

To reduce the number of paths to consider during the computation, several well-known constraints (boundary conditions, continuity condition, monotonic condition, and adjustment window condition) have been applied to the problem to restrict the moves that can be made from any point in the path and so restrict the number of paths that need to be considered. [98]

3. Lower bounding the DTW distance

The adjustment window condition is a lower bounding technique based on the warping window size. This technique is used to speed up DTW calculation. The width of this constraint is often set to 10% of the length of the time series [99]. The following figure illustrates two of the most

frequently used global constraints in the literature, the Sakoe-Chiba band [101] and the Itakura parallelogram [100]. The latter is widely used in the speech recognition.

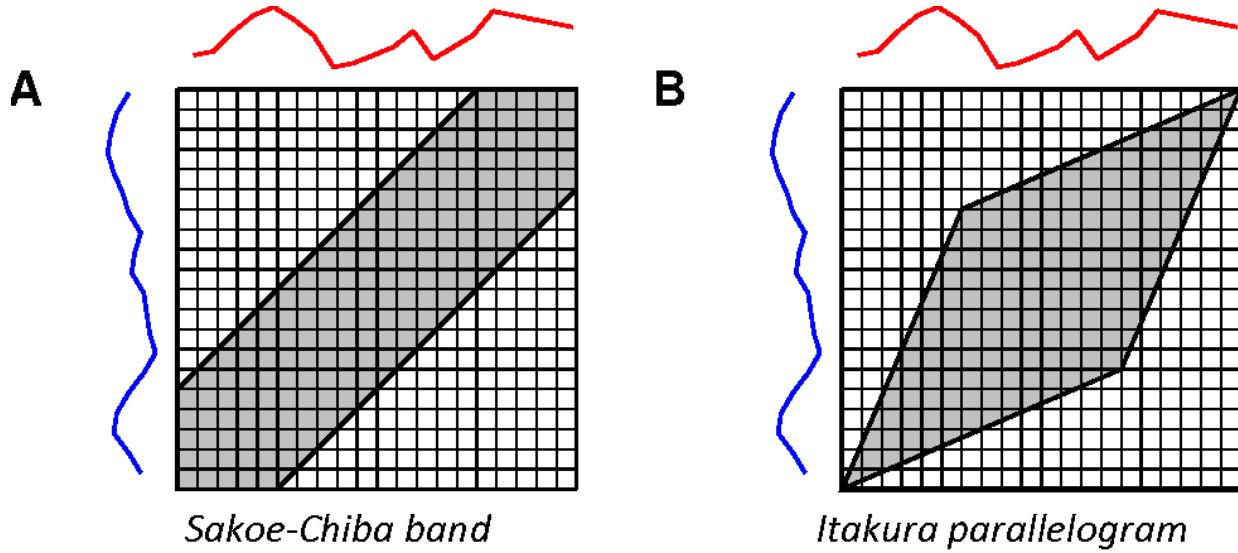


Fig 5.2: Sakoe-Chiba band and Itakura parallelogram [99]

In order to work with warping window size constraint, we need to adapt the original implementation of DTW distance (algorithm 3 described in chapter 4) in such a way that if a_i is matched with b_j , then $|a_i - b_j|$ is no larger than w , a window parameter. In order to make the algorithm work, the window parameter w must be adapted so that $|n - m| \leq w$. The adapted algorithm is described below:

Algorithm 5: DTW Distance with Warping Window Constraint

```

Input: kts1: array [1..n]
Input: kts2: array [1..m]
Input: w: int
Output: score: double
w = max(w, abs(n-m));
i = 1;
costMatrix = [0..n][0..m];
while  $i \leq n$  do
    while  $j = \max(1, i-w) \leq \min(m, i+w)$  do
        costMatrix[i][j] = infinity
costMatrix[0][0] = 0;
i = 1;
while  $i \leq n$  do
    while  $j = \max(1, i-w) \leq \min(m, i+w)$  do
        cost = dist(kts1[i], kts2[j]);
        costMatrix[i][j] = cost + minimum(costMatrix[i-1, j ],
                                           costMatrix[i , j-1],
                                           costMatrix[i-1, j-1])
    return costMatrix[n][m]
```

Hoang Anh Dau [102] claimed that obtaining the best performance from DTW requires setting its only parameter, the maximum amount of warping (w) and thus can produce significant improvements in classification accuracy. In the following section we will make a small experiment to see whether this hypothesis is true or not in case of typing dynamics data.

4. Experiment

Person identification based on the dynamics of typing is a challenging task with applications in various domains ranging from online education to internet banking one of the tasks related to person identification based on the dynamics of typing is person authentication. [103]

4.1. Person authentication dataset

For this task of the person identification open challenge proposed by my own supervisor, in his paper [109] available at “biointelligence.hu/typing.html” website [103], 458 user typing dynamics were recorded with a JS application. In each typing session, the users were asked to type some sentences and the keyboard events key-up, key-down and key-press were captured by the JavaScript application and stored on our web server. In the text file (12 users’ raw data), each record starts with the keyword TYPING PATTERN which is followed by the identifier of the typing session. Each subsequent line corresponds to a keyboard event. These event are: key-press, key-up and key-down. Each line, contains the following pieces of information: [103]

- Type of the keyboard event (key-down/key-up/key-press)
- event.keyCode (the *keyCode* field of the corresponding JavaScript event)
- event.which (the *which* field of the corresponding JavaScript event)
- event.charCode (the *charCode* field of the corresponding JavaScript event)
- event.shiftKey (the *shiftKey* field of the corresponding JavaScript event)
- The value returned by JavaScript's Date.getTime() function, i.e., the number of milliseconds since the 1st of January 1970.

Additionally, we are given the true identity of the users (coded by integer numbers from 1 to 12) for 5 typing pattern per users in a text file (12 users training labels). Each line of this file contains two numbers separated by a comma: [103]

- The identifier of a typing pattern (pattern id for short), and
- The identifier of the user who typed that pattern (user id for short).

There is another text file (12 users test hypothetical labels) contains the *hypothetical* identities of the users for the rest of the typing patterns (i.e., for those typing patterns for which the true identity of the user is not given in the text file (12 users training labels). The hypothetical identities are given in the same format as the true identities (i.e., as pairs of pattern ids and user ids). Our task is to decide if the hypothetical identities in the text file (12 users test hypothetical labels) matches the true identities of the users who typed those patterns. For that we have to provide solutions for

the above task as a list of pairs. The first number of the pair should be the pattern id. The second number of the pair depends on our predictive model. If we predict that the hypothetical user identity for the pattern is the same as the true user identity, the second number should be 1. Otherwise it should be 0.

4.2. Methodology

In order to experiment the warping window size constraint we will run the DTW algorithm of the training data and compare them to the 12 users training data with different w values, $w \in [0, 1, 2 \dots M]$ where M is the length of the longest typing pattern sequence (considered as 1000).

Before running the DTW distance between each pair of keystroke time series, the data must be pre-processed, for that we will use the server side function that we have seen during the last chapter where we had to clean the data, remove same successive events, extract features (KS duration, between KS and a merged version where we will use the average of KS and between KS durations).

For each warping window size w we classify users testing data based on the following schema:

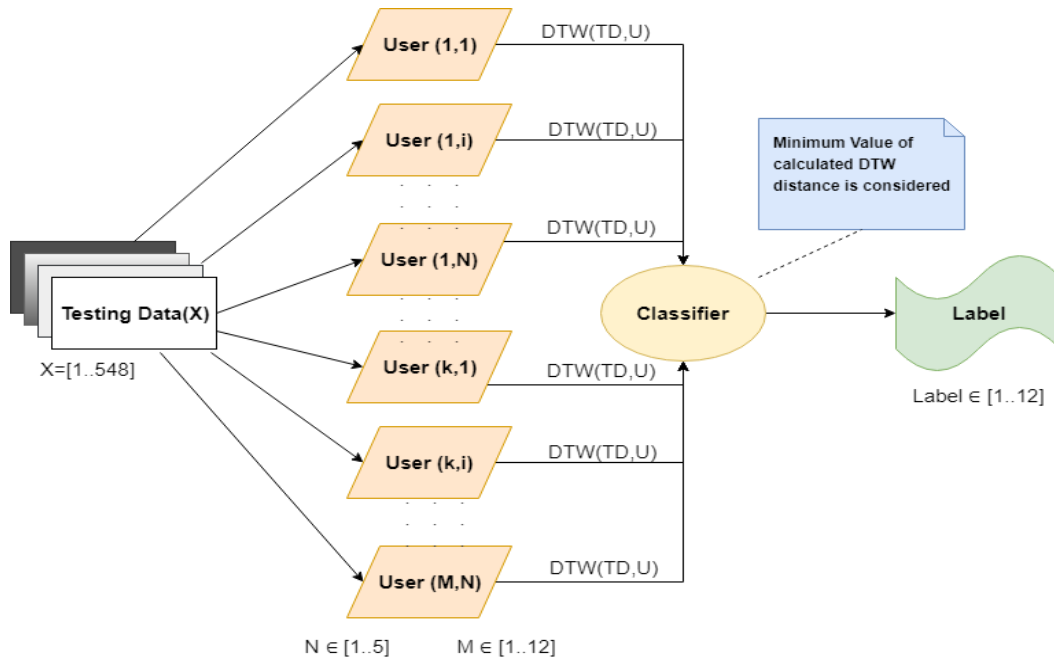


Fig 5.3: Testing data classification process

4.3. Experimental Results

To test the effect of the warping window size to the classification accuracies, we performed an empirical experiment on the dataset with KS Duration, between KS and a merge duration. We vary the warping window size from 0 (Euclidean) to M (no constraint/full calculation) and record the accuracies.

Usually we linearly interpolate all variable-length datasets to have the same length of the longest time series within the dataset and measure the accuracy using the 1-nearest-neighbor with leaving-one-out classification method.

But in our study case, we have used the original variable-length keystroke time series and we didn't linearly interpolate all variable-length datasets to have the same length of the longest keystroke time series within the dataset. The results are shown in the following figures:

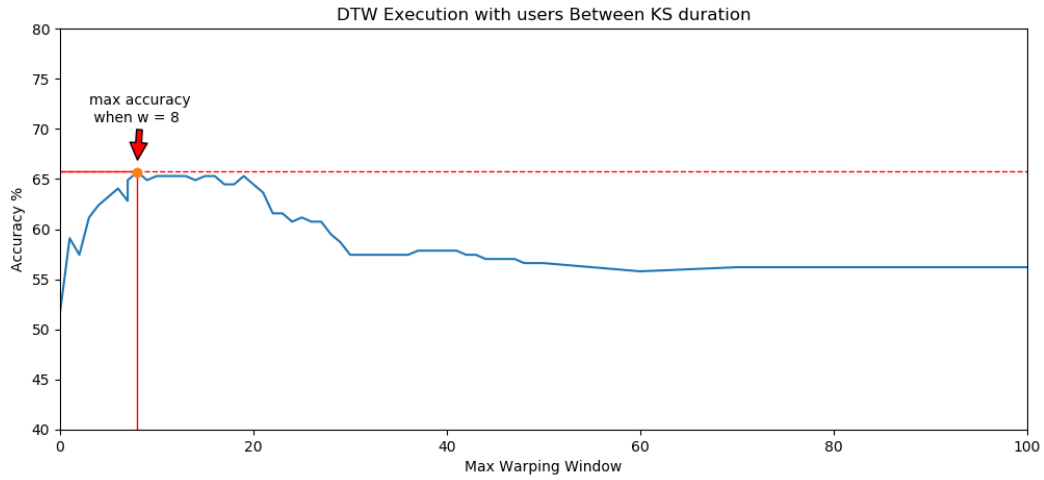


Fig 5.4: The classification accuracies for all warping window sizes and Between KS duration.

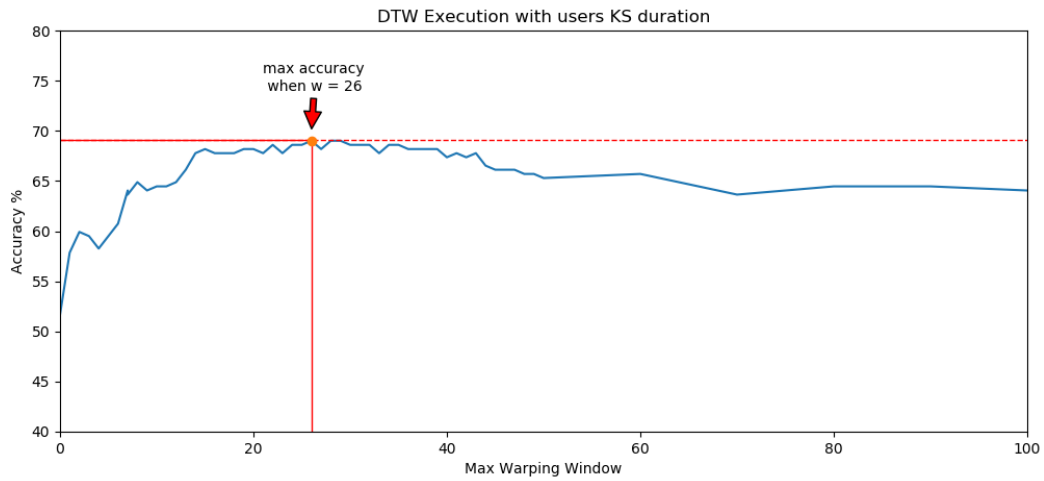


Fig 5.5: The classification accuracies for all warping window sizes and KS duration.

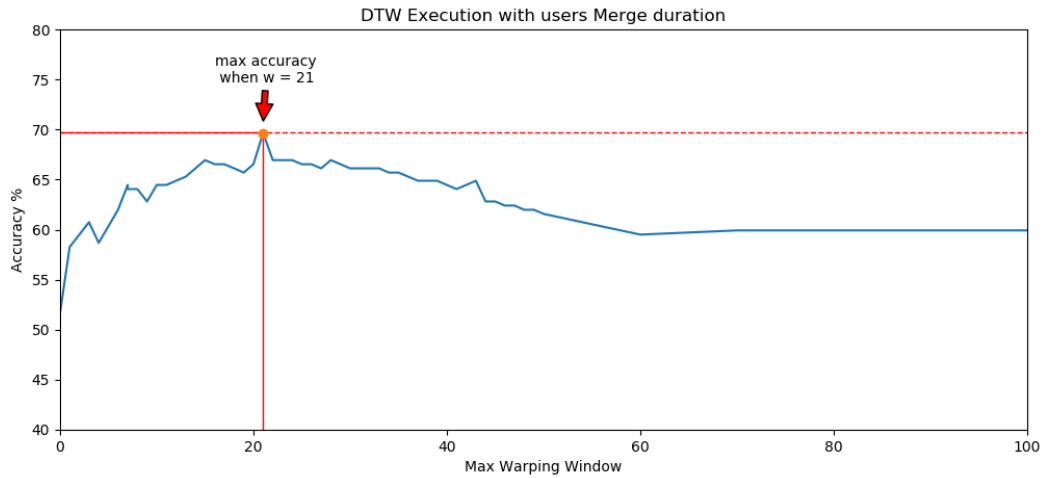


Fig 5.6: The classification accuracies for all warping window sizes and merge duration.

A study made by Ratanamahatana et al. [98] where the same experiment has been proceeded with 6 datasets retrieved from “*UEA & UCR Time Series Classification Repository*” ended by the following results:

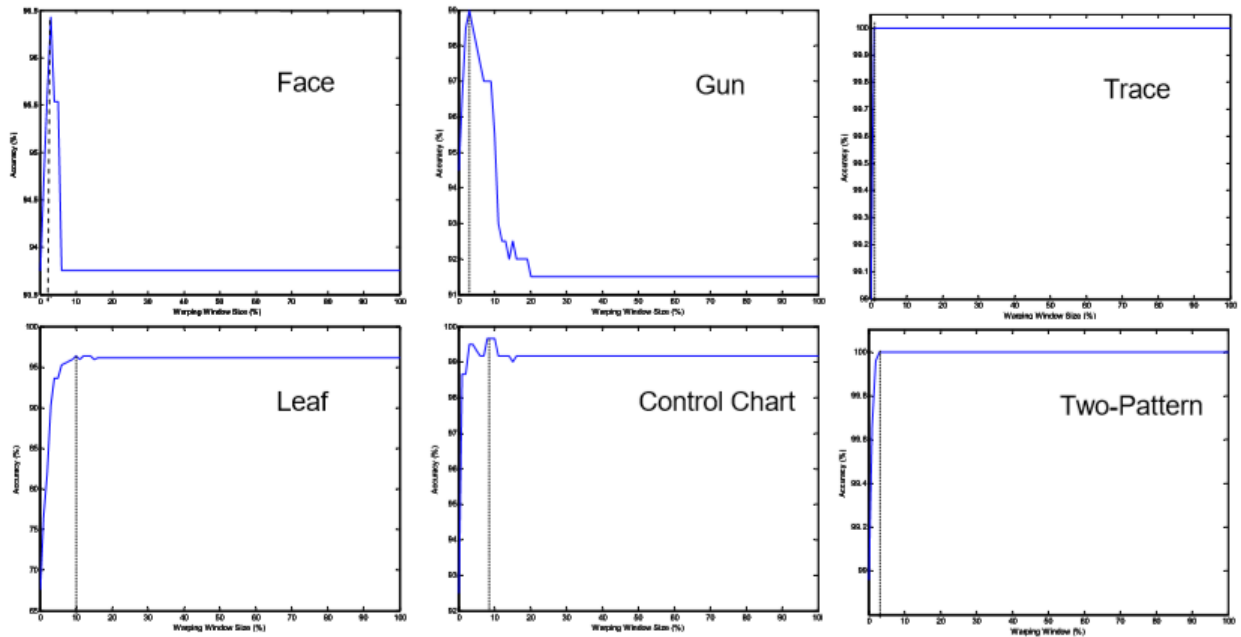


Fig 5.7: The classification accuracies for all warping window sizes. All accuracies peak at very small window sizes. [98]

Wider warping constraints do not always improve the accuracy, as commonly believed [104]. More often, the accuracy peaks very early at much smaller window size, as shown in **Table 5.1** below:

Dataset	Max Accuracy %	Warping Window Size %
Face	96.43	3
Gun	99.00	3
Trace	100.00	1
Leaf	96.38	10
Control chart	99.67	8
Two-pattern	100.00	3
Typing Pattern – KS duration	69.10	2.6
Typing Pattern – between KS	65.70	0.8
Typing Pattern – merge duration	69.67	2.1

Table 5.1: The warping window size that yields maximum classification accuracy for each dataset, using DTW with Sakoe-Chiba band. [98]

From the result found on the paper of Ratanamahatana et al. [98] and our own experiments over the typing pattern person identification challenge, we can conclude that working with a small warping window of 10% and down is a good thing and could be benefit not only to reduce the execution time and memory usage but also rise the accuracy of the classification. We also noticed that working with a larger warping window size could reduce the accuracy of the classification in some cases. That's why it's better to use a little warping.

In order to make our conclusion as a theorem or a fact result, we have to experiment more datasets with variety of constraints, with different dataset sizes and domain of applications.

So to finish the conclusion, the hypothesis claimed by Hoang Anh Dau [102] that says: “*obtaining the best performance from DTW requires setting its only parameter, the maximum amount of warping (w) and thus can produce significant improvements in classification accuracy*” was correct concerning the typing dynamics classification problem.

Chapter 6

Summary

In this last chapter, a conclusion is made about everything that happened through the past chapters, while the reader is also provided with some food for thought in order to understand the advantages that keystroke dynamics could provide, along with future upgrades and ideas for the implementation.

1. Conclusion

Keystroke dynamics is known to be characteristic to individuals that's why keystroke dynamic authentication is an underrated authentication mechanism, which never received the proper media attention that it needs in order to flourish. With its basic concepts, users that interacts with the websites, information systems and high level pertinent systems have the opportunity to access those data in a much secured manner and without installing any additional infrastructure or costs, in hopes of it being more accessible to the wider public. Such a web service implies less to none computation being made on the web-server or client, which makes TyPaVeS an all-around light and easy to use application of keystroke dynamic authentication for any kind of website. TyPaVeS is an open source system, which may attract developers to improve or even make their own versions of TyPaVeS hopefully better, which will also help the spread of keystroke dynamics technologies which can provide so much to our modern society.

Implementing such a system require a deep knowledge in time series classification and anomaly detection techniques. Using the DTW distance as a distance measure between pairwise time series data is one of the best options actually available. Along with the DTW the classifier NN (Nearest Neighbor) played a huge role in the last century for solving time series classification problems and recently we have seen significant progress in improving both the efficiency and effectiveness of time series classification. However, the best solution is typically the NN algorithm with the relatively expensive DTW as the distance measure.

As the DTW causes time complexity problems when the times series are too large, lower bounding the DTW distance by adjusting the warping window size is considered as a solution to reduce the execution time and the memory usage, but adjustment the window condition by using a lower bounding technique performed unexpected results with the classification accuracy, and our experiment over the warping window size show that lower bounding the warping window size by 10% or less gives the best results in term of classification accuracy while having a higher warping

window size reduces the classification accuracy in some study cases and indeed adjustment the window condition decrease the times complexity and memory usage as it will not compute all the values over cost matrix of the DTW algorithm.

2. Food for thought

Keyboard dynamics are only the beginning. The same principles could easily be applied on smart-phones, video-game consoles and basically every electronic device or gadget that involves human-computer interaction, for example a car. There are definitely unique patterns in the way people drive their cars, so a mechanism like that would exclude the possibility of car thefts. To be more precise, there are unique patterns almost in every human-computer interaction, which could authenticate a person. Yes, there are other ways of biometric authentication, but people are not usually happy to provide their characteristics to sensors directly. Behavioral biometrics could solve that, through simple “transparent” sensors which would not affect their everyday routine in any way, in contrast to using a password for example. The development of technologies like that could easily be combined with ubiquitous technologies, smart homes etc.

3. Future work

The next step of considering the application would be to introduce it to the wider public through social networks. A security application that would recognize your personal typing rhythm and deny others from using your account in any way, preventing them to send messages, comments etc. even if someone happened to find an account logged in. As mentioned before, the application was implemented based on a previously created mechanism, which has big room for improvement.

If an application like that would to become commercial, improvement is necessary, in order to achieve acceptable standards in terms of lower error rates a training phase would be benefit to the users, indeed having a multiple typing patterns for the same user could improve the classification accuracy, we could for example have a typing pattern for many cases, when the user is tired or using different keyboard and in order to avoid an expensive calculations as we will deal with multiple typing patterns for each user, extracting a candidate typing pattern that would represent all typing patterns as a single and unique pattern will be a good solution.

Last but not least, we have used only two features while dealing with the typing patterns, so maybe including more keystroke features such as up-up or up-down durations or a combination of them in addition of the already used down-down and down-up durations would increase or perform a better results.

Bibliography

- [1] Shaon Shahnewaz, S. S. (2018, July 10). Fingerprint vs vascular biometrics? What are the differences. Retrieved from <http://www.m2sys.com/blog/important-biometric-terms-to-know/fingerprint-vs-vascular-biometrics-what-are-the-differences/>
- [2] Hiep Nguyen Duc, H. N. (2014, September 4). Biometric Facial Recognition Database Systems. Retrieved from <https://eforensicsmag.com/biometric-facial-recognition-database-systems/>
- [3] MFZBCN CC-BY-SA-3.0. (2012, August 10). Online signature produced with intuous wacom + MATLAB. Retrieved from <https://commons.wikimedia.org/w/index.php?curid=20599626>
- [4] Hogan, M. (2003), "Are you who you claim to be ?", National Institute of Standards and Technology, International Standards Organisation. <http://www.iso.ch/iso/en/commcentre/isobulletin/articles/2003/pdf/biometrics03-03.pdf>
- [5] Ian Muller E. (2018, July 19). Identification vs Verification: What's the Difference?. Retrieved from <https://www.veridiumid.com/blog/identification-vs-verification-whats-the-difference/>
- [6] Hajian-Tilaki K. (2013). Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation. Caspian journal of internal medicine, 4(2), 627-35.
- [7] Zhang, Jian, Yan, Ke, He, Zhen-Yu, and Xu, Yong (2014). "A Collaborative Linear Discriminative Representation Classification Method for Face Recognition. In 2014 International Conference on Artificial Intelligence and Software Engineering (AISE2014). Lancaster, PA: DEStech Publications, Inc. p.21 ISBN 9781605951508
- [8] Tracy V. Wilson. (2005, November 11). How Biometrics Works. Retrieved from <https://science.howstuffworks.com/biometrics.htm>
- [9] Bulatov, Y., Jambawalikar, S., Kumar, P., & Sethia, S. (14-15 November 2002). Hand Recognition System Using Geometric Classifiers. DIMACS Workshop on Computational Geometry,. Piscataway, NJ.
- [10] AnswersDotCom. (n.d.). What are the disadvantages and advantages of biometrics. Retrieved April 8, 2019, from http://wiki.answers.com/Q/What_are_the_disadvantages_and_advantages_of_biometrics.
- [11] Sareen, P. (2014). Biometrics-Introduction, characteristics, basic technique, its type and various performance measures. Int J Emerg Res Manage Technol, 3, 109-19.
- [12] Jain, Anil K., Bolle, Ruud, and Pankanti, Sharath (2006). Biometrics: personal identification in networked society. Vol. 479. Springer Science & Business Media, pp. 6.

- [13] Jain, Anil K., Ross, Arun, and Prabhakar, Salil (2004). An introduction to biometric recognition. In: *Circuits and Systems for Video Technology*, IEEE Transactions on 14.1, pp. 4–20.
- [14] Karnan, Marcus, Akila, M., and Krishnaraj, N (2011). Biometric personal authentication using keystroke dynamics: A review. In: *Applied Soft Computing* 11.2, pp. 1565–1573.
- [15] Jain, Anil K., Dass, Sarat C, and Nandakumar, Karthik (2004). Can soft biometric traits assist user recognition? In: *Defense and Security. International Society for Optics and Photonics*, pp. 561–572.
- [16] Hashiyada, Masaki (2013). DNA Biometrics. Tech. rep. Tohoku University Graduate School of Medicine, p. 8.
- [17] Hill, Robert B (Aug. 1978). Apparatus and method for identifying individuals through their retinal vasculature patterns. US Patent 4, 109, 237, p. 8.
- [18] Jain, Anil K., Hong, Lin, and Pankanti, Sharath (2000). Biometric identification. In: *Communications of the ACM* 43.2, pp. 90–98.
- [19] Srivastava, Prakash Chandra et al (2013). Fingerprints, Iris and DNA Features based Multimodal Systems: A Review. In: *International Journal of Information Technology and Computer Science (IJITCS)* 5.2, p. 88.
- [20] Choudhary, J. (Sep.-Oct2012). Survey of Different Biometrics Techniques. *International Journal of Modern Engineering Research (IJMER)*, ISSN, 2249-6645. Vol. 2, Issue. 5, pp-3150-3155.
- [21] R. Giot, M. El-Abed, and C. (Jul. 2011) Rosenberger, Keystroke dynamics overview, in *Biometrics / Book 1*, D. J. Yang, Ed. InTech, vol. 1, ch. 8, pp. 157–182. [Online]. Available: <http://www.intechopen.com/articles/show/title/keystrokedynamics-overview>
- [22] S. Mondal and P. Bours (2017). A study on continuous authentication using a combination of keystroke and mouse biometrics. *Neurocomputing*, vol. 230, pp. 1–22.
- [23] B. Li, H. Sun, Y. Gao, V. V. Phoha, and Z. Jin (2017). Enhanced free-text keystroke continuous authentication based on dynamics of wrist motion, in *Information Forensics and Security (WIFS), IEEE Workshop on. IEEE*, 2017, pp. 1–6.
- [24] V. Monaco (2018) Public keystroke dynamics datasets. Retrieved from <http://www.vmonaco.com/keystroke-datasets>
- [25] R. Giot, B. Dorizzi, and C. Rosenberger (2015). A review on the public benchmark databases for static keystroke dynamics. *Computers & Security*, vol. 55, pp. 46–61.
- [26] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua (2016). Labeled faces in the wild: A survey, in *Advances in face detection and facial image analysis*. Springer., pp. 189–248.

- [27] Imane Bouacida, I. B. (2018). Sentiment Analysis and Opinion Mining Techniques for Learning Analytics (Master's thesis, Eötvös Loránd University, Budapest, Hungary). Retrieved from http://t-labs.elte.hu/wp-content/uploads/Imane_Bouacida-Thesis.pdf
- [28] Teh, Pin Shen & Teoh, Andrew & Yue, Shigang. (2013). A Survey of Keystroke Dynamics Biometrics. *The Scientific World Journal*. 2013. 408280. 10.1155/2013/408280. pp. 1–24.
- [29] Gaines, R.S., Lisowski, W., Press, S.J. and Shapiro, N. (1980) Authentication by Keystroke Timing: Some Preliminary Results. No. RAND-R-2526-NSF.
- [30] Romain Giot, Mohamad El-Abed, Christophe Rosenberger (2011). Keystroke Dynamics Authentication. *Biometrics, InTech*, chapitre 8, 2011, 978-953-307-618-8.
- [31] Wikipedia, the free encyclopedia. (September 2007). *Biometrics*. Retrieved May 5, 2019, from <https://en.wikipedia.org/wiki/Biometrics>
- [32] N. L. Clarke and S. M. Furnell (March 2007). Advanced user authentication for mobile devices. *Computers & Security*, 26(2):109–119. doi: 10.1016/j.cose.2006.08.008. Retrieved from <http://dx.doi.org/10.1016/j.cose.2006.08.008>.
- [33] Bleha, Saleh Ali, Slivinsky, Charles, and Hussien, Bassam (1990). Computer-access security systems using keystroke dynamics. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.12. pp. 1217–1222. issn: 0162-8828.
- [34] Gaines, R. Stockton, Lisowski, William, Press, S. James, and Shapiro, Norman (1980). Authentication by keystroke timing: Some preliminary results. Tech. rep. DTIC Document.
- [35] Joyce, Rick and Gupta, Gopal (1990). Identity authentication based on keystroke latencies. In: *Communications of the ACM* 33.2, pp. 168–176.
- [36] Leggett, John and Williams, Glen (1988). Verifying identity via keystroke characteristics. In: *International Journal of Man-Machine Studies* 28.1, pp. 67–76.
- [37] Umphress, David and Williams, Glen (1985). Identity verification through keyboard characteristics. In: *International Journal of Man-Machine Studies* 23.3, pp. 263–273.
- [38] R. O. Duda, P. E. Hart, and D. G (2001). *Stork Pattern Classification*. John Wiley & Sons, Inc., second edition.
- [39] Killourhy, K. S., & Maxion, R. A. (2009). Comparing anomaly-detection algorithms for keystroke dynamics. 2009 IEEE/IFIP International Conference on Dependable Systems & Networks. doi:10.1109/dsn.2009.5270346
- [40] S. Cho, C. Han, D. H. Han, and H. Kim (2000). Web-based keystroke dynamics identity verification using neural network. *Journal of Organizational Computing and Electronic Commerce*, 10(4), pp. 295–307.
- [41] S. Haider, A. Abbas, and A. K. Zaidi (2000). A multi-technique approach for user identification through keystroke dynamics. *IEEE International Conference on Systems, Man and Cybernetics*, pp. 1336–1341.

- [42] E. Yu and S. Cho (2003). GA-SVM wrapper approach for feature subset selection in keystroke dynamics identity verification. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 2253–2257. IEEE Press.
- [43] P. Kang, S. Hwang, and S. Cho (2007). Continual retraining of keystroke dynamics based authenticator. In *Proceedings of the 2nd International Conference on Biometrics (ICB'07)*, pp. 1203–1211. Springer-Verlag Berlin Heidelberg.
- [44] S. Cho, C. Han, D. H. Han, and H. Kim (2000). Web-based keystroke dynamics identity verification using neural network. *Journal of Organizational Computing and Electronic Commerce*, 10(4), pp. 295–307.
- [45] B. Hwang and S. Cho (10–16 July 1999). Characteristics of auto-associative MLP as a novelty detector. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, volume 5, pp. 3086–3091.
- [46] Teh, Pin Shen, Yue, Shigang, and Teoh, Andrew Beng Jin (2012). Feature fusion approach on keystroke dynamics efficiency enhancement. In: *International Journal of Cyber-Security and Digital Forensics* 1.1, pp. 20–31.
- [47] Teh, Pin Shen, Teoh, Andrew Beng Jin, Tee, Connie, and Ong, Thian Song (2011). A multiple layer fusion approach on keystroke dynamics. In: *Pattern Analysis and Applications* 14.1, pp. 23–36.
- [48] Teh, Pin Shen, Teoh, Andrew Beng Jin, Tee, Connie, and Ong, Thian Song (2010). Keystroke dynamics in password authentication enhancement. In: *Expert Systems with Applications* 37.12, pp. 8618–8627
- [49] Bolle, Ruud M et al (2004). *Guide to biometrics*. Springer Science & Business Media, pp. 20, 21.
- [50] Al Solami, Eesa (2012). An examination of keystroke dynamics for continuous user authentication. PhD thesis. Queensland University of Technology.
- [51] Stewart, John C., Monaco, John V., Cha, Sung-Hyuk, and Tappert, Charles C (2011). 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.
- [52] Gaines, R. Stockton, Lisowski, William, Press, S. James, and Shapiro, Norman (1980). Authentication by keystroke timing: Some preliminary results. Tech. rep. DTIC Document,
- [53] Khanna, Preeti and Sasikumar, M (2010). Recognizing emotions from keyboard stroke pattern. In: *International journal of Computer Applications* 11.9, pp. 1–5.
- [54] Bartlow, Nick and Cukic, Bojan (2014). User Credential Hardening through Keystroke Dynamics. Tech. rep. West Virginia University.
- [55] Monroe, Fabian, Reiter, Michael K., and Wetzel, Susanne (2012). Password hardening based on keystroke dynamics. In: *International Journal of Information Security* 1.2 (2002), pp. 69–83.

- [56] Revett, Kenneth and Khan, Aurangzeb (2005). Enhancing login security using keystroke hardening and keyboard gridding. In: Virtual Multi Conference on Computer Science and Information Systems.
- [57] Stewart, John C., Monaco, John V., Cha, Sung-Hyuk, and Tappert, Charles C (2011). 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.
- [58] Bello, Luciano et al (2010). Collection and publication of a fixed text keystroke dynamics dataset. In: XVI Congreso Argentino de Ciencias de la Computación.
- [59] Giot, Romain, El-Abed, Mohamad, and Rosenberger, Christophe (2009). 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.
- [60] Li, Yilin et al (2011). Study on the BeiHang keystroke dynamics database. In: Biometrics (IJCB), 2011 International Joint Conference on. IEEE. 2011, pp. 1–5.
- [61] Varun Chandola, Arindam Banerjee, and Vipin Kumar (July 2009). Anomaly detection: A survey. ACM Comput. Surv. 41, 3, Article 15, pp. 1–58.
- [62] Moa Samuelsson, M. S. (2016). Anomaly detection in time series data, a practical implementation for pulp and paper industry (Master's thesis, CHALMERS UNIVERSITY OF TECHNOLOGY, Gothenburg, Sweden). Retrieved from <http://publications.lib.chalmers.se/records/fulltext/242944/242944.pdf>
- [63] Silvia Valcheva, S. V. (2018, March 11). Supervised vs Unsupervised Learning: algorithms, example, difference. Retrieved from <http://intellspot.com/unsupervised-vs-supervised-learning/>
- [64] Alex Irpan, A. I. (2018, February 14). Deep Reinforcement Learning Doesn't Work Yet. Retrieved from <https://www.alexirpan.com/2018/02/14/rl-hard.html>
- [65] Avril Coghlan, A. C. (2010). Using R for Time Series Analysis [graph]. Retrieved from <https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html>
- [66] Ralanamahatana C.A., Lin J., Gunopulos D., Keogh E., Vlachos M., Das G. (2005) Mining Time Series Data. In: Maimon O., Rokach L. (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA.
- [67] Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E. (Aug 2008). Querying and mining of time series data: Experimental comparison of representations and distance measures. Proc. VLDB Endow. 1(2), 1542–1552.
- [68] Bellman, R., Kalaba, R. (1959): On adaptive control processes. IRE Transactions on Automatic Control 4, pp. 1–9

- [69] Agrawal, R., Faloutsos, C., Swami, A. (1993). Efficient Similarity Search In Sequence Databases. In: Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms (FODO). pp. 69–84.
- [70] Berndt, D., Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In AAAI-94 workshop on knowledge discovery in databases 2, pp. 359–370
- [71] Salvador, S., Chan, P. (2007). Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.* 11, pp. 561–580.
- [72] Sankoff, D., Kruskal, J.B. (1983). *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley Publishing Co, Reading, Massachusetts.
- [73] Chen, Y., Nascimento, M.A., Chin, B., Anthony, O., Tung, K.H. (2007). Spade: On shapebased pattern detection in streaming time series. In: IEEE 29th International Conference on Data Engineering (ICDE). pp. 786–795.
- [74] Myers, C., Rabiner, L., Rosenberg, A. (1980). Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28, pp. 623–635.
- [75] Smith, T.F., Waterman, M.S. (1981). Identification of common molecular subsequences. *Journal of molecular biology* pp. 147, 195–197
- [76] Chen, L., Ng, R. (2004). On the marriage of lp-norms and edit distance. In: Proceedings of the Thirtieth international conference on Very large data bases (VLDB'04). vol. 30, pp. 792–803
- [77] Fuad, M.M.M., Marteau, P.F. (2008). The extended edit distance metric. In: International Workshop on Content-Based Multimedia Indexing (CBMI). pp. 242–248
- [78] Boreczky, J.S., Rowe, L.A. (1996). Comparison of Video Shot Boundary Detection Techniques. In: *Storage and Retrieval for Still Image and Video Databases IV*. pp. 170–179.
- [79] Needleman, S.B., Wunsch, C.D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, pp. 48, 443–453
- [80] Chen, L. Ozsu, M.T. (2005). Robust and fast similarity search for moving object trajectories. In: *SIGMOD*. pp. 491–502
- [81] Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E. (Aug 2008). Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proc. VLDB Endow.* 1(2), 1pp. 542–1552
- [82] Vlachos, M., Kollios, G., Gunopulos, D. (2002). Discovering Similar Multidimensional Trajectories. In: *IEEE International Conference on Data Engineering (ICDE)*. pp. 673–684
- [83] Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh (2013). Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2), pp. 275–309.

- [84] Rabiner, Lawrence. (2019). Fundamentals of speech recognition / Lawrence Rabiner, Biing-Hwang Juang. SERBIULA (sistema Librum 2.0).
- [85] Simple Machines. (2015). TypeWatch. Retrieved April 10, 2019, from <http://www.typewatch.net/>
- [86] Intensity Analytics Corporation. (2019). Intensity Analytics: Tick Stream. Key ID. Retrieved May, 2019, from <https://www.intensityanalytics.com/products/TickStream.keyid.aspx>
- [87] Biotracker. (2019). Plurilock DEFEND – Plurilock. Retrieved May 4, 2019, from <https://www.plurilock.com/products/defend/>
- [88] Keytrac, TM3 Software GmbH. (2016). Keyboard biometrics. Retrieved from <https://www.keytrac.net/>
- [89] Trustable password (2000). Retrieved from <http://www.imagicsoftware.com/>.
- [90] Biohec, & Checco Services, Inc. (2002). Keystroke Biometrics Advantage. Retrieved from <http://www.biohec.com/>
- [91] BehavioSec Inc. (2018). BehavioSec: Continuous Authentication Through Behavioral Biometrics. Retrieved from <http://www.behaviosec.com/>
- [92] W3C. (n.d.). HTML & CSS - W3C. Retrieved April 18, 2019, from <https://www.w3.org/standards/webdesign/htmlcss>
- [93] David Flanagan (1998). JavaScript: The Definitive Guide. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 3rd edition, 1998. ISBN 1565923928.
- [94] Christensson, P. (2013, May 23). WAMP Definition. Retrieved 2019, Apr 19, from <https://techterms.com>
- [95] Margaret Rouse, M. R. (2018). What is MySQL? - Definition from WhatIs.com. Retrieved from <https://searchoracle.techtarget.com/definition/MySQL>
- [96] Christensson, P. (2006). PHP Definition. Retrieved 2019, Apr 19, from <https://techterms.com>
- [97] Jyh-Shing Roger Jang, R. J. (2005, February 6). Data Clustering and Pattern Recognition. Retrieved from <http://mirlab.org/jang/books/dcpr/>
- [98] Ratanamahatana, Chotirat & Keogh, E. (2004). Everything you know about dynamic time warping is wrong.
- [99] Kurbalija, V., Radovanovic, M., Geler, Z., & Ivanovic, M. (2014). The influence of global constraints on similarity measures for time-series databases. Knowl.-Based Syst., pp. 56, 49-67.
- [100] Itakura, F. (1975). Minimum prediction residual principle applied to speech recognition. IEEE Trans. Acoustics, Speech, and Signal Proc., Vol. ASSP-23, pp. 52-72.
- [101] Sakoe, H. & chiba, S. (1978). Dynamic programming algorithm optimization from spoken word recognition. IEEE Trans. Acoustics, Speech, and Signal Proc., Vol. ASSP-26. pp. 43-49.

- [102] Dau, H.A., Silva, D.F., Petitjean, F. et al. Data Min Knowl Disc (2018) 32: pp. 1074. Retrieved from <https://doi.org/10.1007/s10618-018-0565-y>
- [103] Krisztian Buza, B. K. (n.d.). Person Authentication. Retrieved January 4, 2019, from <http://biointelligence.hu/typing-challenge/task1/index.php>
- [104] Zhu, Y. & Shasha, D. (2003). Warping Indexes with Envelope Transforms for Query by Humming SIGMOD 2003. pp. 181-192.
- [105] K.Revett, F. Gorunescu, M. Gorunescu, M. Ene, S. Magalhaes, and H. Santos (2007). A machine leaning approach to keystroke dynamics based user authentication. International journal of Electronic Security and Degital Forensics, 1(1): pp. 55-70.
- [106] Sultana, Madeena & Paul, Padma Polash & Gavrilova, Marina. (2014). A Concept of Social Behavioral Biometrics: Motivation, Current Developments, and Future Trends. Proceedings - 2014 International Conference on Cyberworlds, CW 2014. 10.1109/CW.2014.44.
- [107] Gunetti, D. and Picardi (2005). Keystroke Analysis of Free Text. ACM Transactions of Information and System Security (TISSEC), 8(3): pp. 312-347.
- [108] D. Shanmugapriya, Mrs and Ganapathi, Padmavathi. (2009). A Survey of Biometric keystroke Dynamics: Approaches, Security and Challenges. International Journal of Computer Science and Information Security. 5. pp. 115-119.
- [109] K. Buza (2016). Person Identification Based on Keystroke Dynamics: Demo and Open Challenge, 28th International Conference on Advanced Information Systems Engineering (CAiSE'16) Forum.
- [110] Furbush, J. (May, 2018). Machine learning: A quick and simple definition. Retrieved May 8, 2019, from <https://www.oreilly.com/ideas/machine-learning-a-quick-and-simple-definition>