Statistical Modeling of Keystroke Dynamics Samples For the Generation of Synthetic Datasets

Denis Migdal, Christophe Rosenberger

Statistical Modeling of Keystroke Dynamics Samples For the Generation of Synthetic Datasets

Denis Migdal, Christophe Rosenberger

Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

Abstract

Biometrics is an emerging technology more and more present in our daily life. However, building biometric systems requires a large amount of data that may be difficult to collect. Collecting such sensitive data is also very time consuming and constrained, s.a. GDPR legislation in Europe. In the case of keystroke dynamics, most existing databases have less than 200 users. For these reasons, it is crucial for this biometric modality to be able to generate a significant and realistic synthetic dataset of keystroke dynamics samples. We propose in this paper an original approach for the generation of synthetic keystroke data given samples from known users as a first step towards the generation of synthetic datasets. Experimental results show the capability of the proposed statistical model to generate realistic samples from existing datasets in the literature.

Keywords: Keystroke dynamics, Statistical modelling, Synthetic dataset, Data Analysis

1. Introduction

Keystroke dynamics (KD) [1] is a behavioral biometric modality that allows the authentication of individuals through their way of typing a password or a free text on a keyboard. It is a biometric modality which has the advantage of not requiring additional sensor than the keyboard. Many applications concerning keystroke dynamics are possible such as logical access control, behavior monitoring, soft biometrics (i.e. profiling the user) or emotion analysis. This biometric modality also allows the continuous authentication of users through time [2, 3].

User authentication with keystroke dynamics is generally done in real time (*i.e.*, online) in a real world system. Scientists working on keystroke dynamics do not analyze the performance of their system in an online way (*i.e*, by asking users to authenticate themselves in real time and to impersonate other users). Indeed, they work in an offline context by using samples previously collected by other researchers, and stored in a benchmark dataset. A complete list of available keystroke dynamics datasets has been

Email addresses: denis.migdal@ensicaen.fr (Denis Migdal), christophe.rosenberger@ensicaen.fr (Christophe Rosenberger)

made in [4, 5]. As it can be seen, most of datasets have less than 200 individuals and few samples are available for each user. The collection of such datasets is very time consuming, this is the main reason why there is not more very large datasets like for the face modality [6]. This is a crucial problem for the research in this area.

In this paper, our objective is to model real KD data in order to be able generate very large synthetic KD datasets. This approach has been used for the digital fingerprint modality with the SFINGE software [7] as their collection and distribution are regulated in many countries. We believe the KD model could help the research community to create a new dataset of higher quality than the existing ones. We think this work is important, because it is known that KD studies are not fair as (i) acquisition protocols are different between studies [8]; (ii) there is not always a comparative study [9] when authors propose new algorithms; and (iii) there are not always a valuable statistical evaluation [9]. Our work contributes to solve these problems. We show in this paper that is possible to statistically model the KD of users from any existing datasets.

The paper is organized as follows. Section 2 is dedicated to provide some background information on Keystroke dynamics and existing studies for this biometric modality. We present in section 3 the definitions and the components of the analysis process of existing KD datasets. Section 4 is dedicated to the proposed KD generative model. We show its capability to generate similar synthetic keystroke dynamics data from real ones. Last, section 5 concludes this work and gives some perspectives.

This invited article supports and improves the results of the original "Analysis of Keystroke Dynamics For the Generation of Synthetic Datasets" [10].

2. Background

In this section, we provide some background information for the use of keystroke dynamics for authenticating users.

2.1. Keystroke dynamics principle

As any biometric authentication solution, a keystroke dynamic system (KDS) is composed of two main modules: the enrollment and the verification modules. Each user must enroll himself/herself in the KDS in order to compute its biometric reference given multiple samples (*i.e.*, several inputs of the password) acquired during the enrollment step. For each input, a sequence of timing information is captured (*i.e.*, time when each key is pressed or released) from which some features are extracted (*i.e.*, latencies and durations) and used to learn the model which characterizes each user. During a verification request, the claimant types his/her password. The system extracts the features and compares them to the biometric reference of the claimant. If the obtained distance is below a certain threshold, the user is accepted, otherwise he/she is rejected.

First works on KD have been done in the eighties [11], although the idea of using a keyboard to automatically identify individuals has first been presented in 1975 [12]. In the preliminary report of Gaines *et al.* [11], seven secretaries typed several paragraphs of text and researchers showed that it is possible to differentiate users with their

typing patterns. Since then, several studies have been done, allowing to decrease the quantity of information needed to build the biometric reference, while improving the performances [13, 14, 15, 16, 8]. However, most studies are not comparable because they use different datasets or protocols [8, 9].

2.2. Keystroke Dynamics Systems

As the number of collected samples during the enrollment step is low, many Keystroke Dynamics Systems are based on a distance. We aim at computing a distance between two templates K_A and K_B. In this paper, we use 4 distance functions.

- Blesha [17]: We suppose that the template K_A is associated by μ the average value of biometric samples:

$$STAT1 = \frac{(K_B - \mu)^t (K_B - \mu)}{||K_B||.||\mu||} \tag{1}$$

- Hocquet [18]: We suppose that the template K_A is associated by μ and σ the average value of biometric samples and the standard deviation.

$$STAT2 = 1 - \frac{1}{n} \sum_{i=1}^{n} e^{-\frac{|K_B(i) - \mu_i|}{\sigma_i}} \tag{2}$$

- Monrose [19]: The function is given as follows:

$$STAT3 = \sqrt{\sum_{i=1}^{n} (K_B(i) - K_A(i))^2} \tag{3}$$

- BioHashing: This algorithm is a template protection scheme [20] where the biometric template is projected given a secret key and is quantized to generate a binary code (called BioCode). The comparison is realized with the Hamming distance. We apply this protection scheme and compare the templates in the transformed domain.

In the scope of this paper, the BioHashing and Monrose distances between a template (sample) and a set of templates (references) are computed as the minimal distance of the sample with each template in the reference gallery. Moreover, the log function is applied to each of the Monrose distances.

2.3. Datasets

There exist many keystroke dynamics datasets [4]. We decided in this work to focus on fixed text datasets (i.e. where users typed the same passphrase). Datasets have been cleaned to remove incoherent data, e.g. entries in which the user did not type the asked text. This corresponds to 13% of entries in GREYC W, and less than 3 entries for other datasets.

Then, we selected datasets with less than an arbitrary number of elements (i.e. users, and entries per users). We used both 23 and 45 as arbitrary values in this paper. 23 enables to split sets into 5 classes while respecting the Cochran rule, i.e. 80% of the classes

having at least 5 elements [21]. 45 enables to split sets into 9 classes of 5 elements, knowing that 46 is the maximal value that do not discard the GREYC W2 dataset. In order to get comparable sets, the minimal number of elements is also their maximal number of elements: only the first 23, or 45 elements are used.

From the existing fixed-text datasets, only 3 matched our criteria. From these 3 datasets, we build 4 datasets composed of a fixed text Keystrokes for each user (one having 2 fixed Text, 2 datasets are thus created). Table 1 gives the used datasets in this work. Table 2 and Figure 1 give, for each datasets and each Keystroke Dynamics System, the Equal Error Rate and the ROC curve.

Name	**Text**	**# of users (23)**	**# of users (45)**	**Source**
GREYC K	greyc laboratory	120	104	[22]
GREYC W1	laboratoire greyc	79	62	[23]
GREYC W2	sésame	66	46	[23]
CMU	.tie5Roanl	51	51	[24]

Table 1: Description of used datasets.

Distance	**CMU**	**GREYC K**	**GREYC W1**	**GREYC W2**
BioHashing	0.307	0.220	0.201	0.237
Blesha	0.360	0.315	0.303	0.284
Hocquet	0.183	0.146	0.107	0.212
Monrose	0.343	0.281	0.255	0.233

Table 2: Equal Error Rate of used datasets with 45 entries per users.

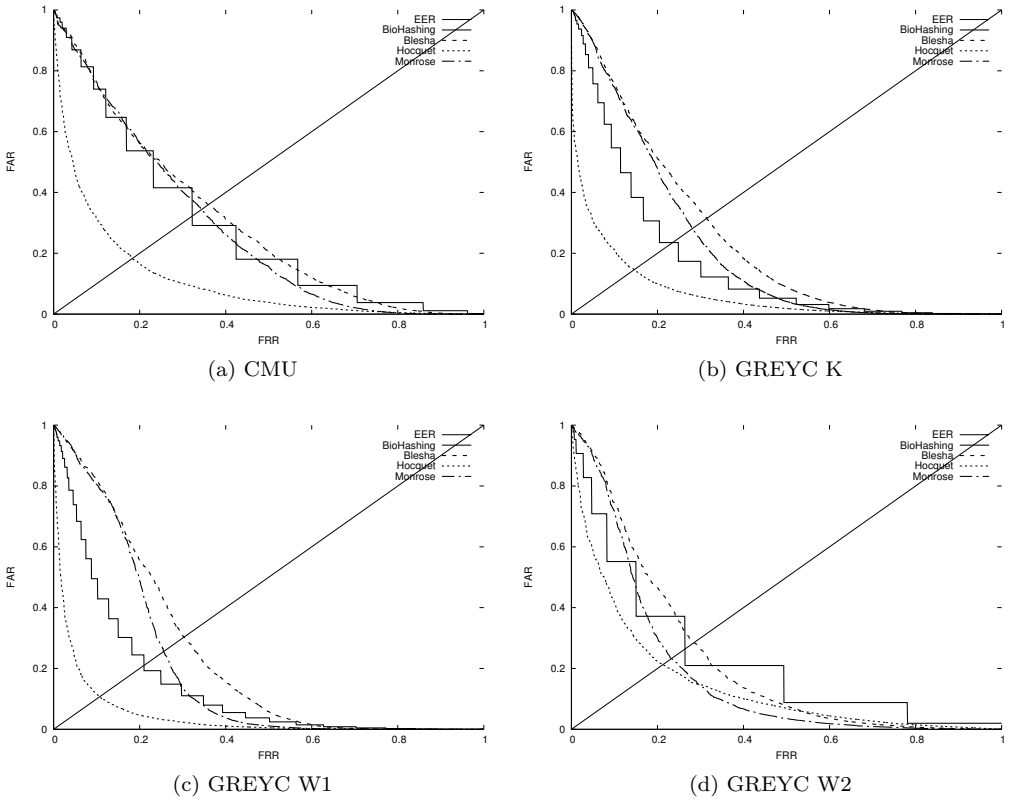
Note that the times in each dataset have been acquired in different ways. In particular, GREYC K used C# programming DateTime which has a resolution of 10.0144ms [1], which explains χ^2's poor results on this dataset. Indeed, some sets of durations have only 8 distinct values which is, when using 45 as the number of elements, less than the number of classes.

2.4. Related works

The generation of synthetic keystroke samples has already been discussed in [25, 26] where authors generated synthetic keystrokes from known users in order to test the robustness of a SVM classifier (used as matching algorithm). Only the uniform and the normal laws have been considered, with the laws parameters directly computed from the mean and standard deviation of the real durations. Authors wanted to generate synthetic keystroke dynamics samples as a naive attack to test the robustness of their presented model.

Keystrokes durations have been analyzed in [27] where authors aim at assisting the detection of synthetic keystroke samples, by detecting aberrant duration. Authors found

[1] `https://manski.net/2014/07/high-resolution-clock-in-csharp/#datetime`



(a) CMU (b) GREYC K

(c) GREYC W1 (d) GREYC W2

Figure 1: ROC curves for the used datasets with 45 entries per users.

out that some durations follow a Zipf's or/and a Benford's(/power) law on the CMU dataset using the Maximum Likelihood Estimator fitness algorithm to estimate the laws parameters. However, these findings do not enable the synthetic generation of keystroke samples as durations are not separated by users and digraphs, and thus cannot generate a duration for a given user and digraph.

In this paper, we aim at generating synthetic keystrokes as a way to replace real keystrokes in KD studies. With this approach, we consider 19 laws to find out that the distribution durations follow a gumbel law more than a normal one. We also show that laws parameters computations from the mean and standard deviation give poor results, and the use of a fitness function is required. Moreover, we are interested in the consistency of the duration between them, to generate keystroke samples as real as it can be.

3. Analysis of real KD datasets

In this section, we analyze the features from KD samples in existing datasets. We first define the formalism we consider in this study.

3.1. Formalism

We define many terms to build the proposed analysis method:

- **Digraph:** $D = [C_0, C_1]$, array of two characters.

- **DigraphTime:** $DT_D = [d_0, d_1, d_2, d_3, d_4, d_5]$, as shown in Figure 3, is an array of 6 durations from 4 times corresponding to the pressure (P) and release (R) times of each character of a Digraph D. A DigraphTime DT_D is defined as partially consistent if the following equations are verified, consistent if the following equations and inequalities are verified, and inconsistent otherwise:
  - $d_0 = d_2 - d_4$;
  - $d_0 = d_1 - d_3$;
  - $d_1 = d_2 - d_5$;
  - $d_3 = d_4 - d_5$;
  - $d_0 \geq 0$
  - $d_1 \geq 0$
  - $d_5 \geq 0$

- **Text:** $T_n = \{D_i\}_{i \in [\![0,n[\![}$, an array of n Digraphs D_i. A text T_n is said consistent if $\forall i \in]\!]0, n[\![, D_{i-1}[1] = D_i[0]$.

- **Keystroke dynamics:** $K = [\{DT_i\}_{i \in [\![0,n[\![}, T_n]$, an array of n DigraphTime DT_i associated to the Digraph $T_n[i]$. Keystroke is said consistent (or partially consistent) if T_n, and all DT_i are consistent (or partially consistent), and if $\forall i \in]\!]0, n[\![, DT_{i-1}[5] = DT_i[0]$.

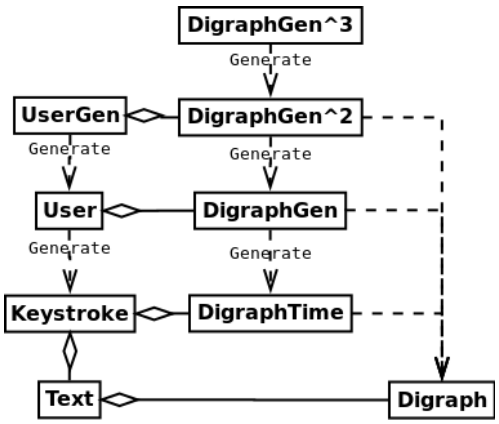


Figure 2: KD Generative model

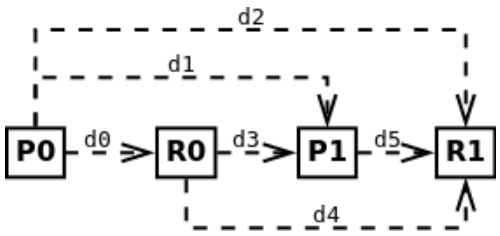


Figure 3: DigraphTime

We propose in this paper a generative keystroke dynamics model. We explain its different components (see also Figures 2 and 3):

- **DigraphGen:** $DG_D() = DT$, generates a DigraphTime for a given Digraph.
- **User:** $U(T_n) = K$, generates a keystroke dynamics sample from a given Text. A User is composed of a set of DigraphGen.
- **DigraphGen^2:** $DG^2_D() = DG_D$, generates a DigraphGen for a given Digraph.
- **UserGen:** $UG() = U$, generates a User. A UserGen is composed of a set of DigraphGen^2.
- **DigraphGen^3:** $DG^3(D) = DG^2_D$, generates a DigraphGen^2 for a given digraph.

3.2. Statistical modelling

As previously seen, generating a keystroke dynamics template from a given text T_n consists in generating an array of DigraphTime, i.e. generating $6 * n$ durations. To be able to generate a keystroke dynamics sample similar to that one user could type, these $6 * n$ durations have to be transformed into a set of assumed independent variables which laws and parameters can then be estimated for a user. We need then to randomly generate durations associated to a given user. In the scope of this paper, only the linear (in)dependency of variables is considered.

3.2.1. Variables (in)dependency

Linearly correlated variables can be transformed into a set of non-linearly correlated variables, through PCA (Principal component analysis), first introduced by Pearson in 1901 [28]. However, we show that durations are not strongly correlated between them, and thus, in the scope of this article, we assume them to be independent. Even if the usage of PCA is irrelevant in such a case, its first step enables the computation of the inter-correlations of two variables by the computation of a correlation matrix. In a correlation matrix $C = \{C_{i,j}\}_{\{i,j\}\in[\![0,n[\![^2}$, $C_{i,j}$ is the linear correlation between the variables i and j.

Reminder: *A correlation matrix $C = \{C_{i,j}\}_{\{i,j\}\in[\![0,n[\![^2}$, with $C_{i,j}$ the linear correlation between the variables i and j, is computed as follows:*

1. *Given a matrix $M = \{M_k\}_{k\in[\![0,K[\![}$ of K entries $M_k = \{M_{k,i}\}_{i\in[\![0,n[\![}$, with $M_{k,i}$ the realization of the variable i for the entry k.*
2. *$\bar{M} = \{\frac{M_{k,i}-\mu_i}{\sigma_i}\}_{i\in[\![0,n[\![,k\in[\![0,K[\![}$ where μ_i is the mean of $\{M_{k,i}\}_{k\in[\![0,K[\![}$, and σ_i, its standard deviation.*
3. *$C = 1/K * \bar{M}^T * \bar{M}$*

To qualify presence of specific correlations between two variables i, j inside m subsets of entries, m correlations matrix $C^l, l \in [\![0, m[\![$ are computed from such subsets. Each element $C_{i,j}$ of the final correlation matrix C is then computed as the mean of each $C^l_{i,j}$: $C_{i,j} = \frac{1}{m}\Sigma_{l=0}^{m-1} C^l_{i,j}$. If each subset corresponds to, e.g. a User, M will be said, in this

paper, "splitted by User", and C will qualify the presence of User-specifics correlations across all Users.

To identify the same correlations between two sets of variables $\{i_x\}_{x\in[\![0,m[\![}, \{j_x\}_{x\in[\![0,m[\![}$, of length m, entries are splitted in m sub-entries $M'_{m*k+x} = \{M_{k,o_x}\}_{o\in\{i,j\}}$. The correlation matrix C is then computed from M'. If each x corresponds to, e.g. a Digraph, M will be said, in this paper, "merged by Digraph", and C will qualify the presence of non-Digraphs-specifics correlations across all Digraphs.

3.2.2. Laws followed by Variables

Once the variables are assumed independent, or transformed in such a way, laws followed by each variable are searched through the following process:

1. Given the realizations of a variable X, and a law law_p with unknown parameters p;
2. Estimate $\hat{p}$ from the median, mean, min, max, or/and standard deviation of X;
3. Estimate p through a fitness algorithm using $\hat{p}$ as a starting point.

In the scope of this paper, we seek to maximize $1-\chi^2(X, law, p)$. The χ^2 test qualifies the capacity of a set of observed values to match a set of expected values. The χ^2 test returns $\chi^2(X, law, p) = 1 - \alpha$, in which α is the p-value, i.e. the probability to obtain the same $1 - \alpha$ score if X follows law_p. If the p-value is below an arbitrary threshold (s.a. 0.05), the hypothesis "X follows law_p" can then be rejected.

However, in the scope of this paper, our goal is not to reject hypothesis, but to select laws that best represent X. The $\chi^2(X, law, p)$ score can then be seen as a score of distance between observed values of X, and the expected values. For the same reason, the number of estimated parameters is not subtracted to the freedom, in order to have comparable values across all laws.

Reminder: *We compute $\chi^2(X, law, p)$ as follows:*

1. *Let $Card(X)$ be the cardinal of X;*
2. *Let $a\%b$ be the rest of the division of a by b;*
3. *$\mathbb{R}$ is divided in $n = \lceil Card(X)/5 \rceil$ subspaces $E_i, i \in [\![0, n[\![$, each expected to contain 5 elements of X. E_{n-1} is expected to contain $Card(X)\%5$ elements of X if $5 \nmid Card(X)$;*
4. *Let $X_i = X \cap E_i$;*
5. *Let $Card(E_i) = 5$, and $Card(E_{n-1}) = Card(X)\%5$ if $5 \nmid Card(X)$;*
6. *Let $Sum = \Sigma_{i=0}^{n-1}(Card(E_i) - Card(X_i))^2/Card(E_i)$.*
7. *Let cdf_f be the cumulative distribution function of the law χ^2 of freedom f;*
8. *$\chi^2(X, law, p) = cdf_{n-1}(Sum)$.*

To qualify the capacity of n subsets of X, $X_i, i \in [\![0, n[\![$, to follow a same law law, but each with different parameters p_i, $s = 1 - \chi^2(X, law)$ is computed as the mean of the χ^2 test applied on each X_i: $s = 1 - \frac{1}{n}\Sigma_{i=0}^{n-1}\chi^2(X_i, law, p_i)$. The higher s is, the more the law law is assumed to fit the observed values. In the scope of this paper, 5 fitness algorithms are used:

- Maximum Likelihood Estimation (R_mle) ;
- Quantile Matching Estimation (R_qme) ;
- Maximum Goodness-of-fit Estimation (R_mge) ;
- The best estimation between R_mle, R_qme, and R_mge (R_max) ;
- $\hat{p}$ (raw) ;

The R_mle, R_qme, and R_mge fitness algorithms are executed through R's `fitdist` function[2]. {1/3,2/3} is used as probs parameter for R_qme. If the fitness algorithm fails to estimate p, p is set to $\hat{p}$, and $1 - \chi^2(X, law, p)$ is assumed to be 0.

In this paper, a set of 19 laws have been tested with the raw estimator, with and without exclusion of aberrant values (here, values that differ from $\pm 3\sigma$ from the median value of X):

- arcsine
- beta
- betaprime
- chi
- chisquare
- raised cosine
- erlang
- exponential
- f
- gamma
- gumbel
- laplace
- logistic
- lognormal
- uniform
- normal
- rayleigh
- student's t
- triangular

From these tested laws, the best 3 are selected, i.e. the 3 laws that maximize $s = 1 - \frac{1}{n}\Sigma_{i=0}^{n-1}\chi^2(X_i, law, p_i)$, and are tested again with the other fitness algorithms. All laws are not directly tested with all fitness algorithms to gain time on the execution, but also due to the fact that all laws (s.a. raised cosine) are not defined in R.

3.3. Experimental observations

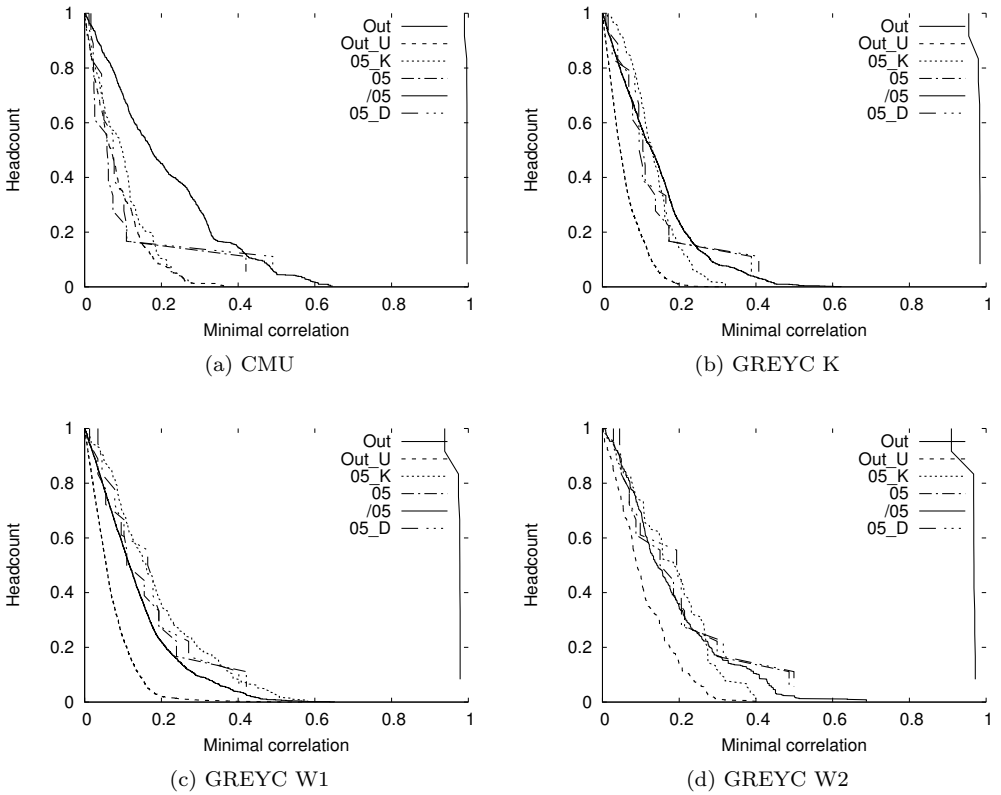
In this section, we first analyze the statistics of real keystroke dynamics from the datasets presented in section 2.3.

3.3.1. Durations correlations

We analyze as a starting point the correlation between durations in a keystroke dynamics sample.

First, diagonals of correlation matrix are discarded. Correlations between two durations $DT_{D_i}[5]$, and $DT_{D_j}[0]$ are discarded if $j = i + 1$, as they are in fact the same duration. Digraph are considered equal if their positions in the keystroke sample are equals.

[2] `https://www.rdocumentation.org/packages/fitdistrplus/versions/1.0-11/topics/fitdist`



(a) CMU (b) GREYC K

(c) GREYC W1 (d) GREYC W2

Figure 4: Number of correlations greater to a minimal value, between durations from different Digraphs (Out, Out_U), durations d_0 and d_5, with durations of the same Digraph (05_K, 05_D, 05), and between durations d_1 to d_4 inside a same Digraph (/05).

As shown in Figure 4, no strong stable correlation has been found between durations from different Digraph, (Out: dataset, Out_U: dataset splitted by User). DigraphTime will be thus assumed independent. Also, no strong stable correlation implying durations d_0 and d_5 of a same DigraphTime has been found (05_K: dataset splitted by User, 05_D: dataset merged and splitted by Digraph, 05: dataset merged by Digraph).

Stable correlations have been detected between durations d_1, d_2, d_3, d_4 of a same DigraphTime ($\overline{05}$: dataset merged by Digraph). It is easy to understand such a result as these durations can be written as $d_x = d_3 + k_x * d_0 + l_x * d_5$ with $l_x \in \{0, 1\}$, $k_x \in \{0, 1\}$, and $\sigma(d_3) \approx 3 * \sigma(d_0 + d_5)$ (see Table 3). In the scope of this paper, DigraphTime is assumed to be computable from 3 independent durations.

3.3.2. Durations laws

For the 6 DigraphTime durations $DT_D[i]$, $i \in [\![0, 6[\![$, the 10 best laws that maximize $1 - \chi^2(DT_D[i], law)$, with parameters depending on the Digraph and User, are presented

Dataset	$\sigma(d_3)/(\sigma(d_0 + d_5))$	$\sigma(d_3)/(\sigma(d_0) + \sigma(d_5))$
GREYC K	3.86	2.98
GREYC W1	3.14	2.41
GREYC W2	2.51	2.03
CMU	6.24	4.96

Table 3: Standard deviation of d_0 durations, compared to the standard deviation of d_3 and d_5 durations.

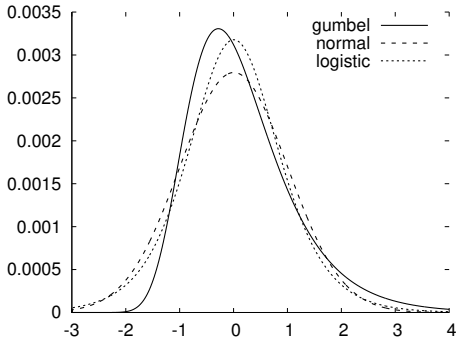


Figure 5: Density function (pdf) of several laws (with median=0, standard deviation=1).

in Table 4. DigraphTime durations will then be assumed to best follow either a gumbel, a normal, or a logistic, which parameters depend on the User and Digraph.

These findings are confirmed in Table 5. The gumbel law seems to best fit d_1 to d_4 durations followed by either the normal or the logistic law. However, for d_0 and d_5 durations, the normal law seems to best fit them, followed by the logistic law and the gumbel law. The exclusion of aberrant values seems to increase the fitness of the law.

As shown in Figure 5, these three laws are quite similar. Contrary to the two other, the gumbel law is asymmetric and possesses a trail that match users' hesitations when typing.

We define the coverage as the headcount of sets for which **1 -**$\chi^2(X, law) > 0.01$. As shown in Figures 6 to 9, sets of 23 elements give better χ^2 scores than with 45 elements, that can be explained by the fact that sets of 45 elements have more classes, and thus the χ^2 test is more strict. GREYC K gives poor results, that can be explained to its d_0 and d_5 durations and the time precision of near 10ms. To the contrary, CMU gives the best results, followed by GREYC W2 and GREYC W1. As expected, the R_max fitness algorithm performs better than other fitness algorithms. Although, R_mle and R_qme perform poorly, they still give a significant increase to the R_max fitness algorithm. Surprisingly, raw fitness algorithm outperforms R_qme.

In order to reduce the number of possible combinations, each duration will be generated with by two laws X, Y (X being used for d_0 and d_5, and Y for d_1 to d_4), but with

Datasets	Rank	d_0	χ^2	d_1	χ^2	d_2	χ^2	d_3	χ^2	d_4	χ^2	d_5	χ^2
CMU	1	normal (3σ)	0.550	gumbel (3σ)	0.262	gumbel (3σ)	0.266	gumbel (3σ)	0.266	gumbel (3σ)	0.285	normal (3σ)	0.546
	2	logistic (3σ)	0.546	logistic (3σ)	0.194	logistic (3σ)	0.200	logistic (3σ)	0.193	logistic (3σ)	0.210	logistic (3σ)	0.546
	3	cosine (3σ)	0.524	normal (3σ)	0.172	normal (3σ)	0.189	normal (3σ)	0.166	normal (3σ)	0.197	cosine (3σ)	0.523
	4	logistic	0.505	laplace (3σ)	0.159	laplace (3σ)	0.167	laplace (3σ)	0.156	laplace (3σ)	0.178	logistic	0.507
	5	normal	0.491	cosine (3σ)	0.149	cosine (3σ)	0.163	cosine (3σ)	0.142	cosine (3σ)	0.162	normal	0.484
	6	cosine	0.438	gumbel	0.127	gumbel	0.133	gumbel	0.123	gumbel	0.142	cosine	0.433
	7	gumbel (3σ)	0.410	logistic	0.081	logistic	0.084	rayleigh (3σ)	0.078	logistic	0.085	gumbel (3σ)	0.403
	8	gumbel	0.388	normal	0.068	laplace	0.076	logistic	0.077	laplace	0.077	laplace	0.391
	9	laplace (3σ)	0.384	laplace	0.063	normal	0.072	laplace	0.067	normal	0.067	laplace (3σ)	0.390
	10	laplace	0.380	cosine	0.057	cosine	0.063	normal	0.055	cosine	0.051	gumbel	0.377
GREYC K	1	normal (3σ)	0.009	gumbel (3σ)	0.149	gumbel (3σ)	0.175	gumbel (3σ)	0.143	gumbel (3σ)	0.157	cosine (3σ)	0.008
	2	cosine (3σ)	0.008	normal (3σ)	0.143	normal (3σ)	0.173	normal (3σ)	0.140	normal (3σ)	0.154	normal (3σ)	0.008
	3	normal	0.008	cosine (3σ)	0.135	logistic (3σ)	0.162	logistic (3σ)	0.137	logistic (3σ)	0.147	normal	0.008
	4	cosine	0.007	logistic (3σ)	0.135	cosine (3σ)	0.153	cosine (3σ)	0.129	cosine (3σ)	0.142	cosine	0.007
	5	logistic (3σ)	0.006	gumbel	0.099	gumbel	0.109	gumbel	0.092	gumbel	0.098	logistic (3σ)	0.005
	6	logistic	0.005	laplace (3σ)	0.088	laplace (3σ)	0.103	laplace (3σ)	0.089	laplace (3σ)	0.096	logistic	0.005
	7	uniform (3σ)	0.004	normal	0.076	normal	0.090	logistic	0.079	logistic	0.084	uniform (3σ)	0.004
	8	gumbel	0.004	logistic	0.074	logistic	0.086	normal	0.072	normal	0.076	gumbel (3σ)	0.004
	9	gumbel (3σ)	0.004	cosine	0.068	cosine	0.075	cosine	0.067	cosine	0.069	gumbel	0.004
	10	uniform	0.004	laplace	0.055	laplace	0.065	rayleigh (3σ)	0.055	laplace	0.057	uniform	0.004
GREYC W1	1	cosine (3σ)	0.149	logistic (3σ)	0.194	logistic (3σ)	0.231	logistic (3σ)	0.164	normal (3σ)	0.188	cosine (3σ)	0.147
	2	normal (3σ)	0.145	normal (3σ)	0.192	normal (3σ)	0.227	gumbel (3σ)	0.159	logistic (3σ)	0.186	normal (3σ)	0.145
	3	logistic (3σ)	0.136	gumbel (3σ)	0.192	gumbel (3σ)	0.220	normal (3σ)	0.153	gumbel (3σ)	0.181	logistic (3σ)	0.135
	4	logistic	0.124	cosine (3σ)	0.171	cosine (3σ)	0.207	cosine (3σ)	0.132	cosine (3σ)	0.164	logistic	0.124
	5	normal	0.119	laplace (3σ)	0.140	laplace (3σ)	0.166	laplace (3σ)	0.123	laplace (3σ)	0.133	normal	0.119
	6	cosine	0.116	logistic	0.114	gumbel	0.140	gumbel	0.089	logistic	0.108	cosine	0.115
	7	laplace	0.095	gumbel	0.110	logistic	0.137	logistic	0.084	gumbel	0.107	laplace	0.096
	8	laplace (3σ)	0.095	normal	0.103	normal	0.131	normal	0.076	normal	0.093	laplace (3σ)	0.095
	9	gumbel (3σ)	0.092	laplace	0.091	cosine	0.113	laplace	0.072	laplace	0.085	gumbel (3σ)	0.092
	10	gumbel	0.091	cosine	0.086	laplace	0.104	cosine	0.063	cosine	0.074	gumbel	0.091
GREYC W2	1	normal (3σ)	0.208	gumbel (3σ)	0.235	gumbel (3σ)	0.264	gumbel (3σ)	0.198	logistic (3σ)	0.226	normal (3σ)	0.210
	2	cosine (3σ)	0.191	logistic (3σ)	0.217	logistic (3σ)	0.250	logistic (3σ)	0.188	gumbel (3σ)	0.219	logistic (3σ)	0.190
	3	logistic (3σ)	0.190	normal (3σ)	0.193	normal (3σ)	0.224	normal (3σ)	0.188	normal (3σ)	0.212	cosine (3σ)	0.187
	4	logistic	0.161	cosine (3σ)	0.179	cosine (3σ)	0.214	cosine (3σ)	0.155	cosine (3σ)	0.173	logistic	0.165
	5	gumbel (3σ)	0.158	laplace (3σ)	0.169	laplace (3σ)	0.179	laplace (3σ)	0.128	laplace (3σ)	0.156	gumbel (3σ)	0.150
	6	normal	0.148	gumbel	0.138	gumbel	0.146	gumbel	0.106	gumbel	0.136	normal	0.148
	7	cosine	0.135	logistic	0.114	logistic	0.124	normal	0.096	logistic	0.121	cosine	0.133
	8	gumbel	0.132	normal	0.093	normal	0.117	logistic	0.094	normal	0.107	gumbel	0.131
	9	laplace (3σ)	0.131	laplace	0.092	laplace	0.115	laplace	0.084	laplace	0.105	laplace (3σ)	0.123
	10	laplace	0.118	cosine	0.082	cosine	0.104	rayleigh (3σ)	0.083	cosine	0.092	laplace	0.115

Table 4: Top 10 results of χ^2 tests with 19 laws, using raw estimator, with (3σ) and without exclusion of aberrant values. $\chi^2 = \mathbf{1}$ -$\chi^2(X, law)$

Datasets	Rank	d_0	χ^2	d_1	χ^2	d_2	χ^2	d_3	χ^2	d_4	χ^2	d_5	χ^2
CMU	1	normal (3σ)	0.685	gumbel (3σ)	0.530	gumbel (3σ)	0.533	gumbel (3σ)	0.533	gumbel (3σ)	0.544	normal (3σ)	0.680
	2	logistic (3σ)	0.677	gumbel	0.503	gumbel	0.511	gumbel	0.503	gumbel	0.515	logistic (3σ)	0.673
	3	logistic	0.668	logistic (3σ)	0.360	logistic (3σ)	0.352	logistic (3σ)	0.338	logistic (3σ)	0.363	logistic	0.665
	4	normal	0.668	normal (3σ)	0.348	normal (3σ)	0.347	normal (3σ)	0.327	normal (3σ)	0.360	normal	0.663
	5	gumbel (3σ)	0.604	normal	0.322	logistic	0.324	logistic	0.309	logistic	0.334	gumbel (3σ)	0.591
	6	gumbel	0.596	logistic	0.321	normal	0.324	normal	0.301	normal	0.324	gumbel	0.577
GREYC K	1	normal	0.011	gumbel (3σ)	0.305	gumbel (3σ)	0.357	gumbel (3σ)	0.287	gumbel (3σ)	0.310	normal	0.010
	2	normal (3σ)	0.011	gumbel	0.296	gumbel	0.350	gumbel	0.282	gumbel	0.302	normal (3σ)	0.010
	3	logistic	0.009	normal (3σ)	0.235	normal (3σ)	0.279	normal (3σ)	0.223	normal (3σ)	0.247	logistic	0.009
	4	logistic (3σ)	0.009	logistic (3σ)	0.229	logistic (3σ)	0.267	logistic (3σ)	0.214	logistic (3σ)	0.238	logistic (3σ)	0.008
	5	gumbel (3σ)	0.009	normal	0.209	normal	0.250	normal	0.201	normal	0.222	gumbel (3σ)	0.008
	6	gumbel	0.008	logistic	0.209	logistic	0.248	logistic	0.194	logistic	0.221	gumbel	0.008
GREYC W1	1	normal (3σ)	0.197	gumbel (3σ)	0.347	gumbel (3σ)	0.408	gumbel (3σ)	0.299	gumbel (3σ)	0.350	normal (3σ)	0.198
	2	normal	0.196	gumbel	0.342	gumbel	0.403	gumbel	0.292	gumbel	0.343	normal	0.196
	3	logistic	0.191	normal (3σ)	0.309	normal (3σ)	0.359	logistic (3σ)	0.257	normal (3σ)	0.297	logistic	0.193
	4	logistic (3σ)	0.186	logistic (3σ)	0.301	logistic (3σ)	0.357	normal (3σ)	0.255	logistic (3σ)	0.294	logistic (3σ)	0.188
	5	gumbel (3σ)	0.161	logistic	0.283	logistic	0.338	normal	0.240	logistic	0.281	gumbel (3σ)	0.163
	6	gumbel	0.155	normal	0.283	normal	0.333	logistic	0.237	normal	0.274	gumbel	0.158
GREYC W2	1	normal (3σ)	0.280	gumbel (3σ)	0.441	gumbel (3σ)	0.491	gumbel (3σ)	0.358	gumbel (3σ)	0.417	normal (3σ)	0.278
	2	logistic	0.267	gumbel	0.419	gumbel	0.462	gumbel	0.338	gumbel	0.408	logistic	0.264
	3	logistic (3σ)	0.265	logistic (3σ)	0.340	normal (3σ)	0.383	normal (3σ)	0.294	logistic (3σ)	0.350	logistic (3σ)	0.260
	4	normal	0.260	normal (3σ)	0.331	logistic (3σ)	0.370	logistic (3σ)	0.284	normal (3σ)	0.345	normal	0.253
	5	gumbel	0.245	logistic	0.319	normal	0.367	normal	0.267	logistic	0.311	gumbel	0.240
	6	gumbel (3σ)	0.239	normal	0.318	logistic	0.362	logistic	0.265	normal	0.307	gumbel (3σ)	0.237

Table 5: Top 6 results of χ^2 tests with 3 laws, using R_max estimator, with (3σ) and without exclusion of aberrant values. $\chi^2 = \mathbf{1}$ -$\chi^2(X, law)$

different parameters. The configuration will be noted X_Y. If X and Y are the same law, the configuration will be noted X.

In our study, we used 7 configurations obtained by combining the normal, and logistic law as X, and the gumbel, normal, and logistic law as Y, and adding the configuration gumbel_gumbel (i.e. gumbel). If the parameters of the laws have been estimated with exclusion of aberrant values, "-3s" is appended to the configuration name.

We can see clearly in Tables 4 and 5 that the estimated laws and parameters for all DigraphTime durations are quite similar for the datasets we used in this study. Thanks to these statistical observations, we propose a generative model of keystroke dynamics data in the next section.

4. Keystroke dynamics generative model

4.1. Principles

As seen in the previous section, DigraphTime durations follow either a gumbel, a normal, or a logistic law which parameters can be estimated for each known User and Digraph. For a given User and Digraph, a DigraphGen can be then implemented as a set of 6 random engines generating the 6 DigraphTime durations with the chosen law and estimated parameters.

The full generative algorithm is thus the following:

- Select two laws, one for d_0 and d_5, one for d_1 to d_4;

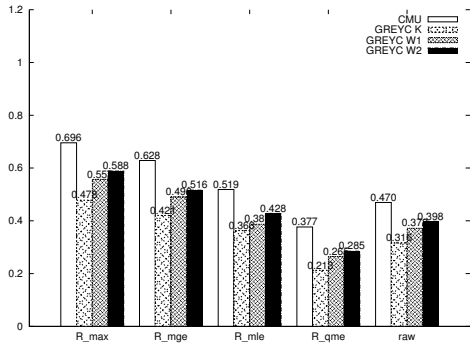


Figure 6: **1 -** $\chi^2(law)$ for gumbel (3σ) with 23 elements per sets

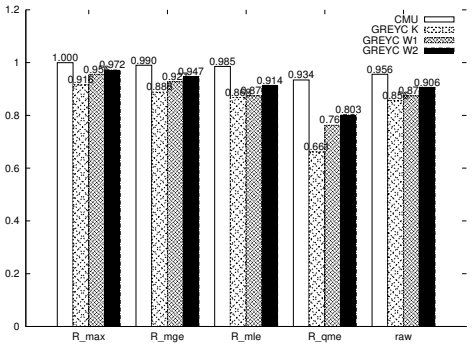


Figure 8: Coverage for gumbel (3σ) with 23 elements per sets

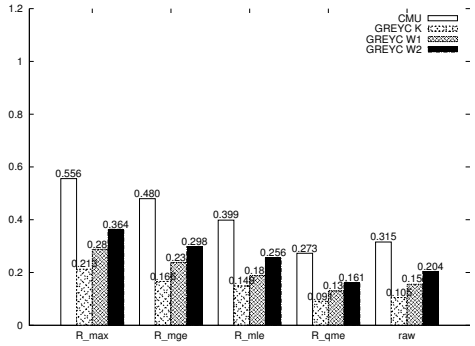


Figure 7: **1 -** $\chi^2(law)$ for gumbel (3σ) with 45 elements per sets

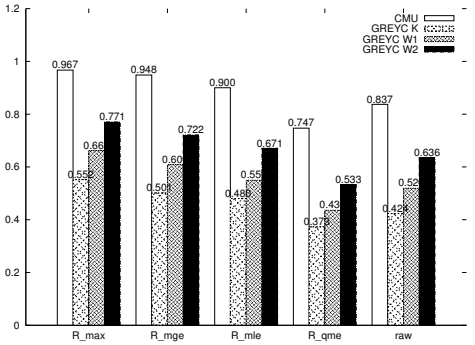


Figure 9: Coverage for gumbel (3σ) with 45 elements per sets

- Estimate the parameters of the durations for each DigraphTime;
- Generate a new Keystroke by randomly generating durations from the chosen laws and estimated parameters;
- Apply a consistency strategy on the generated Keystroke.

We propose 10 consistency strategies, 1 for inconsistent DigraphTime, in which all durations are randomly generated (u), and 10 for partially-consistent DigraphTime, in which 3 durations are computed from the 3 others. The durations to compute can be chosen among the 8 following lists, and be used for all Digraph and User, or be randomly chosen (null) for each new DigraphTime to generate:

- 0: $d_3d_4d_5$
- 1: $d_2d_3d_5$
- 2: $d_2d_3d_4$
- 3: $d_1d_4d_5$
- 4: $d_1d_3d_4$
- 5: $d_1d_2d_5$
- 6: $d_1d_2d_4$
- 7: $d_2d_1d_3$

We also propose an 11th consistency strategy that perform the mean of the 8 strategies from the previous list (m). For each consistency strategy, we propose a fully-consistent version which first applies the consistency strategy, then set to 0 negative d_0, $d5$, and d_1 durations, before recomputing d_2, d_3, and d_4 from the 3 previous duration. Such

strategies are suffixed by 'c'.

Once the DigraphGen created for a given User, the keystroke dynamics of a given Text T_n is generated through the following process:

1. $K[1] = T_n$
2. $\forall i \in [\![0, n[\![, K[0][i] = DT_{T_n[i]} = DG_{T_n[i]}()$.

   Before the consistency strategy application, and if Keystroke is expected to be consistent (or partially consistent), the DigraphTime first duration $K[0][i][0]$ is settled, if exists (i.e. if $i > 0$), to the last duration of the previous DigraphTime $K[0][i-1][5]$.

3. If fully-consistent strategy, d_2 to d_4 recomputed after setting negative d_0, d_1, and d_5 to 0.

4.2. Synthetic dataset generation: protocol

20 synthetic datasets are generated for each real KD datasest, and each possible configuration, i.e. each law configuration L and each consistency strategy CS. The configuration is labelled L.CS. These synthetic datasets are generated so as to contain the same number of users and entries per user than the real dataset from which it is generated (as seen in previous section).

For each synthetic dataset, and each distance function DistFct (matching algorithm), 3 sub-datasets are computed:

- *DataSU:* to qualify the capacity of synthetic Keystroke dynamics to be indistinguishable from real Keystroke dynamics;
- *DataU:* to qualify the KDS performance with real Keystroke dynamics data;
- *DataS:* to qualify, in comparison with DataU, the capacity of synthetic datasets to match the KDS performance that would be expected with real Keystroke dynamics data.

These datasets are composed of legitimate and impostor scores, computed with the distance function DistFct. Legitimate scores are obtained by comparing the reference template with samples from the same user. The 10 first entries of each User are used as templates, and the other entries as samples. Impostors scores are obtained by comparing the reference template of users with samples from other users. DataU is computed from the real dataset, and DataS, from the synthetic one. In DataSU, legitimate scores are legitimate scores of DataU, and impostors scores are the distance, for each User, between real user templates, and its synthetic samples.

We consider the False Acceptance Rate (FAR) describing the ratio of accepted impostor data, the False Rejection Rate (FRR) describing the ratio of falsely rejected legitimate users. The Equal Error Rate (EER) corresponds to configuration of the biometric system when FAR equals FRR. Computed indicators across the 20 synthetic datasets are aggregated by generating the following values:

- mean: the mean of the indicators ;
- error: the difference between the mean of the indicators and an expected value ;
- prec: the maximal absolute difference between the mean and the second greater indicator, and between the mean and the second lesser indicator.

These values can then be aggregated with the following process:

- mean: by the mean of the mean indicators ;
- error: by the absolute mean of the error indicators ;
- prec : by the maximal prec indicators.

4.3. Synthetic dataset generation: results

We present the obtained results of the synthetic generation of KD datasets given real ones.

4.3.1. Indicators

In this study, the durations are assumed independent, and the laws parameters, assumed to be correctly estimated by the fitness algorithm. The equivalency between synthetic keystroke samples and real keystroke samples should be guaranteed by the proposed model, and has thus to be verified.

Three indicators are used to qualify the capacity of the generated synthetic samples to match samples that would have been expected:

- Area Between the Curves (ABC): qualify the capacity of the synthetic datasets to estimate the ROC curves of real datasets (the lesser, the better);
- EER estimation error (EEE): qualify the capacity of the synthetic datasets to estimate the EER of real datasets (the lesser, the better);
- EER of real against synthetic data (ERS): qualify the capacity of synthetic datasets to usurp users from real datasets (the greater, the better).

In order to compare our findings to the related work [25, 26], we added one consistency strategy (6o) where the durations d_1, d_2, and $_4$ are computed, and all durations are positives. We used the normal law (StefN), and the uniform law (StefU), using the raw parameter estimation. As we work on fixed text, the Markov model is not used. We show in the following sections that the uniform law gives poor performances, as expected.

These three indicators are detailed in the following sections. As shown in Figures 10 and 12, best results for the configuration gumbel.5 are found for R_mge and R_qme fitness algorithm, while the raw fitness algorithm gives the worst results. The use of only 23 elements per set seems surprisingly to give slightly better results than we using 45 elements. This might be due to the fact that users' ways of typing evolve with time. The use of R_mge fitness algorithm will thus be ,by default, assumed in the following sections, as for the use of 45 elements per sets.

As shown in Figures 11 and 13, results highly depend on the dataset and the used distance function. For example, GREYC W1 dataset with Blesha distance gives an EER estimation error of 0.069 using gumbel.5, 45 elements per sets, and R_mge fitness algorithm while the best configuration, for this dataset and distance, is normal-3s.1c with an

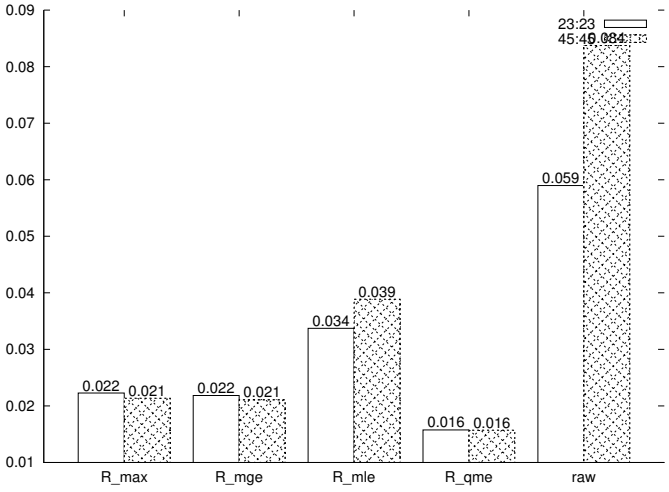


Figure 10: EER estimation error (EEE) using R_mge, gumbel.5, and 45 elements per sets.

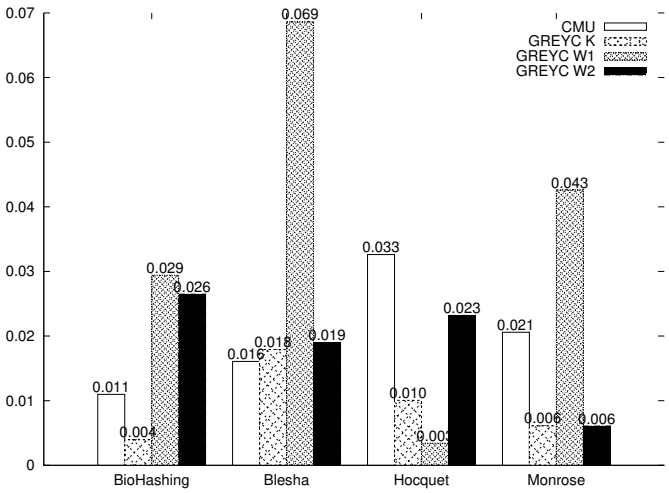


Figure 11: EER estimation error (EEE) using R_mge, gumbel.5, and 45 elements per sets.

EEE of 0.026, which performs poorly, on the same dataset, with the Hocquet distance with an EEE of 0.148.

As shown in Figures 14 and 15, the selection of the configuration is a trade-off between EEE and ERS, although some configurations give both satisfying EEE and ERS.

4.3.2. ROC curve estimations

The Area Between the Curves (ABC), computed from the synthetic (DataS) and real (DataU) entries, qualify the capacity of the synthetic datasets to estimate the ROC

	Sets of 23 elements		Sets of 45 elements	
1	R_qme:gumbel.null	0.045 (0.025±0.095) 0.415 (0.085±0.021)	R_qme:gumbel.5	0.045 (0.027±0.095) 0.337 (0.163±0.009)
2	R_qme:normal_gumbel.null	0.045 (0.026±0.089) 0.410 (0.090±0.021)	raw:gumbel-3s.5	0.045 (0.028±0.089) 0.326 (0.174±0.009)
3	R_qme:logistic_normal.null	0.044 (0.026±0.086) 0.437 (0.065±0.017)	R_qme:logistic_gumbel.null	0.053 (0.029±0.085) 0.433 (0.067±0.009)
4	R_qme:logistic_gumbel.null	0.049 (0.027±0.084) 0.454 (0.054±0.017)	R_qme:gumbel.null	0.049 (0.029±0.080) 0.400 (0.100±0.012)
5	R_qme:gumbel.0	0.045 (0.027±0.080) 0.344 (0.156±0.013)	R_qme:gumbel-3s.5	0.046 (0.029±0.086) 0.354 (0.146±0.008)
6	R_qme:normal.null	0.049 (0.028±0.085) 0.396 (0.104±0.018)	R_qme:logistic_normal.null	0.050 (0.029±0.087) 0.420 (0.080±0.013)
7	R_qme:normal_gumbel-3s.null	0.047 (0.028±0.089) 0.430 (0.071±0.017)	R_qme:normal_gumbel.null	0.049 (0.029±0.086) 0.396 (0.104±0.011)
8	R_qme:gumbel.5	0.046 (0.028±0.087) 0.346 (0.154±0.013)	R_qme:gumbel-3s.0	0.048 (0.029±0.090) 0.346 (0.154±0.012)
9	R_qme:normal-3s.null	0.048 (0.029±0.084) 0.418 (0.082±0.021)	R_qme:gumbel.0	0.047 (0.030±0.079) 0.329 (0.171±0.009)
10	R_qme:gumbel-3s.null	0.048 (0.029±0.085) 0.433 (0.068±0.013)	R_qme:normal.5	0.049 (**0.030**±0.077) 0.323 (0.177±0.010)
Comparison with the related work, using sets of 45 elements				
	With all values		With exclusion of aberrant values	
	raw:StefN.6o	0.057 (**0.035**±0.080) 0.389 (0.111±0.009)	raw:StefN-3s.6o	0.112 (**0.078**±0.093) 0.496 (0.029±0.008)
	raw:StefU.6o	0.261 (**0.165**±0.082) 0.640 (0.140±0.008)	raw:StefU-3s.6o	0.305 (**0.181**±0.072) 0.691 (0.191±0.010)

Table 6: TOP10 configuration that minimize the area between the ROC curves (ABC).
In the first line, the ABC, in the second line the ERS. Each line contains the absolute value, the error, then the precision.

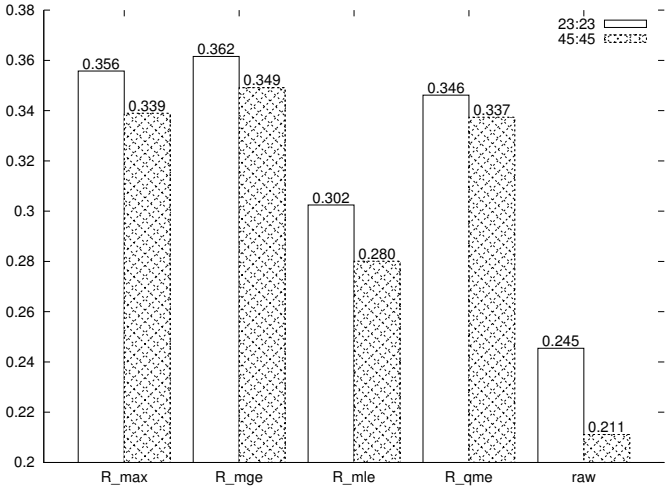


Figure 12: EER of real against synthetic data (ERS) using R_mge, gumbel.5, and 45 elements per sets.

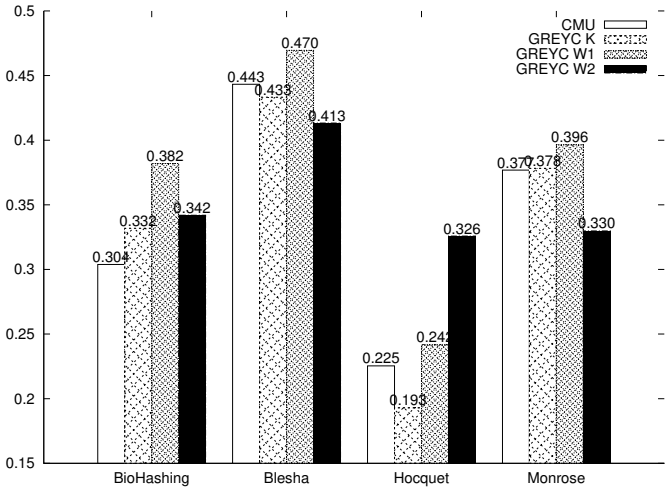


Figure 13: EER of real against synthetic data (ERS) using R_mge, gumbel.5, and 45 elements per sets.

curves of real datasets.

Table 6 shows the best configurations that minimize the ABC. For each configuration, the first line describes the mean the maximal distance between the synthetic and the real ROC curve, then the ABC, then prec, the maximal variation of the synthetic ROC curves relatively to its mean. The second line gives the ERS with its mean, error, and then prec.

As shown in Table 6, the ROC curve can be estimated with a great accuracy (ABC of 0.027 with a prec of 0.095). The bests ABC are obtained with the gumbel law, and

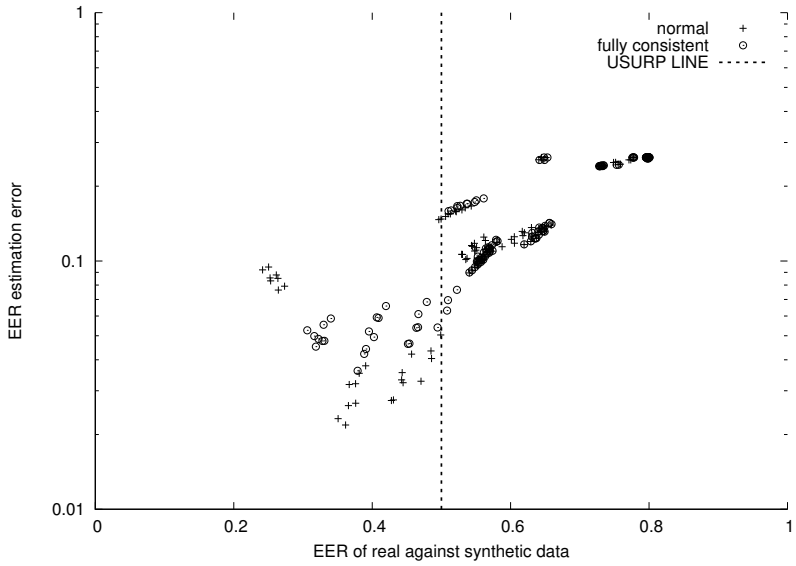


Figure 14: Performances of configurations with sets of 23 elements (using R_mge)

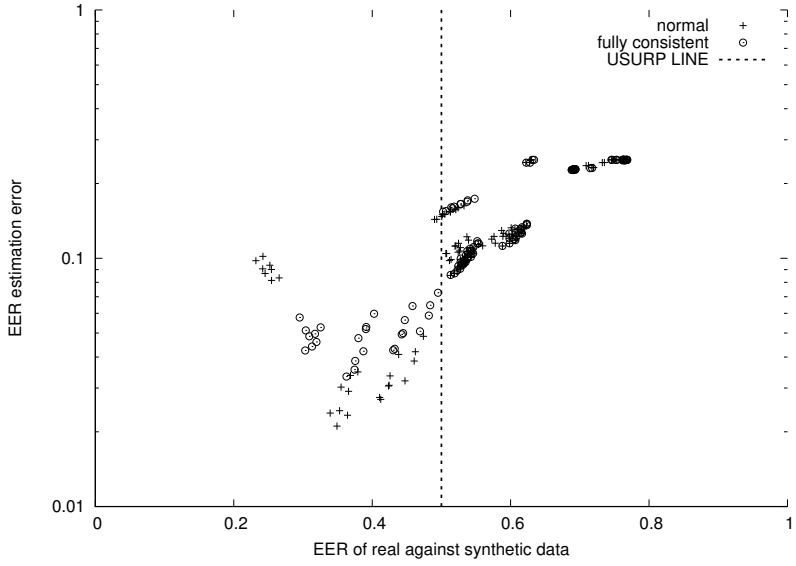
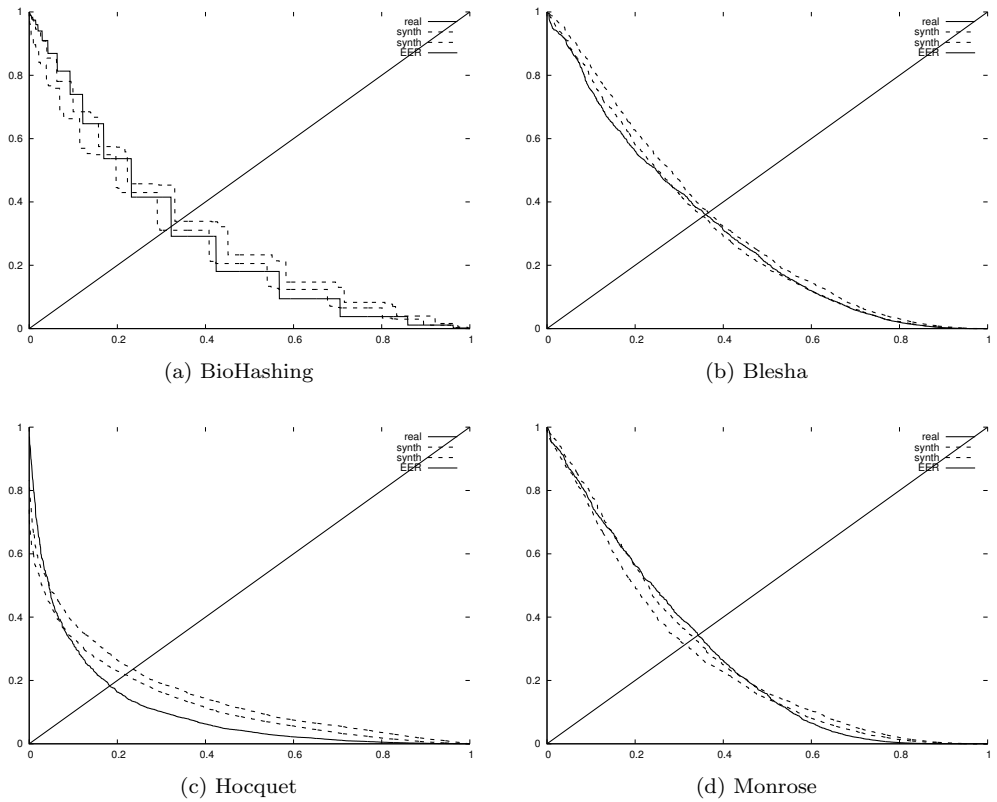


Figure 15: Performances of configurations with sets of 45 elements (using R_mge)

with strategies 5, null, and 0 which, as said in the previous section does not generate d_5, but compute it from the other durations. Removal of aberrant values when estimating the parameters (-3s) does not seem to benefit the ABS. Fully consistant strategies are absents from this top. R_qme is over represented in this top. Best configurations in ABS have lesser performances in ERS (> 0.10 instead of ~ 0.02).



(a) BioHashing (b) Blesha

(c) Hocquet (d) Monrose

Figure 16: ROC curves for the CMU dataset with 45 entries per users, using R_qme and gumbel.5.

The best configurations to estimate the ROC curves of real datasets has been found to be gumbel.5 (using R_qme), followed by gumbel-3s.5 (using raw). The estimation of the ROC curves with gumbel.5 and R_qme is shown in Figure 16

4.3.3. Usurpation of keystroke dynamics

The EER value computed from DataSU (ERS) is used to qualify the capacity of synthetic Keystroke dynamics data to be indistinguishable from real Keystroke dynamics data. As the EER corresponds to configuration of the biometric system when FAR equals the FRR, it is not possible to set a threshold enabling to reject less than EER % of genuine users, without accepting less than EER % impostors. Thus, with an EER of 50%, it is not possible to set a threshold that reject of accept users better than random. With a, EER $> 50\%$, more impostors will be accepted than genuine users.

However, a biometric system with an EER $< 50\%$ can be trivially built from an existing one having an EER $> 50\%$, simply by considering distance scores as similarity

	Sets of 23 elements		Sets of 45 elements	
1	R_max:logistic_normal.nullc	0.498 (0.033±0.013) 0.199 (0.065±0.016)	raw:normal-3s.6	0.500 (0.021±0.011) 0.175 (0.075±0.011)
2	R_qme:normal.4	0.506 (0.034±0.021) 0.182 (0.080±0.013)	raw:normal-3s.6c	0.501 (0.022±0.010) 0.172 (0.077±0.010)
3	R_mle:normal_gumbel.6c	0.505 (0.034±0.019) 0.192 (0.070±0.013)	R_mle:normal-3s.6	0.499 (0.022±0.010) 0.175 (0.074±0.010)
4	R_mle:normal_gumbel.6	0.505 (0.034±0.019) 0.193 (0.069±0.012)	R_mle:normal-3s.6c	0.501 (0.022±0.010) 0.173 (0.077±0.010)
5	R_max:logistic_gumbel.nullc	0.489 (0.034±0.017) 0.212 (0.053±0.014)	raw:normal_gumbel-3s.7	0.510 (0.024±0.011) 0.172 (0.078±0.012)
6	R_qme:normal.6	0.513 (0.035±0.015) 0.178 (0.084±0.013)	raw:normal_gumbel-3s.7c	0.511 (0.024±0.011) 0.171 (0.078±0.012)
7	R_mle:normal-3s.4	0.519 (0.035±0.016) 0.179 (0.082±0.014)	R_qme:gumbel.4	0.500 (0.024±0.012) 0.169 (0.080±0.010)
8	R_qme:logistic_normal-3s.nullc	0.489 (0.035±0.015) 0.207 (0.057±0.015)	R_mle:normal-3s.7c	0.495 (0.025±0.010) 0.176 (0.074±0.009)
9	R_qme:normal.4c	0.510 (0.035±0.023) 0.175 (0.087±0.017)	raw:normal-3s.2c	0.499 (0.025±0.010) 0.173 (0.076±0.011)
10	raw:normal-3s.4	0.518 (0.035±0.013) 0.180 (0.082±0.015)	R_mle:normal-3s.2c	0.499 (**0.025**±0.008) 0.173 (0.077±0.010)
Comparison with the related work, using sets of 45 elements				
	With all values		With exclusion of aberrant values	
	raw:StefN.6o	0.389 (**0.111**±0.009) 0.246 (0.021±0.010)	raw:StefN-3s.6o	0.496 (**0.029**±0.008) 0.177 (0.072±0.010)
	raw:StefU.6o	0.640 (**0.140**±0.008) 0.062 (0.188±0.007)	raw:StefU-3s.6o	0.691 (**0.191**±0.010) 0.028 (0.221±0.006)

Table 7: TOP10 configuration that enables good usurpation(ERS).
In the first line, the ERS, in the second line the EEE. Each line contains the absolute value, the error, then the precision.

scores, i.e. by rejecting users below, instead of rejecting them over, a given threshold. Meaning that for each biometric system with an EER of X, one can build a biometric system with an EER of 1 - X.

In this study, we aim at building synthetic Keystroke dynamics data that are indistinguishable (using the 4 distances functions we study) from real one, i.e. maximizing the minimum of ERS and 1 - ERS, i.e. getting an ERS as close as 50%. Obviously, if the Keystroke dynamics sample contains aberrant values, it would be easily detected. Thus, fully consistent strategies are desired.

Table 7 shows the best configurations that minimize the ERS error (i.e. $|ERS-0.50|$). For each configuration, the first line describes the ERS with its mean, error, and then prec, and the second line the synthetic data EER with its mean, EEE, and then prec.

The best usurpation are obtained with the either the gumbel or the normal law for strategies 6, 7, 4, and 2. None of theses strategies recomputes d_5. Removal of aberrant values when estimating the parameters (-3s) seems to benefit the usurpation. R_max and R_mge are absent from these top. However, as already shown in the previous section,

the best configurations in usurpation have poor results in EER estimation, with an EEE > 0.05, which is still high.

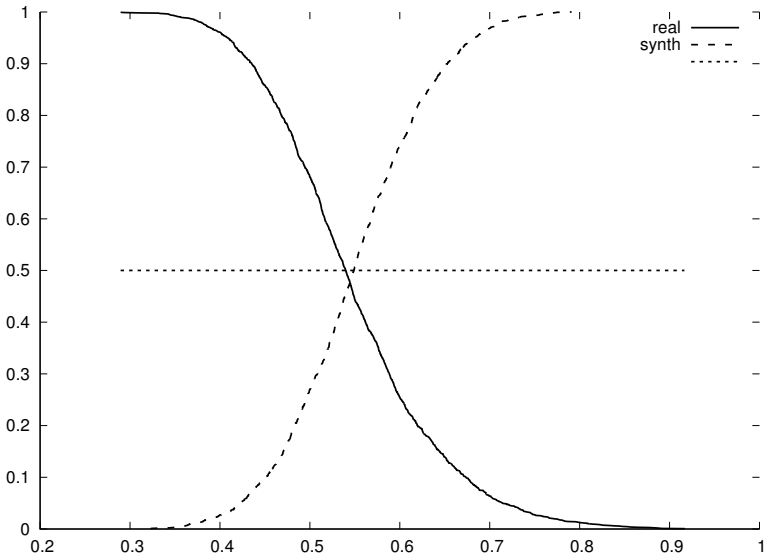


Figure 17: FAR/FRR curves using Hocquet distance of real samples against synthetics samples generated with the configuration logistic_gumbel-3s.nullc using R_mge, with sets of 45 elements.

As shown by the symmetric of the FAR/FRR curves in Figure 17, our proposed Keystroke generation method is thus able to produce synthetic samples that enable identity usurpation of a known user, by imitating its keystroke dynamics.

4.3.4. EER estimations

The difference between the EER values (EEE), computed from the synthetic (DataS) and real (DataU) entries, qualify the capacity of the synthetic datasets to estimate the EER value of the real one. Note that the threshold, in which the EER value is reached, is not taken into account.

Table 8 shows the best configurations that minimize the EEE. For each configuration, the first line describes the mean of the synthetic dataset EER with its EEE, and then prec, and the second line the ERS with its mean, error, and then prec.

As shown in Table 8, the EER value can be estimated with a great accuracy (EEE of 0.016 with a prec of 0.012). The bests EEE are obtained with the gumbel law , and with strategies 5, 2, 7, and 6. Removal of aberrant values when estimating the parameters (-3s) does not seem to benefit the EEE.R_mge and R_max are absent from this TOP. As already shown in previous sections, the best configurations in EEE have lesser performances in ERS (> 0.12 instead of ~ 0.021).

	Sets of 23 elements		Sets of 45 elements	
1	R_qme:gumbel.0	0.258 (0.015±0.016) 0.344 (0.156±0.013)	R_qme:gumbel.5	0.251 (0.016±0.012) 0.337 (0.163±0.009)
2	R_qme:gumbel.5	0.255 (0.016±0.015) 0.346 (0.154±0.013)	R_qme:gumbel-3s.5	0.238 (0.017±0.012) 0.354 (0.146±0.008)
3	R_qme:normal.0	0.264 (0.018±0.015) 0.329 (0.171±0.015)	raw:gumbel-3s.5	0.262 (0.017±0.010) 0.326 (0.174±0.009)
4	R_qme:normal.5	0.260 (0.019±0.017) 0.331 (0.169±0.014)	R_qme:normal.5	0.254 (0.018±0.011) 0.323 (0.177±0.010)
5	R_qme:gumbel.null	0.259 (0.019±0.015) 0.415 (0.085±0.021)	R_mle:normal.2c	0.244 (0.019±0.013) 0.378 (0.122±0.009)
6	R_qme:gumbel-3s.0	0.246 (0.019±0.015) 0.361 (0.139±0.011)	raw:normal.7c	0.247 (0.019±0.012) 0.375 (0.125±0.012)
7	R_qme:normal_gumbel.null	0.261 (0.019±0.016) 0.410 (0.090±0.021)	raw:normal.6c	0.245 (0.019±0.013) 0.380 (0.120±0.008)
8	raw:gumbel-3s.5	0.257 (0.019±0.013) 0.348 (0.152±0.014)	R_mle:normal.6c	0.244 (0.019±0.013) 0.380 (0.120±0.008)
9	raw:gumbel-3s.0	0.261 (0.019±0.016) 0.342 (0.158±0.015)	raw:normal.2c	0.245 (0.019±0.013) 0.378 (0.122±0.009)
10	R_qme:logistic_normal.null	0.257 (0.021±0.012) 0.437 (0.065±0.017)	R_mle:normal.7c	0.247 (**0.019**±0.015) 0.376 (0.124±0.011)
Comparison with the related work, using sets of 45 elements				
	With all values		With exclusion of aberrant values	
	raw:StefN.6o	0.246 (**0.021**±0.010) 0.389 (0.111±0.009)	raw:StefN-3s.6o	0.177 (**0.072**±0.010) 0.496 (0.029±0.008)
	raw:StefU.6o	0.062 (**0.188**±0.007) 0.640 (0.140±0.008)	raw:StefU-3s.6o	0.028 (**0.221**±0.006) 0.691 (0.191±0.010)

Table 8: TOP10 configuration that minimize the mean of EER estimation error (EEE).
In the first line, the EEE, in the second line the ERS. Each line contains the absolute value, the error, then the precision.

The best configurations to estimate the EER of real datasets has been found to be gumbel.5 (using R_qme), followed gumbel-3s.5 (using R_qme).

5. Conclusion and perspectives

In this paper, we presented a method that enables the generation of synthetic keystroke dynamics data from known Users, to either usurp real user KD, or to estimate the EER value of a KDS. These methods have been tested on fixed text, but could be as well applied to free text.

We show that, the best estimation of the EER value of a KDS is meet when using gumbel laws, without exclusion of values, and by computing durations d_5, d_1, and d_2 from other durations instead of generating them (gumbel.5). However, even though some configurations have satisfying performances in both usurpation and EER estimation, our findings show that the generation of synthetic keystroke dynamics is a trade-off between an optimal EER estimation, and an optimal usurpation.

This work constitutes a first step towards the generation of large synthetic Keystroke dynamics datasets. The following step would be the generation of keystroke dynamics data for an unknown user. Such large synthetic Keystroke dynamics datasets could then be used to fairly compare KDS performances, as well as to improve learning-based KDS' performances.

Acknowledgements

Authors would like to thank the Normandy Region for the financial support of this work.

References

[1] R. Giot, M. El-Abed, and C. Rosenberger, "Keystroke dynamics overview," in *Biometrics / Book 1*, D. J. Yang, Ed. InTech, Jul. 2011, vol. 1, ch. 8, pp. 157–182. [Online]. Available: http://www.intechopen.com/articles/show/title/keystroke-dynamics-overview

[2] S. Mondal and P. Bours, "A study on continuous authentication using a combination of keystroke and mouse biometrics," *Neurocomputing*, vol. 230, pp. 1–22, 2017.

[3] B. Li, H. Sun, Y. Gao, V. V. Phoha, and Z. Jin, "Enhanced free-text keystroke continuous authentication based on dynamics of wrist motion," in *Information Forensics and Security (WIFS), 2017 IEEE Workshop on*. IEEE, 2017, pp. 1–6.

[4] V. Monaco, "Public keystroke dynamics datasets," 2018. [Online]. Available: http://www.vmonaco.com/keystroke-datasets

[5] R. Giot, B. Dorizzi, and C. Rosenberger, "A review on the public benchmark databases for static keystroke dynamics," *Computers & Security*, vol. 55, pp. 46–61, 2015.

[6] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua, "Labeled faces in the wild: A survey," in *Advances in face detection and facial image analysis*. Springer, 2016, pp. 189–248.

[7] R. Cappelli, D. Maio, and D. Maltoni, "Sfinge: an approach to synthetic fingerprint generation," in *International Workshop on Biometric Technologies (BT2004)*, 2004, pp. 147–154.

[8] R. Giot, M. El-Abed, B. Hemery, and C. Rosenberger, "Unconstrained keystroke dynamics authentication with shared secret," *Computers & Security*, vol. 30, no. 6-7, pp. 427–445, Sep. 2011.

[9] K. S. Killourhy and R. A. Maxion, "Should security researchers experiment more and draw more inferences?" in *4th Workshop on Cyber Security Experimentation and Test (CSET'11)*, Aug. 2011, pp. 1–8.
[10] D. Migdal and C. Rosenberger, "Analysis of Keystroke Dynamics For the Generation of Synthetic Datasets," in *CyberWorlds*, Singapour, Singapore, Oct. 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01862152
[11] R. Gaines, W. Lisowski, S. Press, and N. Shapiro, "Authentication by keystroke timing: some preliminary results," Rand Corporation, Tech. Rep. R-2567-NSF, May 1980.
[12] R. Spillane, "Keyboard apparatus for personal identification," IBM Technical Disclosure Bulletin, Apr. 1975.
[13] D. Umphress and G. Williams, "Identity verification through keyboard characteristics," *Internat. J. Man Machine Studies*, vol. 23, pp. 263–273, 1985.
[14] F. Monrose and A. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation Computer Syststems*, vol. 16, no. 4, pp. 351–359, 2000.
[15] K. Revett, F. Gorunescu, M. Gorunescu, M. Ene, S. d. M. Tenreiro, and H. M. D. Santos, "A machine learning approach to keystroke dynamics based user authentication," *International Journal of Electronic Security and Digital Forensics*, vol. 1, pp. 55–70, 2007.
[16] H. Lee and S. Cho, "Retraining a keystroke dynamics-based authenticator with impostor patterns," *Computers & Security*, vol. 26, no. 4, pp. 300–310, 2007.
[17] S. Bleha, C. Slivinsky, and B. Hussien, "Computer-access security systems using keystroke dynamics," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 12, pp. 1217–1222, 1990.
[18] S. Hocquet, J.-Y. Ramel, and H. Cardot, "User classification for keystroke dynamics authentication," in *The Sixth International Conference on Biometrics (ICB2007)*, 2007, pp. 531–539.
[19] F. Monrose and Rubin, "Authentication via keystroke dynamics," in *Proceedings of the 4th ACM conference on Computer and communications security*, 1997, pp. 48–56.
[20] A. Teoh, D. Ngo, and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenised random number," *Pattern recognition*, vol. 40, 2004.
[21] R. Sugden, T. Smith, and R. Jones, "Cochran's rule for simple random sampling," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 62, no. 4, pp. 787–793, 2000.
[22] R. Giot, M. El-Abed, and C. Rosenberger, "Greyc keystroke: a benchmark for keystroke dynamics biometric systems," in *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS 2009)*, 2009, pp. 1–6.
[23] R. Giot, M. E. Abed, and C. Rosenberger, "Web-based benchmark for keystroke dynamics biometric systems: a statistical analysis," in *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012 Eighth International Conference on.* IEEE, 2012, pp. 11–15.
[24] K. S. Killourhy and R. A. Maxion, "Comparing anomaly detectors for keystroke dynamics," in *Proc. of the 39th Ann. Int. Conf. on Dependable Systems and Networks*, 2009, pp. 125–134.
[25] D. Stefan, X. Shu, and D. D. Yao, "Robustness of keystroke-dynamics based biometrics against synthetic forgeries," *computers & security*, vol. 31, no. 1, pp. 109–121, 2012.
[26] D. Stefan and D. Yao, "Keystroke-dynamics authentication against synthetic forgeries," in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010 6th International Conference on.* IEEE, 2010, pp. 1–8.
[27] A. Iorliam, A. Ho, N. Poh, S. Tirunagari, and P. Bours, "Data forensic techniques using benford's law and zipf's law for keystroke dynamics," *3rd International Workshop on Biometrics and Forensics, IWBF 2015*, 05 2015.
[28] L. KPFRS, "On lines and planes of closest fit to systems of points in space," in *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (SIGMOD)*, 1901.