Keystroke Dynamics With Low Constraints SVM Based Passphrase Enrollment

Romain Giot, Mohamad El-Abed, Christophe Rosenberger

Keystroke Dynamics With Low Constraints SVM Based Passphrase Enrollment

Romain Giot Mohamad El-Abed Christophe Rosenberger
Laboratoire GREYC, ENSICAEN - Université de Caen, Basse-Normandie - CNRS
romain.giot@ensicaen.fr, {melabed,christophe.rosenberger}@greyc.ensicaen.fr

***Abstract*— Keystroke dynamics biometric systems have been studied for more than twenty years. They are very well perceived by users, they may be one of the cheapest biometric system (as no specific material is required) even if they are not commonly spread and used [1]. We propose in this paper a new method based on SVM learning satisfying operational conditions (no more than 5 captures for the enrollment step). In the proposed method, users are authenticated thanks to keystroke dynamics of a passphrase (that can be chosen by the system administrator). We use the GREYC keystroke benchmark that is composed of a large number of users (100) for validation purposes. We tested the proposed method face to four other methods from the state of the art. Experimental results show that the proposed method outperforms them in an operational context.**

I. Introduction

A. Biometrics systems

The access of entities to controlled resources is managed by authentication systems which provide answers to two questions (i) who is the user ? and (ii) is the user really who he claims himself to be ? In this paper, we are interested in the second case, where we want to verify the identity of the user by keystroke dynamics. Strong authentication uses different factors to authenticate people. It is based on the use of, at least, of two of these parameters: (i) something the user knows; (ii) something the user owns; (iii) something that qualifies the user or its behaviour.

The general process of biometric systems is composed of two mains parts: (i) the enrollment (which consists of registering the user in the system) embeds the data capture, eventually some data filtering, the feature extraction and the learning step with its storage ; (ii) the verification process realizes the data capture, the feature extraction and the comparison with the biometric reference of the supposed user.

B. Keystroke dynamics

The aim of keystroke dynamics systems is to secure the use of password based authentication which suffers of many drawbacks [2]: (i) passwords can be shared between users, (ii) passwords can be stolen, (iii) passwords can be guessed. Keystroke dynamics introduce an additional parameter to the authentication process: something that qualifies the user or its behaviour (the way of typing passwords) in order to strengthen the password authentication. The capture process is presented in Fig. 1. It consists in capturing the time when the keys are pressed and released. In this example, the raw data captured are the time when C and O keys are pressed, then released. The capture process consists in capturing several features when the keys are pressed and released (timestamp of the event, code of the key, etc.). Fig. 1 illustrates an example of the data captured when a user pressed the expression "CO".

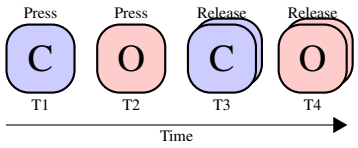

Fig. 1. Information capture in a keystroke dynamics system when pressing C and O keys

The features extraction consists mainly to create latencies (T2-T1, T4-T3, T2-T3) and duration (T3-T1) times with the help of the previous captured times. In our example, T2-T3 is negative because the user press O before releasing C (he types fast), which is not always true. We have chosen this example to show that timing can be negative. The features used are not the same in all the studies. In this paper, the feature extracted data of each capture is the vector resulting in the concatenation of the four previous timing vectors.

II. Background

A. State of the Art

The first research work on keystroke dynamics was realized in 1980 with the report of the *Rand Corporation* [3]. This study proves that individuals could be differentiated by considering their way of typing. Seven secretaries were asked to type three different long texts, and the comparison was done using *statistical methods*.

Many researchers tried to improve the process by both decreasing the number of errors and the number of required data to create the biometric reference of a user. Some of these studies are presented in this section. The first patent was proposed in 1986 [4] and works with passwords (and not long input text), always by using statistical measures to create the user's reference (or model) and compare it with the verification template. Several patents have been done during the last twenty years, and one of the latest [5] have been proposed this year and use *Hidden Markov Models*.

ENSICAEN, 6 Boulevard Maréchal Juin, 14000 Caen Cedex - France

In 1990, Bleha *et al.* [6] tried authentication based on the *user name* with a *static phrase* and use *Bayesian classification* and *distance measures*. They also work on identification process. They argue that the longer is the password, the less is the error rate ; the error rate decreases when number of enrolled patterns increases and results are better with person's name than on passphrases (due to their habit of typing it). Studies using neural networks has begun since 1997 [7]. Authors used latency and duration time of digraphs as features. They obtained an error rate of 0% but use a quite huge number of enrolled patterns (112).

Monrose and Rubin worked on keystroke dynamics for free texts [8]. They also used statistical methods, and propose to split users in different groups in order to speed up timing reconnaissance.

Cho and Hwang [9] proposed, in 2006, to improve the performances of keystroke dynamics by allowing individuals to use *pause* helped by *cues* to improve *unicity*, *consistency* and *discriminability* of their password and render more difficult the forgery of typing dynamics. In [10], Rodrigues *et al.* use *Hidden Markov Model* in their authentication method. By using passwords only composed of numbers, they obtain an *EER* of 3.6%. This study is interesting, because it opens the way on keystroke dynamics in pin code authentication based environment (*i.e.* ATM or cell phone). In [11] authors have tested the efficiency of SVM in the keystroke dynamics. They have tested one-class SVM and two-class SVM (in this case, impostors' data simulating the second class are generated). The performance trade-off and time consumption were better and faster than with neuronal networks, but the experiment was done with only 10 individuals.

In 2007, Hocquet *et al.* [12] proposed to automatically classify the individuals in different classes depending on various parameters and to assign different threshold configuration for each class. This idea is interesting, because it is known that individual thresholds give better results than global ones, but are difficulty applicable in real life (because we do not necessary have impostors patterns). They obtain an *EER* about 2%. The classes creation and threshold configuration is done with a test database. Another study [13] uses various digraph information and time of typing for both user id and password and discretized them into an alphabet of twenty discrete elements. The classification is done by using rough set paradigm. They obtained an *EER* value lower than 1%.

Gaussian mixture modeling of keystroke patterns is used in [14]. Authors also give good information on the way of creating interesting keystroke dynamics database, and how to present the results. The obtained *EER* is around 4.5%. They argue that if passwords have more than 8 letters, the numbers of failures increase.

B. Discussion

Different methods have been proposed in the literature, but the acquisition were different in terms of quantity of users, captures needed for model creation and impostors' data. Most of the time, there were not enough users. Another point concerns the number of captures for the enrollment step that is generally high. Moreover, very few papers focus on passphrase keystroke dynamics.

In the next section, we propose a new method whose aim is to contribute to solve these problems.

III. Proposed Method

We propose in this paper a new method based on SVM learning where the number of captures is limited to 5 for the initial enrollment step. Users are asked to type a passphrase set by the administrator system. For the enrollment, we use a two-classes SVM as machine learning method. The novelty of this method is based on the facts that (i) it requires only five captures to create the model, (ii) the use of pre-processing to discretize the captures.

Suppose we have a training set $\{\mathbf{x}_i, \mathbf{y}_i\}$ where $\mathbf{x}_i$ is enrolled vectors and $\mathbf{y}_i$ the individual. For problems with two classes, with the classes $y_i \in \{-1, 1\}$, a support vector machine [15], [16] implements the following algorithm. First, the training points $\{\mathbf{x}_i\}$ are projected into a space $\mathcal{H}$ (of possibly infinite dimension) by means of a function $\Phi(\cdot)$. The second step is to find an optimal decision hyperplane in this space. The criterion for optimality will be defined shortly. Note that for the same training set, different transformations $\Phi(\cdot)$ may lead to different decision functions.

A transformation is achieved in an implicit manner using a kernel $K(\cdot, \cdot)$ and consequently the decision function can be defined as :

$$f(\mathbf{x}) = \langle w, \Phi(\mathbf{x}) \rangle + b = \sum_{i=1}^{\ell} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b \qquad (1)$$

with $\alpha_i^* \in R$. The values w and b are the parameters defining the linear decision hyperplane. We use in the proposed system a linear function as kernel function.

In SVMs, the optimality criterion to maximize is the margin, that is to say, the distance between the hyperplane and the nearest point $\Phi(\mathbf{x}_i)$ of the training set. The α_i^* which optimize this criterion are obtained by solving the following problem :

$$\begin{cases} \max_{\alpha_i} \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i K(\mathbf{x}_i, \mathbf{x}_j y_j) \\ \text{with constraints,} \\ 0 \ \leq \ \alpha_i \ \leq \ C \ , \\ \sum_{i=1}^{\ell} \alpha_i y_i \ = \ 0 \ . \end{cases} \qquad (2)$$

where C is a penalization coefficient for data points located in or beyond the margin and provides a compromise between their numbers and the width of the margin.

The distance score for the verification test is computed as following:

$$SVM = -prb * prd \qquad (3)$$

were prb stands for the probability accorded to the SVM result and prd stands for the class of the result (-1 or 1). The data used, are the capture vectors discretized in an alphabet of 5 values (same method of Equation 7).

IV. Experimental Protocol

The following section explains the experimental protocol used to collect the data. As there is no big public database for keystroke dynamics, we were in the obligation to create our own one, and decided to make it publicly available at the following address[1].The following subsections present succinctly the benchmark used.

A. Test Population

We have asked to 133 individuals to enroll themselves in our biometric system. They had the possibility to capture their data one or two times a week during more than two months.

B. Acquisition Procedure

Each user has been requested to type the password "greyc laboratory" six times on two different keyboards (this gives twelve captures in total per session) during each session, by alternating the typing from one keyboard to another (*i.e.* keyboard one, then keyboard two, then keyboard one, and so on). We have chosen this password for two mains reasons (i) this is the name of our laboratory, and contribute to its communication, and (ii) this is a not too short password [6] with a good partitioning of the keys on the keyboards (which can help to its discriminability[17]). More information about the database is available in [18].

133 individuals have participated to the capture process, but only 100 of them have provided at least 60 captures. The data used for the statistical evaluations belongs to these users.

The following raw data are captured: (i) the timestamp of the event, (ii) the type of event (press or release), and (iii) the code of the key generating the event. The extracted data computed with the help of the raw data are the total typing timing of a password, and, for each couple of keys, the times RR, RP, PR, and PP (R stands for Release, and P for Press).

Each typing error is also saved in the database. This information gives a good overview on the error rate of the capture process. Even if it is admitted that keystroke dynamics is keyboard dependent, very few studies have tested the implication of the keyboard in the keystroke dynamics. It is an interesting thing to test, because most computer services are accessible in a web-based environment and can be reached everywhere. That implies the fact that users can access (and authenticate) services from different computers, operating systems, and keyboards.

C. Methods From the State of the Art

In this section, we present comparative methods from the state of the art for our experiment. We have tried to use different "families" of algorithms by using (i) statistical based algorithm, (ii) distance based algorithm, (iii) rhythm based algorithm, and (iv) machine learning based algorithm. We note v the test vector, i the size of the template vector, μ the mean vector of the enrolled templates and σ its standard deviation.

1) Statistical Based Algorithm: Two statistical methods are tested. The first one does not take into account the user standard deviation [6]:

$$STAT1 = \frac{(v-\mu)^t (v-\mu)}{||v||.||\mu||} \tag{4}$$

while the second one uses both mean and standard deviation values of vectors [12]:

$$STAT2 = 1 - \frac{1}{n}\sum_{i=1}^{n} e^{-\frac{|v_i-\mu_i|}{\sigma_i}} \tag{5}$$

2) Distance Based Algorithm: The distance based algorithm is based on an euclidean distance [8]:

$$DIST = \min\left(\forall_{u\subset enrol}, \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}\right) \tag{6}$$

3) Rhythm Based Algorithm: This method consists in discretizing keystroke values along five different classes and compute a classical Hamming distance [12]:

$$RYTHM = \frac{1}{n}\sum_{i=1}^{n} abs(class(v_i) - class(\mu_i)) \tag{7}$$

where $class(i)$ is a function returning the class of i (*i.e.* we operate a discretization of the time) along five different classes. To compute the classes, we partition the space in five clusters of the same size 8 between the minimum and the maximum value of the learning dataset. The assigned classes of the whole dimensions of each vector is the number of the cluster.

$$cluster\_width = \frac{max(train\_data) - min(train\_data)}{5} \tag{8}$$

V. Experimental Results

In this section, we present the experimental results of the proposed method face to the four ones from the literature in an operational context (*i.e.* with a maximal number of 5 captures for the enrollment step).

We also analyze the dependency of the keyboard in the authentication process. It is important to test the dependency hypothesis between the keyboard used and the model created. In other words, is the model will change if we change the keyboard? We present also the impact of changing the number of captures to create the model, the use of an individual or global threshold and the dependency to the size of the database, because these parameters are different depending on the studies.

[1] http://www.ecole.ensicaen.fr/~rosenber/keystroke.html

A. Differences Between the Two Keyboards

Table I represents the different *EER* (Error Equal Rate, when the system provides the same rate of genuine rejected and impostors accepted) depending of the keyboard origin of the data used to both train and test sets.

The *EER* of each method is computed by keeping the first ten vectors for enrollment, and all the others for the verification process. The keyboard origin of the vectors can be different. There is no adaptation (see later for more information) mechanism used for this test. When the keyboard source from enrollment and test vectors is different, the computation is done several times by selecting enrolled vectors randomly and averaging the results.

TABLE I

ERROR RATES OF METHODS DEPENDING ON THE KEYBOARD CONFIGURATION. "EERNM" MEANS CAPTURES FROM KEYBOARD "N" FOR ENROLLMENT AND CAPTURES FROM KEYBOARD "M" FOR VERIFICATION, WHERE "1", "2", "A" STANDS RESPECTIVELY FOR KEYBOARD 1, KEYBOARD 2 AND NO DISTINCTION OF KEYBOARD. THE BEST *EER* OF EACH METHOD IS PRESENTED IN BOLD.

Method	EER11	EER12	EER21	EER22	EERaa
STAT1	24.91	23.96	24.73	23.51	25.50
STAT2	17.68	16.55	17.10	16.65	17.58
DIST	27.01	26.00	26.46	25.07	27.56
RYTHM	19.40	20.09	19.25	19.50	19.78
SVM	**10.68**	**10.37**	**10.30**	**11.76**	**11.96**

The columns EER11 and EER22 represent respectively *EERs* when the data belongs only to keyboard one and keyboard two. The column EER12 represents *EER* computed by using keyboard one for training captures and keyboard two for test captures and vice versa for the column EER21. In the column EERaa, we use without any distinction, nor order, data from keyboard one and two.

We can see that results are not exactly equal, depending on keyboard configuration. But, the worst results are not obtained when the data of both keyboards are mixed. 4 times out of 6, the best results are obtained when the test and enroll keyboards are the same, whereas 3 times out of 6, the worst results are obtained when no distinction about keyboard is done. This experiment shows that the implemented solution in the GREYC-Keystroke gives similar results for these two keyboards.

The proposed method outperforms all other methods for each configuration.

B. Action of the Number of Enrolled Templates

An interesting point is the visualisation of the *EER* depending on the number of captures used to create the biometric reference for an individual. In most of the studies, this number is different. Showing this information can facilitate the comparison of different algorithms, when the number of captures used for enrolment is different in the original article.

Algorithms performance varies depending on the number of captures used in order to create models. Most of studies used more than twenty captures in order to create the model, whereas five is really a maximum of typing accepted by a user in order to create his model. Fig. 2 represents the *EER* of different algorithms depending on the number of enrolled patterns. It is clear that the more we use captures, the more the model is representative. For all the methods, less than

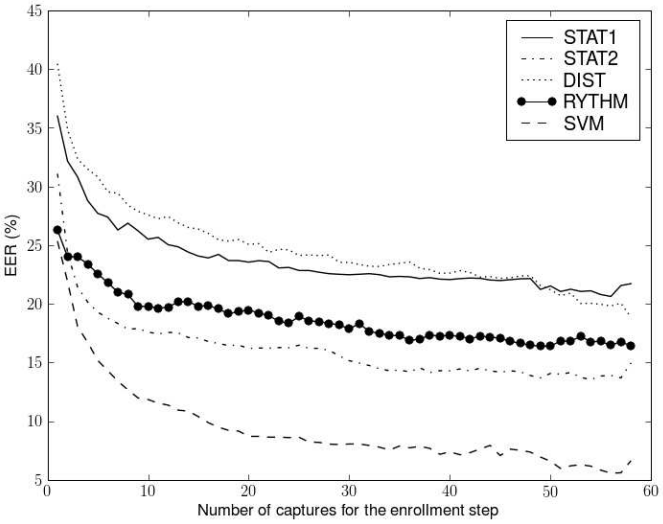

Fig. 2. Evolution of the *EER* of different algorithms by considering the number of patterns used to create the model

ten captures give really bad results, the minimum required seems to be around 40 captures[2]. For some methods, the performances decrease when using more than 50 captures. Once again, the proposed method gives the best results.

C. Adaptation Mechanism

Keystroke dynamics is a behavioral modality, that is why an *adaptation* mechanism can be implemented in order to improve systems' performance [19], [17]. By this way, the model evolutes depending on the evolution of user's typing. Four different kinds of evolution algorithms are tested:

- without adaptation (we keep the first vectors as enrolled vectors for ever), this approach is denoted noted "Classic";
- a method swapping enrolled vectors by releasing the latest one and adding the test vector after verification, called "Adaptive";
- a method adding user's vector after being verified in the list of enrol vectors, noted "Progressive";
- a method based on a combination of the two previous ones, noted "Intelligent": while the number of required enrolled vectors is not reached (set at 15), the progressive method is used, whereas then when the total number is fit, the adaptive system is used. Vectors are added only when they are not too much different from the model.

Table II presents the *EERs* of the described methods by using different adaptation mechanisms. These *EERs* have

[2]but, in this case, the number of patterns used to test the performance is really small

been computed by using: five enrolled vectors, the captured data from both keyboards without distinction and by using a global threshold.

TABLE II

EER OF METHOD BY USING DIFFERENT ADAPTATION MECHANISM AND FIVE CAPTURES TO CREATE THE MODEL BY USING DATA OF BOTH KEYBOARDS. THE BEST *EER* OF EACH METHOD IS PRESENTED IN BOLD.

Method	Classic	Progressive	Adaptive	Intelligent
STAT1	27.7	21.24	23	20.94
STAT2	19.29	15.09	11.71	10.75
DIST	30.81	23.75	25.7	24.65
RYTHM	22.56	15.49	14.36	13.21
SVM	**15.38**	**6.69**	**9.21**	**6.96**
Mean	23.15	16.45	16.8	**15.3**

We can see in this table that using an adaptation mechanism improves the algorithms' performance. For most of the algorithms, the best adaptation mechanism is the "Intelligent" one even if it can use less vectors than the progressive one. So, filtering the captures before adding them in the model improves the performance by reducing the *EER* of approximately 8%. Our method gives again the best results and provide the minimal *EER* of 6% value for the adaptive mode.

The aim of this test is to prove there is an evolution in the way of typing from the user, and not which is the best method to use to create the model A more industrial method, would be to try to adapt the model of the user, if the verification is a success, with all the test vectors (in this case, we need to compare the distance to a predefined threshold), at the risk to add impostors patterns to the model.

D. Dependence of the Threshold

The statical performance of biometrics systems are different between the use of a global threshold configured for the whole system, or a threshold configuration different for each user (or class of users) [14], [20]. The user specific threshold is the threshold minimizing corresponding to its *EER*, which is computed by using its own captures for the *False Reject Rate* and impostors' captures for the *False Acceptance Rate*. Using individual thresholds instead of global ones, is supposed to improve algorithms' performance. Table III presents the improvements in term of *EER* due to the use of individual thresholds. The *EERs* are computed by using: five captures for the model, the "Intelligent" adaptation method, and data from both keyboards without any distinction.

We can see that, for each method, there is a slight improvement of 1.5% by using individual thresholds, but, in order to compute them, it is necessary to have impostors' data which is possible in our case because all the users have the same password, but can not be easily applied in cases different of the passphrase one because every body has a different password. A solution to this problem is presented in [20].

TABLE III

EER OF METHODS WHEN USING GLOBAL AND INDIVIDUAL THRESHOLDS, BY USING DATA OF BOTH KEYBOARDS AND AN ADAPTATION MECHANISM. THE BEST *EER* OF EACH METHOD IS PRESENTED IN BOLD.

Method	EER(global)	EER(individual)	Gains
STAT1	20.94	19.54	1.4
STAT2	10.75	9.22	1.53
DIST	24.65	21.53	3.12
RYTHM	13.21	10.02	**3.18**
SVM	**6.96**	**6.95**	0.01
Mean	15.3	**13.45**	1.85

E. Dependence of the database

Like it is commonly accepted that biometrics systems' performance are dependant of the database, we have decided to plot the evolution of the *EER* in function of the number of users in the database. Fig. 3 presents this evolution. Less

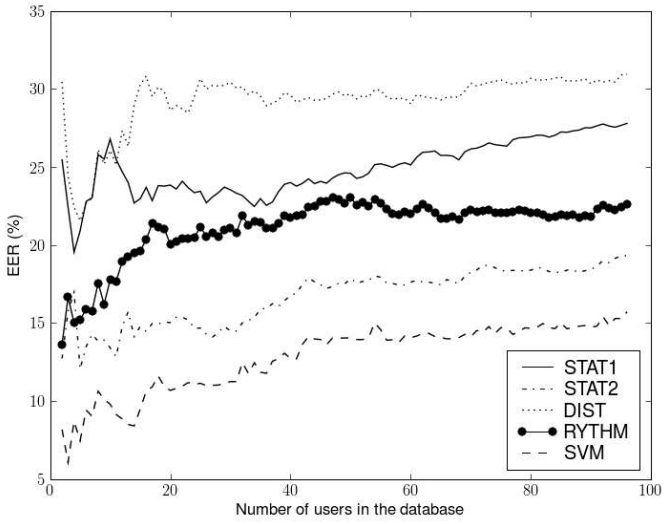

Fig. 3. Evolution of the *EER* of each method depending on the number of individuals in the database.

than 10 users is totally insufficient to test the algorithm's performance. The more there are users in the database, the more the *EER* increases, so EER performance is really dependent on the database size, and especially on the number of users. 50 individuals seems to be the smallest number acceptable (whereas most of the study do not fit this requirement).

F. Discussion

Keystroke dynamics is really sensitive to user's concentration during acquisition, the acquisition error rate is really important. It could be a curb to the spreading of this modality, even if it is one of the cheapest and easiest one to implement. There is no doubt that the performance of a biometric system is due to the number of available captures to create the user's biometric reference. By using five captures, the performances are not as good as attended, whereas by using more captures, the user has to spend more

times in front of the enrolment process, which can be really disturbing, and, also, curb the spreading of this modality.

SVM is the best tested method, but its results are not as good as in the literature [11]. The main reason is the maximal number of captures we use to define the biometric reference that is lower than published papers in that domain. We have to explore on the configuration of the SVM. Even if SVM presents the best results, this method exploits the template of all individuals in the database, this method is so interesting when users types the same passphrase to authenticate themselves.

VI. Conclusion And Perspectives

We showed in this paper that keystroke dynamics is an interesting modality, but there is a lack of available databases to test different issues and improve systems' performances. The performance of methods in the state of the art varies a lot in function of the size of the database[3] in term of number of users (and impostors' data) and number of captures per user (and deviations in the templates). Our performances are lower than in the previous published studies, as we use passphrases instead of free passwords and only 5 captures for the enrollment step. For this context, our method outperforms all other tested methods from the state of the art.

We have seen that individual thresholds improve the performance of systems, one of our future work will be to find ways of configuring easily and quickly without impostors' data. It would be also a good idea to find, for each algorithm, the best extracted features. A more efficient adaptation mechanism is also a thing to explore, always by keeping in mind the improvements of the methods. The number of tested keyboard is really too small, the same kind of study has to be done with more variety of keyboards and computers.

It is also important to emphasize on the fact that our performance are highly dependant of our database on (i) the distribution of the individuals, and (ii) the password itself. On interesting thing would be to do the same experiment, with the same individuals, by using different passwords (on the size and the repartition on the keyboard) and also with the same password and a different panel of individuals.

VII. Acknowledgments

The authors would like to thank the "Basse-Normandie Region" and the French Research Ministry for their financial support of this work. We would like also to thank all the individuals who have participated to the definition of the keystroke dynamics database.

[3]We have seen the *EER* increasing all along the database was growing

References

[1] F. Cherifi, B. Hemery, R. Giot, M. Pasquet, and C. Rosenberger, *Behavioral Biometrics for Human Identification: Intelligent Applications*. IGI Global, 2009, ch. Performance Evaluation Of Behavioral Biometric Systems, pp. 21+.

[2] M. Sasse, S. Brostoff, and D. Weirich, "Transforming the 'weakest link'—a human/computer interaction approach to usable and effective security," *BT Technology Journal*, vol. 19, no. 3, pp. 122–131, 2001.

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

[4] J. D. Garcia, "Personal identification apparatus," November 1986. [Online]. Available: http://www.freepatentsonline.com/4621334.html

[5] V. V. Phoaha, S. Phoha, A. Ray, S. S. Joshi, and S. K. Vuyyuru, "Hidden markov model (hmm)-based user authentication using keystroke dynamics," patent, fev 2009.

[6] S. Bleha, C. Slivinsky, and B. Hussien, "Computer-access security systems using keystroke dynamics," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 12 (12), 1990.

[7] M. Obaidat and B. Sadoun, "Verification of computer users using keystroke dynamics," *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, vol. 27, no. 2, pp. 261–269, 1997.

[8] F. Monrose and Rubin, "Authentication via keystroke dynamics," in *Proceedings of the 4th ACM conference on Computer and communications security*. ACM Press New York, NY, USA, 1997, pp. 48–56.

[9] S. Cho and S. Hwang, "Artificial rhythms and cues for keystroke dynamics based authentication," in *IAPR International Conference on Biometrics*, vol. 5, 2006, pp. 626–632.

[10] R. Rodrigues, G. Yared, C. do NCosta, J. Yabu-Uti, F. Violaro, and L. Ling, "Biometric access control through numerical keyboards based on keystroke dynamics," *Lecture notes in computer science*, vol. 3832, p. 640, 2006.

[11] Y. Sang, H. Shen, and P. Fan, *Parallel and Distributed Computing: Applications and Technologies*. Springer, 2005, ch. Novel impostors detection in keystroke dynamics by support vector machine, pp. 666–669.

[12] 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.

[13] K. Revett, S. de Magalhaes, and H. Santos, "On the use of rough sets for user authentication via keystroke dynamics," *Lecture notes in computer science*, vol. 4874, p. 145, 2007.

[14] 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*, vol. 38, no. 6, pp. 816–826, 2008.

[15] V. Vapnik, "Statistical learning theory," *NY Wiley*, 1998.

[16] B. Scholkopf and A. Smola, "Learning with Kernels: Support Vector Machines, Regularization," *Optimization, and Beyond. MIT Press*, vol. 1, p. 2, 2002.

[17] K. Revett, S. de Magalhães, and H. Santos, "Enhancing login security through the use of keystroke input dynamics," *Lecture notes in computer science*, vol. 3832, p. 661, 2006.

[18] 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)*, 2009, to be published.

[19] 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.

[20] S. Hocquet, J.-Y. Ramel, and H. Carbot, "Estimation of user specific parameters in one-class problems," in *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 449–452.