HAL
open science

# Web-Based Benchmark for Keystroke Dynamics Biometric Systems: A Statistical Analysis

Romain Giot, Mohamad El-Abed, Christophe Rosenberger

## To cite this version:

Romain Giot, Mohamad El-Abed, Christophe Rosenberger. Web-Based Benchmark for Keystroke Dynamics Biometric Systems: A Statistical Analysis. The Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2012), Jul 2012, Piraeus, Greece. pp.1-5. hal-00714251

**HAL Id: hal-00714251**

**https://hal.science/hal-00714251**

Submitted on 3 Jul 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Web-Based Benchmark for Keystroke Dynamics Biometric Systems: A Statistical Analysis

Romain Giot and Mohamad El-Abed and Christophe Rosenberger
Université de Caen, UMR 6072 GREYC
ENSICAEN, UMR 6072 GREYC
CNRS, UMR 6072 GREYC
Email: {romain.giot,mohamad.elabed,christophe.rosenberger}@ensicaen.fr

***Abstract*—Most keystroke dynamics studies have been evaluated using a specific kind of dataset in which users type an imposed login and password. Moreover, these studies are optimistics since most of them use different acquisition protocols, private datasets, controlled environment, etc. In order to enhance the accuracy of keystroke dynamics' performance, the main contribution of this paper is twofold. First, we provide a new kind of dataset in which users have typed both an imposed and a chosen pairs of logins and passwords. In addition, the keystroke dynamics samples are collected in a web-based uncontrolled environment (OS, keyboards, browser, etc.). Such kind of dataset is important since it provides us more realistic results of keystroke dynamics' performance in comparison to the literature (controlled environment, etc.). Second, we present a statistical analysis of well known assertions such as the relationship between performance and password size, impact of fusion schemes on system overall performance, and others such as the relationship between performance and entropy. We put into obviousness in this paper some new results on keystroke dynamics in realistic conditions.**

## I. Introduction

Keystroke dynamics allows users to be recognized based on their way of typing on a keyboard. This is a behavioral modality which has been first experimented in the eighties [3]. It is always an interesting subject of research, as it is a low cost two factors authentication approach. Most consequent keystroke dynamics studies have been evaluated on datasets where users typed the same fixed string [11], [4], [1], while very few of them used different strings for each users [2], [9].

For this study, we want to be placed in the following context. We want to use a web-based application with an authentication system based on static keystroke dynamics. Some studies have already been done on web-based keystroke dynamics [14], [2], [13], [10], but none of them provided the used dataset and their experimental protocols were really differents. Some worked with individual passwords [10], while other used the same strings for each user [14]. In our work, we statistically analyze the behavior of these two approaches.

There is a strong need of a large dataset. We provide a new dataset where users typed both an imposed pair of login and password, a chosen login (their usual one) and password (one chosen by themselves for the experiment). The aim of the dataset is to show the viability of using personal identifiers (*i.e.*, chosen login and password) in native web browser (*i.e.*, using no plug-in or extension of the web browser), because the most recent applications are web-based ones, and systems usually use different logins and passwords for each user. The contribution of this paper is to present this new dataset that is publicly available[1] for testing algorithms in an operational context and experiment keystroke dynamics on this dataset. It is the sole public dataset which satisfy these properties. Additionally, we analyze information provided in this dataset to answer operational questions such as those presented in section III, part B. The paper is organized as follows. Section II presents existing public datasets and the dataset built for the experiment. Section III presents the various experiments. Section IV presents their results. Section V concludes this paper and gives some perspectives.

## II. Public Keystroke Dynamics Datasets

In most studies, researchers use their own dataset which, most of the time, suffers of lack of number of users and sessions. Some keystroke dynamics databases are publicly available in the literature, but none of them provides different login and password for each user. In [4], several users have typed the passphrase "greyc laboratory" on two different keyboards on the same computer during several sessions. 100 users have provided at least 60 samples each on 5 different sessions spaced of one week (most of the time). This database contains the most number of users, but, the number of samples and sessions may be too small to track variability through time. In [11], several users have typed the password ".tie5Roanl" on a single computer during several sessions. 51 users have provided 400 samples each on 8 different sessions spaced of, at least, one day. This database contains a huge number of samples, but the time interval may be too small to track variability on a long period. These two databases are the only ones containing enough samples and users to give statistically significant results. Sadly, they mainly have been used by their own creators, and not by the community. Table I summarizes this information. Even if these two databases are interesting, they do not really fit requirements for realistic studies:

1) We want different logins and passwords per user. This is the most realistic scenario for keystroke dynamics. Real users use different logins and passwords.

[1] http://www.epaymentbiometrics.ensicaen.fr/

2) It is interesting to have different computers and keyboards to grow the variability of the samples (due to shape of keyboard, responsiveness of the computer, precision of the timer, ...).
3) The captures must have been done in a web browser (because nowadays, most of modern applications are available as web-based applications, and not desktop applications. Collecting samples from different browsers allows to track more variability due the browser itself (several browsers exist on all the operating systems).

We have created a web-based application which allows us to capture keystroke dynamics during several sessions. We think that with this dataset, researchers will have an interesting dataset providing a lot of variabilities due to the different factors presented before. The results would not be over optimistic as it may be the case with actual ones. The next section presents the experiment.

## III. Experiment

### A. Proposed Dataset

*1) Acquisition Protocol:* Each week, we have sent an email to the students of our school of engineering and some colleagues of our lab. It asks them to realize the session capture of the week. The first one explains the aim of their participation. During the first session, each user has to choose its own login (we asked them to use the login of their school account, but they have not all respected that), and password. We expect them to type their login as they are used to. So, each user chooses when he/she wants to do the session without any obligation or pressure. Participants have not been rewarded. A session is composed of three different steps. Each step consists in typing several times a pair of login and password. The user interface presents two input fields: one for the login, and one for the password. No typing correction is allowed: if a user presses backspace, the input field is cleared, and the user must type its text from scratch. The password and login the user have to type are displayed near the form and are displayed in a pop-up box at the start of each step[2].

A progression bar is shown at the bottom of the screen. It indicates how many inputs are yet needed to complete the session. As the interface is displayed in a web browser, it is written with Javascript, html and css. Most studies of keystroke dynamics which work on a web browser are written in Java [14], [2], [13]. We have not chosen this language because it imposes the user to install a Java plugin for its browser. For each key event (key press and key release), the timing information is captured through the $timeStamp$ value of the event[3]. We do not track timing information of the key having a code inferior to 48 (except tab, shift, space, ctrl, altgr, and keycode 0 which seems to be present for some punctuated keys) as all as right and left Windows key and keys from F1 to F12. The three acquisition steps are the following ones:

[2]No screenshots for lack of space
[3]https://developer.mozilla.org/En/DOM/Event.timeStamp

TABLE I
Summary of the keystroke dynamics databases. Prop. refers to the proposed dataset. Same means each user types the same password (so impostor are used to type the same password than the user), while different means each user types a different password (and impostors are not used to type it).

| | Size | | Login/password | |
|---|---|---|---|---|
| **Study** | **# users** | **# samples** | **same** | **different** |
| [4] | 100 | 60000 | ✓ | |
| [11] | 51 | 20400 | ✓ | |
| Prop. | 83 | 5185 + 5754/5439 | ✓ | ✓ |

- *Step 1*. Ten inputs of a pair of imposed login and password. This allows us to capture exactly the same thing for all the users as in [4], [11].
- *Step 2*. Ten inputs of the chosen pair of login and password of the user. This simulates the authentication of the user on a system as in [9], [15].
- *Step 3*. For two other users, five inputs of the selected pair of login and password. This allows us to capture ten impostor samples belonging to two other users.

*2) Presentation of the Obtained Dataset:* As the participation was only based on the goodwill of the users, very few of them participated to the study or to each session. That is why only 83 users have participated to the study, whereas the emails were sent to more than 300 students. Sessions were not always done completely. Users have provided a total of 5185 genuine samples (pair of login, password typed by its owner), 5754 impostor samples (pair of login, password typed by a user different of its owner), and 5439 imposed samples (pair of imposed login and password). Most participants are between 20 and 24 years old (mainly students in computer science, chemistry and electronics). Most users are males, which can be problematic to generalize results if males and females have different typing behaviors [6]. Most users have more than 20 impostor samples which allows to obtain good information on False Match Rate. The number of genuine and imposed samples per user is not equally set, there are several users who have provided less than 40 genuine samples. It is difficult to obtain a large keystroke dynamics database with enough quantity of samples per user (which may explain why there are so few publicly available keystroke dynamics databases, and why, most of the time the number of users is relatively small).

Although the obtained dataset is not the largest in terms of number of users involved, it is the only public keystroke dynamics providing different logins and passwords per users. Thus, it is the more realistic one.

### B. Experimental Protocol

We want to answer to the following questions:

1) Does keystroke dynamics' performance behaves similarly on a dataset built with imposed strings, against a dataset built with strings chosen by users themselves? This question is interesting because all public datasets do use imposed strings.

2) Which approach (individual or global threshold) gives better results in terms of performance? This question is interesting, because both approaches are used in the literature and avoid an easy performance comparison.
3) Which features must be used in a score fusion system, in order to improve results?
4) Are password length, entropy and complexity correlated with the recognition performance? This question is interesting, because it has not been studied in the literature (probably because all the passwords are identical). It can give information of how the password must or must not be chosen by the user, in order to strengthen the system.

We have run several experiments using the different subsets (chosen/imposed) to analyze the performance of keystroke dynamics authentication methods. The Equal Error Rate (EER) is individually computed for each user (*i.e.*, EER is computed with the comparison scores of its test samples against its model and real impostors' test samples against its model), and, its averaged value (among all users) is presented under $EER_i$. $EER_g$ presents the EER with the same threshold for all the users (EER is computed with a global intra-scores and inter-scores set). These are two common ways of presenting keystroke dynamics results, but no study analyzed the performance difference between the two approaches. Authentication test is done with only one capture, we do not give another chance if it fails (several tries is a another common way of presenting results [9], [12]). Training is done with 20 samples (two sessions), and testing is done with the remaining samples. We only keep users having at least 20 testing samples (at least two sessions per user). So, we work with users having used the system during at least 4 sessions. As users may use different keys for typing their login or password, the number of pressed characters may be different.

*1) Distance Computation:* In this paper, we have tested only one score computing method. It is based on a Gaussian distribution assumption of the features [8]. Each user provides $N$ samples to build its template. A sample $\mathbf{x}$ is a $n$-dimension vector. The template $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$ is composed of the mean vector $\boldsymbol{\mu}$ and the standard deviation vector $\boldsymbol{\sigma}$ among these features. The distance between sample $\mathbf{x}$ and template $\boldsymbol{\theta}$ is computed using the following formulation:

$$d(\mathbf{x}, \boldsymbol{\theta}) = 1 - \frac{1}{n} \sum_{i}^{n} exp\left(-\frac{|x_i - \mu_i|}{\sigma_i}\right) \tag{1}$$

If a query is not of the same size (different combinations of keys, or use of the mouse to select and erase text, can be the reason of this difference) of the template, we return a distance of 1 (the worst score, 0 being the best).

*2) Statistical Validation:* In order to verify the various answers, we use the Kruskal-Wallis (KW) test [7]. It is a non-parametric (distribution free) test, which is used to decide whether $K$ independent samples are from the same population. More generally speaking, it is used to test two hypothesis: the null hypothesis ($H_0 : \mu_1 = \mu_2 = ... = \mu_k$) assumes that samples have been generated from the same population (*i.e.*, equal population means) against the alternative hypothesis ($H_1 : \mu_i \neq \mu_j$) which assumes that there is a statistically significant difference between at least two of the subgroups. The decision criterion is then derived from the estimated $p - value$ as depicted in equation 2.

$$\begin{cases} p - value \geq 0.05 & accept\ H_0 \\ otherwise & reject\ H_0 \end{cases} \tag{2}$$

*3) Simple Feature Authentication:* We test several features: "rp" (latency between the release of a key, and the pressure of next one), "rr" (latency between the release of two successive keys), "pp" (latency between the pressure of two successive keys) and "pr" (duration of pressure of one key) for both login and password individually, and for each kind of datasets (imposed login/password and chosen login/password). This gives us $4 * 2 * 2 = 16$[4] different experiments.

*4) Score Fusion:* Although feature fusion is often used in keystroke dynamics [5], [1], [11], we have chosen to use a score fusion system [16], [8]. A user sample is composed of several sub-samples (one per kind of extracted features): $\mathbf{x} = (\mathbf{x}^l_{rr}, \mathbf{x}^l_{rp}, \mathbf{x}^l_{pr}, \mathbf{x}^l_{pp}, \mathbf{x}^p_{rr}, \mathbf{x}^p_{rp}, \mathbf{x}^p_{pr}, \mathbf{x}^p_{pp})$ (superscript $l$ and $p$ respectively represent login and password). A template is built for each kind of sample extracted features: $\boldsymbol{\theta} = (\boldsymbol{\theta}^l_{rr}, \boldsymbol{\theta}^l_{rp}, \boldsymbol{\theta}^l_{pr}, \boldsymbol{\theta}^l_{pp}, \boldsymbol{\theta}^p_{rr}, \boldsymbol{\theta}^p_{rp}, \boldsymbol{\theta}^p_{pr}, \boldsymbol{\theta}^p_{pp})$. In this case, the same keystroke dynamics method is used for each extracted features. The final score is the mean (without score normalization) of each of these scores (one for each selected feature), so the fusion rule is: $s_f = \frac{1}{m}\sum_i^m s_i$ with $s_f$ the new fused score, and $s_i$ the comparison score of system $i$ (using features of type $i$), when using $m$ different systems. As an illustration, figure 1 presents the score fusion architecture when pp and rp times from login and password are used.

*5) Study of the performance depending on password size and complexity:* For this experiment, we use the score fusion of all the features of the password. We would like to verify if keystroke dynamics' performance depends on the complexity, the size, or the entropy of the password. Towards this goal, we use again the KW test. For the complexity computation, we have used an existing algorithm which is often used in web applications as depicted in figure 2. The entropy quantifies the expected value of the information contained in the password $\mathbf{p}$. The password contains $P$ unique characters $\{c_1, \cdots, c_P\}$. $p(c_i)$ is the probability of appearance of the character $c_i$ in the password $\mathbf{P}$. Entropy is computed as following:

$$H(\mathbf{P}) = -\sum_{i=1}^{P} p(c_i) \log_2(c_i) \tag{3}$$

## IV. Results

In this section, we present the results of the experiments previously presented. Even if the number of users in the dataset is quite important, only 48 of them provided enough samples to be used in the experiments (note that most of keystroke dynamics studies even use fewer individuals).

[4]number of features * login or password * imposed or chosen

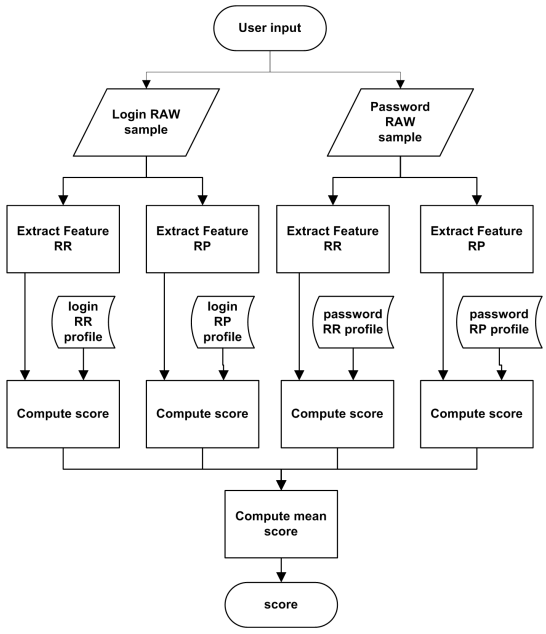


Fig. 1. Score fusion scheme using only RR and RP features.

**Require:** $PASSWORD$
  $SIZE \leftarrow$ number of char of $PASSWORD$
  $LOW$, $UPP$, $NUM$, $OTH \leftarrow$ (0, 0, 0, 0)
  **for** $i = 1$ **to** $SIZE$ **do**
    $CHAR \leftarrow PASSWORD[i]$
    **if** $CHAR \geq$ 'a' and $CHAR \leq$ 'z' **then**
      $SCORE \leftarrow SCORE + 1$
      $LOW \leftarrow 1$
    **else if** $CHAR \geq$ 'A' and $CHAR \leq$ 'Z' **then**
      $SCORE \leftarrow SCORE + 2$
      $UPP \leftarrow 2$
    **else if** $CHAR \geq$ '0' and $CHAR \leq$ '9' **then**
      $SCORE \leftarrow SCORE + 3$
      $NUM \leftarrow 3$
    **else**
      $SCORE \leftarrow SCORE + 5$
      $OTH \leftarrow 5$
    **end if**
  **end for**
  $COEFF \leftarrow SCORE/SIZE$
  $DIVERSITY \leftarrow LOW + UPP + NUM + OTH$
  **return** $COEFF * DIVERSITY * SIZE$

Fig. 2. Compute the complexity of a password (in terme of password security and not typing difficulty).

### A. Simple Feature Authentication

Table II presents the results of the simple feature authentication experiments. Using the KW test, we find that there is no significant difference ($p - value = 0.68$) of performance between the chosen and the imposed datasets. This was a surprising result since users are more likely to type their own *login* and *password* than an imposed ones. We may obtain these results because, even if users have chosen their password, it is not their real password they type several times per day. Using the chosen and imposed datasets, we found that the performance of individual approach outperformed ($p - value << 0.05$) the global approach.

Using both datasets, we found that the *login* outperformed

TABLE II
AUTHENTICATION RESULTS, FOR DIFFERENT EXTRACTED FEATURES FOR EACH KIND OF TEXT. THE BEST RESULT OF EACH LINE IS IN BOLD. THE BEST RESULT OF EACH COLUMN IS UNDERLINED.

| | | Chosen dataset | | Imposed dataset | |
|---|---|---|---|---|---|
| **Type** | **Field** | **EERi** | **EERg** | **EERi** | **EERg** |
| *pr* | *login* | 26.50% | 28.81% | **19.79%** | 21.90% |
| *rp* | *login* | 21.25% | 25.91% | **14.84%** | 20.91% |
| *rr* | *login* | 18.00% | 24.01% | **10.00%** | 15.21% |
| *pp* | *login* | 19.27% | 24.48% | **11.86%** | 18.63% |
| *pr* | *pwd* | **22.21%** | 25.30% | 23.21% | 27.08% |
| *rp* | *pwd* | **18.51%** | 21.56% | 26.63% | 30.54% |
| *rr* | *pwd* | **16.95%** | 19.90% | 22.17% | 27.00% |
| *pp* | *pwd* | **16.45%** | 20.64% | 24.02% | 29.00% |
| | **Mean** | **19.89%** | 22.57% | 19.65% | 23.78% |

($p - values$ are below to 0.05, respectively) the *password* information. This result was also attended since the used logins are much more easier than passwords (hence, users' way of typing the imposed logins would be much more stable than typing the imposed passwords).

A study of the fusion of both information is given in the next section.

### B. Score Fusion

Table III presents the performance of the score fusion when using different features on the chosen dataset. We can see that we can improve the performance by fusing the comparison scores of the right extracted features.

In order to see which feature (or combination of features) gives the best performance result, we use the KW test over the seven sets which combine login with password: the EER values related to the use of "pr", "rr", "pp", "pr" & "rr", "pr" & "pp", "rr" & "pp", and "pr" & "rr" & "pp" informations, respectively. We found that the worst result (with $p - value$ below to 0.05) is obtained by using the "pr" information (the duration of the press of a key which is the most often used feature). We have not selected "rp" which is the worst feature in Table II. The use of all the features ("pr" & "rr" & "pp") outperformed ($p - values$ below to 0.05) the use of "pr" and "pp" informations, while there were no significant performance difference between all the features and "rr", "pr" & "rr", "pr" & "pp", "rr" & "pp" informations. We conclude, that the use of all the features (even if they are redundant) may improve the performance. This generalizes results obtained on a single fixed text [1].

### C. Study of the performance depending on password size, entropy and complexity

Using the KW test, we find that the size ($p - value = 0.019$) and the entropy ($p - value = 0.0062$) of the used passwords have a significant impact on system performance, while there was no impact ($p - value = 0.12$) according to the complexity algorithm. More generally speaking, the average EER value is increased from 10.03% (for users having more than 8 characters) to 15.85% (for the others). Using the entropy information, the average EER value is increased from 10.01% (for those whom the entropy of their passwords is more than

TABLE III
AUTHENTICATION RESULTS, WHEN USING VARIOUS FEATURE FUSION. THE BEST RESULT OF EACH LINE IS IN BOLD. THE BEST OVERALL RESULT IS UNDERLINED.

| Login | | | Password | | | $EER_i$ | $EER_g$ |
|---|---|---|---|---|---|---|---|
| pr | rr | pp | pr | rr | pp | | |
| Login only | | | | | | | |
| ✓ | | ✓ | | | | **15.99%** | 22.42% |
| ✓ | ✓ | ✓ | | | | **14.36%** | 20.72% |
| | ✓ | ✓ | | | | **16.57%** | 23.07% |
| Password only | | | | | | | |
| | | | ✓ | | ✓ | **14.24%** | 17.75% |
| | | | ✓ | ✓ | ✓ | **12.52%** | 16.74% |
| | | | | ✓ | ✓ | **15.36%** | 19.04% |
| Login and password | | | | | | | |
| ✓ | | | ✓ | | | **18.92%** | 23.51% |
| | ✓ | | | ✓ | | **12.37%** | 15.63% |
| | | ✓ | | | ✓ | **11.45%** | 16.15% |
| ✓ | ✓ | | ✓ | ✓ | | **10.25%** | 15.85% |
| ✓ | | ✓ | ✓ | | ✓ | **9.4%** | 19.96% |
| | ✓ | ✓ | | ✓ | ✓ | **10.71%** | 14.95% |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | **08.87%** | 14.08% |
| | | | | **Mean** | | **13.15%** | 18.45% |

2.7) to 16.09% (for the others). The average method is used to fix both thresholds (8 and 2.7). It would be important in the future then to investigate more the way of choosing passwords, which may be considered as a quality measure in keystroke dynamics research field. Such quality information would be useful during the enrollment process in order to enhance the system overall performance.

## V. CONCLUSION

We have presented a new publicly available dataset for keystroke dynamics. This dataset is composed of several users who have a different login and password. We think it is the most realistic keystroke dynamics dataset which is publicly available. We have statistically verified that: (a) presenting EER computed with an individual threshold, gives better result than computing the EER with a global threshold (which explains why a lot of keystroke dynamics studies use this method), (b) using logins gives better performance than using passwords, (c) using all features during the fusion improves the performance, (d) the size and the entropy of the password have an impact on the performance.

Keystroke dynamics is an interesting modality, but, it requires strict conditions during acquisition to avoid capture of noisy samples. This may imply an eduction of the user. As its performance decreases a lot with time, it is necessary to track the time variability into account which will be the next work on this dataset.

## REFERENCES


[1] K. S. Balagani, V. V. Phoha, A. Ray, and S. Phoha. On the discriminability of keystroke feature vectors used in fixed text keystroke authentication. *Pattern Recognition Letters*, 32(7):1070 – 1080, 2011.

[2] S. Cho, D. H. Han, Chigeun Han, and H.-I. Kim. Web-based keystroke dynamics identity verification using neural network. *Journal of organizational computing and electronic commerce*, 10(4):295–307, 2000.

[3] R. Gaines, W. Lisowski, S. Press, and N. Shapiro. Authentication by keystroke timing: some preliminary results. Technical report, Rand Corporation, 1980.

[4] 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)*, pages 1–6, 2009.

[5] R. Giot, M. El-Abed, and C. Rosenberger. Keystroke dynamics with low constraints svm based passphrase enrollment. In *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS 2009)*, pages 1–6, Washington, District of Columbia, USA, Sept. 2009. IEEE Computer Society.

[6] R. Giot and C. Rosenberger. A new soft biometric approach for keystroke dynamics based on gender recognition. *International Journal of Information Technology and Management (IJITM). Special Issue on : "Advances and Trends in Biometrics by Dr Lidong Wang*, 11(1/2):35–49, 2012. Impact Factor : 0.727.

[7] J. J. Higgins. An introduction to modern nonparametric statistics. *The American Statistician*, 2003.

[8] S. Hocquet, J.-Y. Ramel, and H. Cardot. Estimation of user specific parameters in one-class problems. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 449–452. IEEE Computer Society, 2006.

[9] D. Hosseinzadeh and S. Krishnan. Gaussian mixture modeling of keystroke patterns for biometric applications. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(6):816–826, 2008.

[10] C.-H. Jiang, S. Shieh, and J.-C. Liu. Keystroke statistical learning model for web authentication. In *ASIACCS '07: Proceedings of the 2nd ACM symposium on Information, computer and communications security*, pages 359–361, 2007.

[11] K. Killourhy and R. Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. In *IEEE/IFIP International Conference on Dependable Systems & Networks, 2009. DSN'09*, pages 125–134, 2009.

[12] K. Killourhy and R. Maxion. Keystroke biometrics with number-pad input. In *Proceedings of the 40th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN-2010)*, pages 201–210, 2010.

[13] G. Kofi Gagbla. Applying keystroke dynamics for personal authentication. Master's thesis, Department of Telecommunication and Signal Processing Blekinge Insitute of Technology, 2005.

[14] K. Revett, S. T. de Magalhães, and H. M. D. Santos. Developing a keystroke dynamics based agent using rough sets. In *IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology.*, 2005.

[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*, 1:55–70, 2007.

[16] P. S. Teh, A. B. J. Teoh, T. S. Ong, and H. F. Neo. Statistical fusion approach on keystroke dynamics. *Signal-Image Technologies and Internet-Based System, International IEEE Conference on*, 0:918–923, 2007.