FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Behavioural Biometrics in the World Wide Web

Master of Informatics Engineering
Thesis
Report

Leandro Silva
lasilva@student.dei.uc.pt

Supervisors:
Bernardo Patrão
Ernesto Costa

Date: July 01st, 2014

Abstract

Behavioural biometrics such as keystroke dynamics is a reliable human trait that can be used to successfully validate a user claimed identity on a given information system. Apart from its several advantages over traditional authentication mechanisms, the intrinsic time dimension of these distinctive typing patterns also allows for a real time and continuous user authentication.

Being the internet security a recurrent and growing concern, this thesis asserts the actual applicability of such behavioural biometrics technique in a web environment, suggesting an elegant and transparent intrusion detection solution that requires no additional hardware or software. The security of users online is ensured just by the mere consequence of typing.

Keywords

Artificial Intelligence, Behavioural Biometrics, Continuous Authentication, Internet Privacy, Internet Security, Intrusion Detection System, Keystroke Dynamics, World Wide Web

Index

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 STATE OF THE ART 3

2.1. BIOMETRICS 3
2.2. KEYSTROKE DYNAMICS 6
2.3. WORLD WIDE WEB 7
2.4. MARKET ANALYSIS 10

CHAPTER 3 REQUIREMENTS ANALYSIS 13

3.1. MODULAR STRUCTURE 13
3.2. USE CASES 13
3.3. ACTIVITY DIAGRAMS 15
3.4. INTERFACE PROTOTYPES 17
3.5. FUNCTIONAL REQUIREMENTS 19
3.6. NON-FUNCTIONAL REQUIREMENTS 20

CHAPTER 4 DESIGN AND ARCHITECTURE 21

4.1. ARCHITECTURE DESIGN DECISIONS 21
4.2. HIGH-LEVEL SYSTEM ARCHITECTURE 22
4.3. STUDY AND SELECTION OF TECHNOLOGIES 23
4.4. DETAILED SYSTEM ARCHITECTURE 25
4.5. DATA MODEL 29

CHAPTER 5 DEVELOPMENT 34

5.1. INTRUSION DETECTION WEB CLIENT 34
5.2. INTRUSION DETECTION SERVICE 46
5.3. DYNAMIC THRESHOLD 54

CHAPTER 6 VALIDATION RESULTS 57

6.1. OBSERVATIONAL STUDY 57
6.2. RESULTS 58

CHAPTER 7 CONCLUSIONS 69

7.1. RETROSPECTIVE 69
7.2. FUTURE WORK 71

REFERENCES 73

ANNEX 75

Figures List

FIGURE 1 - RECEIVER OPERATING CHARACTERISTIC (ROC) CURVE 5
FIGURE 2 - A GENERAL FRAMEWORK FOR THE KEYSTROKE DYNAMICS EVALUATION PROCESS 6
FIGURE 3 - MAIN USE CASES 13
FIGURE 4 - UC 1.1 – USER TYPING 14
FIGURE 5 - UC 2.2.1 - MANAGE ROLE SETTINGS 15
FIGURE 6 - AD 01 - VALIDATE USER 16
FIGURE 7 – HOME SECTION – CUSTOMER ADMIN DASHBOARD 18
FIGURE 8 – HOME SECTION – USER DASHBOARD 19
FIGURE 9 - FUNCTIONAL REQUIREMENTS 20
FIGURE 10 - NON-FUNCTIONAL REQUIREMENTS 20
FIGURE 11 - HIGH-LEVEL ARCHITECTURE DIAGRAM 22
FIGURE 12 - SYSTEM COMPONENTS DIAGRAM 26
FIGURE 13 - BIOMETRICS GATHERING COMPONENTS DIAGRAM 28
FIGURE 14 - BIOMETRICS GATHERING CORE INTERFACES 29
FIGURE 15 - INTRUSIONDETECTIONWF CLASS DIAGRAM 30
FIGURE 16 - PARTIAL VIEW OF THE TYPEWATCH WEB DATABASE MODEL DIAGRAM 32
FIGURE 17 - FINITE STATE MACHINE DIAGRAM FOR A GIVEN KEY 36
FIGURE 18 - ELEMENTARY FEATURES (DWELLS AND FLIGHTS) FOR THE SEQUENCE "ABBA" 37
FIGURE 19 - COMPOSITE FEATURES (N-GRAPHS) FOR THE SEQUENCE "ABBA" 38
FIGURE 20 - FEATURE EXTRACTION WORKFLOW 39
FIGURE 21 - ENROLMENT PROGRESS NOTIFICATION 45
FIGURE 22 - SUCCESSFUL USER AUTHENTICATION 45
FIGURE 23 - INTRUSION DETECTED 46
FIGURE 24 – RE-AUTHENTICATION PROCESS 46
FIGURE 25 - BIOMETRICS SAMPLE EVALUATION 48
FIGURE 26 - OUTLIER REMOVAL - DWELL DISTRIBUTION COMPARISON (1 SAMPLE) 50
FIGURE 27 - OUTLIER REMOVAL - CUMULATIVE DWELL DISTRIBUTION COMPARISON (10 SAMPLES) 50
FIGURE 28 - PROFILE IDENTIFICATION EMULATOR 57
FIGURE 29 - TIME DURATION COMPARISON (CHROME VS INTERNET EXPLORER) 60
FIGURE 30 - ARTIFICIAL ATTACKS - INTERNAL PROFILE ID 5B… (BOX PLOT) 61
FIGURE 31 - ARTIFICIAL ATTACKS - INTERNAL PROFILE ID 5B… (SCATTER PLOT) 61
FIGURE 32 - EVALUATION RESULTS FOR THE TARGETED 5B… BIOMETRICS PROFILE 64
FIGURE 33 - FRR AND FAR COMPARISON 65
FIGURE 34 - RESULTS GENERALIZATION – 12 USERS 66

Table List

TABLE 1 – ACRONYMS LIST 6
TABLE 2 - COMPARISON MATRIX OF KEYSTROKE DYNAMICS APPLICATIONS 12
TABLE 4 - SAME ORIGIN POLICY 43
TABLE 5 - AVERAGED TIME DURATION OF THE FEATURE EXTRACTION PROCESS PER SAMPLE 59
TABLE 6 - AVERAGED TIME DURATION OF THE SAMPLE EVALUATION PROCESS 59
TABLE 7 - AVERAGED TIME DURATION OF THE THRESHOLD UPDATE PROCESS 60
TABLE 8 - BIOMETRICS SAMPLES USED ON THE ARTIFICIAL ATTACKS AGAINST THE 5B… BIOMETRICS PROFILE 62
TABLE 9 - CONFUSION MATRIX RESULTING FROM THE ARTIFICIAL ATTACKS AGAINST THE 5B… BIOMETRICS PROFILE 62
TABLE 10 - FAR AND FRR RESULTING FROM THE ARTIFICIAL ATTACKS AGAINST THE 5B… BIOMETRICS PROFILE 62
TABLE 11 - FRR AND FAR RESULTING FROM EACH OF THE ARTIFICIAL ATTACKS SETS PERFORMED 63
TABLE 12 - EVALUTIONS PERFORMED FOR THE TARGET 5B... BIOMETRICS PROFILE 64
TABLE 13 - CONFUSION MATRIX RESULTING FROM THE SUPERVISED EVALUATIONS AGAINST THE 5B… BIOMETRICS PROFILE . 64
TABLE 14 - FAR AND FRR RESULTING FROM THE SUPERVISED EVALUATIONS AGAINST THE 5B… BIOMETRICS PROFILE 64
TABLE 15 - FRR AND FAR RESULTING FROM EACH OF THE SUPERVISED EVALUATIONS OBSERVED 65
TABLE 16 - CONFUSION MATRIX GLOBAL AVERAGE RESULTING FROM ALL THE SUPERVISED EVALUATIONS OBSERVED 65
TABLE 17 - RESULTS GENERALIZATION – 12 USERS 66
TABLE 19 - MOST RELEVANT FEATURES FOR THE 5B… BIOMETRICS PROFILE 67
TABLE 20 - LEAST RELEVANT FEATURES FOR THE 5B… BIOMETRICS PROFILE 67
TABLE 21 - FEATURE RELEVANCE FOR THE 5B… BIOMETRICS PROFILE 68

Acronyms List

API	Application Programming Interface
CORS	Cross-Origin Resource Sharing
DOM	Document Object Model
EER	Equal Error Rate
FAR	False Acceptance Rate
FRR	False Rejection Rate
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ID	Identification
JSON	JavaScript Object Notation
KD	Keystroke Dynamics
PasS	Platform-as-a-Service
REST	Representational State Transfer
RIA	Rich Internet Applications
ROC	Receiver Operating Characteristic
SaaS	Software-as-a-Service
SDK	Software Development Kit
SOA	Service-oriented Architecture
SOAP	Simple Object Access Protocol
UDDI	Universal Description, Discovery and Integration
UI	User Interface
WSDL	Web Services Description Language
WWW	World Wide Web

Table 1 – Acronyms List

Chapter 1
Introduction

Internet security has been a frequent and significant subject for both users and organizations in a society that is already dependent on the use of information technologies to accomplish their common daily activities. On an organizational level, this strong dependency reflects the escalation of the service-oriented industries over the traditional manufacturing ones, which goes side-by-side with the contemporary knowledge-driven economic models.

Correspondingly, from a simple end-user perspective, this digital dependency is expressed by the massive adoption of internet services such as e-commerce, online banking, instant messaging, and social networks. These popular services, as virtually every online service, typically require the user to hold – and sometimes expose – an identity online.

This exposure reinforces the importance of having stronger and reliable internet security mechanisms, possibly by means of new authentication approaches and techniques, providing an effective barrier against a wide range of cyber-attacks, which are frequently based on identity theft.

In the process of identity validation for general access control purposes – and considering a wide range of scopes and applications – one can authenticate a user by assessing something the user has, such as a physical token, something the user knows, such as a secret pin or password, or something the user is, such as its biometrics. When running a comparison between them, the fact that biometrics cannot be forgotten, lost, or easily stolen, makes it a promising alternative solution for user authentication, or at least, a complementary one to other traditional methods in a scenario of multi-factor authentication.

Furthermore, behavioural biometrics, such as the gait or the handwritten signature, adds an important time dimension, making it suitable for a continuous authentication process. This opposes to physiological biometrics, such as the human DNA, iris, or fingerprints, which are essentially time invariant [1]. Being keystroke dynamics a type of behavioural biometrics, and knowing that a large amount of online services still involves the user typing as a way of inputting data, it seems a natural approach to try to use such keystroke dynamics as authentication technique, in a will to bring an extra layer of security to internet users.

However, one of the main challenges that arise with the use of behavioural biometrics – and with biometrics in general – is the classic trade-off associated with the interdependent false rejection and false acceptance rates involved in these user authentication attempts. Therefore, it is crucial to minimize these rates, hence, maximizing the evaluation performance accuracy.

Despite of being a highly distinctive biometrics, another factor that may contribute to the rise of these false rejection and false acceptance rates is the low permanence of keystroke dynamics. This is linked to the human behaviour itself, which is susceptible to change over time, and can also be influenced by both environmental and emotional factors.

The main goal of this thesis is to project and develop a solution that takes advantage of keystroke dynamics in order to verify the user claimed identity on a web environment, aiding for the detection of potential illegitimate users.

In more detail, this thesis ought to contribute to the conception and development of an intrusion detection service that is especially tailored for the web. It is to be developed using browser native and standard technologies, being equally of easy system integration, with no additional hardware or software requirements. The protection of the user's privacy and data confidentially are also worth mentioning objectives of this thesis.

Furthermore, the proposed solution may be transparent to the user, non-obtrusive, responsive and accurate in its identity assessments, being conceptually based on the core intrusion detection algorithms i.e. statistical classifiers that are already being applied by an existing desktop-based intrusion detection solution developed by Watchful Software.

In order to validate this thesis, some set of experiments and observational studies are to take place. This validation phase will consist of the actual use of the intrusion detection service, involving at least two different profile groups: a group of people that are familiar with the concept of intrusion detection using keystroke dynamics i.e. the team from the Watchful Software; and an external independent group of people that are not aware of this concept of using keystroke dynamics for user authentication.

This thesis involves the application of several Software Engineering abilities, being focused on important wide topics such as Internet Security and Artificial Intelligence.

This dissertation is organized as follows. After this introduction, in the Chapter 2 it is presented the State of the Art of Behavioural Biometrics and the World Wide Web, including a Market Analysis on the subject. The Chapter 3 follows with the project Requirements Analysis. After this, the Design and Architecture of the system proposed is introduced in the Chapter 4. The actual Development is addressed in Chapter 5. The Validation Results are presented in the Chapter 6, and the thesis Conclusions are presented in the Chapter 7, which also includes a future work description. The References and Annex documentation are placed at the end of this report.

Chapter 2
State of the Art

This chapter presents the state of the art of behavioural biometrics and its association with the World Wide Web, as well as a market analysis on keystroke dynamics applications.

2.1. Biometrics

Biometrics is defined as the measurable behavioural and physiological traits of an individual that are distinctive enough to differentiate it among the population [2].

Biometrics is a technique that has been used for ages, even unconsciously, as we can recognize a known or familiar person just by earing its voice or simply by looking at its face [3]. Fortunately, each one of us is unique, and that fact resembles on many of our physical and behavioural characteristics, therefore, biometrics can take advantage of those unique traits in a scenario of identification validation [4].

In the last decades, automatic systems had been created to make good use of this technique. Think of fingerprint detection, iris recognition, handwritten signature comparison, or keystroke dynamics. [4] [5] Some of these systems have been proven to be a reliable and effective alternative to more traditional authentication techniques. However, despite the overall good performance, there is still plenty of room for improvement.

There are two main categories of biometrics, the physiological ones and the behavioural ones. Both biometrics has its strengths and weaknesses, and some are more suitable for a given scenario than others. There are also environments that favours the use of multi-mode biometrics [6], an approach that combines different biometric techniques to add multiple and distinct layers of protection [5]. However, due to its added complexity, cost, and management effort, it is not always desirable to use multi-mode biometrics, and being that the case, one must know which biometric technique fits better.

2.1.1. Physiological Biometrics

Physiological biometrics concerns with the physical characteristics of the individual. Excluding some extreme cases, these measurable traits cannot be easily stoled or copied, and are stable over time, because they are related to the somewhat immutable distinct aspects of the human body, and focus on properties such as texture, colour, size, shape and composition. The geometry of the face, the fingerprints friction ridges, the iris and retina patterns of the eye, the DNA sequence, and the ear topology, are some common instances.

One of the drawbacks of the application of physiological biometrics techniques is that they can be intrusive to the user. It's true that these techniques typically only imply a one-step enrolment procedure, but that step can be bothersome, and also seen as an invasion of privacy. Imagine the process associated with the obtention of blood samples for the DNA sequence analysis, or the one associated with the measure of the iris or retina of the eye. That could make users feel uncomfortable. Even when the enrolment phase is concluded, its actual use, in most cases, requires additional equipment in order to be applicable.

2.1.2. Behavioural Biometrics

Behavioural biometrics – or behaviometrics – focuses on the unique behavioural characteristics of an individual, and are usually measured for the purpose of identification validation. Some popular techniques include voice, gait, and handwritten signature recognition, as well as mouse and keystroke dynamics.

When compared to the physiological biometric techniques, the behavioural ones are substantially less intrusive to the user. There are two main reasons for this. First, behavioural biometric systems usually don't require additional equipment or hardware. Secondly, its application is typically transparent to the user. This is backed up by the fact that, in most cases, the user does not need to - and it's not desirable to – change its behaviour. Additionally, in some cases the end-user, or the intruder, does not even notice that he is being observed by the biometric security system.

Despite its strengths and strong potential, behavioural biometrics displays some concerns that need to be addressed. It is pointed out in the literature [7] that behavioural biometrics techniques are considered less accurate than the physiological ones, since the target traits are more likely to change over time [1]. This is due to the fact that the human behaviour is in part influenced by emotional and environmental factors, such as stress, fatigue, illness, body injuries, noise, and other distractions [7]. Besides, the continuous practice of the associated activity, or the lack of it, can also be an important influential factor. Nevertheless, to address these limitations, biometric systems employing behavioural biometrics are putting more effort on smarter and more dynamic approaches on the analysis of the features gathered, by adopting a continuous collection and monitorization of the targeted behaviour.

Therefore, one thing to consider is the improvement of the enrolment phase. Typically behavioural biometrics requires an extended registration phase, also called the training phase, where the targeted features are extracted over time in order to build a representative user profile. This is not always possible to accomplish, usually due to environmental and time restrictions. Still, there is a large set of real-world scenarios in which this kind of techniques applies, mostly – but not limited to – logical access related applications. For instance, one may think of business information systems or web-based personal applications, such as social networks or e-mail clients.

Like the physiological biometric characteristics, the behavioural ones – except on extreme situations – cannot be stolen or lost. Matter of fact, they cannot even be easily imitated or reproduced. This is particularly true for behavioural biometrics, because most of these traits observed in the human behaviour are idiosyncratic and neurophysiologically induced, hence, extremely hard to duplicate. The way we write our handwritten signature, the way we walk or the way we type in a keyboard, are perfect good examples.

2.1.3. Biometric System Evaluation

The purpose of a biometric security system is to prevent inside access from potential intruders by measuring and analysing user targeted behavioural or physiological traits. However, due to its limitations, with biometrics, it is not always possible to correctly validate user's access to a system.

There are two major undesirable situations that may occur as a result of an incorrect verification. One is to falsely classify a legitimate user as an intruder to the security system, the other being to falsely classify an intruder as a legitimate user. In the literature [8] [9] [17], these errors are measured by the False Rejection Rate (FRR) and False Acceptance Rate

(FAR), respectively. These are actually excellent accuracy performance indicators for a biometric security system.

FRR and the FAR are interdependent and inversely proportional. There is a valid reason for this. Any classification model designed to validate a user needs to consider a given safety threshold for that same user, and evaluation scores that don't fit in that threshold are expected to be from a potential intruder. The problem here resides in the fact that, when the threshold is too small, legitimate users can somewhat yield a score out of that threshold, due to some natural variations in their behaviour, hence, being falsely rejected by the system. On the other hand, if the threshold is to large, behavioural score evaluations from a potential intruder may fit in the threshold, hence, being falsely accepted as a legitimate user. So, one must carefully adjust the threshold in order to yield good FRR and FRR values.

Depending on the application environment, sometimes it is desirable to improve the FAR over FRR. It means that in some systems, higher FRR values may be easier to bear, because they probably reflect an increase in security, but unfortunately, it may also replicate user efforts to manually reauthenticate – if the system applies a non-passive reaction mechanism – which can be annoying to the legitimate user. On some other systems, it may be preferable the opposite approach – the improvement of FRR over FAR.

Another related standard performance indicator is the Equal Error Rate (EER) [9] [10], which is the value observed when the FRR and FAR values are equal. This is useful to perceive the overall performance accuracy of a given biometric security system. The lower the EER, the better, however, as of today, it's virtually impossible to have a zero EER value.

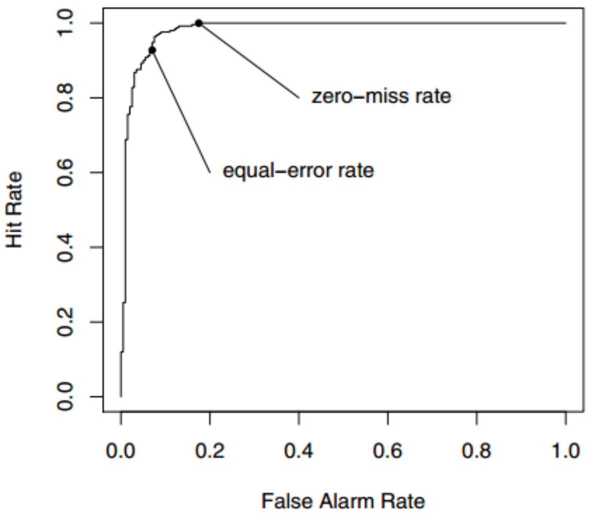


Figure 1 - Receiver Operating Characteristic (ROC) Curve

An extra and useful exercise is to plot all of the possible FRR values – usually as the related Probability of Verification, or Hit Rate, given by the formula $(1 - FRR)$ – against the correspondent FAR values, as shows in Figure 1 [11]. This results in a curve on the graph, called the Receiver Operating Characteristic (ROC curve) [8] [10], which depicts the theoretical system performance for every possible threshold settings, and consequently FRR and FAR pairings. In addition, the resulting curve also depicts another interesting; the area under that line, which is usually referred as the AUC (area under the curve) [11]. In this context, when normalized, this area gives the probability of a randomly chosen legitimate user score to be higher than a score from a randomly chosen intruder. The higher the AUC values, the better the system accuracy. Ultimately, these types of graphs are useful to depict an accuracy performance comparison between two or more biometric systems.

2.2. Keystroke Dynamics

Keystroke Dynamics is the process of analysing the typing rhythms and patterns of an individual using a keyboard or a keypad, in order to distinguish that individual among the population. It is a biometric technique that goes under the category of behavioural biometrics, being the measured behavioural traits usually quantified in terms of duration time of events – such as the hold time of a specific key, or the time elapsed between the release and the depression of two consecutive keys, among others.

As seen earlier, this added time dimension is one of the qualities that differentiate behavioural biometrics from physiological biometrics, and what makes keystroke dynamics a promising solution for continuous identity verification.

2.2.1. Verification Methods

Keystroke dynamics can be applied using two different major methodologies.

In a simpler approach, it can be used as a password hardening, which is a technique that not only verifies if there is a password match, but also if the pattern of typing matches. In fact, most studies on keystroke dynamics address this kind of user authentication [7].

However, more complex systems go beyond password hardening, applying keystroke dynamics in a continuous manner, by analysing the user typing behaviour regularly, even after a successful login authentication [7]. To make this line of thought more formal, here, there are identified two major verification methods: the static-text authentication, as in password hardening; and the free-text authentication, as in continuous authentication.

2.2.2. The Generalized Process for Keystroke Dynamics

The application of keystroke dynamics typically follows a well-structured procedure, consisting of a series of activities, usually defined as follows: feature acquisition, feature extraction, template storage, and classification [14] [10], as shown in figure 2 [14].

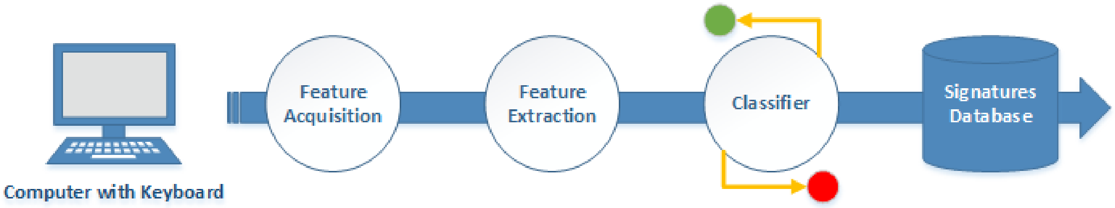


Figure 2 - A General Framework for the Keystroke Dynamics Evaluation Process

The data flow in this process is quite consistent, yet there is a variance in the flow in order to reflect two distinct phases, namely, the enrolment phase, and the testing phase [10]. While the template storage is crucial for the former one, the classification is crucial for the last one. In a sense, initially, a given user process reflects the enrolment phase, then, when a representative user profile is built, the test phase begins. However, this linear thinking is not always observed, and this relates to the following.

Both static-text and free-text approaches follow the same process, but due to its dynamic nature, free-text verification techniques tend to extend the enrolment phase over time. This is related to the more continuous update of the user profile to reflect a potential shift in the behavioural pattern. Thus, the enrolment phase can coexist with the testing phase.

In detail, the enrolment phase consists on the capture, process, and storage of biometric data samples from a given user in order to build a representative biometric profile for that same user [10]. The testing phase consists on the capture, process, and matching of biometric data samples from a given user against the templates stored during the enrolment phase, in order to validate that user in the system [10].

Typically, the classification process yields a decision based on the validation output. Roughly, there are two possible results; the user is considered a legitimate user, or the user is considered an intruder. Based on this, the system can trigger some predefined actions in order to maintain its security. These activities can be either passive, or active. A passive one takes a mere informative approach, while the active one takes an effectible action-based approach.

2.3. World Wide Web

Nowadays, our society prices knowledge higher than never before. Some of that knowledge is extracted from information that takes essentially a digital form, hence, being easier to access and share. This mind set is well-established, and this – together with the boost of the concept of critical data – outlines the desire for improved security mechanisms on existing web-based applications.

The current state of the art of online user authentication services is deeply rooted on the traditional password-based authentication mechanisms. These old-fashion security mechanisms are not secure at all, as passwords are usually bothersome and hard to manage. Even with the new approach of using Unique ID providers such as Google Plus or Facebook, there are still important security flaws within this approach. That is, password-based authentication is a one-set validation process, so they don't provide a continuous user authentication.

A possible solution to address this problem is one that relies heavily on biometrics, more precisely on keystroke dynamics. As of today, despite of being a promising solution, the application of behavioural biometrics on a web context is yet to be fully explored. There are some concerns regarding user safety and privacy that must be taking into account, as well as some more technical limitations or considerations that need to be addressed.

2.3.1. Web Standards

The technical implementation of keystroke dynamics in the World Wide Web – as most other web-based user interaction applications – should follow and comply with the current web standards.

The Word Wide Web Consortium (W3C) is formed by a set of organizations responsible for the proposal and definition of such standard specifications. They provide, in the particular case of HTML5, an "openly-produced and vendor-neutral language" that can be implemented in a "wide range of competing products, across a wide range of platforms and devices". This is a huge advantage to other proprietary or third-party alternatives, such as Adobe Flash or Microsoft Silverlight [15].

The technologies typically involved in client-side user interaction applications are mostly based on HTML, CSS, and JavaScript, which are native technologies supported by virtually any web-browser. These web standards go deep in detail, and include a number of code validations and implementation requirements, as well as some accessibility considerations, in order to provide equal access to users with diverse abilities, hence, also improving overall user experience.

Additionally, one must also not put on hold the implementation of fall-back techniques in order to add support for bleeding-edge browser features that are not yet fully supported by modern browsers, as well as to add support for legacy browsers. This is important to increase the overall compatibility of the resulting web application.

2.3.2. Security

In a web context, it is always important to think of security. This subject gains an added importance when user private data is involved. With keystroke dynamics, the user typing behaviour is the only target of analysis, however, it must be clear to all parts involved that the systems employing such biometrics are essentially concerned with timing of events, and no record of semantic data is involved in process, thus, the user's privacy and confidentiality is respected.

Even knowing this, some security measures are usually needed to protect the system and its biometrics against eventual attacks, as there is recognized some common security weak points on web-based systems or applications. They are in part due to the networking involved.

One of the most popular security attacks on the web is the Man-in-the-Middle attach, where the attacker performs some kind of eavesdropping, being secretly acting, that is, imitating one of two communication end-points actors, thus having full control of the communication process, in which the victim might not even notice that is being attacked. This is typically minimized by relying on some kind of cryptographic measures, protecting the communication of data. One of the strongest security measures used today to address this problem is the use of a more secure communication protocol, for instance, HTTPS.

Authentication and Authorization

Authentication refers to the identification of a user in the web, while authorization deals with the permissions and roles associated with such identified user. Despite the differences, these concepts are associated, as authorization usually depends of authentication.

When a given web-server wants to know who is accessing their website, it typically relies on a username and password based authentication mechanism, so it can identify the person who made the request. This type of mechanism is widely used and accepted, besides its known drawbacks. For instance, it can be bothersome for a user to manage their login information, especially if it is hard to remember, or if the user uses multiple and different login details for multiple websites, which is, yet, in some way, considered a good practice.

In later years, some ideas have been put in practice in an attempt to help users to manage their usernames and passwords. Consequently, some standards and identity providers where especially created to provide a global and unique ID for each user, so they can use it in a wide range of supported websites, for authorization purposes, but also for registration setups. These identity providers can be seen as trusted third-party entities that help to perform the authentication and/or registration process between the user and those websites, saving them time and effort.

Privacy and Confidentiality

Being privacy applied to the person, and confidentiality applied to the data, one can approach user privacy and confidentiality on online biometric security systems by asking a set of appropriate questions.

Is the data, gathered by a biometric system, equal in nature to the one provided by the user? Are users aware that a biometric system is being employed? Do biometric systems store information that can be stoled or reconstructed? How a biometric system can ensure data security during the transmission and storage phases? Who can access such biometric data? Are there any trusted third-party entities involved?

For keystroke dynamics, the answers to these questions can be expressed as follows:

In the current process of keystroke dynamics, there is no semantic value attached to the user data gathered, hence, the text that is actually typed is not tracked by the system. This process is also transparent for the user, so eventually some users might not even be aware that the security system is being employed; however, they can always be informed.

The information stored in the template data storage is virtually impossible to be reconstructed, so if stolen, it is useless. This is due to the fact that no structured semantic text are stored on the system, and the information that is stored only relates to sets containing the timing duration of events, which do not necessarily represent the actual sequence of typing.

In a willing to protect biometric data during its transmission and storage, some security measures are put in practice. The network communication usually relies on encrypted connections, and the data itself can also be encrypted during the storage phase. Regarding the access and visualization of biometrics data, it is typically restricted to administrators. However, as assumed earlier, this biometrics data cannot be reconstructed. This kind of information is usually processed and plotted on dashboards, or listed on reports, for further analysis.

2.3.3. Time Accuracy on the Web

Keystroke dynamics rely on the measure of the typing rhythms and patterns of an individual, and these measures are usually expressed by the time duration of specific events. The resolution of such timing data is usually high, being a millisecond time resolution the lowest desired [16]. Additionally, the variance associated with the time captured must be as minimal as possible. The higher the resolution, the potentially smaller is the associated time variance.

In a web context, the timing of events are typically captured by JavaScript native engines on web browsers – potentially using different browsers – so some relevant time inconsistencies may be found.

The native JavaScript mechanism used to track time returns a timestamp in milliseconds, accessed through the method Date.now(), which is returned as a number that counts the time elapsed since 1 January 1970 00:00:00 UTC. This is a widely used method to retrieve the actual time using a web browser; however there is a new and alternative method that can be also used. It is a routine called *now()*, and it is accessed through the Performance Web API Interface, which belongs to the new High Resolution API. It is an accurate method, with precision to a thousandth of a millisecond, and can be easily implemented, but unfortunately it is not yet fully supported by all major browsers, being also not supported by legacy browsers.

The use of such new time resolution may contribute to the effectiveness of keystroke dynamics based security systems; however, its actual influence not known.

2.4. Market Analysis

Besides the complex and large business decision making process associated with the market analysis, in a simpler approach, it can help to understand the concrete applications of keystroke dynamics in current real-world scenarios. It can also help to depict some trends, user needs, expectations, and to ideate new applications for keystroke dynamics.

Nowadays, the application of keystroke dynamics focuses on password hardening mechanisms or on intrusion detection systems offered by specialized internet security business companies that provide solutions targeted for information technology organizations and institutions – without major geographical restrictions. A recent report [24] shows that keystroke dynamics is being currently applied at both enterprise and government sectors. Additionally, it is noted that, as of today, the US alone has the largest market share, and the Asian-Pacific region owns the fastest growing market share. In overall terms, the keystroke and typing dynamics market is expected to continue to grow over the next years [24]. This is justified by the increasing concern about information security and by the ongoing expansion of its application areas.

2.4.1. Solutions on the Market

Currently, there are only a few business companies with products on the market that deal with intrusion detection and related security systems that employ the use of keystroke dynamics. One of them is Watchful Software.

TypeWATCH by Watchful Software

TypeWATCH is an intrusion detection system developed by Watchful Software, a member of the Critical Group, which is headquartered in Coimbra, Portugal.

This solution provides continuous authentication using free-text techniques, assuring security over the entire session. The enrolment phase is dynamic, being the user profile updated in the data store regularly. The enrolment setup is also really quick, being the user able to use the software in no time.

TypeWATCH has configurable levels of security, so a user can adjust the sensibility of the algorithm according to its needs. It is also possible for the user to hold the verification process for a predefined amount of time in a secure way, as he may feel that he is going to type abnormally during that time. This can be due to some untypical reasons, such as a change in the emotional or a change in the environmental states.

Currently, this software is easy to install, being available for desktop. The company offers a free-trial version for demonstration purposes, and also hosts an online demo and FAQ support section on their web site.

At this time, the company also plans to do some mouse and pointer dynamics research and development activities in order to improve its biometrics application.

2.4.2. Comparison Matrix

The following feature comparison matrix expresses a visual informative comparison between the all the studied solutions on the market. The comparison is structured according to some predefined criteria, which are described as follows:

Authentication Approach: Refers to the methodology applied for user authentication, which can follow a static-text approach, as in password hardening, or a free-text approach, more suitable for continuous authentication.

Enrolment Process: Refers to the way the user is enrolled in a system, which relates to the learning phase, in which the system builds the user biometric profile by gathering some user biometric samples. This can be done following a static approach, building the profile only at the beginning, or following a dynamic approach, by regularly updating the user profile, hence, adapting the profile to the user behavioural changes over time.

Classification Process: Refers to the methodology followed for user identity verification. It can be done using a one-class classifier, which takes new sample for evaluation and matches that sample against the user profile, in order to confirm or refute the association between the user and the given biometric sample. The multi-class classifier takes new sample from an a unknown user and matches it against the profile of all the users enrolled to the system, discovery the user who potentially provided that sample, or marking the sample as belonging to a unknown user, who can be marked as a possible intruder in the system.

Application Environment: Refers to the running environment of the client application. It can be classified as desktop-based application, or as a web-based application, running in a web-browser.

Additional Biometrics: Indicates if there is an additional biometric associated with the product, such as voice recognition or mouse dynamics, besides keystroke dynamics.

Proprietary Software Constraints: Indicates if there are software constraints, such as the installation of proprietary software in the client side – like Flash or Java Applets – in order to run the software.

Dashboard / Configuration Panel: Indicates if the solution provides a dashboard or a centralized administrator interface, in which the administrator can perform some sort of data analysis, configuration management, or monitoring of users and user activity on the application.

Delivery: Refers to the way the solution is delivered to the client. It can be done by local installation, or in the cloud.

Integration & Development: It indicates if the solution provides an API or a SDK, so the customer can customize the solution integration in its own system.

Pricing: Refers to the pricing modality offered by the company for that product. The product can be free, or offered as a trial version, or with a flexible pricing plan, which is one that is automatic adjusted depending of the usage rate.

	TypeWATCH Desktop	AuthenWare	Behavio E.	Behavio W.	BioChec	iAM	TypeSense	KeystrokeID	CVMetrics	KeyTrac	Biotracker	Pluripass
Authentication Approach												
Password Hardening		✓		✓	✓	✓	✓	✓	✓	✓		✓
Continuous	✓		✓	✓					✓		✓	
Enrolment Process												
Dynamic	✓		✓	✓	✓		✓		✓		✓	
Static							✓					✓
Classification Process												
One-Class	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Multi-Class		✓	✓	✓								
Application Environment												
Web-based		✓		✓		✓	✓	✓	✓			✓
Desktop-based	✓		✓					✓	✓			
Additional Biometrics												
Mouse Dynamics		✓	✓						✓		✓	
Other		✓	✓			✓						
Proprietary Software Constraints												
Client-Side						✓	✓					
Dashboard / Configuration Panel												
Admin. GUI Panel	✓											
Log Reports	✓					✓					✓	
Security Level Config.	✓	✓			✓							
Delivery/Deployment												
Cloud-based		✓			✓						✓	
Local Installation	✓	✓			✓						✓	
Integration & Development												
API										✓		
SDK					✓							
Pricing												
Free Trial	✓	✓								✓		
Flexible Plans						✓				✓		

Table 2 - Comparison Matrix of Keystroke Dynamics Applications

2.4.3. Research Conclusions

From the research, it is possible to observe that there is market for the type of the solution proposed, as there are only a few solutions that are simultaneously web-based and targeted for free-text.

Chapter 3
Requirements Analysis

The chapter introduces the requirements functional and non-functional for this project, as well as some complementary artefacts, such as Interface Prototypes and major Use Cases.

The initial phase of the requirements analysis started with a reunion with some stakeholders, particularly with the Project Manager, and the Product Owner. After that meeting, the high-level-requirements were defined, and later improved and specified. This served as the base for the detailed specification of the Requirements here introduced.

3.1. Modular Structure

In order to better define the project requirements, the functional requirements were grouped into modules. Some of the modules are associated with system-level requirements, while others correspond to user-level requirements.

In the analysis of user-level requirements, there were identified two main actors of the system, the User, and the Customer Administrator. The functions performed by the Customer Administrator are grouped into the Customer Administrator Dashboard module, while some of the functions performed by the User are grouped in the User Dashboard module, especially the ones related with application monitoring and application settings.

This modular structure helps to map some functional requirements with the functional modules that are depicted in the Customer Administrator Dashboard and User Dashboard interface prototypes. Some of these sets of user-level functions are also described with the help of User Cases.

3.2. Use Cases

This section describes the system actors and the most relevant Use Cases of the project.

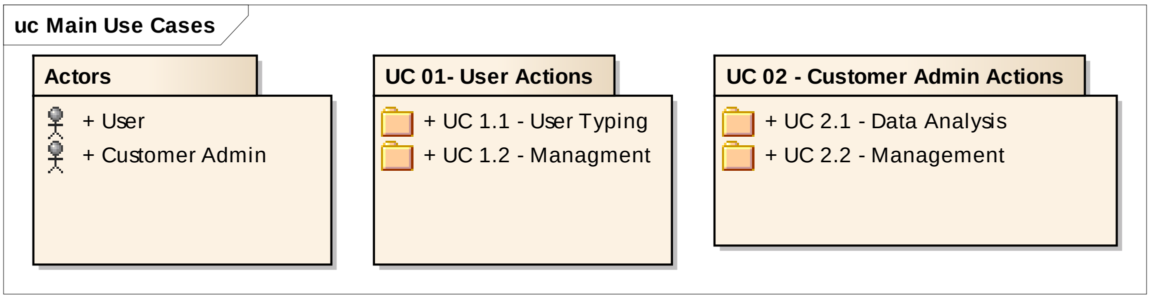


Figure 3 - Main Use Cases

There are two main actors that interact with the system, and are described as follows.

- **Customer Admin:** Refers to the customer administrators, the ones responsible for the administration of the application TypeWATCH web on the customer side. They can interact with the system by a Customer Admin Dashboard, being able to configure some application and user related settings.
- **User:** Refers to the end-users of the web application using TypeWATCH Web. They are the ones who produce keystroke dynamics by typing on targeted web text

inputs. They can also interact with the system by a User Dashboard, being able to access some user logs and statistics and to configure some user related settings.

The use cases can be grouped in two major groups, the User Actions and the Customer Admin Actions.

3.2.1. User Actions

This sub-section shows some of the use cases involving the end user and it helps to give a more visual perspective to the proposed requirements.

UC 1.1 – User Typing

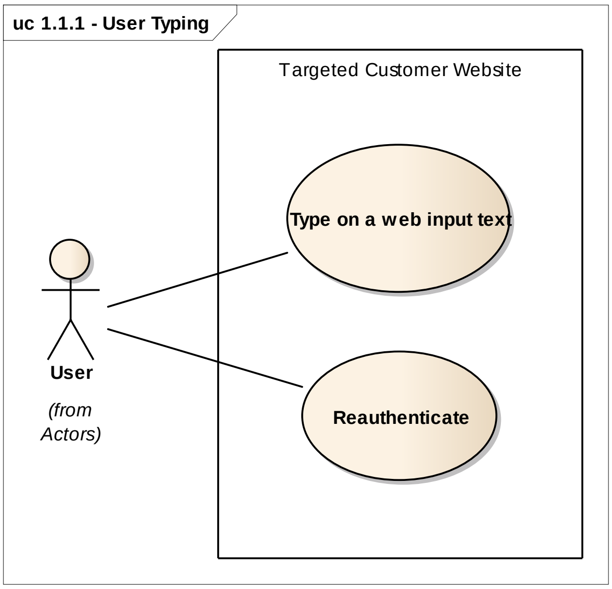


Figure 4 - UC 1.1 – User Typing

This figure depicts the uses cases regarding the activity of typing on a web input belonging to the Customer website. This activity is associated with the following two main use cases.

- **Typing on a web input**: represents the user case of typing on a web input targeted for continuous keystroke dynamics analysis. This analysis is performed by the TypeWATCH system while the user types, in order to validate the user identity.
- **Re-authentication**: represents the user case of user re-authentication in the customer website, and in this scenario can occur when the user receives an alarm resulting from a user identity validation performed by the TypeWATCH web system that returned a “Fail” result. This indicates that the user typing on the customer web-page is considered a potential intruder, so the user needs to perform a predefined re-authentication action, in order to confirm the identity claimed initially.

3.2.2. Customer Admin Actions

This sub-section shows some of the use-cases involving the customer administrator, and helps to give a more visual perspective to the proposed requirements.

UC 2.2 – Management

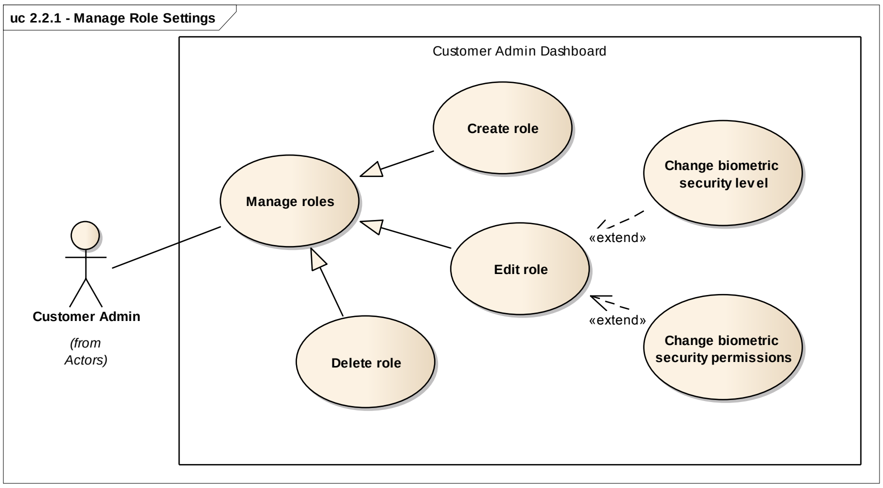


Figure 5 - UC 2.2.1 - Manage Role Settings

This figure depicts the uses cases regarding the management of Roles settings, which are associated with the actions that can be performed in the Customer Admin Dashboard Role section. Here the Customer Administrator can perform the following three use cases scenarios:

- **Create a Role:** represents the use case of creating a role. A role can be created by giving it a name. The other remaining role properties are defined with system default values.
- **Edit a Role:** Refers to the use case in which the Customer Admin edits a selected role.
- **Delete a Role:** Refers to the use case in which the Customer Admin deletes a selected role.

The “Edit Role” use case can be extended by the following use cases: “Change the biometric security level” and “Change biometric security permissions”.

- **Change the biometric security level:** represents the use case in which the user changes the application biometric security level to “Moderate”, corresponding to the lowest biometric security level defined by the TypeWATCH Web system.
- **Change the biometric security permissions:** represents the use case in which the user changes the application biometric security level to “High”, corresponding to the intermediate biometric security level defined by the TypeWATCH Web system.

3.3. Activity Diagrams

There is a main activity diagram in this system, the one corresponding to the process of user validation by the, collection, analysis and classification of the user typing behaviour.

AD 01 – Validate User

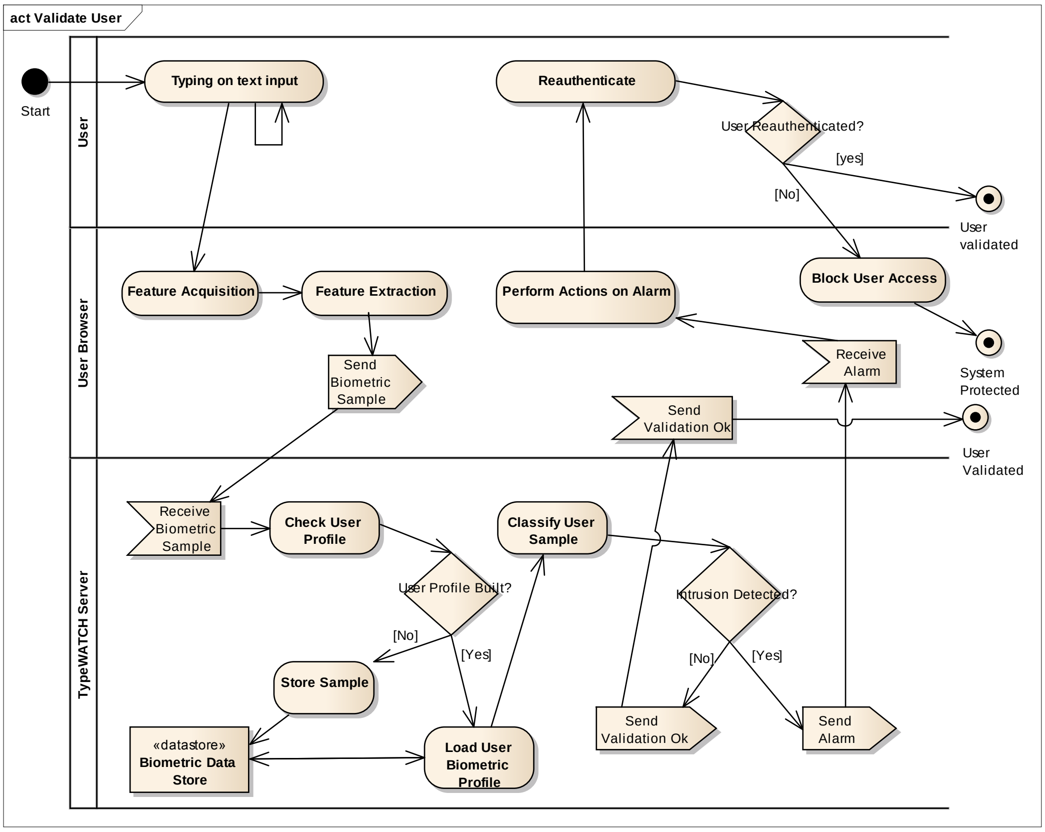


Figure 6 - AD 01 - Validate User

This figure illustrates the "Validate User" diagram related to the process of validating a user in the system by analysing the user keystroke pattern while the user types on a text input in the application website. This process is described in detail as follows:

- **Typing on text input:** An activity that refers to the user typing with the keyboard on a text web input targeted for keystroke dynamics analysis in the application web site.

While the user performs the "Typing on text input" activity, the application running in the user browser performs some activities in order to build a biometric data sample for the user.

- **Feature Acquisition**: An activity that refers to the process of structuring of capturing user keystroke by the application running in the browser.
- **Feature Extraction**: An activity that refers to the process of structuring the keystroke related events in predefined biometric features.

Once the biometric sample is complete, it sent to the TypeWATCH Web server in order to be classified.

- **Check User Profile:** An activity in which there is verified if the biometric profile of the user is built, meaning that the initial enrolment phase is concluded.

If the biometric profile of the user is not built yet, then it means that the user biometric profile cannot yet be used for sample classification, so the sample received should be added to user biometric profile

- **Store Sample:** Refers to the activity of storing the user biometric sample in the data store by adding the sample to the stored user biometric profile.

If the biometric profile of the user is built, it means the user biometric profile can be used for sample classification.

- **Load User Biometric Profile**: An activity that refers to the loading of the user biometric profile that is store in the data base.

Once the user biometric profile is loaded, it is ready to be used in the classification process of the received user sample

- **Classify User Sample:** Refers to the process of testing the received user sample against the loaded user biometric profile, in order to classify the user sample. This classification process yields a score, and that score determines if the sample belongs to the user, or not if a potential intruder was detected.

If the potential intruder was detected, then an alarm is generated, and sent to the User web Browser. In there is no potential intruder detected, the user is considered a valid and legitimate user, and a positive validation message is sent to the User Browser.

If an alarm is received in the User Browser, the predefined actions on alarms are performed.

- **Perform Actions on Alarm:** Refers to a defence mechanism implemented in the User Browser that will apply some predefined security measures in order to protect the system from the potential intruder.

The actions in the activity "Perform Actions on Alarm" can require the user to reauthenticate.

- **Reauthenticate:** An activity in which the user executes a predefined security procedure in order to be re-authenticated in the application's system.

If the user succeeds to reauthenticate in the system, then user validated, but if the user fails to reauthenticate in the application' system, the user access to it could be denied.

- **Block User Access:** Refers to the process of blocking the user access to the application' system, in order to protected from a potential illegitimate user access.

3.4. Interface Prototypes

In this section there are introduced the most relevant interface prototypes for this project. They are grouped into two major sets. The ones related with the Customer Admin Dashboard interface, and the ones related with the User Dashboard interface.

3.4.1. Customer Administrator Dashboard

The most important interface prototypes for the Customer Admin are "Home" and "Role" interface prototypes, and are described as follows.

Home Section – Customer Admin Dashboard

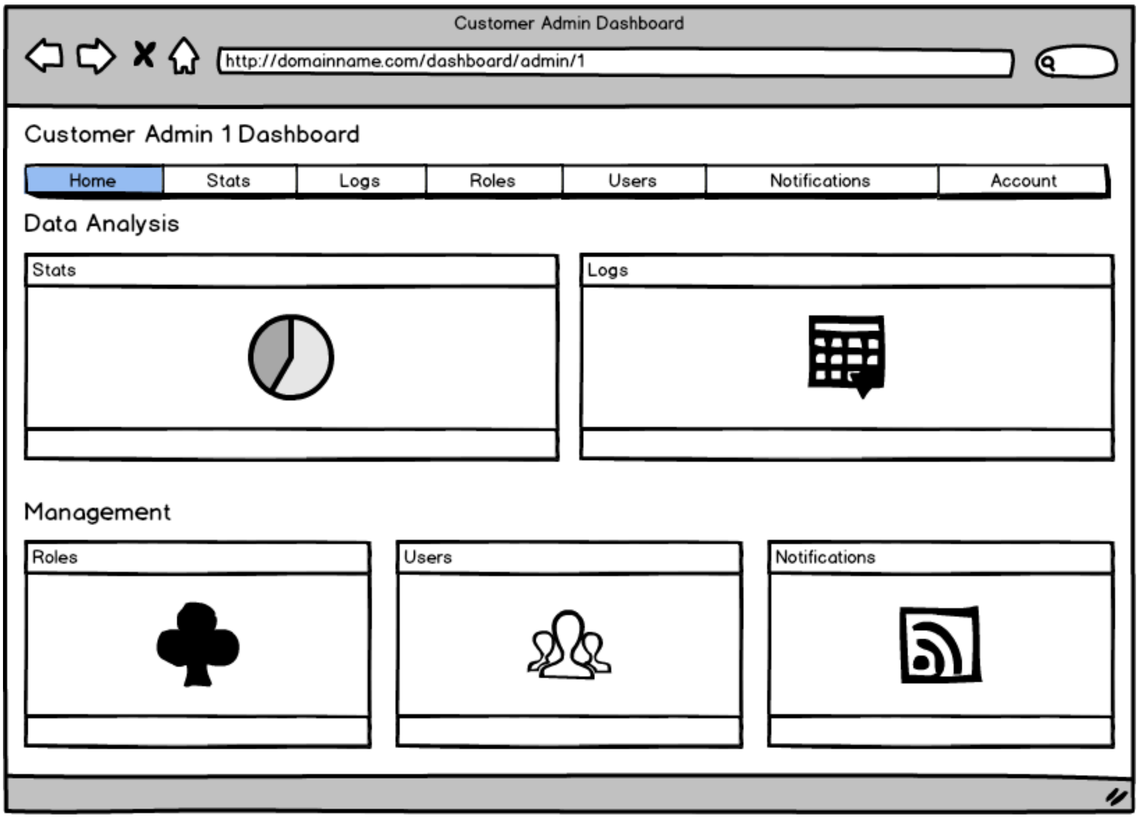


Figure 7 – Home Section – Customer Admin Dashboard

This figure depicts the interface prototype for the Home page in the Customer Admin Dashboard. Here the Customer Admin can access to some Data Analysis and Management related sections.

The Stats page is dedicated to the visualization of detailed statistical data regarding the user identity validation attempts, while Log page is dedicated to the visualization of detailed log data regarding the users' identity validation attempts, which also may include log data related to the actions on alarms performed by the users.

The Roles page is dedicated to the management of Roles, which can be listed, created, edited or deleted by the Customer Admin. The Users page is dedicated to the search and listing of the users of the application. Here the Customer Admin can select users form the list and assign them to an existing Role, which as defined in the Role page. The Notification page is dedicated to the configuration of the push notifications that can be received by the Customer Admin. Apart from enabling and disabling some predefined alarm related push notifications, the Customer Admin can also choose how to receive and view these notifications.

3.4.2. End-User Dashboard

The most important interface prototypes for the End-User are "Home" and "Security" interface prototypes, and are described as follows.

Home Section – User Dashboard

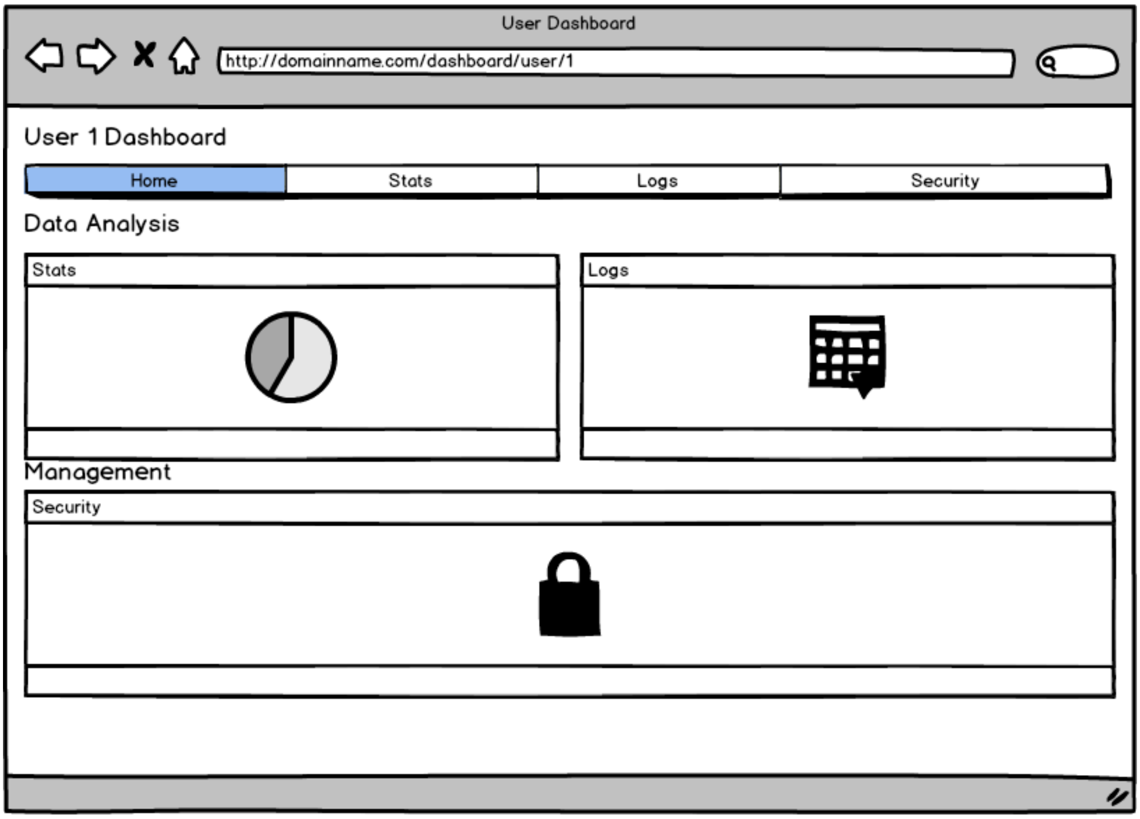


Figure 8 – Home Section – User Dashboard

This Figure 8 – Home Section – User Dashboard depicts the interface prototype for the Home page in the User Dashboard. Here the User can access to some Data Analysis and Management related sections.

The Stats page is dedicated to the visualization of detailed statistical data regarding the user identity validation attempts, while Log page is dedicated to the visualization of detailed log data regarding the user identity validation attempts, which also may include log data related to the actions on alarms performed by such user.

The security page is dedicated to the management of some Biometric Security settings that are applied to the user.

3.5. Functional Requirements

This section introduces the set of functional requirements. They are grouped in a modular structure.

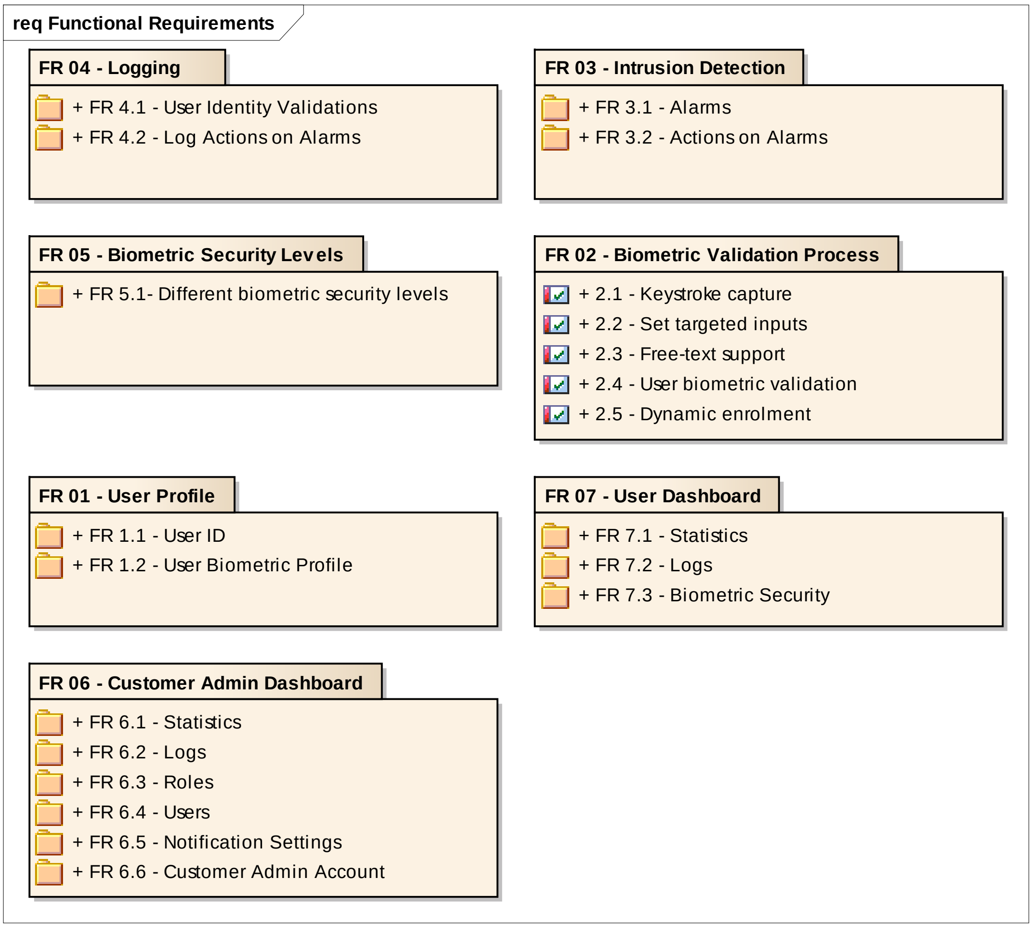

req Functional Requirements
FR 04 - Logging
+ FR 4.1 - User Identity Validations
+ FR 4.2 - Log Actions on Alarms
FR 03 - Intrusion Detection
+ FR 3.1 - Alarms
+ FR 3.2 - Actions on Alarms
FR 05 - Biometric Security Levels
+ FR 5.1- Different biometric security levels
FR 02 - Biometric Validation Process
+ 2.1 - Keystroke capture
+ 2.2 - Set targeted inputs
+ 2.3 - Free-text support
+ 2.4 - User biometric validation
+ 2.5 - Dynamic enrolment
FR 01 - User Profile
+ FR 1.1 - User ID
+ FR 1.2 - User Biometric Profile
FR 07 - User Dashboard
+ FR 7.1 - Statistics
+ FR 7.2 - Logs
+ FR 7.3 - Biometric Security
FR 06 - Customer Admin Dashboard
+ FR 6.1 - Statistics
+ FR 6.2 - Logs
+ FR 6.3 - Roles
+ FR 6.4 - Users
+ FR 6.5 - Notification Settings
+ FR 6.6 - Customer Admin Account

Figure 9 - Functional Requirements

3.6. Non-Functional Requirements

This section introduces the set of non-functional requirements. They are grouped in a modular structure.

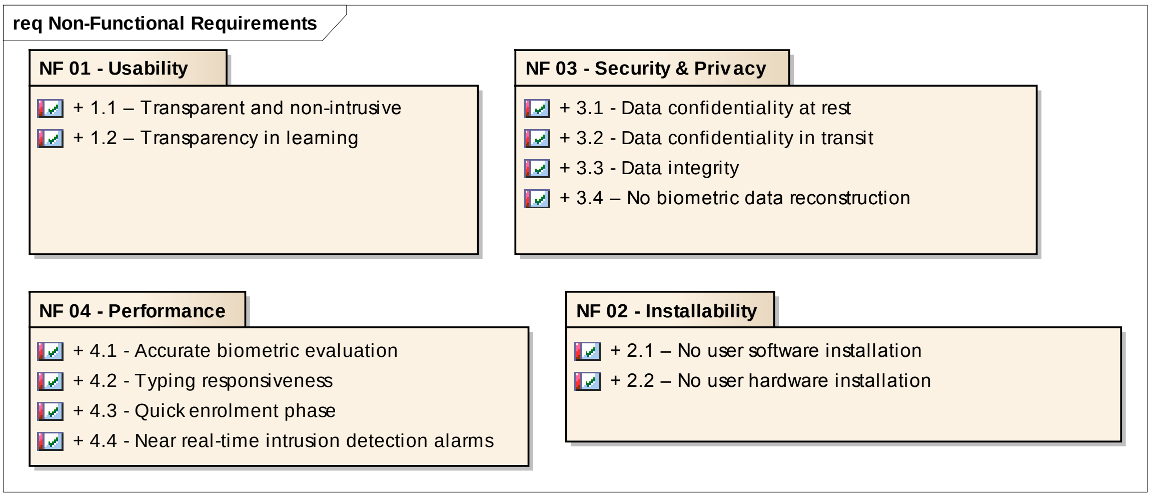

req Non-Functional Requirements
NF 01 - Usability
+ 1.1 – Transparent and non-intrusive
+ 1.2 – Transparency in learning
NF 03 - Security & Privacy
+ 3.1 - Data confidentiality at rest
+ 3.2 - Data confidentiality in transit
+ 3.3 - Data integrity
+ 3.4 – No biometric data reconstruction
NF 04 - Performance
+ 4.1 - Accurate biometric evaluation
+ 4.2 - Typing responsiveness
+ 4.3 - Quick enrolment phase
+ 4.4 - Near real-time intrusion detection alarms
NF 02 - Installability
+ 2.1 – No user software installation
+ 2.2 – No user hardware installation

Figure 10 - Non-Functional Requirements

Chapter 4
Design and Architecture

The chapter presents the Design and Architecture of the TypeWATCH Web system. Here, some architectural decisions are justified, and are followed by the presentation of high and detailed levels of architecture design. To end this chapter, the data model is also introduced.

4.1. Architecture Design Decisions

The system architecture designed for this web-based application is backed up by some architecture design decisions.

Asynchronous Communication

The TypeWATCH server offers an Intrusion Detection service that is to be consumed asynchronously by a JavaScript client API that runs in background on the customer web-site, which is accessed by the end-user. The communication between the client JavaScript API's and the exposed web-service is asynchronous. This avoids both the web-page from being refreshed on each request, and the loss of in-memory application data.

Easy Integration with Customer Systems

The system architecture should facilitate the integration of TypeWATCH core system with existing customer systems. This implies that no web-service, server proxy, or similar component is to be installed on the customer side. The communication between a web application and TypeWATCH system is done directly, without passing though the customer existing servers.

The JavaScript client application that runs on the user side – the one responsible for the user biometrics gathering and the handling of resulting intrusion alarms returning by an Intrusion Detection web-service – is at first, made available to the customer, so it can add a proper license key, and extend – programmatically – a set of functionalities in order to integrate this functionalities with its server system.

Once this is setup by the customer, the JavaScript application is ready to be included in the customer's web-page, so it can be transferred though HTTP from the customer server to the end-user side, in order to run on the end-user client browser.

Multitenant Architecture

The services of a multitenant application share only a single secure virtual computing environment. This means that, in this particular case, all customers that use this server application share the same running environment, and the data base server instance. This facilitates a possible integration of the biometrics profiles with multiple customer webs site or applications, and it also facilitates possible data mining related tasks.

4.2. High-Level System Architecture

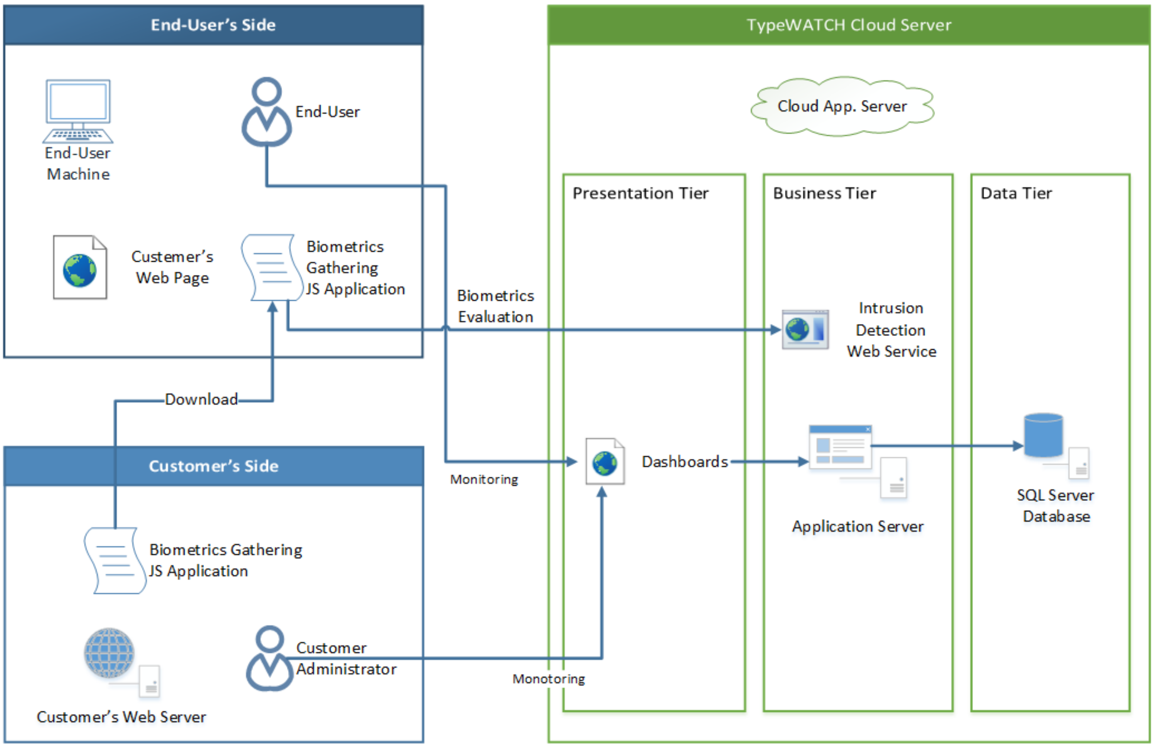


Figure 11 - High-Level Architecture Diagram

The system is divided in three main areas. The End-User's Side, the Customer's Side and the TypeWATCH Server. The Figure 11 - High-Level Architecture Diagram shows the key system actors, the key components, and the associations between these components.

4.2.1. End-User's Side

In this End-User area, a computer with a keyboard and internet connection is required by the end-user to access the customer's web site page. This web-site page includes the JavaScript client application that is transferred from the customer's web server.

- **End-User**: A human actor that interacts with the system, by using a customer website, or by using its dedicated TypeWATCH dashboard.
- **User Machine**: The computer used by the end-user. It requires a keyboard, and a web browser.
- **Biometrics Gathering JS Application:** A JavaScript application that is transferred from the customer's server when the end-user accesses a customer web-page. This application is then executed on the web page, and is mainly responsible for capturing the user typing behaviour, processing the biometrics data, sending the biometric data to the TypeWATCH server, and for handling the alarms and the actions to be performed on alarms.

4.2.2. Customer's Side

This is the area that holds the customer's system, which includes the customer's web server, and where the JavaScript client application is installed.

- **Customer Admin:** The human actor responsible for the administration and monitorization of the respective TypeWATCH Web dashboard / administration console.
- **Customer System:** The costumer existing system application, which is loosely coupled with the TypeWATCH server side.
- **Biometrics Gathering JS Application:** The JavaScript application that is installed on the customer's web site, and integrated with the customer's system. It is the same JavaScript application that is transferred to the end-user browser when it loads the customer web-site.

4.2.3. TypeWATCH Server

This are represents the TypeWATCH core system, which runs on the cloud. This block follows a client-server architecture known as multi-tier architecture. There are three main tiers, the Presentation Tier, the Application Tier, and the Data Tier.

Presentation Tier

The presentation tier serves as a graphical and functional interface to users, in this case, the end-users and the customer's administrators. It consists of the User Dashboards and the Customer Dashboards.

- **User Dashboard:** A graphical user interface intended for the End-User, in which the end-user can manage and configure some application related user settings, and to access its user validation logs and related validation statistical data.
- **Customer Dashboard:** A graphical user interface intended for the Customer Administrator, in which the administrator can manage and configure application related user settings, and to access overall user settings, and to access overall user validation logs, and related validation statistical data. It also permits the administrator to receive notifications about the user activities in the system.

Business Tier:

This is the tier that is, mostly, responsible for the application logic of the TypeWATCH Web server, and it is mainly composed by the following components.

- **Application Server:** A major component, responsible for the business logic functions of the system, and for the management of existing web-services or web servers, and data store connections.
    - **Intrusion Detection Web Service:** It is a specific kind of web server application that exposes the Intrusion Detection web service through a well-defined interface. It handles the requests that are sent by the client API, and returns back possible intrusion detection alarms.
- **Data Tier:** It is the tier responsible for the access to and persistence of data.
    - **SQL Server Database:** A database server that holds the user biometric profiles, the application and user registry, licenses, evaluation logs, and other relative data.

4.3. Study and Selection of Technologies

It is important to run a study on the tools and technologies that may be useful in the development of the solution prototype, in order to make the appropriate choices. The study is influenced by some of the goals and objectives proposed in this thesis, by the software requirements proposed, and by the architectural decisions that were made during the system architecture design. The following study presents some the tools and technologies considered for this project.

Biometrics Gathering Web Application

Web Standards (JavaScript)

Conceptually, on the user side, there is the end-user, its computer with a keyboard, and web browser. Here, there is a web application module that deals with the capture and structure of the user's biometric data. It is important to run a study of the technologies that should be used in the development of this module in order to obey to the following criteria:

- The end-user shall use the application without installing any software on its machine.
- The application shall comply with the current web standards.

For this, the following technologies were considered:

- **Adobe Flash:** A technology from Adobe Systems that can be used as a software platform to develop application capable of running in the browser. It's extremely popular in video and audio streaming web-applications, but in other applications, such as Rich Internet Applications (RIA's), or animation oriented web-applications, its use and its popularity have been declining over the years. The Flash applications can be programmed using Action Script, but there are some proprietary tools that are helpful in the development of such applications. In short, it is a proprietary technology that can be installed in the browser as a plug-in.
- **Microsoft Silverlight:** An application framework developed by Microsoft Corporation that is suitable for the development web-applications, mainly Rich Internet Applications, and applications that focus on animations and streaming of video and audio. It also supports asynchronous communications. It is a proprietary technology, and should also be installed in the computer machine as a plug-in.
- **JavaScript:** A dynamic and interpreted programming language that is native on major web browsers and that complies with the web-standards. It is an extremely popular and non-proprietary scripting language that can be used to manipulate the Document Object Model (DOM), and to capture user interactions in the form of events. In addition, the JavaScript Object Notion (JSON), which is a lightweight data-interchange format that is getting extremely popular whiten web application, is native JavaScript. This programming language has a large adoption base, and there are thousands of scripts, micro-frameworks, and unobtrusive plugins written in JavaScript that were developed by the online community, that can be used for free in commercial applications. It is a scripting language suitable for any kind of web applications, but it's particularly useful for event-orient applications.

After the analysis of these technologies, it was decided that the application module that runs on the web browser should be written in JavaScript.

Development Framework

Web Services (Windows Communication Foundation using SOAP)

The solution to be developed follows a Service-oriented Architecture (SOA), in which a group of functionalities are abstracted with the loose coupling concept in mind. This architecture is platform, vendor, and technology independent, and provides a high-level of interoperability between different systems over a network.

Typically, there are the service providers, who publish the web-service by using a standard description language that describes both the purpose of the service and the service interface. This is the information that is then looked up by service consumers who are interested in using the service.

In part, these are characteristics that sum up the advantages of using web-services in this project.

Regarding the web-services implementation, they can be implemented following two different set of specifications:

- **WSDL/SOAP/UDDI-based web-services:** These web-services rely on an architecture in which the service is described by the WSDL specification, is located by the IDDI specification, and actually access using the SOAP protocol specification.
- **REST-compliant web-services:** These web-services are resource-oriented. They take advantage of the HTTP capabilities to find the location and the actions to be performed on a given resource.

In the architecture designed for this project, the nature of data that is exchange over the designed web-services is not resource-oriented, so the web-services based on SOAP are more suitable to be used in this project.

For the implementation of the web-service, the proposed technology is the Microsoft Windows Communication Foundation framework, which is a server-side framework that is part of the Microsoft .NET framework stack. With this, it is possible to define SOAP endpoints. These endpoints support the communication of JSON serialized data, which is a light-weight data-interchange format based on the JavaScript notation, the language used to create the data on the client-side.

Web Pages (ASP.NET)

The ASP.NET framework allows the development of regular server-side web-applications, with seamless integration with the web-page view. This is to be used in the development of the administration consoles.

The current TypeWATCH is already using some of these technologies, so there is in-house knowledge on how to implement such applications, and coherence between this project and the current TypeWATCH product could be maintained by using the same technologies.

4.4. Detailed System Architecture

The detailed system architecture includes the system components, its interfaces, and the data model.

4.4.1. System Components

The following diagram illustrates main TypeWATCH Web system components.

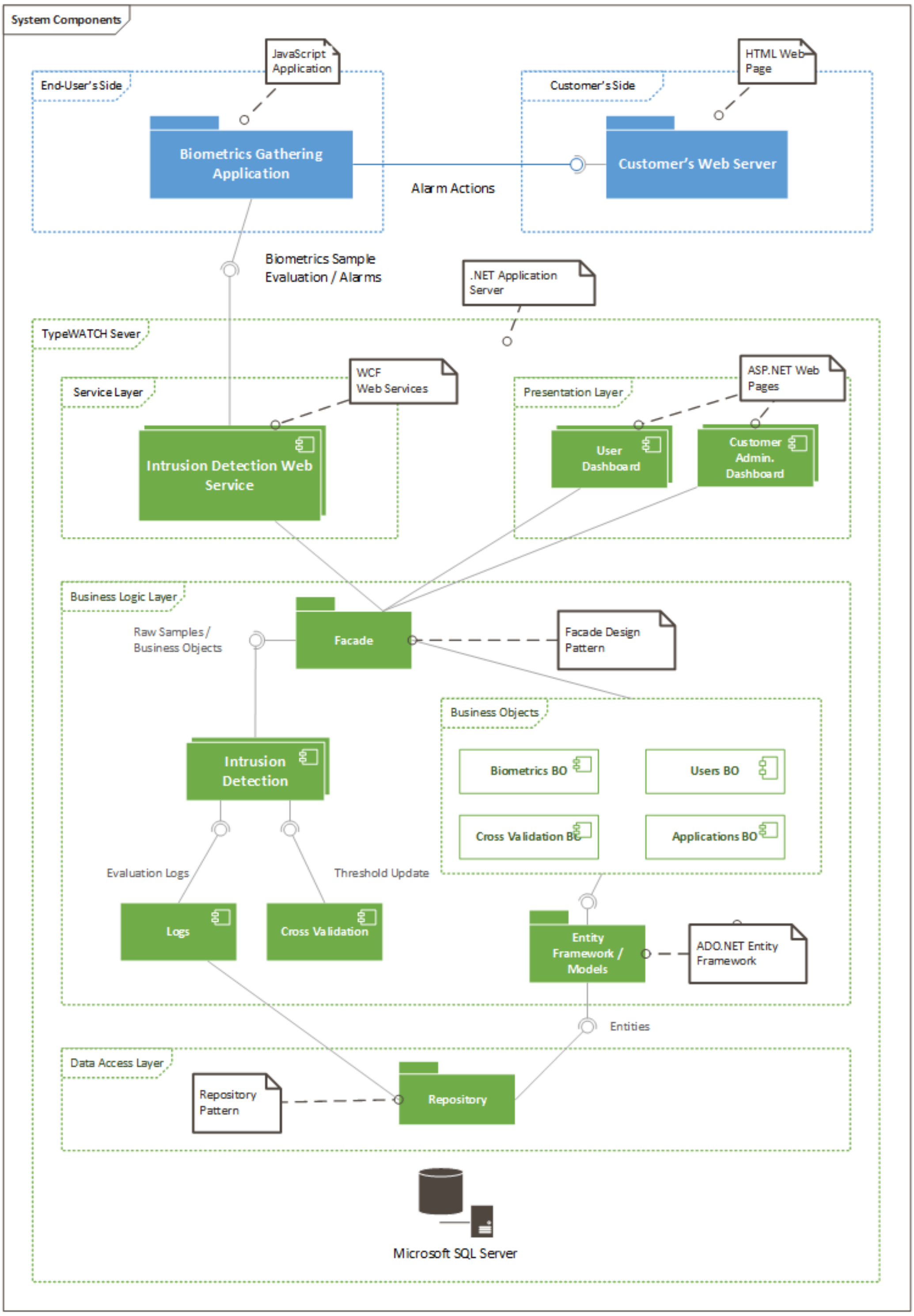


Figure 12 - System Components Diagram

This is the global view of the TypeWATCH Web system architecture, its main components, interfaces, and the interaction between them.

The Biometrics Gathering component is captures and processes the user biometrics data on the customer web-site, which was previously served by the Customer's web server.

The Biometrics Gathering client API sends an asynchronous request through the Intrusion Detection Web Service interface in order to evaluate the user biometrics sample, and then handles the response, and if there is an alarm, it triggers potential predefined security measures on the customer's server.

The Web Service interface is exposed using the Windows Communication Foundation framework stack, by defining SOAP endpoints. These endpoints support the communication of JSON serialized data. The interface Biometrics Evaluation receives a Sample – including the Biometrics Sample, the Application Identification, and the User Identification. This interface returns an Evaluation Result. It contains the Enrolment Progress, and an Alarm, if any.

The request for a biometrics sample evaluation received at the service layer by the Intrusion Detection Web Service components is actually performed in the business logic layer. Some data mappings are also performed at the service layer, in order to decouple the service and business logic models. The interaction between the service layer components and the business logic components are done through a Facade. The Intrusion Detection component is the one responsible for the actual evaluation, using the support of some business objects components.

If there is a need for the update of the user's dynamics threshold, the Cross-Validation components is used. Additionally, all evaluation logs are persisted by the Logs component.

The entity models used by the business objects are generated by an object-relational mapper from the database model. These models are persisted through a Repository component that implements a set of instance of the Repository Interface suggested by the Repository Design Pattern.

The persistence of that is done by an instance of the Microsoft SQL Server.

Biometrics Gathering Components

A global view of the Biometrics Gathering Application is presented on Figure 13 - Biometrics Gathering Components Diagram by its main components, interfaces, and the interaction between them.

This Biometrics Gathering application is initiated at the page load, by starting the Sample Builder component. This main component invokes the Feature Extraction component, which in turn invokes the Feature Acquisition components.

The Feature Acquisition component is responsible for the capture of keyboard events, and guarantees the integrity of these captures by using Finite State Machines for each key bind. The events captured are buffered and then to the Feature Extraction component. This component is the responsible for the filtering of events and for the extraction of biometrics features from these events. The features extracted are sent to Sample Builder component.

This Sample Builder component receives the extracted features, and gathers user and application related that. The gathering of this data is from the responsibility of the User Identification and Application Identification components, respectively.

The Sample Builder assembles the biometrics features, the user identification, and application identification into a biometrics sample, and sends it to the Intrusion Detection service by using the Communication Client component. This component consumes the exposed Intrusion Detection web-service interface by making asynchronous XMLHttpRequest requests. The data sent includes the Biometrics Sample, the Application Identification and the User Identification. The response is received.

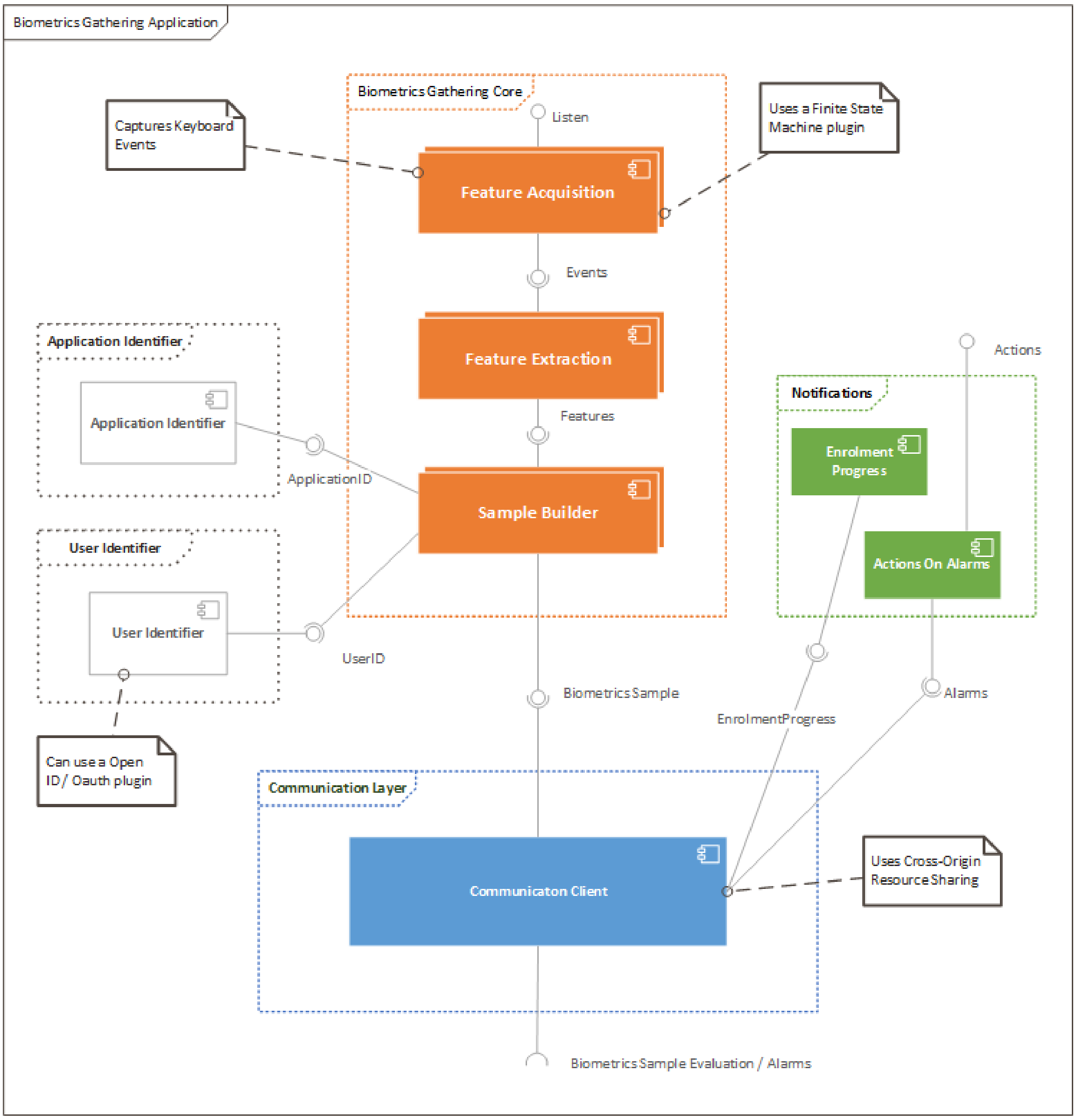


Figure 13 - Biometrics Gathering Components Diagram

The data received is sent to the Notifications component which is responsible for the display of information notifications and for the triggering of security measures as a protection mechanism to mitigate potential intrusions.

4.4.2. Interfaces

Biometrics Gathering Core Interfaces

The Interfaces shows is a sequence diagram that shows the sequence of messages that are sent across the internal interfaces of this Biometrics Gathering core component.

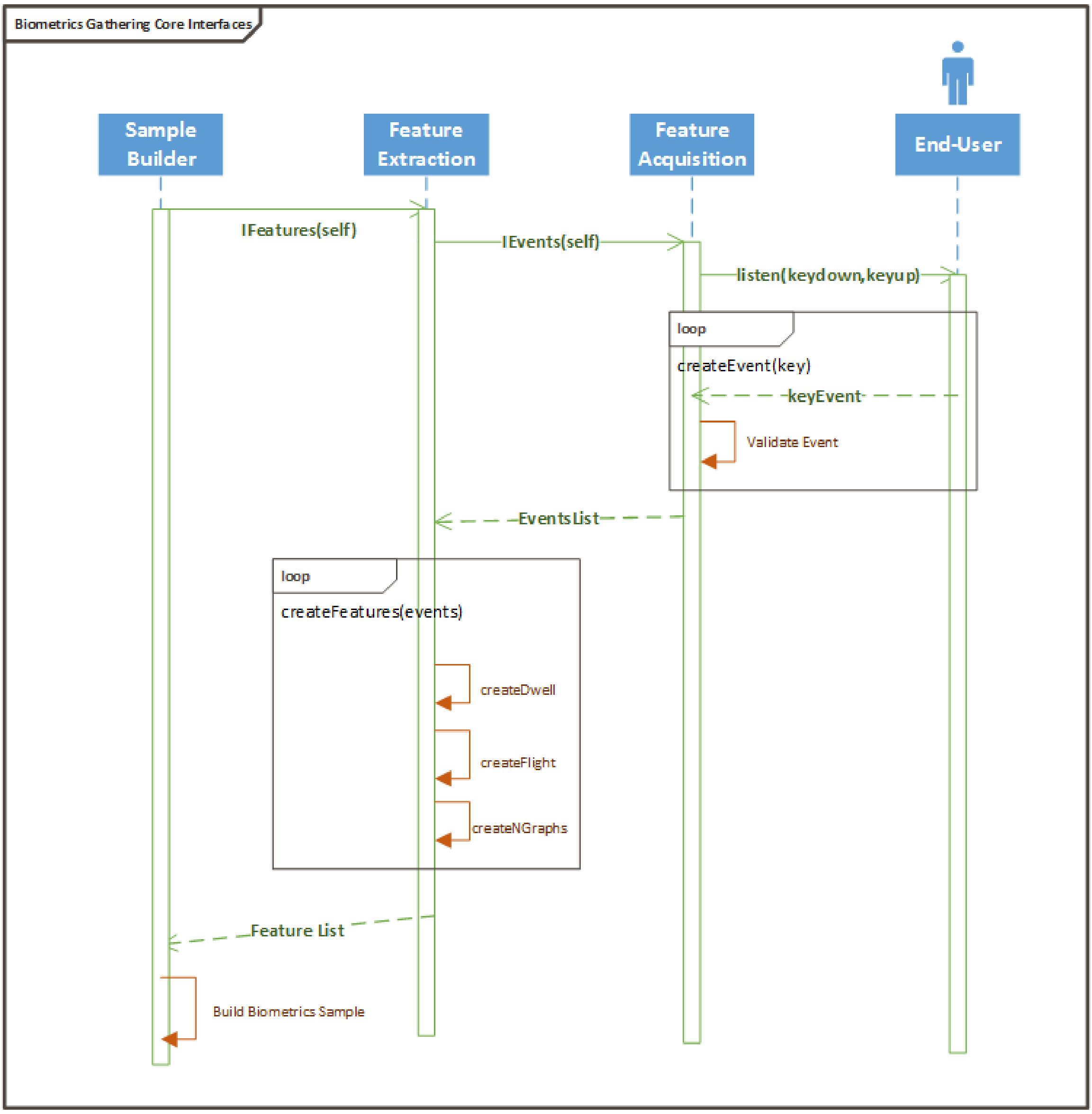


Figure 14 - Biometrics Gathering Core Interfaces

A Sample Builder sub-component is instantiated when the application starts. It invokes the Features Extraction sub-component, which in turn invokes the Feature Acquisition component. This last one binds keyboard events and captures them, recording the current timestamp. It returns the list of captured events to the Feature Extraction component, which filters and builds biometrics features from the events list. The features are then sent to the Sample Builder.

4.5. Data Model

This sub-section introduces the data-model of the system components, by presenting the system classes, and the database schema.

4.5.1. Class Diagram

The Figure 15 - IntrusionDetectionWF class diagram shows the UML class diagram containing the relevant classes of the Intrusion Detection Core Component.

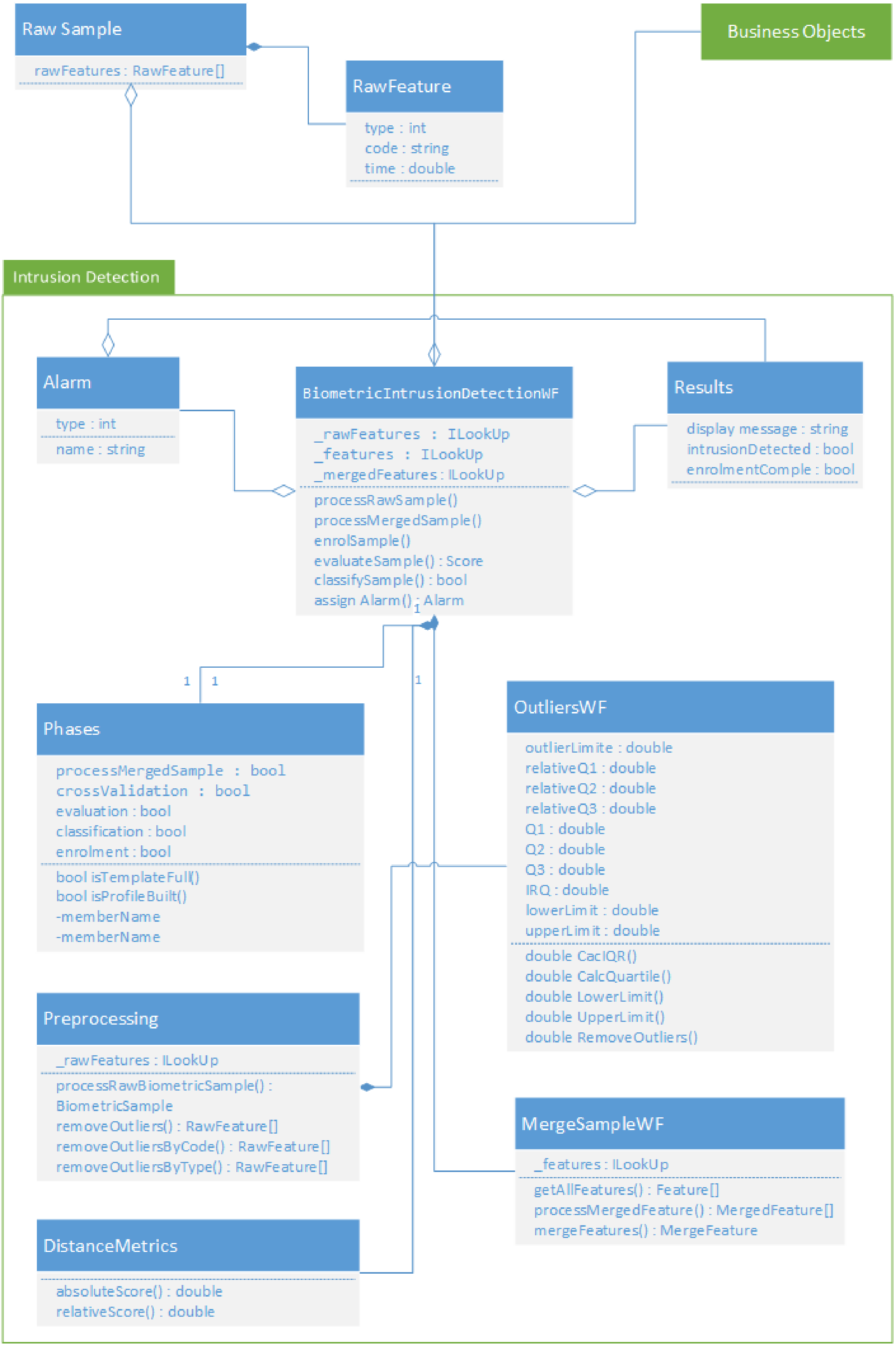


Figure 15 - IntrusionDetectionWF class diagram

The classes on this diagram are described as follows:

“BiometricsIntrusionDetectionWF” class

This class is the main class of this component, and is responsible for the classification of the user biometrics sample. It receives this raw biometrics sample (RawSample) for evaluation and also two business objects. The BiometricsBO and the UserBO, which have relevant metadata and business rules that support the classification of the given biometrics sample.
All the main tasks of this process are of the responsibility of this main class.

“Phases” class

This class is that infers the actual workflow of this intrusion detection component. This class defines which tasks should be performed during this evaluation process, and also asserts the actual state of the target biometrics profile.

“Preprossessing” class

This class represents a pre-processing module that takes a raw biometrics sample, applies an outlier removal, and merges the common features of the sample. It returns a processed sample, ready to be classified.

“OutliersWF” class

This class is responsible for the removal of the outlier features that are possibly present in the raw biometrics sample received. It returns a biometrics sample without the outliers that were found.

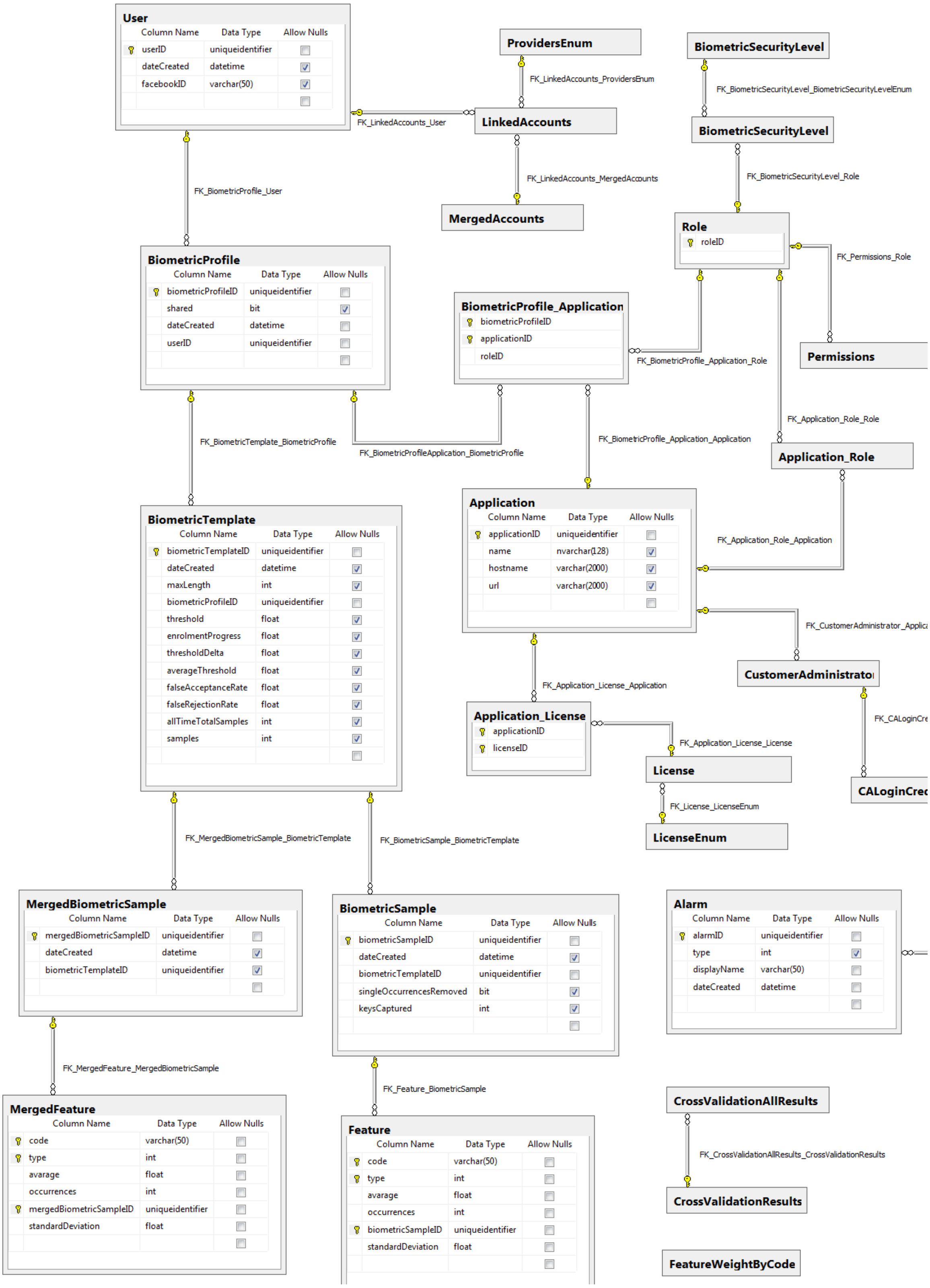
“MergeSampleWF” class

This class has the responsibility of creating a merged sample from the user’s biometrics template, in order to use it as the target biometrics sample in the biometrics sample evaluation.

“DistanceMetrics” class

This class holds the actual evaluation models for this intrusion detection module. It has two main methods associated with it, which corresponds to the two evaluation models that are used by this Intrusion detection component. That is, the absolute score model, and the relative score model.

4.5.2. Database Diagram



User

Column Name	Data Type	Allow Nulls
🔑 userID	uniqueidentifier	☐
dateCreated	datetime	☑
facebookID	varchar(50)	☑
		☐

BiometricProfile

Column Name	Data Type	Allow Nulls
🔑 biometricProfileID	uniqueidentifier	☐
shared	bit	☑
dateCreated	datetime	☐
userID	uniqueidentifier	☐
		☐

BiometricProfile_Application

- 🔑 biometricProfileID
- 🔑 applicationID
- roleID

Role

- 🔑 roleID

Application

Column Name	Data Type	Allow Nulls
🔑 applicationID	uniqueidentifier	☐
name	nvarchar(128)	☑
hostname	varchar(2000)	☑
url	varchar(2000)	☑
		☐

BiometricTemplate

Column Name	Data Type	Allow Nulls
🔑 biometricTemplateID	uniqueidentifier	☐
dateCreated	datetime	☑
maxLength	int	☑
biometricProfileID	uniqueidentifier	☐
threshold	float	☑
enrolmentProgress	float	☑
thresholdDelta	float	☑
averageThreshold	float	☑
falseAcceptanceRate	float	☑
falseRejectionRate	float	☑
allTimeTotalSamples	int	☑
samples	int	☑
		☐

Application_License

- 🔑 applicationID
- 🔑 licenseID

MergedBiometricSample

Column Name	Data Type	Allow Nulls
🔑 mergedBiometricSampleID	uniqueidentifier	☐
dateCreated	datetime	☑
biometricTemplateID	uniqueidentifier	☑
		☐

BiometricSample

Column Name	Data Type	Allow Nulls
🔑 biometricSampleID	uniqueidentifier	☐
dateCreated	datetime	☑
biometricTemplateID	uniqueidentifier	☐
singleOccurrencesRemoved	bit	☑
keysCaptured	int	☑
		☐

Alarm

Column Name	Data Type	Allow Nulls
🔑 alarmID	uniqueidentifier	☐
type	int	☑
displayName	varchar(50)	☐
dateCreated	datetime	☐
		☐

MergedFeature

Column Name	Data Type	Allow Nulls
🔑 code	varchar(50)	☐
🔑 type	int	☐
avarage	float	☐
occurrences	int	☐
🔑 mergedBiometricSampleID	uniqueidentifier	☐
standardDeviation	float	☐
		☐

Feature

Column Name	Data Type	Allow Nulls
🔑 code	varchar(50)	☐
🔑 type	int	☐
avarage	float	☐
occurrences	int	☐
🔑 biometricSampleID	uniqueidentifier	☐
standardDeviation	float	☐
		☐

ProvidersEnum

BiometricSecurityLevel

LinkedAccounts

BiometricSecurityLevel

MergedAccounts

Permissions

Application_Role

CustomerAdministrato

License

LicenseEnum

CALoginCred

CrossValidationAllResults

CrossValidationResults

FeatureWeightByCode

FK_LinkedAccounts_ProvidersEnum
FK_BiometricSecurityLevel_BiometricSecurityLevelEnum
FK_LinkedAccounts_User
FK_BiometricProfile_User
FK_LinkedAccounts_MergedAccounts
FK_BiometricSecurityLevel_Role
FK_Permissions_Role
FK_BiometricProfile_Application_Role
FK_Application_Role_Role
FK_BiometricTemplate_BiometricProfile
FK_BiometricProfileApplication_BiometricProfile
FK_BiometricProfile_Application_Application
FK_Application_Role_Application
FK_CustomerAdministrator_Applica
FK_Application_License_Application
FK_CALoginCre
FK_Application_License_License
FK_License_LicenseEnum
FK_MergedBiometricSample_BiometricTemplate
FK_BiometricSample_BiometricTemplate
FK_MergedFeature_MergedBiometricSample
FK_Feature_BiometricSample
FK_CrossValidationAllResults_CrossValidationResults

Figure 16 - Partial view of the TypeWATCH Web Database model diagram

The Figure 16 - Partial view of the TypeWATCH Web Database model diagram shows the partial view of the database schema.

This database model diagram highlights the main database entities of the system, showing its attributes, data types, primary keys, and foreign keys relationship between them.

From this database model diagram is possible to observe that a given user can have multiple Biometrics Profiles associated with an Application. For instance, there can be one biometrics profile for each application, a biometrics profile shared between some applications – or both. Similarly, a Biometrics Profile can have multiple Biometrics Templates associated with a Biometrics Profile. This enables the possibility of assigning a different biometrics template according to some environment information – e.g. computer in used, type of the target web-site, type of keyboard, and so on. This is not a requirement; however, this design decision was made taking the possibility of such future features in mind.

Each Biometrics Template can have two different types of samples. It can have a set of regular Biometrics Samples and a set of Merged Biometrics Samples. A Merged Biometrics Samples is a reduced and condensed sample resulting from the merged of biometrics Features from a given amount of Biometrics Samples.

A given Biometrics Sample can hold a set of Features, each one defined by the biometrics Feature's "code", "average", "standard deviation", and "occurrences". Similarly, a set of Merged Features can be assigned to a given Merged Biometrics Sample.

Chapter 5
Development

The Development phase consists of the implementation of a part of the Project Requirements. The full set of requirements, when translated into an Architectural view, defines an Intrusion Detection Service – with a client and server-side, two monitoring Dashboards applications, and a Database to persist the related data.

From these, the Intrusion Detection Service and the correspondent Database were planned for actual implementation.

5.1. Intrusion Detection Web Client

In an application relying on keystroke dynamics, the feature acquisition is the starting point when it comes to use such behavioural biometrics as a way to correctly identify an individual among a population. This gathering process holds a number of concerns and considerations that must be addressed, being user privacy and data confidentiality the ones that automatically pop out in end-users minds when they face an application of this nature.

Therefore, it is important to make sure that no semantic information is saved or misused, and to ensure that it is virtually impossible to reconstruct the biometrics data into the original typed text. Being this an application script running on a web browser, it is vital to protect the access to the sensitive run-time application data, as well as the access to executable functions.

To address this, several techniques were used. To start with, the key related events that are captured are identified by their key codes, and not by their corresponding character. For security reasons, these key codes are then encrypted. However, encrypting the data is not enough. Another partial solution that helps is to make difficult the reading of code by the human eye, so following a good common practise, when deployed, the code is compressed and obfuscated. Compressing is the process of removing lines and trailing spaces. Obfuscating is the process of changing code variable and function names. Compressing the code also makes the files a lot smaller, which reduces download overheads [16].

Other security measures were applied, perhaps, stronger ones. The access to the run-time code is restricted by defining a hierarchy of scopes, being the variables and functions of that scope not accessible from outside of it. There is as well a particular case of this, which happens when inside a scope, exists an object that comes from outside the scope, for instance, a parameter. In this situation, all the variables or functions contained on that referenced object are private, excepting the ones that are declared using the "this" keyword. This is how interfaces between components are defined in this biometrics gathering application.

All this scope handling works carefully together with the data life cycle. With JavaScript, when an object oriented approach is taken, the equivalent to an object class is a function, which only exists during its executing life cycle. In some modules, this time span is extremely short, so some of the sensitive data only exists during a very short time. As an example, when the sample is full, all the feature processing takes only around 10 milliseconds. It is important to remind tough, that the access to all the data involved in the application is protected by some of the techniques here introduced. Dependency Injection is another one.

In order to keep that data alive and protected in the workflow between the different modules, Dependency Injection is used. It is a software design pattern that enables inversion of control by decoupling function and responsibility between modules, classes, or other functions. In this particular application, this means that, when a module is instantiated or a function is called, some required object data must be passed as a parameter, instead of being built inside the function or module itself. Besides the typical advantages of using this recommended software design pattern, one important advantage stands out. That is, even if it is possible to invoke a function or to initiate a module from a malicious user script, it would first need to construct the object parameter that is required to run the module or function properly. This approach troubles the attempts to replicate the execution of the application in such a malicious way.

5.1.1. Feature Acquisition

Being keystroke dynamics the detailed timing information that can be extracted from the press and release of keyboard keys when the user is typing, it is crucial to capture these timings in an accurate and consistent way.

JavaScript, the scripting language used in this application supports the binding of key related event listeners that are triggered when a user types on a web page using a keyboard. These listeners can be bound into any existing input field on the Document Object Model of the page. For this, a query can be made to select the desired input fields, being them text inputs, password inputs, text areas, or other HTML elements of input text-based type.

In this scenario, it is possible to bind three types of events, being them “key down”, “key press” and “key up”. The event listeners on this application listen only to “key down” and “key up” events, because the “key press” event occurs somewhere between the two. As the goal here is to extract the time elapsed while user is holding down a particular key, “key down” and “key up” were chosen. It is also important to notice that “key press” does not bind to non-character keys, such as modifier keys like “shift” or “control”. It also provides only the corresponding character key, not the corresponding key code. These two situations are both not intended.

When it comes to typing, it is not a natural typing behaviour for a user to repeatedly hold the same key for a considerable long time. In fact, this is only most likely to happen on gaming scenarios, where the gamer needs to hold the same key, or set of keys, for a significant time. For instance, when a user is typing the band name “abba”, it is assumed that the user would only type the second “b” in the sub-sequence “bb” only after the release of the same character key that was used to type the first “b” in that same sequence. Further studies must be carried to understand other typing behaviours that does not follow this pattern, and to understand if such behaviour could be positively used to distinguish a typist.

The earlier mentioned assumption helps to ease a problem that happens when a user presses a key for a considerable long time. In this particular situation, a number of continuous “key down” events would be triggered. For instance, when a user holds the key corresponding to the character “b” for a long time, it is typed a sequence containing several “b” characters until the user releases the key. Here the problem is that in this sequence only one corresponding “key up” event would be triggered.

This would cause the pairing of “key down” and “key up” events harder to achieve later in the feature extraction phase, because there would exist a “key down” event assigned to each character “b”, and only one “key up” event assigned to the whole sequence of characters “b”. In practise, this particular “key up” event would correspond to the last typed character "b". By this approach, after the pairing of events, only one "key down" – "key up" pair would be availed, being the other ones discarded.

An elegant and alternative solution for this problem is the assignment of a finite state machine to each key on the keyboard. The finite state machine that is used is a JavaScript plugin that enables the creation of custom events and states. In this case, there are the "Start", "Keydown", "Keyup" and "Clear" events, being the list of states composed by "None", "Open" and "Closed" states.

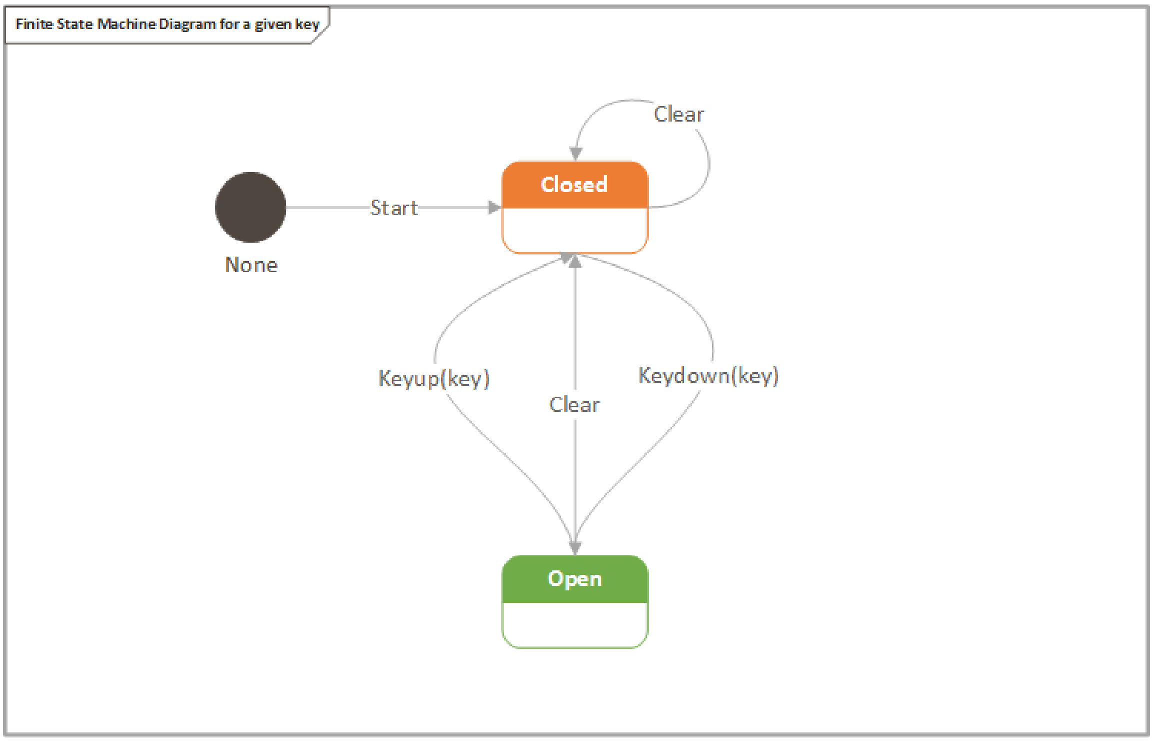


Figure 17 - Finite State Machine Diagram for a given key

Initially, a finite state machine is created in a "None" state. Then it is started by triggering the "Start" event, and it goes from "None" to "Closed", being the finite state machine ready to be used. When a user presses a corresponding key, a "Keydown" event is triggered on the state machine, being the state changed from "Closed" to "Open". This means that, for that particular key, the application will not accept future "Keydown" events from that same key until the corresponding "Keyup" event is triggered. When that event finally happens, the state is changed back to "Closed". This approach avoids the problem surrounding the handling of events that result from the pressing of a key for a considerable long time, helping to ensure that there is consistency between keystroke events.

By default, these event listeners listen to any key on the keyboard. So there is one state machine for each key. All events are accepted, except the ones discarded by the machine states itself. The events captured contain the key code, the timestamp, and additional meta-data that are kept for later analysis. For instance, it is possible to know if the event corresponds to a modifier or special key, as "shift" or "control" keys.

Regarding the timestamp accuracy, the Performance Web API Interface is preferably used, having a precision of a thousandth of a millisecond. However, this API is only supported by modern major browsers, so when it is not supported, the feature acquisition module falls back to the native browser DateTime object, which has a millisecond time resolution. Despite of a lower time resolution, this may turn out to be irrelevant, as the time duration spectrum the extracted features typically start up at a few dozens of milliseconds.

This process of feature acquisition retains all these "key down" and "key up" events in a buffer, then, when the 125th "key up" event is accepted, the events in the buffer enter the feature extraction phase, which occurs in the "feature extraction" module. The communication between the modules then made using a well-defined interface.

5.1.2. Feature Extraction

On the "feature extraction" phase, there is a module responsible for processing the events that were capture on the previous "feature acquisition" phase. These events are received, and then relevant information is extracted and compiled from them.

Besides from the list of "keydown" and "key up" events, this module also receives a list of "key codes" corresponding to the order of "key down" events that were earlier captured and accepted. This is now the base for the extraction algorithm that takes place. The complexity of this main routine is O(n), being "n" the length of the list of key codes.

So, for each element on the "key codes" list, a sub-routine based on Dynamic Programming is run in order to calculate a Dwell, Flight, Digraph, Trigraph and Fourgraph, whenever is possible. Each feature then consists of a code (the encrypted key code), the time duration of the feature, and the type of the feature.

Elementary features (dwells and flights) for the sequence "abba"

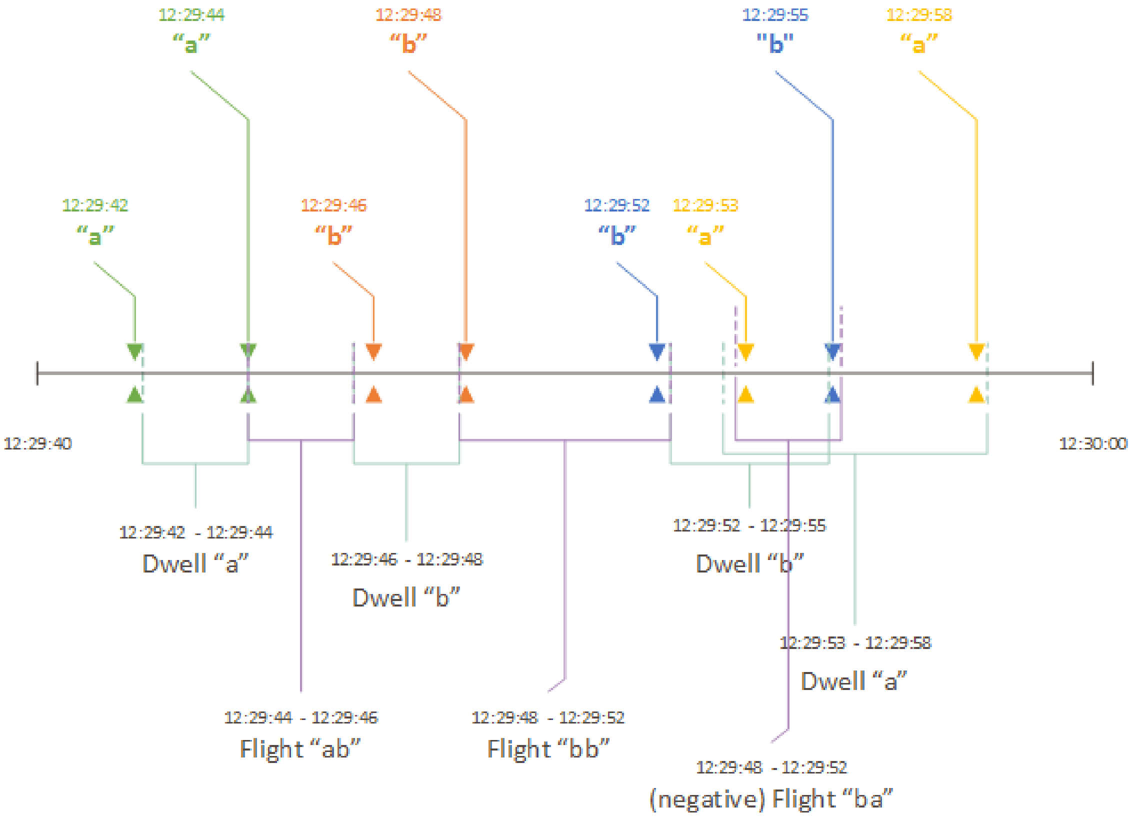


Figure 18 - Elementary features (dwells and flights) for the sequence "abba"

As shown in Figure 18 - Elementary features (dwells and flights) for the sequence "abba", a Dwell is constructed by calculating the elapsed time between the appropriate "key down" and "key up" timestamps for that key code. A Flight is constructed by calculating the elapsed time between the appropriate “key up” timestamp of the current key code, and the appropriate “key down” timestamp of the subsequence key code.

As shown in Figure 19 - Composite features (N-Graphs) for the sequence “abba”, a Digraph is constructed by calculating the sum of the current Dwell and Flight times. A Trigraph is constructed by calculating the sum of the current Digraph time with the previous Digraph time. Finally a Fourgraph is constructed by calculating the sum of the current Digraph time with the previous Trigraph time.

Composite features (N-Graph) for the sequence “abba”

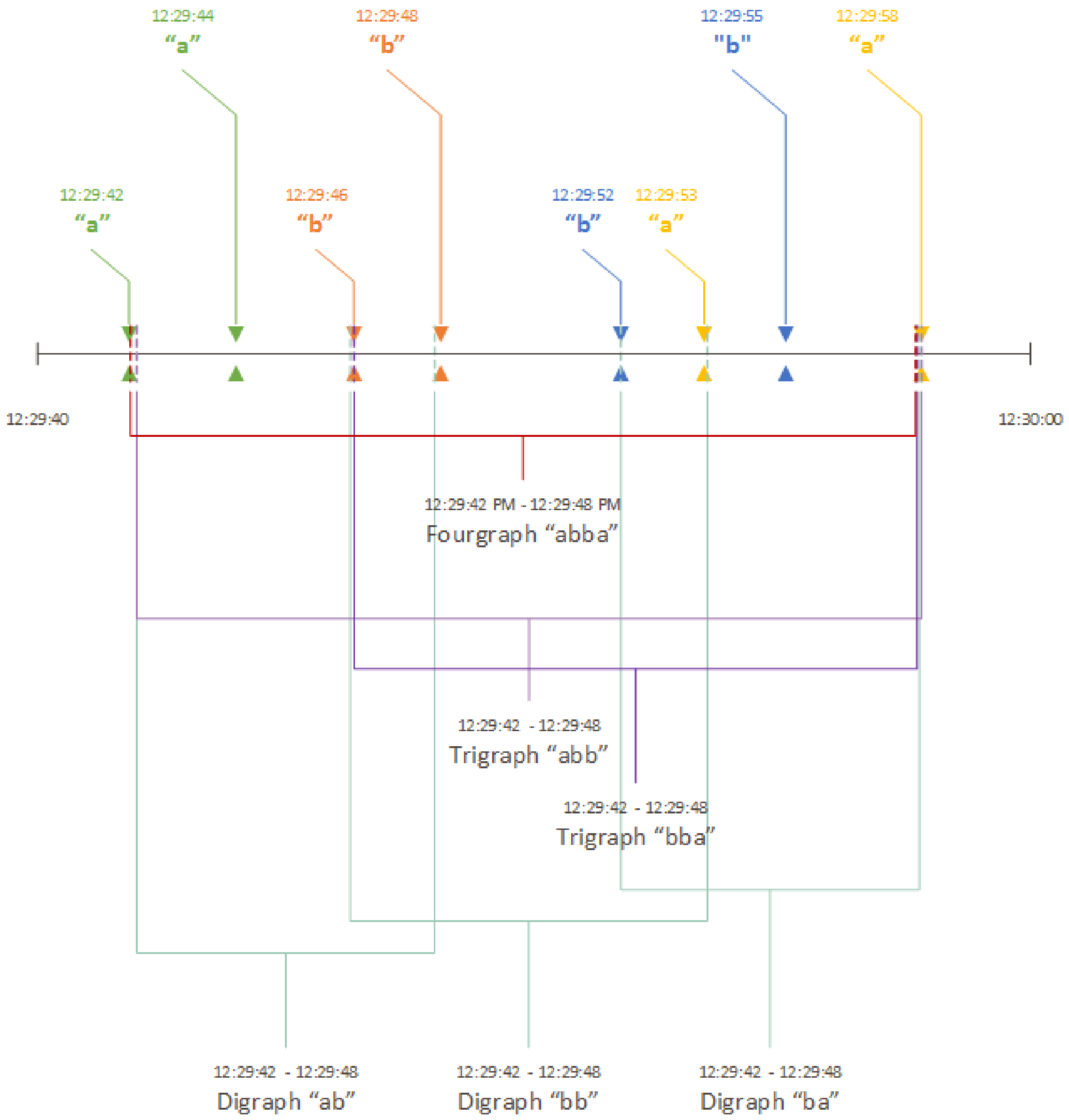


Figure 19 - Composite features (N-Graphs) for the sequence “abba”

Feature Extraction Workflow

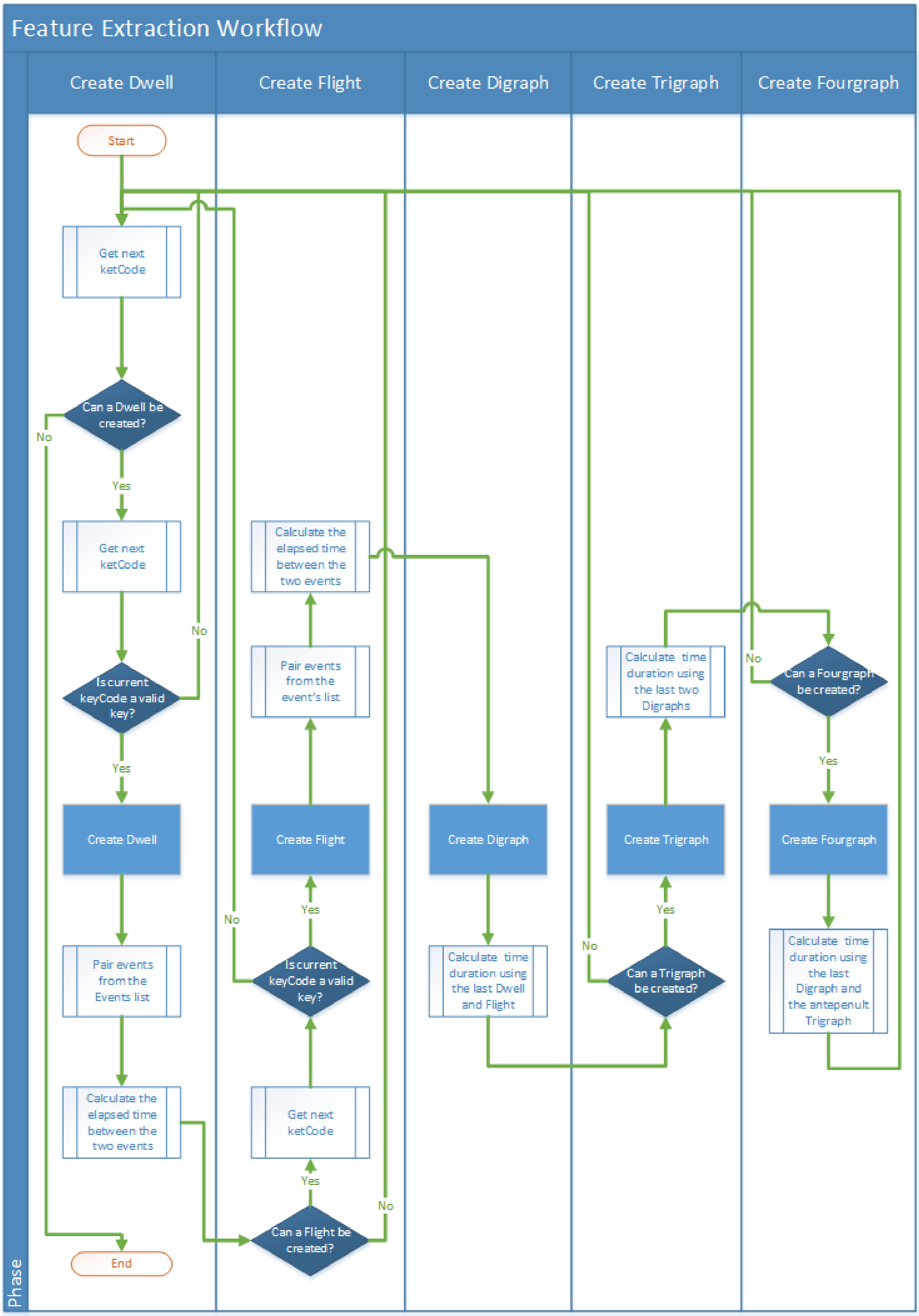

Feature Extraction Workflow
Create Dwell
Create Flight
Create Digraph
Create Trigraph
Create Fourgraph
Phase
Start
Get next ketCode
Can a Dwell be created?
No
Yes
Get next ketCode
Is current keyCode a valid key?
No
Yes
Create Dwell
Pair events from the Events list
Calculate the elapsed time between the two events
End
Calculate the elapsed time between the two events
Pair events from the event's list
Create Flight
Yes
No
Is current keyCode a valid key?
Get next ketCode
Yes
Can a Flight be created?
No
Create Digraph
Calculate time duration using the last Dwell and Flight
Calculate time duration using the last two Digraphs
Create Trigraph
Yes
No
Can a Trigraph be created?
No
Can a Fourgraph be created?
Yes
Create Fourgraph
Calculate time duration using the last Digraph and the antepenult Trigraph

Figure 20 - Feature Extraction Workflow

All features excepting the Dwells need to use the subsequent key code character on the list in order to be constructed. Additionally, the Trigraphs can only be constructed from the second iteration on, and the Fourgraphs from the third iteration on. This can be viewed as a sliding window of a length of 4 in which the current iteration index corresponds to the penultimate position of that sliding window. The window initially starts with a length of 1.

This process of constructing the features runs quickly because a technique of Dynamic Programming is used, in which previous features are used to calculated new ones with a time lookup complexity of O(1).

This algorithm, which is fully described in annex, has a time complexity of O(n), being "n" the length of the key code characters list. The output of this algorithm consists on one array of features for each type of features extracted, which can be Dwells, Flights, Digraphs, Trigraphs, and Fourgraphs. All these arrays of features are merged to a common feature list, which is then shuffled and sent to the Sample Builder module.

5.1.3. User Identification

In order to verify the user claimed identity, first we need to identify the user. This keystroke dynamics based intrusion detection system is primarily intended to be used after the user successfully logs in on the web site, protecting him in a continuous fashion thought out the whole session. This is achieved by continuously evaluating biometrics samples from that claimed identified user against its biometric profile.

A possible alternative model would consist on the evaluation of biometrics samples that belong to an unidentified user. The outcome of such evaluations would drive to the assignment of a potential identity to such unidentified user, as a result of evaluations against a set of biometrics profiles.

Being the first model introduced the one proposed, it is imperative to know beforehand the claimed user identity in order to verify it using its biometrics profile. Considering that the regular user authentication procedure occurs between the user and the web site system, the biometrics intrusion detection system only need to receive an identification token from that web site, so it does not store or manage any user password related data. However, in order to facilitate the possible usage of the same user biometric profiles between different web sites or applications, a solution based on Unique User ID's was adopted.

By doing this, a user can be authenticate on the web site by using OpenID or OAuth 2.0 providers such as Google Plus or Facebook. This is also a common practice in many web sites today. Still, this is regarded as an optional solution, and an existing ID provided by the web site itself could also be used.

In this biometrics gathering application, the user can then log in using Facebook or Google Plus accounts, being the user registered in the intrusion detection system by such provided Unique ID. A problem may arise within this approach, and relates to the linking of different accounts. For instance, if a user logs in using a Facebook account for the first time, the correspondent Unique ID is not yet registered on the intrusion detection system and a new profile for the user ID is created. The problem is that this same user could already have logged in using the Google Plus account, hence could already have a biometrics profile assigned. A typical solution for this problem, but yet to be implemented, is to ask the user to link all of its accounts (Google Plus and Facebook in this case), in order assign just one biometrics profile to that user.

Currently, as the application is running as a Chrome extension on the Facebook page, Cookies are used to read the user Facebook Unique ID token, which is then used to create the biometric profile for that user. However, this is only a temporary solution that was implemented and used on a controlled test environment only.

This module of user identification is responsible to send the user Unique ID token and its related data to the Sample Builder module in order to be included in the biometrics sample that was produced by the user for evaluation.

5.1.4. Application Identification

A web site that wants to consume the Web Service API for intrusion detection needs to be identified and authorized by the intrusion detection system itself. For this, the web site application client needs to hold a valid license API key. The process of generating, assigning and testing this kind of license was not implemented; however, the system supports it, and is designed with that feature in mind.

Currently, for testing purposes, the web application only runs on Facebook, using a Chrome plugin, so no API key license is being check on the web service for intrusion detection. However, hostname and other website related data are being collected on this web application.

5.1.5. Sample Builder

The Sample Builder module is the main module of this web based biometrics gathering application. It is responsible for assembling the user biometrics sample data, the user identification data, and the application identification data.

As referred earlier, this biometrics gathering application runs on a web browser and needs to continuously capture the keystroke events that the user produces while typing. The biometrics data produced by the user is sent to the intrusion detection web service in chunks, so that it can re-authenticate the user periodically. The periodicity of these evaluations is a function that depends on the user typing activity, as each sample is considered closed after the user produces a predefined number of "key up" events.

Currently this value is set to 125 – which in practise – may or may not correspond to the typing of 125 characters, as some keyboard events may be discarded by a finite state machine.

A lower value would increase the rate of sample evaluations; however, the performance accuracy of the intrusion detection system would decrease, as less biometrics data would be sent for evaluation. On the other hand, a higher value would result in an opposite scenario. The value used yields good results and was also chosen because it is the same value that is used on the desktop flavour of TypeWATCH, which is the main inspiration for this web based intrusion detection system.

The Sample Builder module is therefore the central point of this biometrics gathering application, where the all information needed to construct the user biometrics sample is gathered and prepared for further evaluation. Once the sample is ready, it is sent to the communication module, which is the client that will consume the intrusion detection web service API.

5.1.6. Intrusion Detection Web Service Client

This module is the one responsible for the communication between the biometrics gathering web application and the exposed intrusion detection web service API. This module receives from the Sample Builder module samples that are ready to be sent for evaluation.

As stated earlier, these evaluations are made periodically while the user produces real-time biometrics data on the web site using its keyboard. An asynchronous and non-blocking communication is the best fit for this scenario, because on each request for sample evaluation, the application can continue to gather more user biometrics data without waiting blocked for a response from the web service, and also avoiding the need to make a full page refresh to update the page as a result from the response is received. Without this, the user would be constantly bothered with recurrent page refreshes, which could also lead to the loss of important in memory data in the web page. These two considerations are in fact two keys aspects in this application.

Being JavaScript the scripting language used in this web application, the technology used to make such asynchronous and non-blocking requests is the XMLHttpRequest. This is the standard API for JavaScript that enables the invocation of HTTP or HTTPs requests to a given web server.

However, there is a problem that appears in this context, and it is closely related to security restrictions imposed by web browser security policies. It is also linked to the option of using the XMLHttpRequest API between different domains, but this is hard to avoid due to some particular aspects that shape the architecture wanted for this whole intrusion detection system.

One of the main concerns during the architecture design phase was to make the integration of the intrusion detection system with existing customer systems as easy, fast and cost-effectively as possible. So, it is a requirement that the only thing needed to integrate this intrusion detection system with the customer systems is a JavaScript application that is included on the customers web page, being the communication between the web application and the intrusion detection system done directly, without passing though the customer existing servers. By doing this, there is a total independence between the customer server-side and the intrusion detection server-side.

The problem here introduced is that now the biometrics application needs to make browser HTTP cross-domain requests, as it runs on the customer web site, which is hosted on a different domain from the one the intrusion detection web server is hosted. This is a scenario that is not allowed.

In detail, cross-origin domain writes are in fact allowed, but cross-origin domains read are not. This means that if a host A makes a cross-domain request to host B, the request would be received by B, being this considered a write, but the script on A would not be able to read the response from B, as it would be blocked by the browser.

So, in the current scenario, if the communication using the browser was made to the customer web server instead – assuming it is on the same domain of the web page – this problem would not occur, however this would then imply communication between the customer server-side and the intrusion detection server-side. This server-to-server cross domain communication is not restricted by the same-origin-policy that is applied by web-browsers. However, this scenario would difficult the deployment and integration of such intrusion detection service with existing customer systems.

So, in the current scenario, the communication between the web based biometrics gathering application and the intrusion detection web server using the XMLHttpRequest API is then forbidden, as they are hosted on different domains.

A host domain on the internet is defined by its URL (Uniform Resource Locator). The scheme of the HTTP URL' is composed by the transfer protocol, http or https, which is followed by a colon and two back-slashes. Then there is the host, and then the TCP port, which is preceded by a colon. The default port for http is the 80, and for https is the 443, but both can be omitted on the URL. Additionally the URL may end with an absolute path and/or a query. Two hosts are considered to be on different domains, if the protocols, hosts or ports differ.

Same Origin Policy

The Table 3 - Same Origin Policy shows actual examples of domain origin comparisons.

URL	Same Origin?	Reason
http://one.example.com/dir2/b.html	Yes	= Protocol, = Host, = Port
http://one.example.com/dir/inner/a.html	Yes	= Protocol, = Host, = Port
https://one.example.com/secure.html	No	!= Protocol
http://one.example.com:81/dir/etc.html	No	!= Port
http://two.example.com/dir/other.html	No	!= Host

Table 3 - Same Origin Policy

However, communication and exchange of data is crucial on a web environment, so there are natural and standard ways to allow secure communication between hosts seating on different domains.

The natural approach is the usage of native and standard HTML element tags. For instance, when a web page makes a request for an image to be displayed that is hosted on a different domain, a *img* HTML tag is used, which enables the browser to make such HTTP GET request in a secure way. The data receive can then be considered safe if the content type on the HTTP HEAD matches the content type of data received. Similarly, the same security control actions occur when the web page makes a HTTP POST request by using the native and standard *form* HTML tag.

Likewise, when the request content type is a JavaScript file, the request is made possible by including the file source on a HTML *script* tag, and by doing this, once the file is downloaded via HTTP GET, it is granted permission for its execution. Then file is then executed. However, it is run on the current web page domain, not on the domain from which the file originated. So, when executed, if that script file wants make requests using XMLHttpRequest to a host domain different than the one where the script is running, the web browser will block such requests.

This is much the current situation with the biometrics gathering application. It is composed by JavaScript files, so they can be downloaded and run without problems on the web page just by being included on a HTML *script* tags. This happens naturally because the request made is a HTTP (e.g. GET) request using standard HTML tags. However, after the script files are loaded, and executed, the scenario is different, as the goal now is to make cross-domain HTTP requests to send data using the XMLHttpRequest API instead, which is forbidden.

As the data format that is exchanged in these requests is serialized JSON (JavaScript Object Notation), one popular solution to overcome this cross-domain problem is the use of JSONP (JSON with Padding). This technique makes possible to grab a script block from cross-domain sites by including – behind the scenes – a HTML *script* element tag with the source destination, which in turn, by means of a forced HTTP GET request, will return a response from that source containing a script block. This script will execute an existing call back function whose name can be specified as a parameter on such HTTP GET request. This call back function will receive the desired JSON data as function parameter.

However, this technique does no work with HTTP POST requests, which are the ones that are intended to be used by the biometrics gathering application to send data to the intrusion detection web service. The HTTP GET method was not designed to send data, but to request data. It uses a query composed by parameters and its corresponding values to fetch the intended data from the target web server. On the contrary, HTTP POST method was especially designed to send data. The amount of data that can be sent with a HTTP POST is much larger than the one that can be sent using the HTTP GET method, yet, on both cases, the limits also depend on the browser used and on the server configurations. So, this JSONP technique ended up not being adopted.

CORS (Cross-Origin Resource Sharing)

The solution adopted for this problem relies on the use of CORS (Cross-Origin Resource Sharing) [18] [19], which is a new mechanism that allows a web page to make cross-domain XMLHttpRequest requests by defining exceptions to the same origin security policy. This new specification defines a set of headers that are exchanged between the client and the server, allowing the server to relax the cross-domain restrictions for all, or some external domains, and also for all, or some, HTTP verbs. The "Access-Control-Request-Method", the "Access-Control-Request-Headers", and the "Access-Control-Allow-Origin HTTP" are some of the headers that are added to the request.

The definition and configuration of these permissions is done on the server side. The client browser then asks for permissions by sending first a HTTP OPTIONS request. This is called a prefight request and it is needed when the request type involved intends to make changes on the server. That is exactly what happens with the HTTP POST requests whose content type is set to "application/json".

If the server approves the intended request by the browser, it responds positively, by setting the "Access-Control-Allow-Origin HTTP" header to the web page current origin. The browser reads the response, from the HTTP OPTIONS request, and if there is an origin match, the browser will not block the intended request. So, after this, the actual HTTP POST request is made by the client to the server. This works, therefore, as a client-server agreement.

However, this can create some security issues. It is important to define on the server side which hosts are allowed to make such requests, and in what terms these requests will be made. To overcome this potential security hole, an authentication process must be implemented. It is true that the exposed web services can restrict its functionality, by defining required parameters. One of them could be the requirement for the inclusion of a valid API Key, which would be mandatory for each request. The purpose of the key would be to verify the identity of the host, and to set a set of authorized actions for the application living in that host. The key would first be assigned to the client and included in the JavaScript biometrics gathering application. This API key solution is part of the design of this system, but was not yet implemented.

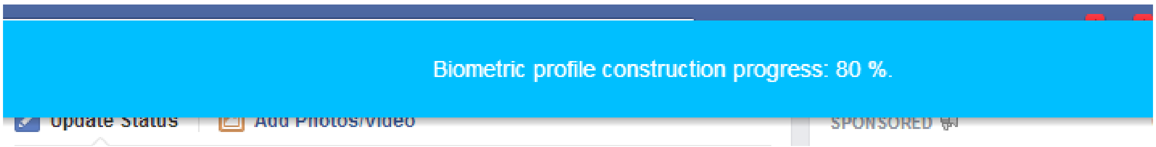
In the current scenario, after surpassing all of this connection issues, the intrusion detection web server responds to the sample evaluation request by returning an evaluation result.

5.1.7. Intrusion Detection Alarms

When the biometrics gathering application makes sample evaluation request to the intrusion detection web service, it receives back a response. Depending of if the user is or is not on the enrolment phase, a different response is generated.

Enrolment Progress Notification - Example

If the user is on the enrolment phase, it means that its biometrics template is not complete, hence, not ready for being target of evaluations. In this case, and for testing purposes, the result does not contain the actual evaluation result, but rather the indication of the current construction progress of the user biometrics template. A message pops up to showing that indication.

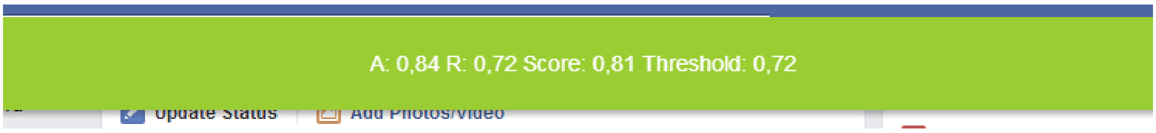


Biometric profile construction progress: 80 %.

Figure 21 - Enrolment Progress Notification

Successful User Authentication - Example

On the other hand, if the user has a biometrics template completed, and ready for evaluation, an actual evaluation result is yield. If an intrusion was actually detected, the result returned contains the type of alarm generated. Otherwise, and again, for testing purposes, a message pop ups indicating the evaluation success, by showing the evaluation scores.



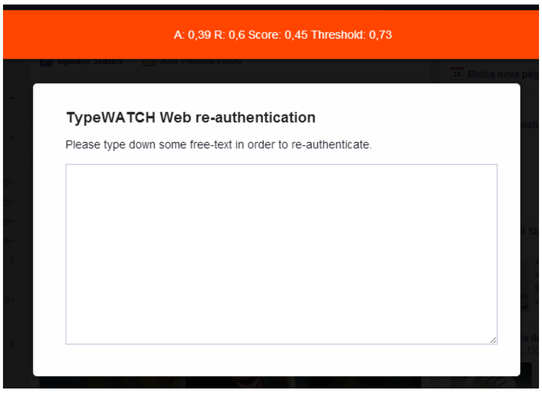
A: 0,84 R: 0,72 Score: 0,81 Threshold: 0,72

Figure 22 - Successful user authentication

There are 3 types of alarms. The "Yellow" alarm, the "Orange" alarm, and the "Red" alarm. These alarms can then be assigned to some actions that are triggered on this client side. These actions can be extended to fit the needs of the web site customer – and being this done with JavaScript – they can easily be translated into calls to the server that hosts the web page in order to initiate some possible security measures to prevent the intruder to take malicious actions.

Intrusion Detection - Example

The current application consists on a Chrome extension running on Facebook. When it faces a "Red" alarm, it blocks the screen and presents a modal window to the user. The user is prompted to type another sample in order to regain access to its Facebook page. It will only regain access if he manages to receive a subsequent successful sample evaluation.

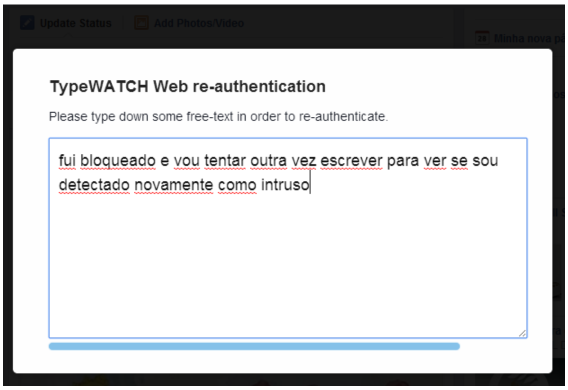


A: 0,39 R: 0,6 Score: 0,45 Threshold: 0,73

TypeWATCH Web re-authentication

Please type down some free-text in order to re-authenticate.

Figure 23 - Intrusion Detected



TypeWATCH Web re-authentication

Please type down some free-text in order to re-authenticate.

fui bloqueado e vou tentar outra vez escrever para ver se sou detectado novamente como intruso

Figure 24 – Re-Authentication Process

5.2. Intrusion Detection Service

The purpose of the intrusion detection service is to provide means to evaluate the users biometrics samples in order to verify their claimed user identify, in a continuous manner, over the entire user session. The web sites that want to use this service need to include a client application in order to consume this service. That application is the biometrics gathering application, and it needs to comply with the web service contract that is exposed by the intrusion detection web server.

5.2.1. Intrusion Detection Web Service API

The web service that is exposed is based on the Microsoft's Windows Communication Foundation technology (WCF), which is an API for .NET that enables the creation of connected, service-oriented applications.

This service-oriented application, the intrusion detection system, currently defines two different services.

Application Registry Web Service

One of the services, the "Application Registry" service, provides a way for an application or web site to be registered in the intrusion detection system. This registry is to be done by the system administrators. A license key should be assigned in order to identify and authorize further evaluations requests for that application. However, the management of these licenses is yet to be implemented.

Intrusion Detection Web Service

This is the main web service that is exposed by the system and is the one responsible for evaluating the users' biometrics samples in order to detect potential intrusions.

The web service contract requires the API client application to send a user's biometrics sample for evaluation, the identification of the user, and identification of the web-site / application. The web service then delegates the process of such request to the intrusion detection business logic layer. After the process is completed, the web service returns to the client application a response that is to be parsed by the API client application.

5.2.2. Application Authentication

This is the module responsible for the verification of the web-site / application identity, by validating a license key that is assigned to the web site or application that comes in the request of a the Intrusion Detection Web Service call.

This module is also responsible for the association of new users to that given web site or application registry entry.

However, this module is yet to be fully implemented.

5.2.3. User Identification

This module is the one responsible to verify if a user identification that received is already registered on the system, for a given application.

By default, if the user is not yet registered in the system using that web-site or application, a new biometrics profile is created and assigned to the user. On the other hand, if the received user identification matches a registry entry on the system for a given web site or application, is biometrics profile is returned.

This module should be also responsible for the linking of different OpenID / OAuth accounts to the same user, but is not implemented yet.

5.2.4. Biometrics Sample Evaluation

This is the core module of this biometrics intrusion detection system. It is responsible for the evaluation and classification of the biometrics sample that it receives.

The actual workflow of this sample evaluation process depends on a preliminary and quick analysis of some business logic configuration parameters in conjunction with the existing data on the target user biometric profile.

To accomplish this, these decision making criteria are gathered into one single object class that is used to assert the state of the target biometrics profile, and to define which tasks should be performed during this evaluation process. For instance, this can be used to recognize the phase, or phases, in which the target user biometrics profile is, i.e. enrolment phase, evaluation phase, or classification phase. This can also be used to decide if an additional biometrics sample pre-processing is required before the sample evaluation, or if a dynamic or static threshold is to be used on the biometrics sample classification, or even to simply recognize if the target biometrics template is full.

All of these assertions are translated into tasks that are subsequently delegated to the appropriate sub-modules.

During this workflow, there is also the need to maintain the raw, the processed, and the target merged biometrics samples indexed by one or more attributes, such as "code" or "type". To implement this, dictionary structures are used, being the performance lookup on the search, reunion and intersection of features operations highly improved.

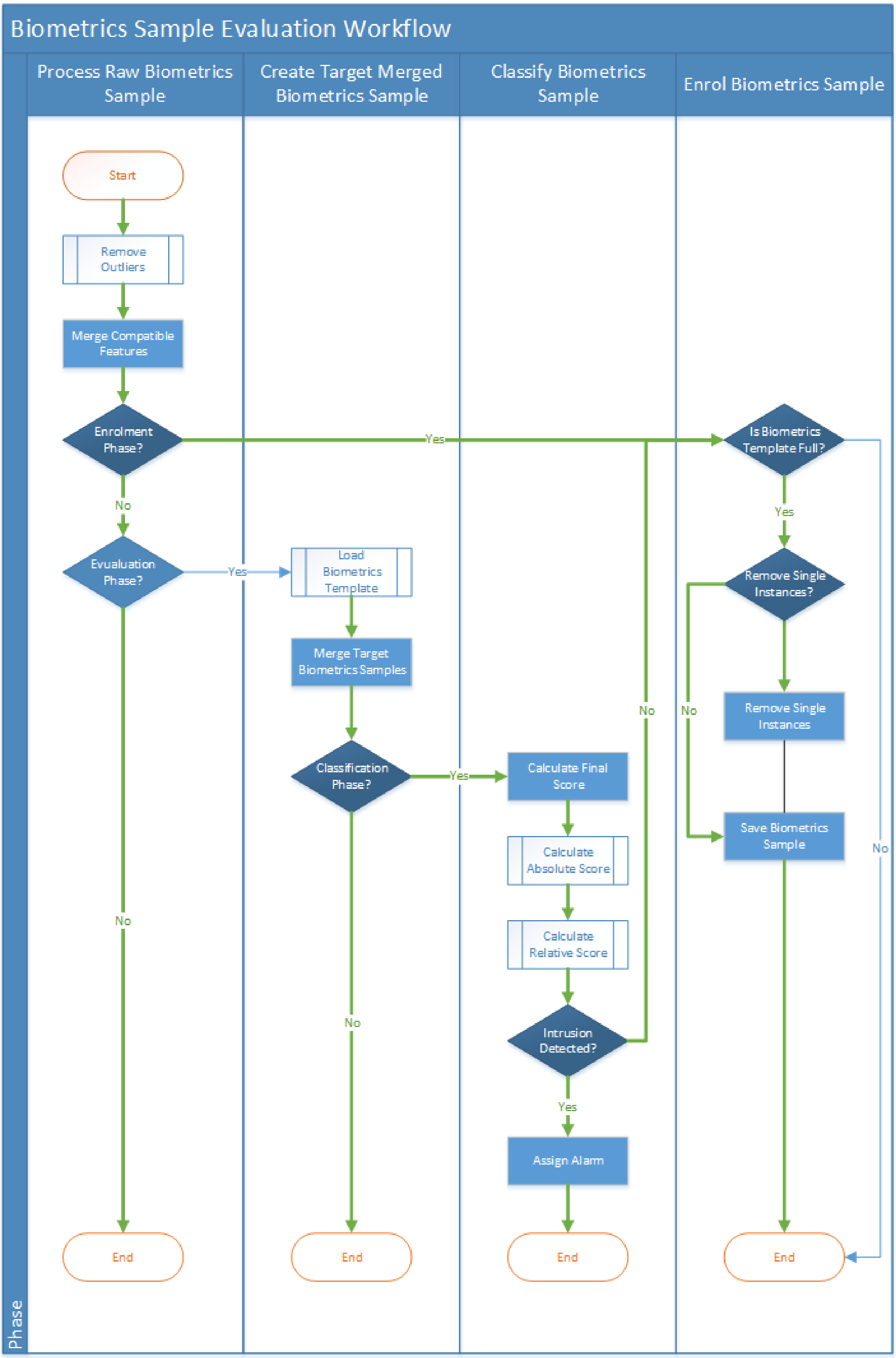


Figure 25 - Biometrics Sample Evaluation

5.2.5. Data Pre-Processing

Raw Sample Processing

The user biometrics sample received for classification comes in a "raw" state. Further processing is needed before the actual sample evaluation. To start with, an outlier removal procedure is performed, and then a merge of common features is made.

Outlier Removal

The process of outlier removal is an important one. Outlier features represent, ideally, the smaller fraction of the user biometrics features that are considered to be atypical among the set of features. Despite of being features that were produced by the user, they are usually the result of a circumstantial abnormal typing behaviour, i.e. a random pause or hesitation on typing, or even the result of measurement errors, so they should be removed.

Being the time duration property the fundamental characteristic of a feature, this property is the one that is used as the discrimination factor. The outliers are the ones with time duration values that are too extreme or distant from the majority of the others ones. Therefore, when all the features are sorted by its time property value, the features to be removed are the ones with the lower and higher values.

To find out which are the outlier features on a raw sample, a popular descriptive statistics technique is applied. This technique consists on the analysis of the statistical dispersion of the sample by finding out its Quartiles (Q_n) and the Interquartile Range (IQR) for the particular sample. It is important to note that this technique does not assume any typical distribution.

To calculate the Quartiles, the sample must be sorted by the desired property value, which in this case is the time duration of the features.

The quartiles are defined as follows:

The First Quartile (Q1),

The Q1, also called the lower quartile is the point that splits the lowest 25% of the data from the remaining 75%.

The Second Quartile (Q2)

The Q2 represents the median value, being the point that splits the first half of the data (50% of the sample data) from the second half of the data (50% of the sample data).

The Third Quartile (Q3),

The Q3, also called the upper quartile, is the point that splits the highest 25% of the data from the remaining lowest 75% of the data.

IQR

The Interquartile Range is given by the range between the upper and lower quartiles.

$$IQR = Q3 - Q1$$

These quartiles and interquartile values are useful to detect outliers in a given data set. The standard approach to classify a feature as an outlier is given by the following formula:

$$Outlier \notin](Q1 - (1.5 * IQR), Q3 + (1.5 * IQR))[$$

This means that features with a time value that falls below Q1 - 1.5(IQR) or above Q3 + 1.5(IQR) are considered outliers in the sample.

The outlier removal procedure is run several times. First it is run for each feature that shares, simultaneously, the same "code" and "type". For instance, all digraphs "ab" features, or all trigraphs "ert" features. The identified outliers are removed. This outlier removal procedure is then repeated for all the features that share the same "type" attribute value. For instance, all dwells or flights. Once more, the outlier features that are identified are removed.

It is important to note that the process of finding the outliers in a given feature list can only be run if the list contains 4 or more features. This is due to the fact that it is not possible to calculate the quartiles and the interquartile range with less than 4 features in the list. However, when this happens, it usually only happens when the procedure is run on a list made by features that share, simultaneously, the same "code" and "type".

After all this outlier removal procedure is complete, it is observed that, on average, the total amount of outliers removed represents around 10% of the total features on the original sample.

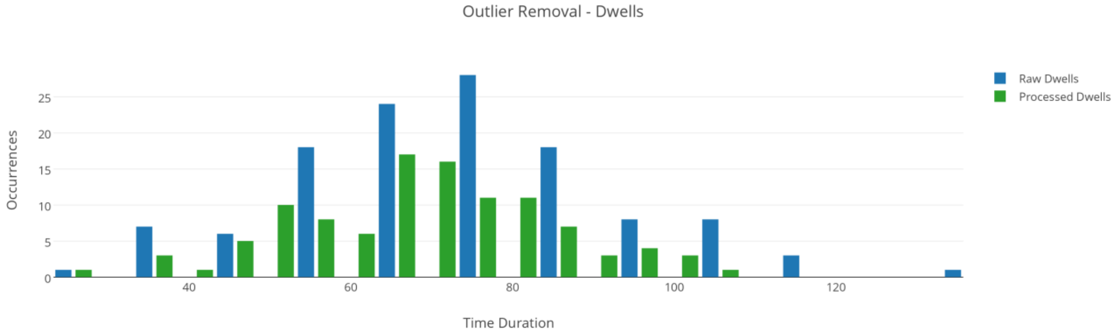


Figure 26 - Outlier Removal - Dwell Distribution Comparison (1 sample)

The Figure 26 - Outlier Removal - Dwell Distribution Comparison shows a comparison of the time duration distribution of the dwells features before and after the process of outlier removal in a given biometrics sample.

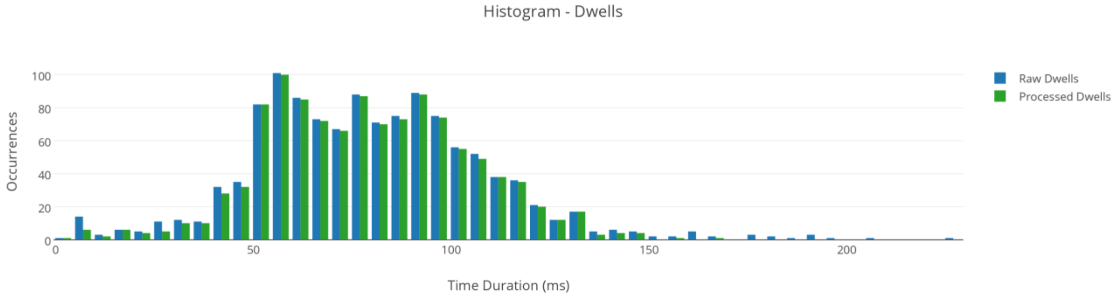


Figure 27 - Outlier Removal - Cumulative Dwell Distribution Comparison (10 samples)

The figure 27 shows a comparison of the time duration distribution of the dwells features before and after the process of outlier removal in in a set of 10 contiguous biometrics samples

Similar figures are available in annex for the remaining type of features.

Merge Raw Features

This raw sample processing includes a process of data reduction, in which similar features are grouped together into one representative feature, in order to produce a smaller, yet descriptive biometrics sample.

Once this process is completed, the biometrics sample to be evaluated is now composed by the new set of features that were created. This processed biometrics sample is now almost ready to be evaluated against the target user biometrics template.

Merge Biometrics Samples from Template

To evaluate a user biometrics sample against its biometric profile template, a selection of biometrics samples targeted for evaluation is proposed. Usually, the most recent *n* samples are the ones to be considered, as they reflect that most recent typing behaviour of the user. The actual number of samples to be selected is defined by a business logic configuration parameter. Currently, this parameter is set to 15, being then the most recent 15 biometrics samples the ones chosen for evaluation.

Before the evaluation proceeds, a data reduction is made in which similar features from all the selected biometrics samples are grouped together.

It is important to note that not all merged features that are created are added to the final merged sample. The merged features with an occurrence value lower than a predefined value are discarded, because they are considered to be not representative enough, due to its lower occurrence frequency. This value is a business logic configuration parameter, so can be easily adjusted.

5.2.6. Classification

The classification model used to classify a biometrics sample is a one-class statistical classifier, being the unary-class defined as the *legitimate* class. This class represents a population of biometrics data that belongs to the legitimate user.

This one-class classifier is based on supervised learning, so the classification of new data is then a function that depends on prior knowledge, being this, a limited training data set that is built during the training phase – i.e. enrolment phase. It is assumed that a training data set contains only data that belongs to the legitimate user. This training data set is also only a fraction of all the data that is classified as *legitimate*; however, it is a dynamic training data set.

The actual sample classification is a process in which two distinct evaluation models are applied in order to classify the new data by measuring the similarity between the biometrics sample to be evaluated and the target merged biometrics sample from the user profile. One of the models calculates the Absolute Score, and the other calculates the Relative Score. Before these two models are applied, an additional procedure takes places.

Both of these models do not evaluate the entire biometrics sample, but only evaluate pairs of compatible features, that is, the features that are present in both samples – the one to be evaluated and the target one. To achieve this, a filtering procedure is run in which the two lists of features are intersected, producing a tuple list of shared compatible features. These pairs of features are then the inputs of each of the evaluation models. The output generated by each of the model is also of the same type, being in this case the evaluation scores.

Absolute Score

The Absolute Score is the result of an evaluation model based on distance metrics. It calculates the similarity between pairs of features (F_i, MF_i) by measuring the distance between the "average" time duration value attribute of a feature (F_i) from the sample to be evaluated against the "average" time duration value attribute of the corresponding merged feature (MF_i) from the target user merged biometrics sample that represents the user biometrics template.

For each pair of compatible features to be evaluated, an acceptance interval is defined. This interval defines a lower and upper threshold around the "average" time duration attribute value of the merged feature that belongs to the target user merged biometrics sample.

If the value of the feature to be evaluated lays out of the interval, either by being lower than the lower threshold or higher than the upper threshold, the feature to be evaluated is considered not similar. Otherwise, it is considered similar. If it is considered similar it will have a score D_i of 1. If not, the feature will have an assigned D_i score of 0. Each feature evaluated will have then an individual score assigned.

As seen, each individual evaluation feature score depends on the result of such distance metrics (D_i), however, the final absolute individual score (S_i) also depends on its frequency on the sample to be evaluated. This frequency, which ends up being a weight (W_i), corresponds to the "occurrences" attribute value of the feature to be evaluated. This is to afford the fact that, all the original instances of a given feature were previously merged into a new correspondent one that represents all of those original instances of such given feature.

The final absolute score of the sample evaluation is the ratio between the sum of all individual absolute scores and the maximum possible sum of all individual absolute scores i.e. in a situation where all features were considered to be similar to the corresponding target ones. Therefore, being the absolute score a ratio, it is always defined by a value between 0 and 1, inclusive. For instance, and giving extreme examples, if all of the individual features would be evaluated as similar, the ratio would be equal to 1. Likewise, if none of the individual features would be considered similar, then the ratio would be equal to 0. The higher the absolute score, the similar the biometric sample to be evaluated is to the merged biometrics sample produced from the user biometrics template.

Relative Score

The relative score model is associated to the fact that it is frequent to observe that, the set of keys – or sequence of keys – that are typed by a user in a fastest or slowest way are usually nearly the same. In other words, there are a significantly stable set of keys – or sequence of keys – that, on average, the user tends to type faster, and a stable set of keys – or sequence of keys – that, on average, the user tends to types in a slower way. This pattern is frequently observed regardless of the variance on the speed of typing yield by the user. This variance can occur due to some psychological or environmental factors, such as stress, fatigue or environmental noise or distractions.

The relative score is a result of an evaluation model based on such pattern. The distance metrics used is the relative disorder between the correspondent pairs of features. To calculate such disorder, the elements of these pairs must be individually sorted by the "average" duration time attribute value, producing two independent ordered set of features.

In detail, this evaluation model relates the pairs of compatible features by finding out the relative displacement position between those compatible features on the lists. A final

Relative Score is produced, being defined with a normalized value between 0 and 1, inclusive. The higher relative score, the higher similarity degree.

Final Score

Each of the two models to be applied return a score value, a decimal value comprehended between 0 and 1. These scores are then weighted to produce the evaluation Final Score. The weights that are used are business logic parameters that can be easily configured. Currently, those are defined as such:

- 0.75: the Absolute Score.
- 0.25: the Relative Score.

The Absolute Score model as a higher weight because it showed to be a more discriminative model the Relative Score model. However, both models are important, yet, for different reasons.

Concrete Classification

The goal of this classifier is to classify a given unclassified user biometrics sample as being legitimate, or not. This classifier takes the Final Score evaluation result and a given decision threshold as the input for the classification.

A biometric sample is classified as *legitimate* by asserting the following expression:

$$Final\ Score\ \in [threshold, 1], \qquad threshold \in [0,1]$$

Which means that the final score must be above a given threshold in order to the biometrics sample be classified as *legitimate*. However, if this is not asserted, the biometrics sample is not considered to be a legitimate one, so a potential intruder is detected.

5.2.7. Intrusion Detection Alarms

When a potential intrusion is detected, an alarm produced. Currently, there are 3 distinct degrees of severity associated with the potential intrusion detection, and these severity degrees are assigned to 3 different intrusion detection alarms. These alarms are defined as follows:

- "Yellow" or type 1 alarm
- "Orange" or type 2 alarm
- "Red" or type 3 alarm

The least severe is the type 1 alarm, the most severe is the type 3 alarm, and the type 2 alarm stands in the middle. A type 0 alarm can be also defined, but it is actually a non-alarm, or a "Green" alarm, and it can be assigned when no intrusion is detected.

This alarm schema was considered as a way of fighting the problem that results from the classification of biometrics sample whose evaluation result stands too close to the decision threshold. These evaluations denote a degree of uncertainty. When an evaluation result is in this condition, it is considered to be in a grey area. The potential intrusions that are assigned with the type 1 and type 2 alarms ate the ones that belong to this area.

The Final Score previously used for the classification is the criteria used to differentiate and assign a type of alarm to a given evaluation result. The assigned of such alarms depend on some business logic configuration parameter values. These values are relative to the decision threshold used to classify the user biometrics sample, and act as the delimiter for the alarms to be generated.

The "Yellow" parameter is currently set to 0. The "Orange" is set to 0.025 and, the "Red" one is set 0.05.

All this procedure resembles to a multi-class classifier, however, it is not, because the classification is made using only target features that are labelled with the legitimate class.

5.2.8. Enrolment Process

The user biometrics enrolment process is divided into the following two different phases:

- Automatic Enrolment
- Dynamics Enrolment

The Automatic Enrolment phase is mandatory and corresponds to the phase in which the user builds its biometrics template from scratch.

Within the Dynamic Enrolments phase, a new biometrics sample is only added to the user biometrics template if the user biometrics sample is classified as *legitimate*. Regarding the enrolment policy, it is applied a first in first out policy, however, a biometrics sample is only removed from the biometrics template to give space to a new one when to total number of biometrics sample reaches a given limit. This limited is defined by a business logic configuration parameter and it is currently set to 250 biometrics samples.

Currently, when the Automatic Enrolment phase is complete, the user biometrics template enters in the Dynamic Enrolment phase. All of this can be controlled by business logic configuration parameters that define when the Automatic Enrolment phase is completed, and when the Dynamic Enrolment phases is initiated. At this time, the Automatic Enrolment phase is completed when the user biometrics profile enrols a given number of biometrics samples, being this number currently set to 15. For the Dynamic Enrolment phase, the control parameter is also set to 15.

5.3. Dynamic Threshold

The classifier used by this intrusion detection system classifies the biometrics samples using distance metrics. The final evaluation score is compared with a given threshold. Biometrics samples whose evaluation scores stand above the threshold are classified as *Legitimate*. The biometrics sample is considered an outlier otherwise. A central problem with this kind of classification models is related with the difficult to set a good decision threshold.

The definition of such threshold is crucial to regulate the False Acceptance and False Rejection Rates. If the threshold is set too high, there will be an increase on the False Rejection Rate, and a decrease on the False Acceptance Rate. On the other hand, if the threshold is set too low, the False Acceptance Rate will increase, and the False Rejection Rate wills decrease.

In order to define a potential optimal threshold, a dynamic threshold is used instead. There is a threshold for each biometrics template. This threshold is updated periodically, and is calculated by means of a set of artificial attacks using the K-Fold Cross-Validation technique.

K-Fold Cross-Validation

The Cross-Validation is a model validation technique that is used to evaluate the accuracy performance of given statistical-analysis.-based prediction model. It is used to realize how well the model will behave when testing against an independent data set – other than the original training set. This is called generalization.

To apply this technique, a training data set and an independent data is need. However, one of the advantages of using K-Fold Cross-Validation technique is that it actually does not require additional data than original training data set. This original training data set is simply divided into K folds, being one of the folds used as the testing set i.e. the independent set, and the remaining K – 1 folds used as the training data set. The prediction model is applied using this data, returning an evaluation result. This whole process is applied K times, by choosing a different fold as the independent data set. Each fold is only used once as the testing set, but can be used more than once as part of the training set. The testing and the training sets are disjoint ones.

The K evaluation results that are produced during this process are usually average to find a final evaluation result, which corresponds to an evaluation score of the prediction model. The recommend K value when it comes to assess the accuracy performance of a prediction model is 10. However, this is not a strict rule, and for this particular purpose, a 16-Fold-Cross Validation technique is used instead, since the evaluation module currently in use uses a merged sample that are constructed from 15 biometrics samples, in order to be used as the target biometrics sample.

In the current scenario, this cross-validation technique is also useful to discern some evaluation patterns by means of artificial attacks. The one-class classifier that is used by this intruder detection system defines a threshold to make such classification decision.

When cross-validation is used to make artificial attacks using randomly chosen intruder samples from different users i.e. samples that are assumed to be outliers, against the user biometric template, it is possible to discern some sub-clusters of data containing those outlier samples. The values of the evaluation results given by the application of this K-Fold Cross-Validation technique are then the criteria that form these sub-clusters.

These are called sub-clusters because they are part of the bigger cluster, that is, the cluster that is formed by the all the intruder sample evaluation results. It is important to note that these sub-clusters may overlap. However, the central problem here is that the big outlier cluster formed by this sub-clusters may also overlap with one other important cluster, the one formed by the evaluation results from the artificial attacks made against the user biometrics profile by using only biometrics sample that are assumed to belong to the legitimate user. This overlap corresponds to a grey area, in which evaluation scores that stand in this area may induce evaluation errors, that is, False Rejections and False Acceptances. The goal here is to minimize these errors, adjusting them as intended. Therefore, this K-Fold Cross-Validation technique is applied in order to find an optimal threshold value that minimizes such evaluations errors.

A unique source of legitimate target biometrics samples are used on both artificial attacks sets. Currently, the legitimate biometrics samples correspond to the most recent 15 biometrics samples from the user profile.

Additionally, the current amount of biometric samples used to attack the target user biometrics samples are set to:

- 105 intruder artificial attacks – composed by 15 randomly chosen biometrics samples from each of the 7 randomly chosen intruders.
- 105 randomly chosen legitimate artificial "attacks".

These values where chosen because the False Acceptance Rates are calculated from the 105 intrusion artificial attacks, and the False Rejection Rates from the 105 self-artificial attacks. The False Acceptance Ratio is calculated by counting the total intruder biometrics samples that were classified as legitimate using the current threshold, and the total intrusion attacks performed. Similarly, the False Rejection Rate is calculated by counting the total self-biometrics samples that were classified as outliers, and the total self-attacks performed.

Being this procedure a little time and processing expansive, it is important to choose the lowest value possible that is higher than 100, because these rates are given as a percentage value.

However, this amount of samples is only used when available. When this is not the case, the lowest value of both could be used instead. This would ensure that an equal amount of samples were used in both artificial attacks sets. This is called sample stratification. However, as the goal is to use 105 samples whenever it is possible, this stratification strategy is dropped, in order to ensure more accurate FAR and FRR values.

Threshold Update

The update of the threshold is made as a result of such artificial attacks. The resulting FAR and FRR values are compared. If the values are equal, nothing changes. However, if the FRR is higher than the FAR, the threshold is decreased by a certain amount. Otherwise, it is decreased by the same amount. This quantity is a business logic configuration value and is currently set to 0.01.

Feature Analysis

A 16-Fold Self-Cross-Validation technique was also applied to try to understand the importance that each feature has for a particular user. This was done by calculating, on each fold, the individual score of each feature evaluated, and the prevalence of such feature in the biometrics sample. Additionally, for each feature, there is a sample evaluation score with, and without the feature. All the average of all of these property results, the correspondent standard deviation and the total occurrences registered for further analysis.

This procedure is run, simultaneously, for each feature, but also for each type of feature.

Chapter 6
Validation Results

The validation exercise is crucial to assess some of the anticipated goals for this thesis. The main objective of this thesis is to evaluate the actual and concrete applicability of Behavioural Biometrics in the World Wide Web for the purpose of user authentication.

Being the science of Keystroke Dynamics a highly studied field on Behavioural Biometrics, it is surprising how it stills such an unfamiliar authentication technique to the majority of internet end-users. Some of the existing theoretical knowledge base was put into practise, by means of a web-based proof of concept, and is now the target of validation.

The goal of this chapter is to try to answer to the following question:

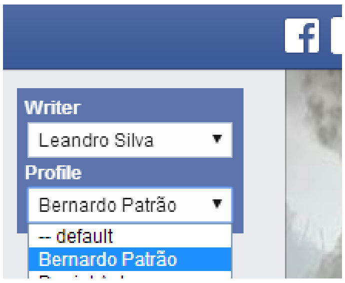
If an intrusion detection system relies on keystroke dynamics analysis, can it be successfully used on an internet environment?

6.1. Observational Study

The validation process of this thesis was based on some observational studies. These studies were not strictly formal ones due to some logistical and time limitations. However, it was possible to run the desired planned experiments with a small set of people, using their on computers – when possible – on their own usual environment.

Experiment

The people that contributed to this experiment were asked to type freely on a Facebook web-page, using the Chrome extension biometrics gathering application. They were asked to type whatever they wanted, with no time restrictions, and wherever they wanted, including on the Facebook chat. Real Facebook accounts were only used when the users were using their only personal computers. Otherwise, in order to protect the user's privacy, an emulation process was conducted.



Writer
Leandro Silva
Profile
Bernardo Patrão
-- default
Bernardo Patrão

Figure 28 - Profile Identification Emulator

In this emulation process, people were asked to select from a list, the identification of the user who is typing, and the identification of the user targeted profile. This made the supervised attacks to the user profiles a really easy, simple, and quiet process.

Environment

All computers used were laptops with the QWERTY layout. The browser in used was an up-to-date version of Google Chrome (v35). This browser was the only one used because this experiment required the installation of the Chrome extension i.e. the biometrics gathering application.

The language used in this experiment was the English and the Portuguese languages.

The observation studies occurred in two different environments:

- **A**: one at the offices of a Software developer company, with in-house employees, using their own computers. This experiment occurred over the course of 8 work days, during the labour hours.
- **B**: one at the living room of University students, using a single computer. This experiment occurred at two separated days, during 2 hours, after labour hours.

However, due to some logistical limitations that translated into a scarcity of input data, the data collected from these two experiments were merged into one.

People Involved

There were a total of 17 people involved in these two experiments:

- **A:** 12 people (7 people completed the experiments process)
- **B:** 5 people (5 people completed the experiments process)

Therefore, there were only a total of 11 people formally involved in these experiments. The data analysed within this observational studied resulted from these 11 people typing activity.

People Profile

- The 7 people from the experiment **A** all work at the same Software developer company – and also of the same work team – so they all use a computer on a daily basis – they even used their own work computers. Despite of the Portuguese being the native written, read, and talked language, they all also know how to talk, read, and type properly in the English language.
- The 5 people from the experiment **B** are all university or former university students that use a computer on a daily basis. Despite of the Portuguese being the native written, read, and talked language, they all also know how to talk, read, and type properly in the English language.

Data Collected

There were collected a total of 679 biometrics samples from the all the 12 persons involved who completed the enrolment phase.

6.2. Results

The results extracted from these experiments are mostly related with time and accuracy performance, and the ratio of outliers removed.

6.2.1. Time Performance Results

The following time performance metrics were taken into account.

Feature Extraction Time

These are the average time duration that the biometrics gathering application takes to process a biometrics sample i.e. extract the features. Detailed time information is available in the document *"Validation Results – Behavioural Biometrics in the World Wide Web"* in annex.

Internal Profile ID										
F1…	32..	17…	27…	4A…	29…	94…	CD…	DA…	D8…	5B…
Individual Average (ms)										
0,006	0,008	0,004	0,011	0,012	0,012	0,008	0,007	0,011	0,009	0,009
Global Average (ms)										
0,009										

Table 4 - Averaged time duration of the feature extraction process per sample

It was observe that, on average, a feature extraction of a sample takes around 10 milliseconds. However, this value may depend on the browser and machine used, and also on their processing activity load.

Ultimately, this is an excellent result, and it does not seem to any visible impact on the responsiveness of the target web-site.

Sample Evaluation

This is the time that it takes to send a biometrics sample evaluation to the intrusion detection service, process a sample evaluation request, and yield a response to back to the client.

Detailed time information is available in the document "Validation Results – Behavioural Biometrics in the World Wide Web" in annex.

Internal Profile ID										
F1…	32..	17…	27…	4A…	29…	94…	CD…	DA…	D8…	5B…
Individual Average (ms)										
1,019	1,111	0,916	0,933	1,199	1,158	1,168	0,633	0,801	1,171	1,421
Global Average (ms)										
1,048										

Table 5 - Averaged time duration of the sample evaluation process

It was observed that, on average, a biometrics sample evaluation process takes around 1 second. However, this value may also depend on network and server loads.

This is a good result, since a user may take – at the best case – around 20 seconds to type the 125 required characters to build a biometric sample for evaluation. Further analysis on this topic must be conducted.

This extra 1 second is then just a fraction of the time it takes to gather enough data to build a biometrics sample.

Threshold Update

These are the averaged time duration that the intrusion detection service takes to update the dynamic classification threshold.

Internal Profile ID										
F1…	32..	17…	27…	4A…	29…	94…	CD…	DA…	D8…	5B…
Individual Average (ms)										
33,703	32,790	33,703	34,480	34,701	35,094	36,328	29,042	31,945	45,470	65,676
Global Average (ms)										
37,539										

Table 6 - Averaged time duration of the threshold update process

It was observed that, on average, a dynamics threshold update process takes half to 1 minute. This process involves a cross-validation technique, so this time value largely depends on the number of samples involved in the process. On the worst case, this process takes slightly more than a minute to complete. However, this value may also depend on the server load.

This is a good result, knowing that this procedure of threshold updating is only set to occur from time to time e.g. once a day, or a few times a day, for a given user, as a way of tuning the intruder detection classifier decision quality. This decision is on the estimated False Acceptance and False Rejection rates, for the given user.

Time Duration Comparison

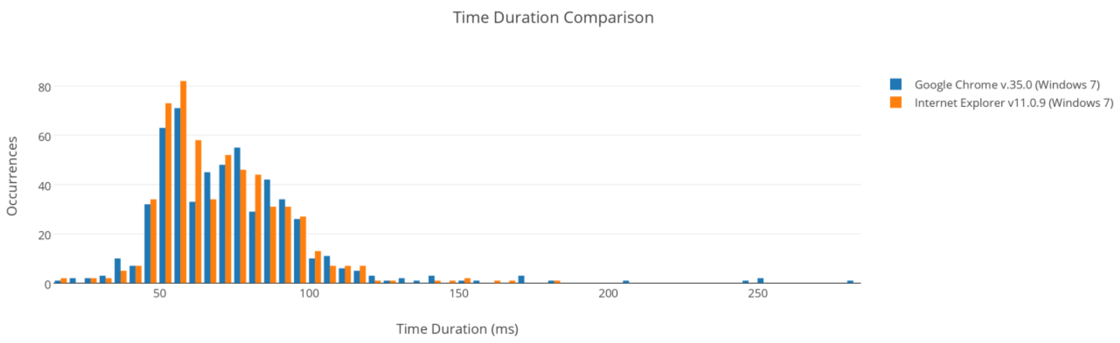


Figure 29 - Time Duration Comparison (Chrome vs Internet Explorer)

A time duration comparison was run in order to determine if the used of browser has a significant impact in the time duration values of the simpler features (dwells). This was done by comparing the time duration distribution of the dwells obtained from 2 different sets of 5 similar biometrics samples that resulted from the typing of the following sentence:

- "Let's type down some words in order to understand if the browser has some significant influence on the time duration accuracy".

The Internet Explorer v11.09 browser was used to produce a set of 5 similar biometrics samples, and the Google Chrome v.35.0 browser was used to produce the other set f 5 similar biometrics samples.

From this particular experience, it is not possible to take a clear and definitive conclusion about the influence of the browser on the time duration values of the features extracted. However, these two distributions have in fact a similar shape.

6.2.2. Accuracy Performance Results

Regarding the accuracy performance, the False Rejection and False Acceptance rates, and the Results Generalization of the prediction model, were the main performance metrics evaluated.

Artificial Experiment Results

Two sets of artificial attacks were made using cross-validation in order to calculate possible FAR and FRR for each user. A total of 679 samples were collected from the 12 persons involved. This gives an average of 56.6 biometrics samples per person.

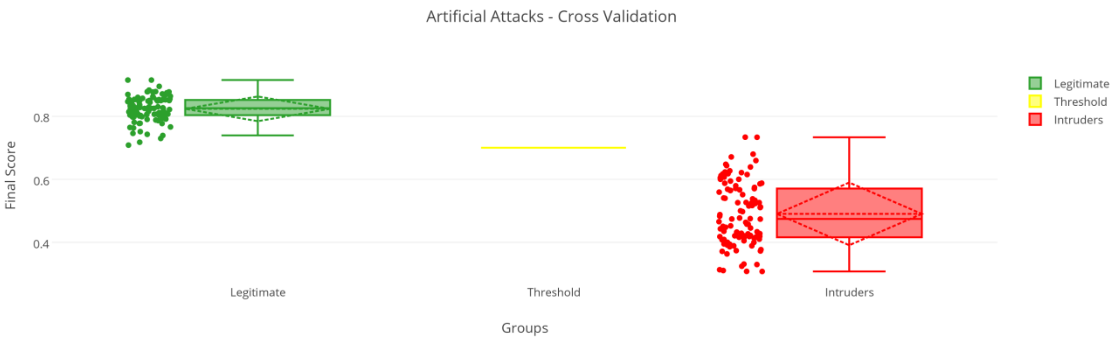


Figure 30 - Artificial Attacks - Internal Profile ID 5B… (Box plot)

The Figure 30 - Artificial Attacks - Internal Profile ID 5B… (Box plot) shows an example of a set of artificial attacks against the user biometrics profile identified by the 5B... internal profile ID. It is possible to observe, that only a few attacks end up resulting in False Acceptance or False Rejection Rates. The same artificial attacks showed on Figure 30 - Artificial Attacks - Internal Profile ID 5B… (Box plot) are also plotted on the Figure 31 - Artificial Attacks - Internal Profile ID 5B… (Scatter Plot)

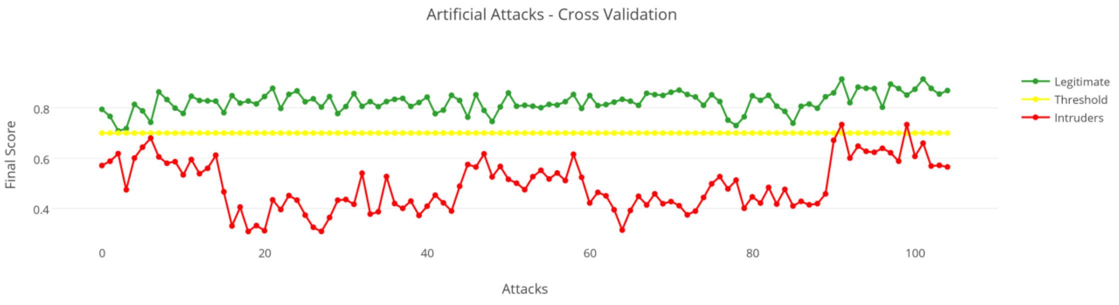


Figure 31 - Artificial Attacks - Internal Profile ID 5B… (Scatter Plot)

For this particular set of artificial attacks, there was used a total of 210 biometrics samples, 105 from the legitimate user profile, and 105 from intruders. At the time, the dynamic targeted threshold was set to 0.70, as showed in the Table 7 - Biometrics Samples used on the Artificial Attacks against the 5B… Biometrics Profile below:

Artificial Attacks - Example

Legitimate Samples	Intruder Samples
105	105

Threshold
0,700

Table 7 - Biometrics Samples used on the Artificial Attacks against the 5B… Biometrics Profile

Confusion Matrix

From these set of artificial attacks, a confusion matrix can be drawn to illustrate the prediction model accuracy performance – for this particular user.

True Acceptances	False Acceptances
105	2
False Rejections	**True Rejections**
0	103

Table 8 - Confusion Matrix resulting from the Artificial Attacks against the 5B… Biometrics Profile

From this analysis it is possible to obtain the corresponding False Acceptance and False Rejection Rates.

False Rejection Rate	False Acceptance Rate
0,000	0,019

Table 9 - FAR and FRR resulting from the Artificial Attacks against the 5B… Biometrics Profile

A similar analysis was made for all the people that participate in this experiment.

From all the artificial attacks performed, it is possible to present the correspondent FAR and FRR, as well as the global averaged values.

FRR and FAR Values

Table 10 - FRR and FAR resulting from each of the Artificial Attacks sets performed gives the average of FAR's and FRR's values, which are substantially low. However, these are the results of artificial attacks that were performed with not enough legitimate classified data.

A fact worth of mention is that the legitimate – and intruder – samples that were used, were already classified, that is, they were previously evaluated by the intrusion detection model, classified as legitimate and added to the respective biometrics template. However, the samples belonging to a legitimate user that were not classified as legitimate are usually discarded, hence, not added to the user biometrics template, so they are not used in the corresponding artificial attacks.

This is clear limitation of this technique that is reinforced by the fact that potentially legitimate data – that was discarded – is not being considered in these attacks. However it is hard to guarantee a 100% correctly classified data set to use in such dynamic and periodical artificial attacks. However, a fair point of this approach is that randomly chosen old samples can be used in this process, so the adaptability of the prediction model is put into test.

Targeted Internal Profile ID	False Rejection Rates	False Acceptance Rates
5B…	0,000	0,019
DA…	0,000	0,000

CD…	0,000	0,029
32…	0,000	0,000
27…	0,045	0,019
94…	0,063	0,029
29…	0,038	0,010
D8…	0,000	0,000
F1…	0,000	0,000
4ª…	0,000	0,010
17…	0,019	0,029
E0…	0,000	0,000
Total	**Average FRR**	**Average FAR**
12	0,014	0,012

Table 10 - FRR and FAR resulting from each of the Artificial Attacks sets performed

Another important advantage of this method is that it is an automated process, so it does not require a human controlled, time expensive, real-time experiment. The outputs of this technique are also already being used to update the targeted user dynamic decision threshold, so no additional processing is actually required.

As referred earlier, this artificial attack technique uses Cross-Validation to infer the FAR and FRR values, however, despite of being similar, this procedure is not the regular K-Fold Cross-Validation technique that is used to evaluate the perdition model, which uses only K samples of the legitimate user instead, and no intruder's samples are used. The resulting evaluation metrics gives only a prediction accuracy score for the targeted legitimate user.

Supervised Experiment Results

Another method of deducing the False Acceptance Rate and False Rejection rates is to analyse the actual evaluation results from the supervised experiments. In such experiments, it is important to guarantee that a user typing is in fact who he claims to be before the evaluation begins, and to guarantee that the premeditate attacks are in fact performed by actual intruders. This was supervised by observing the experiments, and by using an identification tool, in which the target and typist profiles were identified before the actual evaluation, in order to differentiate a regular evaluation from a premeditate attack

Due to some logistical limitations, only a few evaluations were recorded. It was not possible to group a significant number of people, and due to the fact that these experiments involve a lot of typing, is was not possible to collect a large amount of biometrics sample data. Still, a total of 438 biometrics samples from 5 distinct persons in this supervised. This gives an average of 87.6 biometrics sample per person.

Evaluation Results – Example

The Figure 32 - Evaluation Results for the targeted 5B… Biometrics Profile depicts the evaluation results of a given user. A total of 169 evaluations were made.

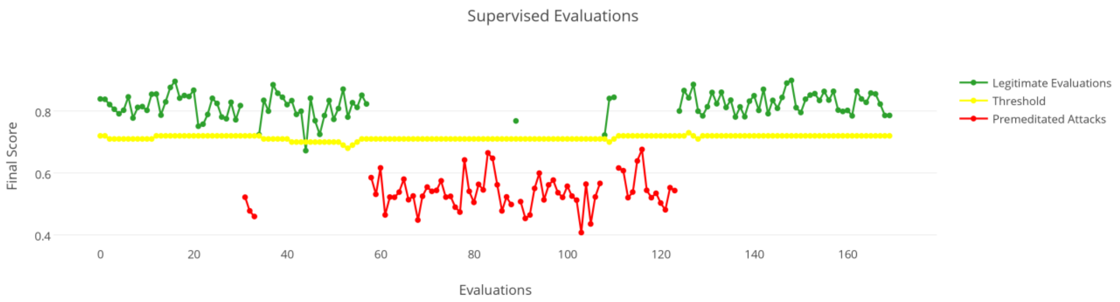


Figure 32 - Evaluation Results for the targeted 5B… Biometrics Profile

Legitimate Evaluations	Premeditated Attacks
104	65

Table 11 - Evalutions Performed for the target 5B... Biometrics Profile

Confusion Matrix

The following confusion matrix resulted from such evaluations:

True Acceptances	False Acceptances
103	0
False Rejections	**True Rejections**
1	65

Table 12 - Confusion Matrix resulting from the Supervised Evaluations against the 5B… Biometrics Profile

FRR and FAR

From this confusion matrix is possible to calculate the False Rejection and False Acceptance Rates.

False Rejection Rate	False Acceptance Rate
0.001	0,000

Table 13 - FAR and FRR resulting from the Supervised Evaluations against the 5B… Biometrics Profile

A similar analysis was conducted for all the people that participate in this experiment.

Global average FRR and FAR Values

From all the supervised evaluations, it is possible to present the correspondent FAR and FRR, as well as the global averaged values.

The results on Table 14 - FRR and FAR resulting from each of the Supervised Evaluations observed show a 1.88 % of False Rejections and 0.04 % of False Acceptances. These are really good results, however, there were not collected enough data to evaluated the classifier, so further analysis could be conducted in order to validate it with more confidence.

Nevertheless, the resulting accuracy performance metrics are really good indicators, and they compared with other experiments later on this chapter.

Targeted Internal Profile ID	False Rejection Rates	False Acceptance Rates
5B…	0,001	0,000
29…	0,033	0,021
D8…	0,017	0,000
17…	0,043	0,000
E0…	0,000	0,000
Total	**Average FRR**	**Average FAR**
5	0,018	0,004

Table 14 - FRR and FAR resulting from each of the Supervised Evaluations observed

Confusion Matrix - Global Average

From all the supervised evaluations, it is possible to present a global averaged confusion matrix, given in percentage values

True Acceptances	False Acceptances
98.12 %	0.05 %
False Rejections	**True Rejections**
1.88 %	99.5 %

Table 15 - Confusion Matrix Global Average resulting from all the Supervised Evaluations observed

FRR and FAR comparison

The European standard for access-control systems (EN-50133-1) specifies a false alarm rate of less than 1% for this type of solution [20]. Additionally, the current version of TypeWATCH for Desktop has an average FRR of about 1.54 % and a FAR average of 2.26 % [21]. Further comparisons can also be made with several experiments referenced in the literature [22], which shows a FRR and FAR values between 0 and 20 %.

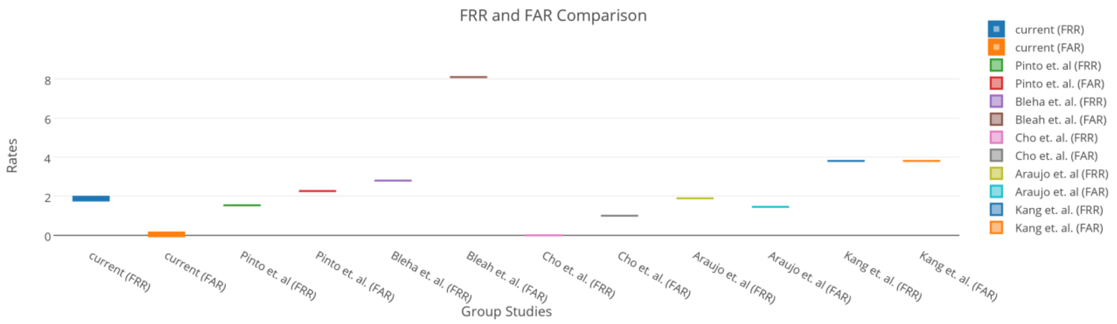


Figure 33 - FRR and FAR Comparison

The Figure 33 - FRR and FAR Comparison shows a comparison of FRR and FAR values between the results of the supervised experiments performed on this thesis, the results of the TypeWATCH desktop version, and a selection from results that were referenced in a paper by K. Killourhy and Roy Maxion [22].

Results Generalization – 12 users

For each user, a 16-Fold Cross-Validation was run in order to evaluate how the results of the Final Score resulting from the Absolute Score and Relative Score models will generalize to a given independent data set. This helps to estimate the accuracy of the prediction these models in practice. Please recall that the Final Score is defined between 0 and 1, and the higher the value, the better.

The results are illustrated as follows:

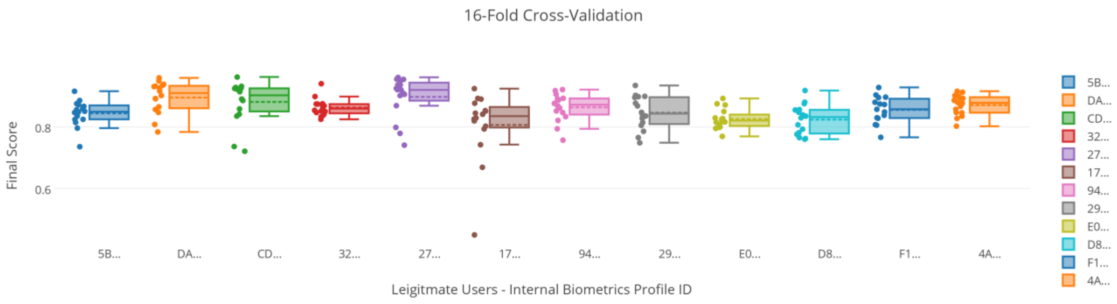


Figure 34 - Results Generalization – 12 Users

Internal Biometrics Profile ID	Averages	Standard Deviations
5B...	0.844	0.039
DA...	0.896	0.050
32...	0.902	0.067
27...	0.860	0.027
17...	0.835	0.110
94...	0.872	0.044
29...	0.843	0.052
E0...	0.820	0.031
D8	0.832	0.046
F1	0.857	0.042
4A	0.877	0.032
Average	0.857	0.044
Standard Deviation	0.025	0.022

Table 16 - Results Generalization – 12 Users

The results on Table 16 - Results Generalization – 12 Users

show that the prediction model used has on average performance score of 0.857, and an associated standard deviation of around 0.044. These are not excellent results; however, they seem to be good enough to yield excellent False Acceptance and False Rejection Rates, as seen earlier.

6.2.3. Feature Analysis

From the 16 K-Fold Cross Validation evaluations, it was possible to determine the most and least potentially relevant features for the given users.

The Table 17 - Most Relevant Features for the 5B… Biometrics Profile shows the top ten most potentially relevant features (sorted by Score Average and Occurrences frequency).

The Dwell of a "space" is the most relevant feature, which is understandable, as it is the character that separates the words in a sentence.

Ignoring the Dwell of a "space" feature for a moment, it can be observed the two most common features are the two Dwells related with the letter "e" and "t", respectively. There are studies that support the idea that the most common letters of the English language are these very same two letters, "e" in first place, and "t" in the second place [19]. The letter "e" is also the second most common letter on the Portuguese Language, being the letter "a" the most common [20]. The Portuguese and English languages were the ones used in these experiments.

Character	Type	Score Average	Occurrences	Prevalence Avg.
Space	Dwell	1	**347**	0,141742178
E	Dwell	1	**154**	0,061568791
T	Dwell	1	**132**	0,052468283
I	Dwell	1	**101**	0,042095166
R	Dwell	1	**62**	0,026732801
t_space	Digraph	1	**38**	0,018131347
Y	Dwell	1	**22**	0,018055075
C	Dwell	1	**18**	0,018962067
B	Dwell	1	**14**	0,016683475
e_a	Flight	1	**12**	0,018649788

Table 17 - Most Relevant Features for the 5B… Biometrics Profile

It is also noteworthy the fact the firsts non Dwells on the list are a Digraph related with the "t" letter and the "space" character, and a Flight related with the "e" and "a" letter, which were all referenced above.

Character	Type	Score Average	Occurrences	Prevalence Avg.
h_i	Digraph	0,333333333	**6**	0,011930642
V	Dwell	0,333333333	**6**	0,012641046
u_space	Digraph	0,333333333	**6**	0,011613314
t_h_e	Trigraph	0,333333333	**6**	0,012058965
h_i	Flight	0,333333333	**6**	0,011930642
c_e	Digraph	0	**2**	0,010869565
r_space	Digraph	0	**2**	0,010050251
c_e	Flight	0	**2**	0,010869565
l_e	Digraph	0	**2**	0,0125
o_n	Flight	0	**2**	0,0125

Table 18 - Least Relevant Features for the 5B… Biometrics Profile

The Table 18 - Least Relevant Features for the 5B… Biometrics Profile shows the top ten most potentially relevant features (sorted by Score Average and Occurrences frequency).

However, this analysis is not conclusive, as it must be also run in parallel a similar process, using also intruder biometrics data in order to perceive which features are the most and least relevant features exclusively for the target user e.g. features that are rank differently in two the parallel studies.

From the 16 K-Fold Cross Validation evaluations, it was also possible to determine the potential feature relevancy for each type of feature for the given users.

It can be observed from the Table 19 - Feature Relevance for the 5B… Biometrics Profile that the most relevant features are the Dwell features. This observation aligns with a study carried for the desktop version of the TypeWATCH developed by Watchful software [21].

Type	Score Average	Occurrences	Prevalence Avg.
Dwell	**0,949066**	**1586**	0,646142637
Digraph	**0,794487**	**492**	0,195799949
Flight	**0,775209**	**346**	0,137373925
Trigraph	**0,551111**	**52**	0,020683489

Table 19 - Feature Relevance for the 5B… Biometrics Profile

Chapter 7
Conclusions

In this chapter, the objectives proposed on this thesis are restated by balancing goals and contributions, the problems encountered and the solutions proposed, and by stressing some important findings and observations. In addition, the future work is exposed by listing pending tasks, new thoughts or ideas, and improvement suggestions.

7.1. Retrospective

The main goal of this thesis was to come up with a proof of concept that states behavioural biometrics analysis as an effective user authentication technique for web-based applications.

In more detail, this translated into the development of a web-based intrusion detection service that is built around keystroke dynamics.

The inherent objectives proposed in this thesis were actually achieved, as the application created showed to fulfil the both the time and accuracy performance requirements. It is also an application that has a quick and dynamic enrolment phase, and responds to potential intrusions in near-real time.

This intrusion detection service is also of easy integration with existing web-sites, requiring no additional hardware or software installation. Its use is transparent to the end-user, being non-obtrusive, as it works just by the mere consequence of typing, thus, notably enhanced the security of users online.

In fact, it was showed that these measurable traits of the human behaviour, particularly the ones resulting from the habit of typing on a keyboard, are actuality valuable data that can be successfully used as an additional layer of protection to the existing security mechanisms employed by web applications, in order to give logical access control to the legitimate users.

Even though it a great potential, keystroke dynamics are yet to be fully explored in real world scenarios. As a matter of fact, there are only a few products on the market that use keystroke dynamics for user authentication[1], despite of all of the optimistic remarks that have been presented by the numerous research studies carried out on this field over the last two decades.

Similarly, this thesis supports the idea that the intrusion detection services of the future, can, in fact, consider keystroke dynamics analysis as an effective model for user identity validation, particularly on the internet, targeting web sites that involve the typing of free-text.

However, the provision of a generic service of this kind – in a web environment – may involve a cautious and well thought out system architecture design, especially if it is to be delivered in a SaaS model.

In this scenario, the most suitable architecture design is a client-server architecture that requires an interoperable client application that performs instant and asynchronous i.e. non-blocking communications between the client and the server side. This enables the evaluation

[1] Most of them are actually targeted for one-step verification processes only e.g. password hardening

of biometrics data in near-real time – providing a continuous and instant protection – and without refreshing the web page – thus, avoiding the loss of in-memory data.

A technical difficulty that may arise from this architecture design relates with the same-origin policy that is applied by web-browsers. As communications are to be made directly from the web page to the intrusion detection server[2], this technical difficulty occurs because the reading of data that comes from cross-domain HTTP requests response is blocked by the browser when the communication is done using the standard XMLHttpRequest API. The appropriate solution to deal with this situation is the use of CORS (Cross-Origin-Resource Sharing), which makes it possible to control, on the server side, which cross-domain requests are allowed to be performed.

In addition, the client-server architecture model is actually a requirement, because the evaluation process needs to be run in a separated tier from the web client application, as a way of protecting it from local browser access. There are other important reasons to support this, for instance, it enables a centralized control of processes and related data, and makes it possible to take advantage of a multitenant architecture, which enables the application of evaluation models that use intruder biometrics data to improve the model performance accuracy.

Typically, the prediction models applied by statistical classifiers – which are the base of this intrusion detection system – and other classification systems based on machine learning, use the False Rejection and False Acceptance Rates as reference indicators to ultimately measure their performance accuracy. However, it is also important to measure the results generalization ability of the prediction model, and the time performance related indicators.

In order to evaluate the intrusion detection developed for this thesis, a validation phase was conducted. The time performance and the performance accuracy of the system, the generalization ability of the prediction model, and the observable impact on the responsiveness - and overall usability – of the target web page were the proposed success indicators.

Despite of the low amount of the data collect – due to some logistical limitations – it was possible to obtain fair success indicators.

Regarding the time performance, this intrusion detection system show excellent indicators, as it takes only around 10 milliseconds to extract the features, and only around 1 second to return an actual evaluation. The time needed to run a user biometrics template threshold update procedure is about half to one minute per user, which is actually not bad, as this procedure does not need to be run on real time, so it can just be run occasionally, as a way of improving the classification model for that given user.

Regarding the performance accuracy, from the supervised experiments – using 5 persons – it can be observed that the average False Rejection Rate is of 0.04 % and the False Acceptance Rate is of 1.88 %. The results from other validation procedure – by means of artificial attacks, using a total 12 persons – indicates an even lower False Rejection Rate, which is of 0.014 %, and equally an even lower False Acceptance Rate of exactly 0.0 %. However, these approaches suffer from some limitations, as both don't use enough data to make actual assumptions – particularly the supervised evaluation approach.

[2] Without passing through the server of the origin domain, for service deployment and delivery reasons

Despite of these limitations, it is still possible to estimate how the prediction model in will generalize its evaluation results to upcoming biometrics samples. For this, a procedure based on K-Fold Cross-Validation for the 12 persons was used, which returned a decent and promising averaged normalized score of 0.857.

Recall that these experiments were made in the Facebook web-page by a small set of people who know how to – and actually – use the site on a near-daily basis. The computers used were most of the time always the same for the same people, and the activities performed were mostly related to free-texting, messaging, and chatting, using both the English and Portuguese languages. Therefore, several and intensive different test set on different environments sets must be made in order to perceive the actual performance of this intrusion detection system.

Nevertheless, all of the people that were involved were please and fascinated by the performance of this intrusion detected system. What is even more interesting is that the people who were not familiar with the concept find it to be really exotic, kind of magical and revolutionary, being really overwhelmed by this technology.

This enforces the idea that such a promising solution has its space on the market, and it is surprising how keystroke dynamics is still such an unknown concept for the general public.

7.2. Future Work

This thesis instated a connection between quite a few science fields and different areas of knowledge, laying a solid foundation for the application of behavioural biometrics in the World Wide Web. However, there are still many loose ends to take care of.

First off, there is still the need of running more intensive validation experiments, perhaps using a larger group of people, using different environments, different keyboards, and different web browsers. From the top of this, a extend analysis of biometrics features could be made in order to determine which features are more relevant to which users. This would allow the improvement of the current classification model, by assigning dynamic and optimal weights to such biometrics features. Additionally, a similar study for key combinations is also on the future plans.

Still in regard to the current classification model, it would be interesting to apply a different dynamic enrolment policy, as currently, the model simply uses the k most recent user biometrics samples in order to evaluate a new biometrics sample – being k a configurable and predefined business logic parameter. A different dynamics enrolment policy would provide a higher adaptability in the case of abrupt user legitimate behavioural deviations. The application of a more sophisticated dynamic threshold update policy would also be interesting as well.

Another possibility for this intrusion detection system is the use of an alternative classification model, perhaps one based on a probabilistic classifier e.g. a Naive Bayes classifier, instead of a statistical one. To accomplish this, this would roughly only require a discretization of the biometrics data, in order to construct the actual probabilities on a training phase, using both intruder and legitimated classified data. It is important to note however, that the current model already uses intruder and legitimated classified data in order to update the dynamics decision threshold for each user.

The current intrusion detection system was already designed to support the use of different templates for each user profile, and similarly, the assignment of different profiles on the same web site to a given user. This would enable the possibility of assigning a different biometrics template according to some environment information – e.g. computer in used, type of the target web-site, or type of keyboard, and also to enable the use of share biometrics profile across different web-sites.

These last features are all part of configuration, monitoring and support tools i.e. the end-user and customer administration dashboards – that were studied and designed in the both the requirements and design phase, however, there were not set to be implement, yet. This is then a task for future work.

An important goal to future is to try to decrease the amount of characters needed to perform a biometrics sample evaluation, without decreasing significantly the overall system performance accuracy. Currently, this amount is set to 125 characters, which in some use cases might be too much.

There also future plans of using also mouse and pointer in conjunction in keystroke dynamics for the same intrusion detection purpose.

As a personal feature request, it would be really interesting for this intrusion detection system to support the detection of the language that is actually being type. This would be based on statistical analysis as well.

This thesis asserts that security of users online can actually be ensured just by the mere consequence of typing. However – and fortunately – there is still a lot of room for further improvements.

References

[1] F. Bergadano, D. Gunetti e C. Picardi, "User Authentication through Keystroke Dynamics," em *ACM Transactions on Information and System Security, Vol. 5, No*, Torino, Italy, ACM, 2002, p. 367–397.

[2] A. Jain, A. Ross e S. Prabhakar, "An Introduction to Biometric Recognition," Circuits and Systems for Video Technology, IEEE Transactions on, 2004.

[3] National Science and Technology Council (NSTC) Subcommitee on Biometrics, "Biometrics History," 7 August 2006. [Online]. Available: http://www.biometrics.gov/documents/biohistory.pdf. [Acedido em 15 November 2013].

[4] F. Monrose e A. D. Rubin, "Keystroke dynamics as a biometric for authentication," Elsevier Science B.V., 2000.

[5] A. Klokova, "Comparison of Various Biometrics Methods," Electronics and Computer Science, University of Southampton, Southampton, United Kingdom, 2010.

[6] H. Kang, Y. Han, H. Kim, W. Choi e Y. Chung, "An Empirical Study of Multi-mode Biometric Systems Using Face and Fingerprint," em *Information Security Applications*, Springer Berlin Heidelberg, 2004, pp. 348-354.

[7] D. Gunetti e C. Picardi, "Keystroke Analysis of Free Text," ACM, New York, USA, 2005.

[8] J.-D. Masters, "Keystroke Dynamics as a Biometric," University of Southampton, Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science, 2009.

[9] Griaule Biometrics, Understanding Biometrics, 2008.

[10] Padma e Manivannan, Comparative and Analysis of Biometric Systems, International Journal on Computer Science and Engineering, 2011.

[11] V. S. Valencia, "Biometric Covariate Analysis using Partial Area Under Curve," University of Arizona, 2010.

[12] R. Moskovitch, "Identity Theft, Computers and Behavioral Biometrics," Intelligence and Security Informatics, 2009. ISI '09. IEEE International Conference on, Dallas, TX, USA, 2009.

[13] W3C. Retrieved , "HTML 5 A vocabulary and associated APIs for HTML and XHTML, W3C Working Draft 12 February 2009".," 12 February 2009. [Online]. Available: http://dev.w3.org/html5/spec/Overview.html#relationship-to-flash-silverlight-xul-and-similar-proprietary-languages. [Acedido em 17 February 2009].

[14] F. C. L. S. L. H. R. L. M. Araujo, "User Authentication Through Typing Biometrics Features," em *IEEE Transactions on Signal Processing*, IEEE Transactions, 2005, pp. 851-

855.

[15] Business Wire, "Market for Keystroke & Typing Dynamics Canvassed by GIA in Insightful Report Now Available at MarketPublishers.com," 27 May 2013. [Online]. Available: http://www.businesswire.com/news/home/20130527005121/en/Market-Keystroke-Typing-Dynamics-Canvassed-GIA-Insightful. [Acedido em 27 November 2013].

[16] JVANEYCK, "Cross Domain Requests in JavaScript," 7 1 2014. [Online]. Available: http://jvaneyck.wordpress.com/2014/01/07/cross-domain-requests-in-javascript/. [Acedido em 2014].

[17] Mozilla Developer Network, "HTTP access control (CORS)," 18 4 2014. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS. [Acedido em 2014].

Annex

[1] Internship Planning – Behavioural Biometrics in the World Wide Web

[2] State of the Art – Behavioural Biometrics in the World Wide Web

[3] Requirements Analysis – Behavioural Biometrics in the World Wide Web

[4] Study and Selection of Technologies – Behavioural Biometrics in the World Wide Web

[5] Design and Architecture - Behavioural Biometrics in the Wold Wide Web

[6] Development - Behavioural Biometrics in the Wold Wide Web

[7] Validation Results - Behavioural Biometrics in the Wold Wide Web