**TecnoLógicas**
ISSN-p 0123-7799
ISSN-e 2256-5337
Vol. 23, No. 47, pp. 197-211
Enero-abril de 2020

Artículo de investigación/Research Article

TecnoLógicas

# Identity Verification in Virtual Education Using Biometric Analysis Based on Keystroke Dynamics

## Verificación de identidad en la educación virtual mediante análisis biométrico basado en la dinámica del tecleo

Daniel. Escobar-Grisales [1],
Juan. C. Vásquez-Correa [2],
Jesús F. Vargas-Bonilla [3] y
Juan Rafael Orozco-Arroyave [4]

Recibido: 13 de septiembre de 2019
Aceptado: 05 de diciembre de 2019

Cómo citar / How to cite
D. Escobar-Grisales, J. C. Vásquez-Correa, J. F. Vargas-Bonilla, J. R. Orozco-Arroyave, "Identity verification in virtual education using biometric analysis based on keystroke dynamics", *TecnoLógicas*, vol. 23, no. 47, pp. 197-211, 2020. https://doi.org/10.22430/22565337.1475



[1] MSc. in Electronics and Telecomunications Engineering, Faculty of Engineering. Universidad de Antioquia, Medellín-Colombia, daniel.esobar@udea.edu.co
[2] MSc. in Telecommunications engineering, Faculty of Engineering, Universidad de Antioquia; Pattern, Recognition Lab. Friedrih-Alexander-Universität, Erlangen, Nürnberg- Germany, jcamilo.vasquez@udea.edu.co
[3] PhD. in Cibernetcs and Telecommunications· Faculty of Engineering. Universidad de Antioquia, Medellín-Colombia, jesus.vargas@udea.edu.co
[4] PhD. in Computer Science, Faculty of Engineering, Universidad de Antioquia, Pattern Recognition Lab. Friedrih-Alexander-Universität, Erlangen, Nürnberg- Germany, rafael.orozco@udea.edu.co

**Abstract**

Virtual education has become one of the tools most widely used by students at all educational levels, not just because of its convenience and flexibility, but also because it can expand educational coverage. All these benefits also bring along multiple issues in terms of security and reliability in the evaluation the of student's knowledge because traditional identity verification strategies, such as the combination of username and password, do not guarantee that the student enrolled in the course really takes the exam. Therefore, a system with a different type of verification strategy should be designed to differentiate valid users from impostors. This study proposes a new verification system based on distances computed among Gaussian Mixture Models created with different writing task. The proposed approach is evaluated in two different modalities namely intrusive verification and non-intrusive verification. The intrusive mode provides a false positive rate of around 16 %, while the non-intrusive mode provides a false positive rate of 12 % In addition, the proposed strategy for non-intrusive verification is compared to a work previously reported in the literature and the results show that our approach reduces the equal error rate in about 24.3 %. The implemented strategy does not need additional hardware; only the computer keyboard is required to complete the user verification, which makes the system attractive, flexible, and practical for virtual education platforms.

**Keywords**

Biometrics, Identity verification, Keystroke dynamics, Virtual Education.

**Resumen**

La educación virtual se ha convertido en una de las herramientas más utilizadas por los estudiantes en todos los niveles educativos, no solo por la comodidad y la flexibilidad, sino también por la posibilidad de ampliar la cobertura educativa en una población. Todos estos beneficios traen consigo múltiples problemas de seguridad y confiabilidad a la hora de evaluar el proceso de aprendizaje del estudiante, ya que las estrategias tradicionales de verificación de identidad, como la combinación de nombre de usuario y contraseña, no garantizan que el estudiante matriculado en el curso realmente realice el examen. Por lo tanto, es necesario diseñar un sistema con otro tipo de estrategia de verificación para diferenciar un usuario válido de un impostor. Este estudio propone un nuevo método de verificación, basado en el cálculo de distancias entre los modelos de mezclas gaussianas creados con diferentes tareas de escritura. El enfoque propuesto es evaluado en dos modalidades diferentes llamadas verificación intrusiva y verificación no intrusiva. El modo intrusivo proporciona una tasa de falsos positivos de 16 %, mientras el modo no intrusivo provee una tasa de falsos positivos de 12 %. Además, la estrategia propuesta para verificación no intrusiva es comparada con un trabajo previamente reportado en la literatura y los resultados muestran que nuestro enfoque reduce la tasa de error en aproximadamente un 24.3 %. La estrategia implementada no necesita hardware adicional, solo es requerido el teclado del computador para realizar la verificación, lo que hace que el sistema sea atractivo y flexible para ser usado en plataformas de educación virtual.

**Palabras clave**

Biometría, dinámica de tecleo, educación virtual, verificación de identidad.

## 1. INTRODUCTION

Virtual Education (VE) offers multiple benefits, not only because of its convenience and flexibility for students and teachers, but also because it can improve educational coverage, especially in remote areas with limited access to resources. Nevertheless, the quality of virtual education is controversial although, according to the U.S. Department of Education [1], online students achieve a better performance than those who take face-to-face classes. Furthermore, online students tend to be self-motivated, self-disciplined, and self-directed, which makes VE a very popular modality nowadays.

The freedom students experience in VE also produces security and reliability issues, especially when giving tests and exams According to Bretag [2], fraud in VE is higher and more worrying than in traditional education. For instance, 95 % of the students in Israel and 69 % in Korea admitted to committing fraud in virtual exams or tests and the trend is similar in the rest of the world [2]. For this reason, virtual tests are not used in evaluations such as admission exams or final tests by universities.

In general terms, biometric systems can be classified into two approaches: verification and identification [3]. In identification, the biometric features of a user are compared to multiple users in a database in order to find the identity of the user among all the individuals. In verification, a previously registered user logs-in to the system and the biometric features of the user are compared with the biometric features of the register.

Depending the similarity of the features, the system may decide whether the user is valid or not.

Keystroke Dynamics (KD) analysis is a very good option to capture biometric information to control who has access to certain information or platforms. One of the main advantages of KD is that it does not require the use of additional hardware, i.e., the identity of a user can be verified with a regular keyboard computer.

KD analysis started in the 20th century when telegraph operators had to transmit dozens of words in a short period of time, developing a distinctive rhythm that was captured by the operators on the other side of the line to identify who was transmitting [4]. Later, in 1990, Joyce and Grupta [5] extracted specific digital signatures to identify users based on their KD. The authors asked users to type their username and password 8 times to compute a curve with the average time they took to enter the data. At a later login, the system compared the average curve with the new curve generated in the new login. Then, the system detected whether the user was valid or an impostor based on a measure of similarity between the two curves. The system was evaluated with 30 valid users and 27 impostors. As a result, there was a total of 30 valid access attempts and 810 intruder access attempts. The authors reported a False Positive Rate (FPR) of 0.25 % and a False Negative Rate (FNR) of 16.0 %. The system had several usability issues since the user was requested to type the data correctly.

The system was biased by cases where a user deleted wrong characters. A similar strategy was proposed in [6] to identify 173 users based on their KD. The users attended a programming course at the Helsinki University, and the data were extracted based on their programming exercises. The authors created a student profile based on the average hold time when pressing any key, the average hold time when pressing a particular key, the average time when pressing two particular keys, and a combination of the three previous times. The similarity between the evaluation sample and the database was measured with the Euclidean distance.

The authors reported accuracies of up to 97 %. In [7], the authors proposed a model to verify user identity with features extracted when users typed a password on a smartphone. The authors asked 94 different users to type the password ".tieRoanl" in order to extract features such as pressure

when touching the screen, coordinates of the pressing point, and times when the finger presses or releases the screen. The authors computed several statistical functionals from the keystrokes and obtained a set of 155 features. The most important features were selected based on a minimum Redundancy Maximum Relevance (mRMR) algorithm. The selected features included pressure and coordinates. The authors reported an accuracy of 97.4 % using a Support Vector Machine (SVM) classifier. In recent years, identity verification based on KD has captured the attention of the research community. For instance, a keystroke dynamic application was presented in [8]. In the study, the authors created a keyprint (typing fingerprints) to authenticate users in online courses. The aim of a keyprint is to capture few data with specific characteristics of a user's KD; therefore, only data with unusual values of typing dynamics are considered. The authors claim that this system is suitable for verification but not for identification. They also showed that two samples from the same user are very unlikely to be exactly the same; therefore, to determine the similarity between the samples, a $t$-test ($\alpha = 0.05$) is enough. The decision is made based on the equal error rate (EER), i.e., where FPR and FNR are the same. The authors reported an accuracy of 80 %, but the main drawback of the approach was that users needed to type least 964 characters to be correctly identified.

A strategy to authenticate a user identity based on KD is proposed in [9]. Where the identity of the users is verified by comparing enrollment and log-in information. 63 users where asked to type 5 items of personal information: name, last name, email address, nationality, and national ID. The database comprised 12 genuine accesses and 12 impostor access per user to enroll, for a total of 7560 samples. Six genuine samples were used to register the user; and the rest, to log in. The authors tested different features and, but the best result was obtained using time between key press and release and the difference between the time of pressing a key and releasing the following one. With these features and a classifier based on the Modified Scaled Manhattan distance, they obtained an EER of 2.4 %. In [9], this result was achieved because the identity of users was verified using KD when they typed data such as name, email address, and other information. As users are similar with these data, the KD will probably not vary from one sample to the next, which allows systems to verify users' identify in a more accurate way.

### 1.1 Contribution of this study

This paper proposes a methodology to verify the identity of students based on their KD. The proposed approach is tested on two different modalities: intrusive and non-intrusive. The first mode considers the case when the subject is aware of being tested, and the second mode considers the case where the subject is not aware of the verification process, and then a different writing task is required to verify the identity of the subject. The features extracted from the writing tasks are used to create Gaussian Mixture Models (GMM). Those models are compared using probabilistic distances to make the decision whether a user is valid or not. The main difference of the proposed method with respect to others reported in the literature is that our approach is based on probabilistic models instead of the direct comparison of feature sets. The results indicate that it is possible to detect intruders with accuracies of up to 89 %, measured in the EER.

## 2. MATERIALS AND METHODS

### 2.1 Participants

A total of 170 subjects (116 male) participated in this study. The average age was 24 years old. The subjects were asked to perform 5 different tasks which were

designed to capture the KD over different regions of the keyboard. Most users were undergraduate students from the University of Antioquia. Users with higher education attainment were also considered.

In addition, 20 of the 170 users performed different tasks in two different sessions. Table 1 details participants' information.

Table 1. Demographic information of the participants in this study
Source: Created by the authors.

| | Male | Female |
|---|---|---|
| Subjects | 116 | 54 |
| Age ($\mu \pm \sigma$) | 23 ± 5.8 | 24 ± 7.1 |
| Students | 102 | 44 |
| Graduate | 6 | 7 |
| MSc | 4 | - |
| PhD | 4 | 3 |

## 2.2 Data collection

Each user of the database performed 5 different tasks the first 4 tasks were designed to capture specific movements on the keyboard. For instance, task 1 captures long horizontal displacements. In this task, the user typed the sentence "*El sapo de mi casa come queso, zapallo y xoubas*". Here, the characters of each word follow other characters on the opposite side of the keyboard; thus, it is possible to define the user's dynamics while moving from one side to the other. Fig. 1a shows the keyboard regions involved in this task. The arrow indicates displacements between the two regions. Similarly, task 2, "*En un pueblo un niño juega afuera y tu vejez es notable*", aims to capture short displacements along the horizontal axis. These displacements are shown in Fig. 1b. Task 3, "*La leña esta partida, la tijera se ha roto, yo quiero jugar y reír, dale a la gata sus gatitos y las fresas y las patatas del huerto*", connects characters in the middle row of the keyboard with some in the top row, defining top vertical displacements. This task is shown in Fig. 1c.

Finally, task 4, "*La vaca flaca, las lañas malvas, las jacas blancas, a la sal acabas la salsa, zancada flaca*", requires the user to connect characters in the middle row with characters in the bottom row, defining the lower vertical displacements, as shown in Fig. 1d. To define users' KD in normal conditions, we considered the task 5, which has a total of 500 characters. This task was extracted from the novel *Frankenstein or the modern Prometheus* by Mary Shelly [10].

Table 2 details the size of each task.

Table 2. Length of each task
Source: Created by the authors.

| Tasks | Character count | Word count |
|---|---|---|
| 1 | 54 | 11 |
| 2 | 56 | 12 |
| 3 | 133 | 28 |
| 4 | 92 | 17 |
| 5 | 518 | 90 |

## 2.3 Methods

A user registers in the platform by typing the first 4 tasks previously described in the Fig 1. When the user types, the system returns data from the KD. The user-model is created with the KD data. When the user logs into the platform, s/he should type one of the 5 tasks following a procedure similar to the one completed during registration stage. A login model is created per user and compared to the model created during registration stage. Finally, if the distance between these two models is short, the user is classified as valid; otherwise, the user is classified as an intruder. The general methodology is summarized in Fig. 2. The next subsections detail the methods applied at each stage of the methodology.

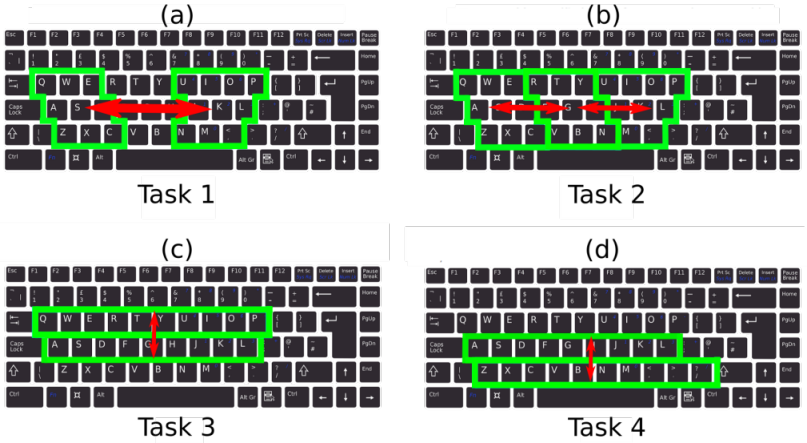


Fig. 1. (a) Task 1, long displacement on the horizontal axis; (b) Task 2, short displacement on the horizontal axis; (c) Task, 3 top displacement on the vertical axis; and (d) Task 4, lower displacement on the vertical axis. Source: Created by the authors.

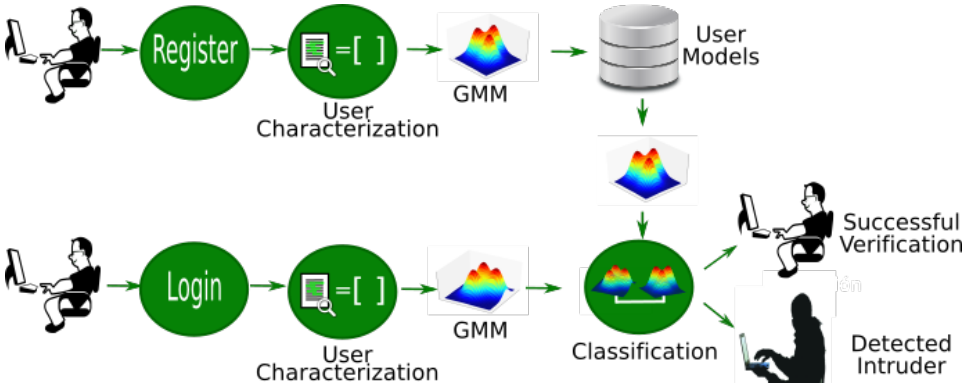


Fig. 2. General methodology implemented in this study. The upper part of the methodology shows the registration stage; and the lower part, the log-in stage. Source: Created by the authors.

### 2.4 Raw information extracted from the computer to model KD

Computers can provide the ASCII code of the characters that are typed when a text is written. They can also store the time the keys were pressed (P) and released (R). Table 3 shows an example with the raw information that can be extracted when the word "Hola" is typed.

### 2.5 User Characterization

The objective of this stage is to find a feature matrix $X \in \mathbb{R}^{n,k}$ associated to each user. $n$ refers to the number of segments, and $k$ is the number of extracted features. Fig. 3. Describes the feature matrix $X$. Note that each task might have a different number of segments (rows), but the number of features is fixed. The characterization process is divided into two parts: segmentation and feature extraction.

#### *2.5.1. Segmentation*

Each row in $X$ refers to a specific segment of the text that the user has typed. These segments were based on a tri-graph model, which consists of small packets with the information of three consecutive characters. A similar strategy was considered in another study for identity verification based on speech signals [11].

For our analysis a sliding window of 5 tri-graphs, with an overlap of 3 tri-graphs, was used, as shown in Fig. 4.

### *2.5.2 Feature extraction*

A total of six-time series are created when the user types each character: three when the key is pressed and three when the key is released. These times are shown in Table 3. With this information, it is possible to extract 2 main features: **Hold time**, which is the time between press and release of a key; and **Flight time**, which is the time between pressing a key and pressing the next one, as described in Fig 5. The thirteen features that are extracted per segment are described below:

- Total Hold Time ($T_{HT}$): the sum of the hold times of the characters.
- Average Hold Time ($A_{HT}$): the sum of the hold times of the characters divided by the number of characters.

Standard Deviation of the Hold Time ($\sigma_{HT}$): the deviation of the Hold times with respect to $A_{HT}$.

Table 3. Example of data captured by the platform. Word= "hola", p: press, r: release
Source: Created by the authors.

| Key | Code | Operation | Time(ms) |
|---|---|---|---|
| H | 72 | P | 3301 |
| O | 111 | P | 3524 |
| H | 72 | R | 3556 |
| O | 111 | R | 3612 |
| L | 108 | P | 3644 |
| L | 108 | R | 3692 |
| A | 97 | P | 3716 |
| A | 97 | R | 3820 |

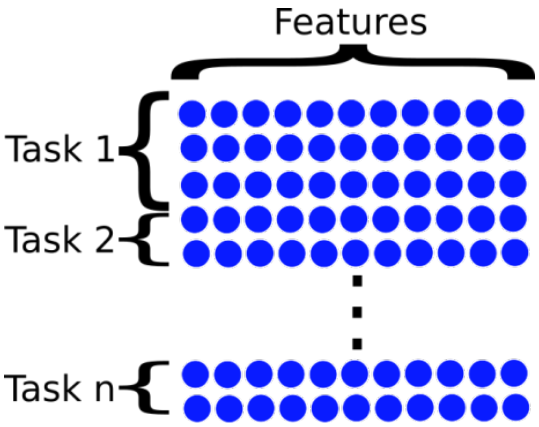


Fig. 3. Feature matrix $X$ of each user. Source: Created by the authors.

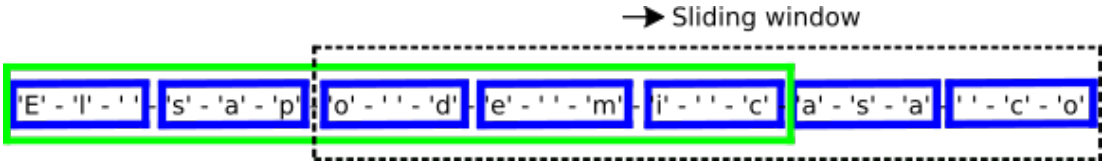


Fig. 4. Segmentation of the sentence: "*El sapo de mi casa co*". Trigraphs (blue), segment1 (green), and segment 2 (dotted line). Source: Created by the authors.

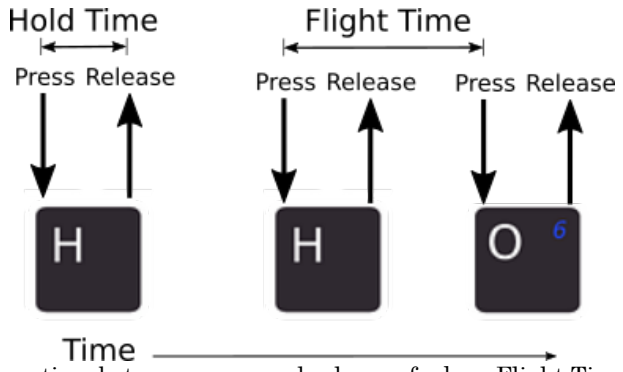


Fig. 5. Hold Time: time between press and release of a key. Flight Time: time between pressing a key and pressing the next one. Source: Created by the authors.

- Strong Key ($S_K$): the code of the key, with shorter hold time.
- Time Strong Key ($T_{SK}$): the minimum hold time.
- Weak Key ($W_K$): the code of the key, with longer hold time.
- Time Weak Key ($T_{WK}$): the maximum hold time.
- Total Flight Time ($T_{FT}$): the sum of flight times of the characters.
- Average Flight Time ($A_{FT}$): the sum of the Flight times of the characters divided by the number of characters.
- Standard Deviation of the Flight Time ($\sigma_{FT}$): the deviation of the Flight times with respect to $A_{FT}$.
- Strong Key in Flight ($S_{KF}$): the code of the key, with shorter flight time.
- Time Strong Key in Flight ($T_{SKF}$): the minimum flight time
- Weak Key in Flight ($W_{KF}$): the code of the key, with longer flight time.
- Time Weak Key in Flight ($T_{WKF}$): the maximum flight time.

Once the feature matrix has been created per user, it is necessary to find a representation to model the distribution of the features. The models created in the registration stage are compared with those created in the log-in stage. We considered Gaussian Mixture Models (GMM) to create those models and the Bhattacharyya distance to compare them, as explained below.

### 2.6 Gaussian Mixture Model

A GMM is a probabilistic model created to represent a population from a linear combination of Gaussian distributions.

Each Gaussian of the GMM models a specific group of samples in a population [12-13]. Equation (1) shows the mathematical expression for a GMM of a multivariate random variable $\boldsymbol{x}$, which corresponds to the sum of $M$ Gaussian distributions, weighted by a parameter $C_m$.

A compact way to represent GMM models is indicated in (2).

Three parameters should be estimated in the GMM modeling approach: weight $C_m$, mean vector $\boldsymbol{\mu_m}$, and covariance matrix $\boldsymbol{\Sigma_m}$. $m$ is the index for the Gaussians. These parameters are estimated using the Expectation–Maximization (EM) algorithm. The total number of Gaussians $M$ must be defined before starting the estimation procedure, and it can be done according to the Bayesian Information Criterion (BIC) [14], which measures the quantity of information lost when the model is used.

However, in case of problems where there is no prior knowledge of the data, the number of Gaussian distributions is found experimentally [11].

$$f(x) = \sum_{m=1}^{M} \frac{C_m}{(2\pi)^{D/2}|\sum_{\boldsymbol{m}}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_m})^T \sum_{m}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_m})\right] \quad (1)$$

$$f(x) = \sum_{m=1}^{M} C_m \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu_m}, \sum_{\boldsymbol{m}}) \quad (2)$$

### 2.7 Classification

Each user is represented by a GMM.

Thus, to calculate the similarity between two models (registration: $f_i(x)$ and log-in: $g_i(x)$), we can use the Bhattacharyya distance ($D_{bha}$), where $\boldsymbol{\mu_m}$ and the $\boldsymbol{\Sigma_m}$ of each GMM are taken into account [15].

$D_{Bha}$ can be expressed as in (3):

$$D_{Bha} = \mu_{Bha} + \sum_m \quad (3)$$

where the first term considers the mean vectors of the GMMs, and the second term is the covariance matrix. As indicated in Equation (3), the similarity measurement between the two models, $f_i(x)$ and $g_i(x)$, considers the mean vectors and the covariance matrix separately. Mean vectors are compared in (4), while the covariance matrix is considered in (5).

Finally, depending on the similarity of both models, it is possible to classify the user's identity. If the user is valid, the distance between the two models is expected to be less than the distance resulting from an impostor. However, it is necessary to define a threshold $U$ to decide whether a user is valid or an impostor.

This distance measurement has been considered in previous studies where GMM models resulting from speech recordings are compared [16].

Figures 6 and 7 show the flowchart of the registration and login stages, respectively. The number of components $M$ and the decision threshold $U$ are found in the training and development stage explained below.

## 3. EXPERIMENTS AND RESULTS

The test stage aims to evaluate the performance and usability of the system in two different modes: intrusive and non-intrusive verification. In the intrusive mode, the user is aware that his/her identity is being verified through the keyboard. On the other hand, in the non-intrusive mode the user does not know that is being verified.

### 3.1 Experiment 1: intrusive mode

In the intrusive mode, two sessions are required because the registration and log-in writing tasks are the same, then we use the first session to register and the second to log-in the user. In this case only 20 of the 170 users have two sessions; therefore, this experiment was conducted with 20 users.

$$\mu_{Bha} = \frac{1}{8}\sum_{i=1}^{M}\left\{(\boldsymbol{\mu_{fi}}-\boldsymbol{\mu_{gi}})^T\left[\frac{\sum_{\boldsymbol{fi}}+\sum_{\boldsymbol{gi}}}{2}\right]^{-1}(\boldsymbol{\mu_{fi}}-\boldsymbol{\mu_{gi}})\right\} \quad (4)$$

$$\sum_{Bha} = \frac{1}{2}\sum_{i=1}^{M}\left[ln\frac{\frac{\sum_{\boldsymbol{fi}}+\sum_{\boldsymbol{gi}}}{2}}{\sqrt{|\sum_{\boldsymbol{fi}}||\sum_{\boldsymbol{gi}}|}}\right] \quad (5)$$

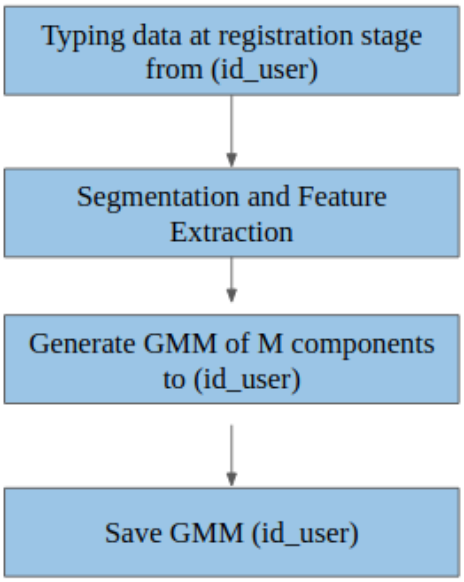


Fig. 6. Flowchart of the registration stage. Source: Created by the authors.

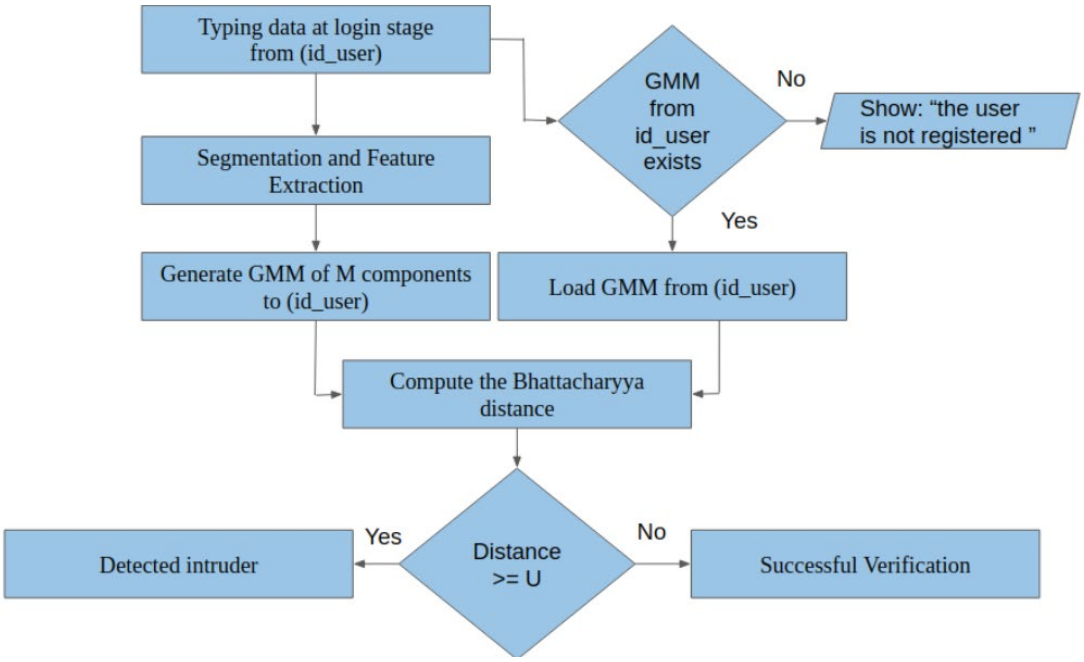


Fig. 7. Flowchart of the login stage. Source: Created by the authors.

For this experiment a cross validation strategy was carried out with 5 folds (Subject independent in each fold).

Therefore 4 subjects were considered for the test and 16 were considered for the training. The Fig.8. shows the test and train sets for each fold.

### *3.1.1 Training of the GMM-based model*

The training stage is considered to find the optimal hyper-parameters of the classifier that makes the decision. The number of Gaussian components ($M$) were optimized following a grid search strategy between 1 and 50 in steps of 3 (with selection criterion in the minimum EER).

The threshold $U$ was optimized between 0 and 1 up to steps of $10^{-3}$ (selection criterion also in the EER).

These parameters are found for each fold. In each fold we consider the best M where the final EER was optimal. For this modality of intrusive verification, the optimal point is in $M = 34 \pm 3.356$ which is the median of the best M in each fold. The $U$ value was also varied from 0 to 1 for each $M$ and the average of the best thresholds along the folds is $U = 0.148 \pm 0.006$.

### *3.1.2 Test of experiment 1*

The results of this experiment are shown in Table 4. The performance is measured in terms of FPR and FNR. The usability of the method is measured in terms of the Cost to a User to Enroll (CUE) and the Cost to a User to Authenticate (CUA) [17]. These costs refer to the number of keys required to be pressed to do the registration or authentication procedure. The registration model is generated with the first 4 tasks; therefore, the CUE is 314 keystrokes.

The log-in model is generated with Task 3 and Task 4 as it is indicated in Table 4. The minimum EER is obtained with task 3, however this is the task with the highest CUA. Tasks 1 and Task 2 do not have the minimum keystrokes required to perform their modeling with a GMM with $M = 34$. A minimum of $2M + 1$ keystrokes is needed in order to estimate the GMM's covariances, therefore these tasks were not included in the analysis.

Table 4. Performance and usability metrics when generating login models with known tasks. Source: Created by the authors.

| Log-in Tasks | FPR (μ±σ) | FNR (μ±σ) | EER (μ±σ) | CUA |
|---|---|---|---|---|
| **3** | **11.8 ± 4.2** | **19.5± 4.1** | **15,7± 4.2** | **133** |
| 4 | 10.3 ± 3.6 | 22.5 ± 5.1 | 16,4 ± 4.4 | 91 |

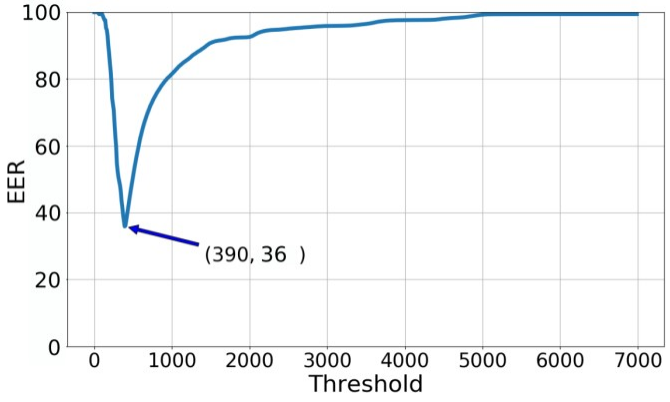


Fig. 8. EER when varying the decision threshold. Source: Created by the authors.

### 3.2 Experiment 2: Non-intrusive mode

In this experiment, tasks 1, 2, 3 and 4 were used to generate the user registration model, in the same way as in the previous experiment. The difference with the previous experiment is that the login model is generated with the task 5. The task 5 is divided into 5 equal length chunks. For each chunk, the distance between the registration and log-in models is computed. To decide whether a user is valid or not, the average distance is estimated for the 5 chunks and compared to the decision threshold $U$.

For this experiment there are 170 different users. A cross validation strategy similar to that developed in the previous experiment was used. The only difference with respect to the previous experiment is that here we addressed a 10-fold cross validation strategy, then 153 subjects are used for the training stage and 17 subjects are used in the test stage (subject independent in each fold).

#### *3.2.1 Training of the GMM-based model*

The strategy for the training of the GMM-based model in this experiment is the same as the previous experiment. For this experiment the optimal hyper-parameters are: $M = 36 \pm 6.074$ and $U = 0.013 \pm 0.021$.

#### *3.2.2 Test of the experiment 2*

Table 5 shows the performance and usability of the system by varying the number of chunks used to decide

whether the user is valid or not. In this case the CUE is the same as the previous experiment, because the same tasks are used to create the registration model.

### 3.3 Experiment 3: comparison with another methodology in non-intrusive mode

In the literature there are several works of biometric verification based on keystroke dynamic, but few verify identity of the user in a non-intrusive way. The methodology proposed in [6] is a work of identity verification in non-intrusive mode.

We implemented this methodology with the 170 users of our database. In [6], the authors propose to create a student profile based on the average Hold time when pressing different keys. The similarity between registered and log-in samples is calculated by the Euclidean distance and a training set was taken to optimize the decision threshold.

Table 5. Performance and usability metrics, generating login models with unknown tasks by the register model, using an average distance. Source: Created by the authors.

| Used Chunks | FPR (μ±σ) | FNR (μ±σ) | EER (μ±σ) | CUA |
|---|---|---|---|---|
| 1 | 13.5 ± 0.9 | 13.6± 0.5 | 13.6 ± 0.7 | 104 |
| 2 | 18.8 ± 0.7 | 9.6 ± 0.4 | 14.2 ± 0.6 | 208 |
| 3 | 12.9 ± 1.0 | 10.4 ± 0.5 | 11.7 ± 0.8 | 312 |
| 4 | 14.1 ± 1.0 | 10.0 ± 0.5 | 12.1 ± 0.8 | 416 |
| 5 | 14.7 ± 1.0 | 9.8 ± 0.4 | 12.3 ± 0.7 | 520 |

This methodology was adapted to the problem of non-intrusive verification, using tasks 1 to 4 as register tasks and the task 5 for log-in. Fig. 9 shows the EER when varying the decision threshold from 0 to 7000 with steps of 10. Fig. 9 shows an EER of 36 % when the threshold is 390. This is the minimum EER obtained using the methodology proposed in [6] for a non-intrusive verification approach. As it can be observed, the approach proposed here, based on GMM models, is more accurate and reliable than other approaches reported in the literature.

## 4. CONCLUSIONS

This study proposed a method for identity verification based on the statistical modeling of KD using GMMs. The main application of the proposed approach can be in virtual education platforms to verify the identity of a student when he/she is performing a test. The system was evaluated in two modes: (1) intrusive mode, which is text dependent, and (2) non-intrusive mode, which is text independent. (i.e., the user is not aware that his/her identity is being verified).

In the intrusive mode, the user logs-in the system with one of the tasks used in the registration stage. Since the log-in is performed with a fixed task, the user knows that the identity is being verified.

This mode showed an EER of 15.7 %. The usability of this mode was evaluated and showed a CUA of 133 keystrokes. This mode can be modified by changing the log-in tasks. For instance, the registration and log-in stages can be performed by typing the username and password. In this case the access to the system is the same than in the traditional manner. However, our proposed system provides an additional security layer because the user has to provide the username and the password with a valid KD to enter the system. The main drawback of the proposed approach is that the verification is only performed when the user logs-in the platform. If the valid user logs-in the platform but the exam is performed by an intruder, the system will not be able to detect the fraud.

In the non-intrusive mode, the log-in task is independent on the tasks used in the registration stage. In this case the user is not aware that is being verified. This mode achieved an EER of 11.7 %. This mode can be used during evaluation activities because the identity of the user can be constantly verified without interrupting the activity. Although this mode presents a higher CUA compared to the other mode, this is not a problem because the verification can be performed based on any text typed by the user including those texts written during the examination.

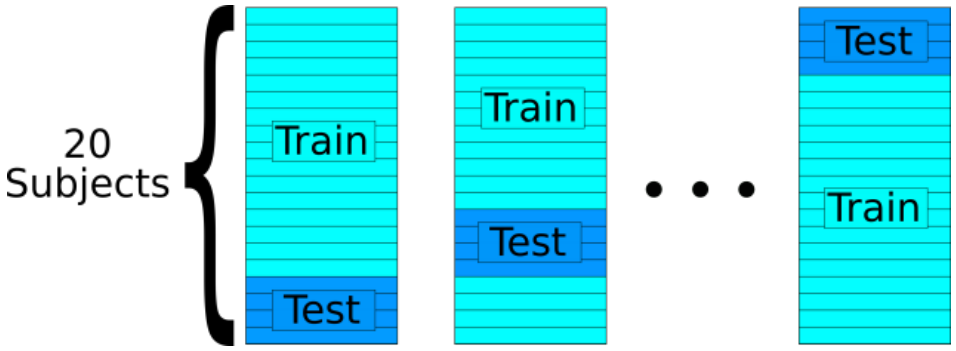


Fig. 9. EER when varying the decision threshold. Source: Created by the authors.

This study also compares the proposed methodology with the one presented in [6], which also has a non- intrusive approach.

Varying the decision threshold, the minimum EER obtained is 36 %. When the methodology proposed here is used, the EER is reduced down to 11.7 %. Therefore, with the methodology proposed in this work it is possible to improve the EER by 24.3 %, compared to the result obtained using the methodology proposed in other work for non- intrusive verification.

For future work, we want to use i-vectors to verify identity as it was presented with speech signals in [18-19].

Also, it is possible to use another machine learning algorithm like Support Vector Machine (SVM) or Recurrent Neural Networks (RNN) to classify whether a user is valid or not.

Experiments with n-grams models should also be addressed in the future.

## 5. ACKNOWLEDGEMENTS

The authors acknowledge to the GITA research group of the faculty of Engineering of the University of Antioquia. Also acknowledge to Ingeni@ of the faculty of Engineering of the University of Antioquia and the company Pratech S.A.S.

## 6. REFERENCES

[1] B. Means, Y. Toyama, R. Murphy, M. Bakia, and K. Jones, "Evaluation of Evidence-Based Practices in Online Learning: A Meta-Analysis and Review of Online Learning Studies,", U.S Department of Education, Estados Unidos, Report ED-04- CO-0040 Task 0006, 2009. Available: URL

[2] T. Bretag, *Handbook of Academic Integrity*. Singapore: Springer Singapore, 2016. https://doi.org/10.1007/978-981-287-098-8

[3] A. K. Jain, A. Ross, and S. Prabhakar, "An Introduction to Biometric Recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 4–20, Jan. 2004. https://doi.org/10.1109/TCSVT.2003.818349

[4] W. L. Bryan and N. Harter, "Studies in the physiology and psychology of the telegraphic language.," *Psychol. Rev.*, vol. 4, no. 1, pp. 27–53, 1897. https://doi.org/10.1037/h0073806

[5] R. Joyce and G. Gupta, "Identity authentication based on keystroke latencies," *Commun. ACM*, vol. 33, no. 2, pp. 168–176, Feb. 1990. https://doi.org/10.1145/75577.75582

[6] K. Longi, J. Leinonen, H. Nygren, J. Salmi, A. Klami, and A. Vihavainen, "Identification of programmers from typing patterns," in *Proceedings of the 15th Koli Calling Conference on Computing Education Research - Koli Calling '15*, Koli Finland, 2015. pp. 60–67. https://doi.org/10.1145/2828959.2828960

[7] S. Krishnamoorthy, L. Rueda, S. Saad, and H. Elmiligi, "Identification of User Behavioral Biometrics for Authentication Using Keystroke Dynamics and Machine Learning," in *Proceedings of the 2018 2nd International Conference on Biometric Engineering and Applications - ICBEA '18*, Amsterdam, 2018. pp. 50–57. https://doi.org/10.1145/3230820.3230829

[8] J. R. Young, R. S. Davies, J. L. Jenkins, and I. Pfleger, "Keystroke Dynamics: Establishing Keyprints to Verify Users in Online Courses," *Comput. Sch.*, vol. 36, no. 1, pp. 48–68, Jan. 2019. https://doi.org/10.1080/07380569.2019.1565905

[9] A. Morales, M. Falanga, J. Fierrez, C. Sansone, and J. Ortega-Garcia, "Keystroke dynamics recognition based on personal data: A comparative experimental evaluation implementing reproducible research," in *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, Arlington, 2015. pp. 1–6. https://doi.org/10.1109/BTAS.2015.7358772

[10] M. W. Shelly. *Frankestein or, the modern Prometheus*. London: Penguin, 2007. Available: URL

[11] D. Yu and L. Deng, *Automatic Speech Recognition*. London: Springer London, 2015. https://doi.org/10.1007/978-1-4471-5779-3

[12] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models," *Digit. Signal Process.*, vol. 10, no. 1–3, pp. 19–41, Jan. 2000. https://doi.org/10.1006/dspr.1999.0361

[13] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 1, pp. 72–83, 1995.

https://doi.org/10.1109/89.365379

[14] M. Nishida and T. Kawahara, "Speaker model selection based on the Bayesian information criterion applied to unsupervised speaker indexing," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 4, pp. 583–592, Jul. 2005. https://doi.org/10.1109/TSA.2005.848890

[15] P. Mahalanobis, "On the generalized distance in statistic", *National Institute of Science of India*, vol 2, no 1, pp. 49-55, Apr. 1936. Available: URL

[16] T. Arias-Vergara, J.C. Vásquez-Correa, J. R. Orozco-Arroyave, J. F Vargas-Bonilla and E. Nöth, *"Parkinson's Disease Progression Assessment from Speech Using GMM-UBM", Proceedings of Interspeech,* pp 1933-1937, San Francisco, 2016. Available: URL

[17] A. Peacock, X. Ke, and M. Wilkerson, "Typing patterns: a key to user identification," *IEEE Secur. Priv. Mag.*, vol. 2, no. 5, pp. 40–47, Sep. 2004. https://doi.org/10.1109/MSP.2004.89

[18] N. Garcia-Ospina, J.-R. Orozco-Arroyave, and J.-F. Vargas-Bonilla, "Speaker Verification System for Online Education Platforms," in *2018 International Carnahan Conference on Security Technology (ICCST)*, Montreal, 2018. pp. 1–5. https://doi.org/10.1109/CCST.2018.8585602

[19] X. Jiang, S. Wang, X. Xiang, and Y. Qian, "Integrating online i-vector into GMM-UBM for text-dependent speaker verification," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Kuala, 2017. pp. 1628–1632. https://doi.org/10.1109/APSIPA.2017.8282293