Analysis of Template Update Strategies for Keystroke Dynamics

Romain Giot, Bernadette Dorizzi, Christophe Rosenberger

Analysis of Template Update Strategies for Keystroke Dynamics

Romain Giot*, Bernadette Dorizzi† and Christophe Rosenberger*
*GREYC - CNRS, Université de CAEN, ENSICAEN
Caen, France
Email: {romain.giot,christophe.rosenberger}@greyc.ensicaen.fr
†Institut Télécom; Télécom SudParis
Email: bernadette.dorizzi@it-sudparis.eu

***Abstract*—Keystroke dynamics is a behavioral biometrics showing a degradation of performance when used over time. This is due to the fact that the user improves his/her way of typing while using the system, therefore the test samples may be different from the initial template computed at an earlier stage. One way to bypass this problem is to use template update mechanisms. We propose in this work, new semi-supervised update mechanisms, inspired from known supervised ones. These schemes rely on the choice of two thresholds (an acceptance threshold and an update threshold) which are fixed manually depending on the performance of the system and the level of tolerance in possible inclusion of impostor data in the update template. We also propose a new evaluation scheme for update mechanisms, taking into account performance evolution over several time-sessions. Our results show an improvement of 50% in the supervised scheme and of 45% in the semi-supervised one with a configuration of the parameters chosen so that we do not accept many erroneous data.**

I. Introduction

The aim of biometrics is to verify the identity of individuals using different personal characteristics (mainly morphological and behavioral). All biometric modalities are subject to an intra-class variability (leading to a high False Non Match Rate) due to several facts: acquisition of a reflex for behavioral biometrics [1], use of different sensors for enrollment and verification acquisitions, noise due to the acquisition environment (humidity, lightning, stress...) or enrollment procedure that is inefficient. These problems can be by-passed by using different methods. It is possible to operate a multiple tries authentication scheme for modalities having a high False Non Match Rate (in [2], two verifications are done before rejecting a user). We can use multibiometric system [3] in order to improve the performance of each single system. We can also choose features that are more stable over time (for example, in on-line signature recognition, pen inclination is subject to a high time variability but not the coordinates of the trajectory [4]). Another possibility is to use a template update procedure to reduce the time variability [5], [6]. This is the approach we have chosen to explore in this paper.

For the main modalities (face or fingerprint, for example), the model (or template) of a user is constituted of a single sample, or a gallery containing several samples. The verification of a user is done by computing a comparison score between the query sample and the samples stored in the model. When several samples are available in the user gallery, one has to use a score merging function (max, $mean$, ...) to produce the final comparison score. In this paper, we are mainly interested to the keystroke dynamics modality [7], [8]. In this modality, the user identity relies on the way he types a fixed string on a keyboard. In this case, one uses several samples of the user keystroke sequence to build a model of the user. This model corresponds to several moments which estimate the density function of the keystroke dynamics of the user ($mean$, $standard\ deviation$, $median$, ...). For this reason, many samples are needed to build the model.

We can observe two kinds of template update procedures in the literature. In the so-called *supervised* methods, an operator adds new samples in the model of a user over the time. This procedure insures that only *genuine* samples are added and easily allows improving the system's performance. However, such procedure is expensive in time and money because:

- we need to enrol the users during several sessions, and they need to be cooperative;
- we need an operator to verify the respect of the enrollment procedure.

The other way is named *semi-supervised* [6] and consists in using samples captured during the use of the system and considered by the system as highly genuine. In this case, we add samples with an high probability of belonging to the user. However, an impostor sample with a high probability of being a genuine sample can be appended to the model. The more the classifier does error, the more we can add impostor samples in the model.

Using template update procedures is interesting for modalities where the biometric data evolves a lot with time. Figure 1 presents the evolution of a keystroke typing pattern depending on time. It has been computed as following. For each user, we have computed, in chronological order, the dispersion measure of the samples. At each time i, the reference is composed of the N previous samples. We have set in Figure 1(a) N to 10, in Figure 1(b) N to 25 and

in Figure 1(c) N to 50 (the number of samples captured during a session). The graphics present the mean value and the standard deviation of this dispersion measure for all the users. For the iteration m (the first iteration starts at the value N), we compute the mean value of its test window of size N: $\mu_m = \frac{1}{N}\sum_{i=m-N}^{m-1} s_i$ (s_i is the ith sample). Then, for each sample in the window, we compute its difference against the mean: $d_i = \mu - s_i,\ \forall i \in [m-N; m-1]$. The dispersion measure is the mean of the euclidean norms of the d_i: $dispersion = \frac{1}{N}\sum \|d_i\|_2$.

From these figures, we can conclude that, the more the user types the password, the more he types it consistently (the mean value of the dispersion measure decreases progressively for all the window sizes). This behaviour is globally the same for all the users (the standard deviation also decreases). There is a deviation from the initial samples. This facts confirms the interest of using template update mechanisms in order to substitute the old unstable samples by more recent, and, therefore, more stable ones. Our review of the literature has led us to the conclusion that there are at this moment no commonly adopted protocol allowing to evaluate the quality of an update mechanism when used on several time sessions. Therefore our first contribution, in this paper, is to propose a protocol for evaluating keystroke dynamics template update systems. This protocol could also be used for other modalities (especially for behavioral ones) for which databases captured on several sessions are availables. In a second time, we will propose a new semi-supervised keystroke dynamics template update mechanism and its evaluation using our new framework.

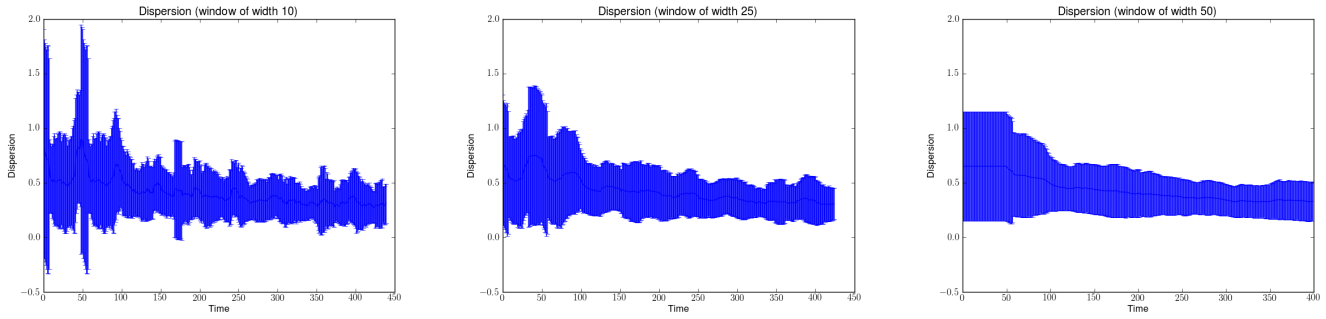
This paper is organized as follows. Section II presents previous works on template update in biometrics in general. Section III presents our new evaluation protocol and our new scheme for semi-supervised keystroke dynamics template update. Section IV presents the results which are discussed in Section V and Section VI concludes the paper.

II. Background

This section presents previous works in the literature of template update mechanisms for biometrics. Different template update strategies have been presented in [5]. These strategies have been proposed by keeping in mind the necessity to reduce intra-class variability. The problem of template update is presented as a problem of cache replacement strategies. These replacement strategies are *"First In First Out"*, *"Least Frequently Used"* and *"Least Recently Used"*. These names are self expressive and do not need more explanations. Authors in [5] also presented an extended replacement algorithm which uses the relevance of each sample in the gallery for selecting the old sample to be replaced by the newly authenticated test sample. This relevance depends on the age of the sample and its ability to accept test samples. These methods are relevant for modalities (like face or fingerprint) where the matching of the test sample is done with each element of the gallery, but they are not well adapted to dynamic modalities like on-line signature or keystroke dynamics (for example "Least Frequently Used" scheme is a nonsense in keystroke dynamics (as well as in any system which does not use nearest neighbour distance computation)), Supervised update in presented in [9]. Three different strategies are presented: model adaptation, score adaptation and compound which combines the two previous procedures. Model adaptation is used to provide user specific model and score adaptation is used to normalize scores differently depending on the acquisition conditions (in their study, they know they have three different condition acquisitions). Adaptive biometric systems can automatically update the classifier as well as its hyper-parameters. In [10], for video based face recognition and supervised learning, a fuzzy ARTMAP neural network classifier (which is able to be used for incremental learning) is used in conjunction to a dynamic particle swarm optimization system (which allows to update the hyper-parameters of the classifier) and a long term memory (which allows to not forget old samples). In [6], the authors present a complete overview of adaptive biometrics as well as a template update approach based on self-training and co-training (in the multibiometric case). Tested samples labeled with a high genuine probability are appended to the template. The template update procedure is done on a batch way. We can find an overview of template update studies for biometric systems in [11]. In this paper we can see that most studies are related to face and fingerprint recognition systems. The number of samples per user is not really important.

Concerning the keystroke dynamics modality, Kang *et al.* presented two supervised template update methods (named "*growing window*" and "*moving window*") [12]. Growing window refers to a method which permanently appends the newest sample to the gallery. This way, the number of samples in the gallery grows with time and the model encodes a long term variability. Moving window refers to a method which permanently replaces the oldest sample of the gallery by the newest one. This way, the number of samples in the gallery remains constant and the method only catches local variability. Note that after each modification of the gallery, the model must be re-estimated. In [12], the term *"fixed window"* refers to the fact of not using template update. Authors in [12] have validated their experiments on a database of 21 users. They only used genuine attempt samples during the update procedure[1] and their results show that an improvement can be observed when using these template update solutions. We do not know if impostor samples have been tested before the template update or after (this is an important point, because template evolves and the evolution may add more impostor acceptation). Anyhow, this study shows that in a *supervised* environment, template update works quite well for keystroke dynamics. In [2], a semi-supervised template update method

[1]It is not explicitly written, but looking at figure 2 and 4 suggests they have only used genuine attempts



(a) Window of size 10 (b) Window of size 25 (c) Window of size 50

Figure 1. Evolution of the dispersion measure between sample t and samples $[t-N;t-1]$ with $N \in \{10, 25, 50\}$

is also used for keystroke. The update procedure is similar to the moving window. Authors show that using a template update procedure improves performances if the update is not systematically applied (because, in the other case, impostor patterns are added to the new template which progressively drifts from the real user way of typing). They do not give information on the presentation order of the patterns neither on the decision method used for taking the decision to update or not.

We can see from this state of the art that there are two main points to take into account when designing a template update procedure: (i) what is the template update procedure ? (ii) how do we take the decision of applying a template update procedure or not ? Note also that the experiment of the state of the art have mainly be done in a supervised way for keystroke dynamics and with databases collected on a short time span for most studies.

III. Proposed Method

We present in this section our proposition to the problem of adaptation for keystroke dynamics. The first contribution is to propose an evaluation protocol of template update mechanisms for dataset providing enough data of different sessions. The protocol allows analysing the performance at different time steps instead of evaluating the performance at only one time slot as done in most studies.

A. Evaluation Methodology

We suggest different scenarios for update procedures benchmarking.

1) Samples presentation: Depending on the biometric modality, we can have far more impostor samples than genuine user's ones. In order to truly test the adaptation performance, we must know several things:

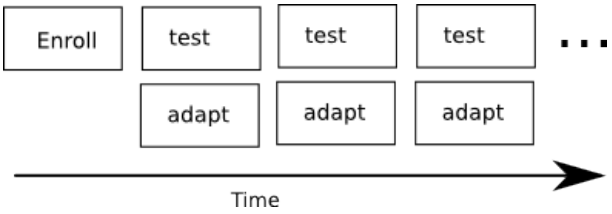
- What is the sample presentation order during the testing? If all the user's samples are presented before the impostors' ones, the updated template should be more robust (and give better performances) than if impostor's samples are interleaved with user's ones.
- What is the percentage of presented impostors samples? The more we present impostor's samples, the more the probability of including them in the template is high. It seems that this information is rarely presented in the literature.
- What is the skill of the impostors ? This information is particularly interesting for some modalities where impostor sample can be an imitation of the user sample (i.e., signature recognition where an impostor can try to imitate a signature). In our case, all impostor samples are random forgeries (every body types the same thing).

For these reasons, we argue that it could be useful to consider different scenarios which are of progressive difficulty. For each session and user, we build an ordered pool of test samples, which will be tested against the user template during this session. The difficulty of the problem is proportional to the ratio of genuine and impostor's data in the pool. Each pool is created accordingly to these points:

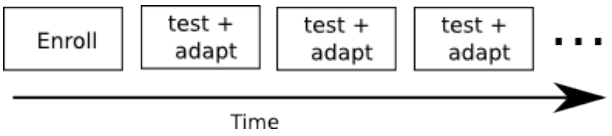
- the choice of using a genuine or impostor sample depends on a chosen impostor sample rate;
- the presentation order between genuine and impostor samples is totally random and linked to the probability of impostor sample appearance (i.e., with an impostor probability of α, for each position of the pool, the probability of having an impostor sample is of α and the probability of having a genuine sample is of $1-\alpha$);
- the test samples are presented in a chronological order in order to keep the bias due to the learning of password typing (both for genuine users and impostors).

This way, we do not favour any kind of sample (genuine or impostor) and it reflects a more realistic scenario.

In [13], authors have chosen to use three different ways of presenting samples (i) presenting impostors' samples first, (ii) presenting the samples randomly, and, (iii) presenting genuine samples first. But, they do not give more information on it, and we do not know if the chronological order is



(a) Separate update and test strategy. This is the offline way



(b) Joint adapt and test strategy. This is the online way

Figure 2. Evaluation strategies under several sessions. First session serves for enrollment. This is an extended version of the strategies presented in [14] using several sessions.

respected (which may be of less importance for their study on fingerprint, than in our on keystroke dynamics).

2) Sessions Implication: The first session serves for building the template of each user (we need several samples to compute the user's template for the keystroke dynamics).

As we work with a dataset providing several sessions, we want to keep the chronological relation of the sessions in our results. In [14], authors present a way of evaluating the performances with only two sessions (one for building the template, one for testing) without necessarily respecting the chronological order of the samples. We have extended this proposition to a database with several sessions. Figure 2 presents the evaluation strategies we have adopted.

We launch the template update procedure one session after the other. During one session, for each user, we test each sample of its pool of testing samples (which can be genuine or impostor samples) against his template. We store the obtained scores in two classes: the intra (comparison with a genuine sample) and inter (comparison with an impostor sample) scores. The aim of this sets is to compute the various error rate on-line. Depending on the comparison of this score against the adaptation threshold, we apply the template update procedure (so an impostor sample can be added to the enrolled samples). Choosing this update threshold is yet an open problem. The best choice is to choose a threshold which authorize very few impostor samples. This threshold could be computed thanks to a development pool and its FMR and FNMR values.

During the experiment, we catch two kinds of errors:

- The *on-line* error rate computed on the fly during a session. This error rate is a realistic one, because it corresponds to a real use of the system.
- An *off-line* error rate can be computed using the samples from the next session, evaluated with the templates obtained at the end of the present session and without using any template update mechanism. It is the most commonly used method in the literature.

Require: pools of samples computed
Require: update decision threshold
{Compute initial templates}
for all $user$ **do**
  Use samples of session 0
  Build template of $user$
end for
for $session = 1$ to last session - 1 **do**
  $intra1 \leftarrow []$
  $inter1 \leftarrow []$
  {Launch update procedure and compute performances inline}
  for all $user$ **do**
    while sample pool of $user$ not empty **do**
      $sample \leftarrow$ Get next from $pool$
      $score \leftarrow$ Compare $sample$ to $user$'s model
      if $sample$ belongs to $user$ **then**
        Store $score$ in $intra1$
      else
        Store $score$ in $inter2$
      end if
      if $score <= threshold$ **then**
        Apply template update for $user$ with $sample$
      end if
    end while
  end for
  $FMR1, FNMR1 \leftarrow$ Get internal error rate
  {Compute performances after wise}
  $intra2 \leftarrow []$
  $inter2 \leftarrow []$
  for all $user$ **do**
    Use $session + 1$ as test session
    Compute $intra$ scores and append them to $intra2$
    Compute $inter$ scores and append them to $inter2$
  end for
  $FMR2, FNMR2 \leftarrow$ Get hypothetic error rate
end for
Display $FMR, FNMR$ depending on session

Figure 3. Algorithmic view of the template update algorithm and its evaluation.

In this case, the error rate we are tracking is the couple of False Match Rate (FMR) and False Non Match Rate (FNMR) of each session for a given acceptance threshold and the Error Equal Rate (EER) (for global performances). The EER corresponds is the intersection point when FNMR and FMR are plotted on a graph. Figure 3 presents a summary of the template update evaluation algorithm, when several sessions are considered.

As we have already said before, the difficulty of a scenario lies in the ratio of impostors samples tested against the

template (it seems this point is not considered in the literature):

- Very few impostors samples are tested against the template: it is an easy scenario where they are few attacks.
- Same quantity of impostors samples than genuine samples tested against the template: this is an averaged scenario.
- Many impostors samples are tested against the template: it is a hard scenario where more attacks than usage occur.

The order of sample presentation is totally random, for this reason, it is necessary to repeat the operations several times. Therefore, the final results corresponds to averaged results on several runs of the evaluation process.

B. Dataset and Description of our Keystroke Dynamics System

For the keystroke dynamics modality, very few public datasets with several sessions are available. This is really problematic, because we know that this modality is subject to high intra-class variability and that the data are changing continuously with time (see figure 1). Our own dataset consists of five sessions [7] for one hundred users. Each user typed the password “greyc laboratory" on two different keyboards on the same machine twelve times per session during five sessions. For most of the users, the sessions are spaced of, at least, one week. Even if the number of sessions is far larger than the number of sessions of common fingerprint databases, we find its specifications insufficient for such an experiment (remember we want to track typing evolution all along the time). That is why we have used another database [8], consisting of height sessions for fitfy-one users providing fifty samples per session. Each user has provided 400 samples. We hope to track an important variability between sessions. The password is “.tie5Roanl". Each session is spaced of, at least, one day.

We have chosen a statistical keystroke dynamics method presented in [15]. Each sample is composed of the duration of the key press of each key (H), the delay between each key press (DD), and the delay between a key release and the next key press (UD). So, a sample is encoded as following $s = [H_0, DD_0, UD_0, H_1, DD_1, UD_1, ...]$. To build a model, we need several samples: $gallery = [s_1, s_2, ..., s_N]$, where N is the number of samples in the gallery. We then compute the following information:

- $mean$, the *mean* value of the gallery;
- $median$, the *median* of the gallery;
- std, the *standard deviation* of the gallery;
- max, the *maximum* value between $mean$ and $median$;
- min, the *minimum* value between $mean$ and $median$.

Naturally $Card(s_1) = Card(s_2) = ... = Card(s_N) = Card(mean) = Card(median) = Card(std) = Card(max) = Card(min)$, with $Card(.)$ the size of a vector.

The comparison score between the verification sample v and the model is computed by counting the number of time values

1: $score \leftarrow$ Compare $sample$ with $template$
2: **if** $score <= decision\_threshold$ **then**
3: User is accepted
4: **if** $score <= update\_threshold$ **then**
5: Update $template$ with $sample$
6: **end if**
7: **end if**

Figure 4. Summary of the semi supervised template update procedure

which verifies the following equation:

$$res = min * (0.95 - \frac{std}{mean}) <= v <= max * (1.05 + \frac{std}{mean}) \quad (1)$$

res is an array of 0 and 1. The first 1 is replaced by a 1.5, and the final score is computed as following:

$$score = 1 - mean(res) \quad (2)$$

C. Template Update

Some template update procedures have been proposed in the state of the art for keystroke dynamics. In this work, we have chosen to implement, in a semi-supervised way, two of them, which have been extensively applied in a supervised way [12]. Note that in this supervised context, no impostor samples can be appended to the model. Our paper presents their behavior in a semi-supervised way, where errors in the classifier can imply adding some erroneous samples in the model. The update procedure is applied only if the matching score between the sample and the template is above an *update threshold* if the sample as been considered as genuine by the *acceptance threshold*. This update threshold may be different from the *acceptance threshold*. A summary of an authentication procedure using a template update procedure can be seen in Figure 4.

1) Moving window: It consists in adding the new sample to the user’s gallery while releasing the latest one. The number of samples in the gallery remains constant.

2) Growing window: It consists in adding the new sample to the user’s gallery. The number of samples in the gallery keeps growing.

IV. Results

We have computed the EER of two update methods for different scenarios, corresponding to several impostor rates and different decision update thresholds. Figure 5 and Figure 6 respectively present the EER for various update thresholds and impostors rates. Because of a lack of place, we have only presented the results of the first and last sessions. Colors closer to dark blue correspond to an EER of 0% while colors closer to red correspond to an EER of 50%. These figures confirm that:

- The more impostor samples are tested the lower are the performances (because the probability of including impostor samples is more important). This is confirmed by looking at the bottom of the figures.

- Using a high update threshold gives lower results (we are using a distance score) when there is a high probability of including impostor samples. This is confirmed by looking at the right-bottom corner of the figures.
- Performance degrades with time when the decision update threshold is too low (i.e., when template update is almost never done). This is confirmed by looking at the left of the figures of the last session.

For the next experiment, we have therefore chosen a scenario with 20% of impostor samples which is quite realistic; an *update threshold* of 0.0 and an *acceptance threshold* of 0.2. This is a secure scenario where we want mainly to reduce the FMR. The baseline system is a system using no template update procedure (we have named it "'none"). Figure 7 presents the EER, FMR and FNMR for each session evaluated in online and offline way. Session 0 is used to configure the initial models. As computing the offline EER at session 7 is impossible (because we do not have a session 8 to use), we do not have a session 7 for the offline computation. From Figure 7 we can make the following observations for these values of the parameters:

- Without any template update mechanism, the EER becomes worse with time. This implies the need to always test keystroke dynamics methods with several sessions and the need of good template update mechanisms. If this requirement is not set, as in most keystroke dynamics studies, the results would be far better than in a real use case.
- When choosing the EER as error rate, growing window is always better than no adaptation. Moving window is always better than growing window. This shows that performing keystroke dynamics absolutely requires template update mechanisms, but, this strategy must to track only recent inputs and forget old behaviors. The old samples quickly become not any more representative of the user and conserving them decreases the results.
- Results can be interpreted differently when using an other error rate than the EER: a couple of FNMR/FMR for example. In the case of the EER, we present the results using different acceptance threshold for each session (which is not possible without a human intervention in the real life), while in the second case, all the parameters are fixed at the beginning of the experiment. Results seem to be better in the first case than in the second one (look at the difference of behaviour of growing window between the two kinds of error rates). The difference may be explained by the different number of genuine and impostor scores.
- With the selected thresholds, the FNMR does not evolve a lot when using no adaptation (it even seems to slightly reduces). The FNMR also reduces when using the moving window, while it augments when using the growing window. It means that, in the case of keystroke dynamics, using data on a too long period seems to be a bad choice.
- With the selected thresholds, the FMR grows a lot when using no adaptation, whereas it reduces significantly with any of the template update systems.
- Online and Offline EER computation do not show a lot of difference. The online method may be more interesting, because it gives a supplementary session for computing performance, and, reduce computation time (because scores are already computed for the adaptation process).
- Even if template update for keystroke dynamics is a necessity, it is a difficult problem, mainly due to the poor performance of this modality. The score distribution of genuine and impostor scores overlaps a lot and any choice of the thresholds will imply making many mistakes.

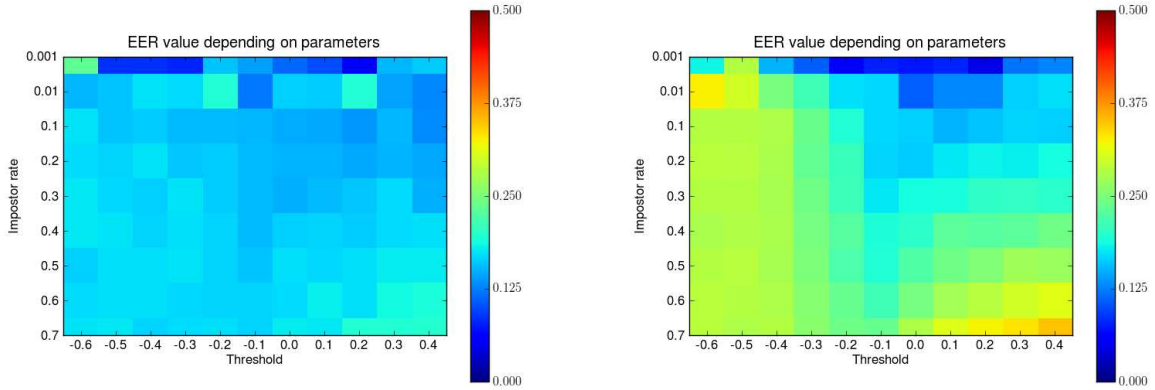
V. Discussion

Some of our results have been presented using the EER. This value gives a good overview of the performances, but it is not realistic because it needs to have configured the thresholds with the right value. In a real world depending on the level of security required, another functioning point (other values of FMR and FNMR) could be chosen. A future work would be to test our update methodologies in this context. By considering our previous results, we can argue that the EER is not a good error rate for the evaluation of template update mechanisms, because it could give better results which could not be obtained in a real case (because of the necessity to change the thresholds values). FNMR and FNMR given configuration thresholds would be better.

Another question of interest is the way of computing the results. We have chosen to display the performance rate per session. We think this is better than giving only one error rate for the whole data set, because the data are acquired through time and vary relative to time. This temporal information allow to see if performances increase or decrease over time. By the way, when computing performance with only the samples of one session, we are sure not to alter the results with the information of previous or next sessions. But, maybe one session is a too long or too short period.

A future work would be to explore the various ways for quantifying the error rate of various template update procedures. The random way of selecting verification captures may complicate such analysis. So, for the moment, presenting the evaluation results remains a problem to solve.

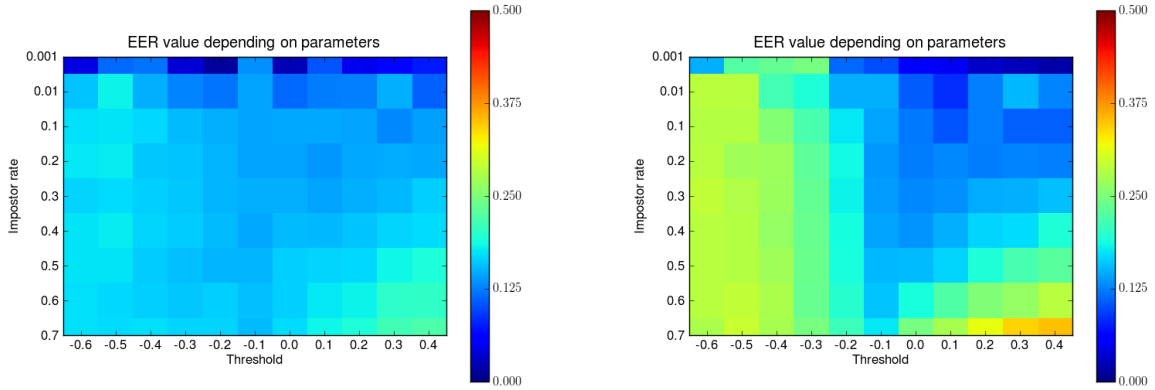
As everybody knows, results are greatly dependant of the used dataset. Scientific community does not have another keystroke dynamics datasets with this high number of sessions. It is a problem, because results could be slightly different with a dataset created with a different protocol. For example, during a session, passwords have been typed almost in one time (i.e., no pause during typing). This could reduce the intra-session variation and augment the inter-session variation.



(a) Session 1

(b) Session 7

Figure 5. EER value for different configurations for each session with the growing window procedure



(a) Session 1

(b) Session 7

Figure 6. EER value for different configurations for each session with the sliding window procedure.

VI. Conclusion

In this paper, we have proved the possibility of using template update methods in a semi-supervised scenario. Such scenario is much more difficult because impostor patterns can be added to the new template of an user. Due to these errors, the template may derive from the user way of typing and may reject him more easily (while it accepts more easily impostors). In order to guaranty a good behavior of the system, we need to limit the impostors authentication attempts and to set a relatively strict decision update threshold. In a supervised way, we obtain an improvement around 50%. while we obtain an improvement of 45% with a semi-supervised way with the appropriate configuration.

One issue in the biometric template update is the evaluation procedure. We hope to have open a new path in the evaluation of semi-supervised template update systems. This study has for objective to observe the behavior of template update on behavioral modalities. In the future, our aim will be to evaluate other keystroke dynamics methods and other template update procedures. An automatic configuration of the different thresholds would also be interesting.

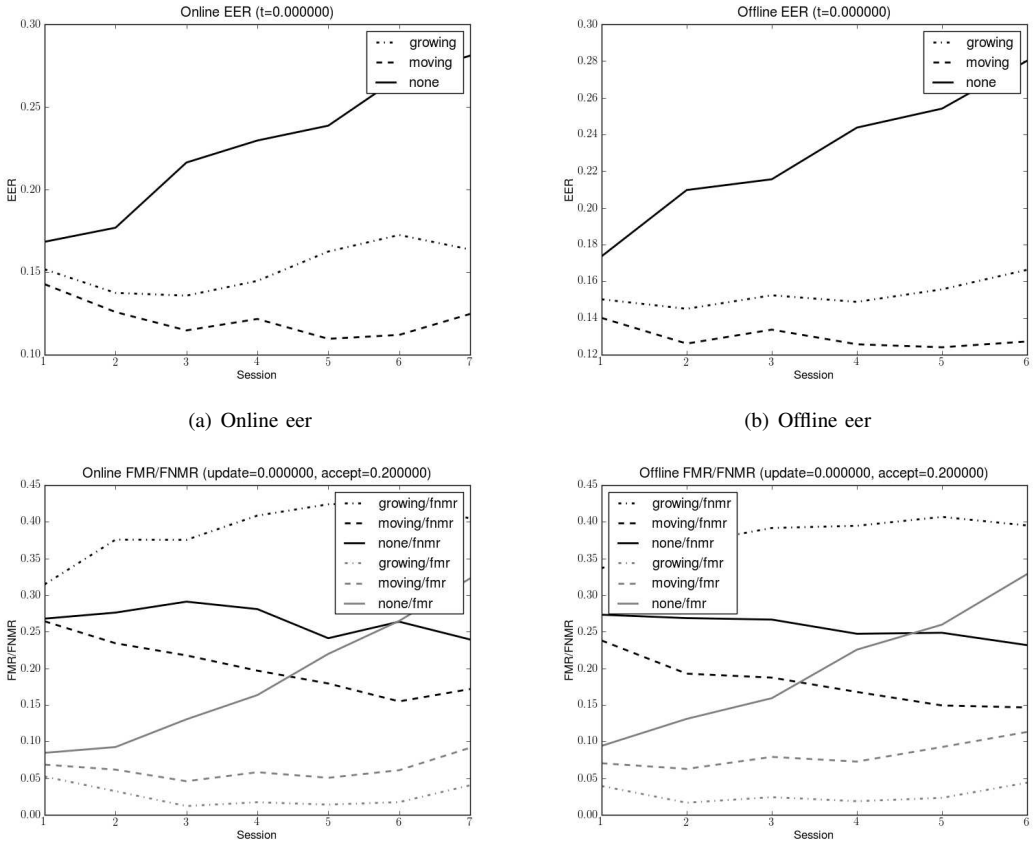
References

[1] R. Yampolskiy and V. Govindaraju, "Behavioural biometrics: a survey and classification," *International Journal of Biometrics*, vol. 1, no. 1, pp. 81–113, 2008.

[2] L. Araujo, J. Sucupira, L.H.R., M. Lizarraga, L. Ling, and J. Yabu-Uti, "User authentication through typing biometrics features," *IEEE Transactions on Signal Processing*, vol. 53, no. 2 Part 2, pp. 851–855, 2005.

[3] R. Giot, B. Hemery, and C. Rosenberger, "Low cost and usable multimodal biometric system based on keystroke dynamicsand 2d face recognition," in *IAPR International Conference on Pattern Recognition (ICPR 2010)*. Istanbul, Turkey: IAPR, 8 2010.

[4] N. Houmani, S. Garcia-Salicetti, and B. Dorizzi, "On assessing the robustness of pen coordinates, pen pressure and pen inclination to time variability with personal entropy," in *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*, 2009.



(a) Online eer

(b) Offline eer

(c) Online FMR/FNMR

(d) Offline FMR/FNMR

Figure 7. This figure presentes various error rates (EER, and FMR and FNMR given an acceptance threshold) for an update threshold (0.0) evaluated in online and offline ways.

[5] T. Scheidat, A. Makrushin, and C. Vielhauer, "Automatic Template Update Strategies for Biometrics," Tech. Rep., 2007.

[6] G. L. M. Fabio Roli, Luca Didaci, *Advances in Biometrics*. SpringerLink, 2008, ch. Adaptive Biometric Systems That Can Improve with Use, pp. 447–471.

[7] R. Giot, M. El-Abed, and R. Christophe, "Greyc keystroke: a benchmark for keystroke dynamics biometric systems," in *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS 2009)*. Washington, District of Columbia, USA: IEEE Computer Society, Sep. 2009, pp. 1–6.

[8] K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in *39th Annual International Conference on Dependable Systems and Networks (DSN-2009)*, 2009, pp. 125–134.

[9] N. Poh, J. Kittler, S. Marcel, D. Matrouf, and J. Bonastre, "Model and score adaptation for biometric systems: Coping with device interoperability and changing acquisition conditions," in *Proceedings of Internal Conference on Pattern Recogntion (ICPR 2010)*, 2010.

[10] J.-F. Connolly, E. Granger, and R. Sabourin, "An adaptive classification system for video-based face recognition," *Information Sciences*, vol. In Press, Corrected Proof, pp. 1–21, 2010.

[11] A. Rattani, B. Freni, G. Marcialis, and F. Roli, "Template update methods in adaptive biometric systems: A critical review," in *Internal Conference on Biometrics 2009 (ICB 2009)*, 2009.

[12] P. Kang, S.-s. Hwang, and S. Cho, "Continual retraining of keystroke dynamics based authenticator," in *Proceedings of ICB 2007*, ser. Lecture Notes in Computer Science, S.-W. Lee and S. Li, Eds., vol. 4642. Springer Berlin / Heidelberg, 2007, pp. 1203–1211.

[13] C. Ryu, H. Kim, and A. K. Jain, "Template adaptation based fingerprint verification," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, 2006.

[14] N. Poh, R. Wrong, J. Kittler, and F. Roli, "Challenges and research directions for adaptive biometric recognition systems," in *Advances in Biometrics*, 2009, pp. 753–764.

[15] T. de Magalhaes, K. Revett, and H. Santos, "Password secured sites: stepping forward with keystroke dynamics," in *International Conference on Next Generation Web Services Practices*, 2005.