Online Signature Verification with New Time Series Kernels for Support Vector Machines

Christian Gruber, Thiemo Gruber, and Bernhard Sick

University of Passau, Institute of Computer Architectures
{gruberc, grubert, sick}@fmi.uni-passau.de

Abstract. In this paper, two new methods for online signature verification are proposed. The methods adopt the idea of the longest common subsequences (LCSS) algorithm to a kernel function for Support Vector Machines (SVM). The two kernels *LCSS-global* and *LCSS-local* offer the possibility to classify time series of different lengths with SVM. The similarity of two time series is determined very accurately since outliers are ignored. Consequently, LCSS-global and LCSS-local are more robust than algorithms based on dynamic time alignment such as Dynamic Time Warping (DTW). The new methods are compared to other kernel-based methods (DTW-kernel, Fisher-kernel, Gauss-kernel). Our experiments show that SVM with LCSS-local and LCSS-global authenticate persons very reliably.

1 Introduction

Authentication of a person's identity is an everydays issue. Often, signatures are used for verification. However, in most cases a signature is compared to a single reference signature with the naked eye only. Electronic, typeface-based techniques (so-called *offline signature verification*) can easily be outsmarted. Since authentication by signature is more widely accepted than any other technique (e.g., fingerprint or iris scan), a signature verification system that ensures a high level of security must be developed. Biometric signature verification systems that are based on the dynamics of a person's signature and not on its image are substantially more suitable for a reliable authentication (so-called *online signature verification*).

Support Vector Machines (SVM) are very popular since a few years. As they provide very good results for various pattern recognition problems, they also seem to be a good choice for online signature verification. Compared to most methods used for signature verification such as Hidden Markov Models (HMM) or Dynamic Time Warping (DTW), SVM, which are based on the principle of structural risk minimization, have various advantages such as a convex objective function with very fast training algorithms. On the other hand, SVM typically are applied to data sets containing feature vectors of fixed length and not to problems dealing with time series of variable length such as in online signature verification. In the following, the terms time series and sequence will be used equivalently.

D. Zhang and A.K. Jain (Eds.): ICB 2006, LNCS 3832, pp. 500–508, 2005.

2 Related Work

Only a few researchers in the area of online signature verification apply SVM as their usage poses a major problem: "Classic" kernel functions such as linear, polynomial, or Gauss-kernels require input vectors of fixed length. Since almost all signatures have different lengths, a way has to be found to deal with input vectors of different lengths.

One approach is to extract a fixed set of features, such as average velocity, pen-up time, etc., from each signature and to present these input vectors to an SVM utilizing kernels as mentioned above (so-called *static kernels*). This approach is described in [1]. SVM with classic kernels are also used in [2], but just for the fusion of two preceding classifiers' decisions (ensemble technique).

The other approach is to use kernels for sequences (so-called *dynamic kernels*). As far as we know, SVM with dynamic kernels have not been applied to online signature verification yet. Several publications such as [3, 4] provide a survey and a comparison of various kernel functions that are suitable for time series classification. Two categories of kernels can be distinguished: Within the first category are kernels based on distance measures. [5] introduced the *Time Alignment Kernel* for handwritten digit recognition. [6, 7, 8] developed kernels based on DTW for speech recognition and handwritten character recognition. The second category comprises kernel functions based on probabilistic models such as HMM or Gaussian Mixture Models (GMM). [9] introduced the *Fisher-kernel* which maps an input sequence onto a score vector with fixed length that is obtained from the parameters of the underlying probabilistic model. Fisher-kernels are used for classification of DNA-fragments, speaker-independent classification of spoken letters, or speaker identification. Another probabilistic kernel based on the Kullback-Leibler divergence of two GMM was described in [10]. This kernel was applied to speaker authentication as well as image classification.

3 LCSS-Based Kernels for SVM

3.1 Kernel Functions for Online Signature Verification

Given a binary classification problem $\mathcal{L} = \{(\mathbf{x}_i, y_i)\}$ with class labels $y_i \in \{-1, 1\}$, $i = 1 \dots l$, an SVM (see, e.g., [14]) classifies a test sample $\mathbf{t}$ by $y_t = sign\left(\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{t}, \mathbf{x}_i) + b\right)$with parameters $\alpha_i, b \in \mathbb{R}$. $K(\cdot, \cdot)$ is a kernel function which is the inner product of the samples transformed into a higher-dimensional space. In this feature space, an SVM tries to separate the two classes linearly. If a kernel function satisfies Mercer's conditions, the resulting kernel matrix is positive semidefinite, and the objective function that has to be optimized to determine the α_i and b is convex.

HMM are *probabilistic models* often used for online signature verification. Kernel functions based on HMM can either be applied to raw data or to features extracted from those data. In the former case a large number of hidden states is needed to get reasonable results. The consequence is a slow training phase. In the latter case complex pre-processing measures must be applied (e.g., segmentation of sequences and/or approximation with local models).

Approaches based on *distance measures* such as DTW can also be used as kernel functions. However, most of these methods are very sensitive to varying signal offsets or different ranges. Also, most of these methods do not deal appropriately with the type of outliers that appears within signatures (e.g., additional loops). We need kernel functions that do not require time-consuming pre-processing and meet our requirements concerning outliers.

3.2 LCSS-Global

In this section, a method that determines the similarity of two sequences with different lengths is introduced. In contrast to other methods such as DTW, non-matching subsequences (*gaps*) of these two sequences are ignored.

Subsequence Similarity: We are given two sequences $X = (x_1, \ldots, x_n)$, $Y = (y_1, \ldots, y_m)$ with $x_i, y_i \in \mathbb{R}$, $n, m \in \mathbb{N}$, $n \leq m$ and $\gamma, \epsilon \in \mathbb{R}^+$ with $\gamma \leq 1$. The sequences X and Y are called (γ, ϵ)-*similar*, if subsequences $X' = (x_{i_1}, \ldots, x_{i_{\lfloor \gamma \cdot n \rfloor}})$ and $Y' = (y_{j_1}, \ldots, y_{j_{\lfloor \gamma \cdot n \rfloor}})$ exist for all $k = 1, \ldots, \lfloor \gamma \cdot n \rfloor - 1$ with $i_k \leq i_{k+1}$ and $j_k \leq j_{k+1}$, such that $y_{j_k} - \epsilon \leq x_{i_k} \leq y_{j_k} + \epsilon$ for all $k = 1, \ldots, \lfloor \gamma \cdot n \rfloor$. The parameter γ determines the length of the subsequence of corresponding data points, ϵ controls how close (regarding the y-axis) matching points have to be. An example of two time series matched with LCSS-global is shown in Fig. 1. Similar points are matched, dissimilar subsequences (gaps) are ignored. The elements of X' and Y' are temporally ordered just as in X and Y, but they may consist of discontiguous fragments of the original sequences. Sketches for appropriate algorithms that determine such subsequences based on dynamic programming are set out in [12].

LCSS-Global Similarity: The overall similarity of two sequences X, Y based on LCSS-global with a user-defined $\epsilon \in \mathbb{R}^+$ is given by

$$Sim_\epsilon(X, Y) = \{max\, \gamma \mid X, Y \text{ are } (\gamma, \epsilon)\text{-similar}\}. \tag{1}$$

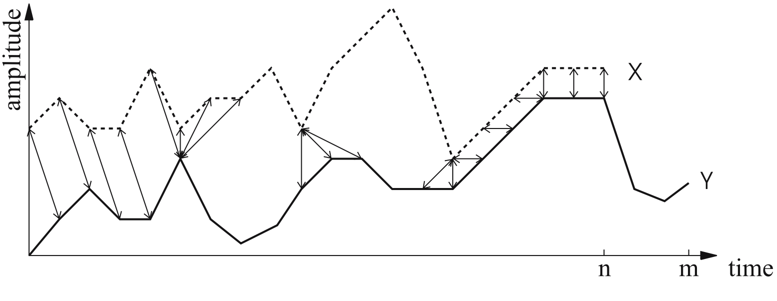

Fig. 1. Matching of two time series with LCSS-global

3.3 LCSS-Local

In LCSS-global, sequences must be rescaled to a common range in order to get suitable matchings. But, global rescaling may not reflect the similarity of two given sequences perfectly. LCSS-local allows the application of different local scaling functions to different subsequences of the time series. Additionally, very short discorresponding gaps are not ignored if they do not exceed a predefined length. Longer, discorresponding gaps are regarded as outliers and not matched. The algorithm is based on the method described in [13] which detects correlations of two sequences in three steps:

1. Computation of all ϵ-similar subsequences of a fixed length,
2. Iterative fusion of two subsequences to one longer subsequence, and
3. Determination of the longest common subsequence length.

Atomic Matchings: We are given two sequences $S = (s_1, \ldots, s_w)$ and $T = (t_1, \ldots, t_w)$ with equal length $w \in \mathbb{N}$ and $\epsilon \in \mathbb{R}^+$. S and T are ϵ-similar if $t_i - \epsilon \leq s_i \leq t_i + \epsilon$ for all $i = 1, \ldots, w$. Such a correlation is called an *atomic matching* of S and T.

In order to find all atomic matchings of two time series $X = (x_1, ..., x_n)$, $Y = (y_1, ..., y_m)$ with different lengths, they are split into all possible contiguous subsequences $\widetilde{S}_i = (x_i, \ldots, x_{i+w-1})$ and $\widetilde{T}_j = (y_j, \ldots, y_{j+w-1})$ with user-defined length w with $i = 1, \ldots, n - w + 1$ and $j = 1, \ldots, m - w + 1$. Subsequently, every subsequence $\widetilde{S}_i$ and $\widetilde{T}_j$ is rescaled to a specified interval by linear mapping functions f_i and g_j (with $f_i(x) = a_i x + b_i$ and $g_j(y) = c_j y + d_j$) in order to align the different ranges (e.g., rescaling to $[0, 1]$). With these local transformations, corresponding subsequences can now be found.

The rescaled subsequences S_i and T_j are compared pairwise and checked for ϵ-similarity. If two subsequences S_u and T_v are ϵ-similar, this is called an *atomic matching* (S_u, T_v) *of length* w of the two time series X and Y at points $u, \ldots, u + w - 1$ on X and $v, \ldots, v + w - 1$ on Y.

Longer Subsequences: Having computed all atomic matchings of a certain length w, the matching subsequences of X and Y must be fused to longer subsequences. Consider two atomic matchings (S_{i_1}, T_{j_1}) and (S_{i_2}, T_{j_2}) with $i_1 < i_2$ and $j_1 < j_2$. With *length*() denoting the length of a (sub-)sequence, these atomic matchings can be fused to longer subsequences if either the following conditions 1 and 3 or 2 and 3 hold (cf. Fig. 2):

1. The subsequences S_{i_1} and S_{i_2} are nonintersecting on X, that is, $i_1 + length(S_{i_1}) \leq i_2$. Moreover, the distance of two atomic matchings must be smaller than a given parameter ϱ: $i_2 - i_1 + length(S_{i_1}) \leq \varrho$. These conditions must also hold for T_{i_1} and T_{i_2}. It should be noticed that the gap between the two subsequences is included in the longer subsequence.
2. The two atomic matchings (S_{i_1}, T_{j_1}) and (S_{i_2}, T_{j_2}) intersect on both time series with the same length: $d = i_1 + length(S_{i_1}) - i_2 = j_1 + length(T_{j_1}) - j_2$.

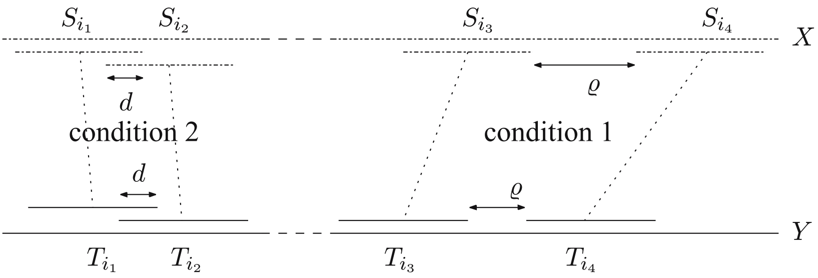

Fig. 2. Construction of longer subsequences

3. The parameters a_i, b_i, c_j and d_j used for scaling and translation are approximately equal: $|a_i - c_j| < \kappa_1$ and $|b_i - d_j| < \kappa_2$, with user-defined κ_1, κ_2.

Subsequently, longer subsequences can again be fused with other subsequences already fused as long as conditions 1 and 3 or 2 and 3 hold.

Longest Common Subsequences: Given a set with k pairs of matchings $\mathcal{S} = \{(S_1, T_1), \ldots, (S_k, T_k)\}$ that has been determined as described above. Now that subset $\mathcal{S}' = \{(S_{l_1}, T_{l_1}), \ldots, (S_{l_h}, T_{l_h})\}$ of $\mathcal{S}$ must be found for which

- The end points S_{l_i} and T_{l_i} precede the start points S_{l_j} and T_{l_j} on X and Y, respectively $(1 \leq i < j \leq h)$. That is, the corresponding subsequences do not overlap.
- The total length of all subsequences in this subset $\sum_{i=1}^{h} length(S_{l_i}) + \sum_{i=1}^{h} length(T_{l_i})$ is maximal.

With $\circ$ denoting the sequential composition of two subsequences, the subsequences $X' = S_{l_1} \circ \ldots \circ S_{l_h}$ and $Y' = T_{l_1} \circ \ldots \circ T_{l_h}$ are called the *longest common subsequences* of X and Y. X' and Y' may have different lengths.

LCSS-Local Similarity: Let X' and Y' be the longest common subsequences of X and Y, then the similarity $\mathcal{S}im_{w,\epsilon,\varrho}(X,Y)$ of X and Y computed with LCSS-local is given by

$$\mathcal{S}im_{w,\epsilon,\varrho}(X,Y) = \frac{length(X') + length(Y')}{length(X) + length(Y)}. \quad (2)$$

3.4 Extension to Multivariate Time Series

Multivariate time series $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ and $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_m)$ with $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^D$ with $D \in \mathbb{N}$ $(D \geq 2)$ can be processed with LCSS-global and LCSS-local as well. Therefor, $\mathcal{S}im_{\epsilon}(X_l, Y_l)$ or $\mathcal{S}im_{w,\epsilon,\varrho}(X_l, Y_l)$ $(l = 1, \ldots, D)$ are computed separately for each dimension l. Then, the similarity of the multivariate time series X and Y is computed by the average of the corresponding similarity measures.

3.5 LCSS-Global and LCSS-Local as Kernel Functions

As a kernel function $K(\cdot,\cdot)$ can be seen as a similarity measure of two samples, $Sim_\epsilon(X,Y)$ and $Sim_{w,\epsilon,\varrho}(X,Y)$ can be used as kernel functions:

$$K_{global}(X,Y) = \frac{2 \cdot Sim_\epsilon(X,Y) \cdot \min\{length(X), length(Y)\}}{length(X) + length(Y)}, \tag{3}$$

$$K_{local}(X,Y) = Sim_{w,\epsilon,\varrho}(X,Y). \tag{4}$$

Equations (3) and (4) do not define kernel functions that satisfy Mercer's conditions. I.e., $K(\cdot,\cdot)$ is not guaranteed to be positive semidefinite.

4 Experiments

The Biometric Smart Pen BiSP (see [15] for details) is a novel ballpoint pen for the acquisition of biometrical features based on handwriting movements which does not need a specific writing pad. For the verification of individuals by means of handwritten signatures the pen is equipped with sensors which measure the dynamics of pressure on the refill in three dimensions and the finger kinematics by means of tilt angels of the pen.

4.1 Database and Experimental Setup

For the following experiments, signatures of 71 persons have been recorded with the BiSP. The number of signatures available for a specific person varies from 6 to 40. In order to provide significant results, reference models are created only for those persons who provided at least 10 signatures (i.e., 63 persons).

The methods LCSS-global and LCSS-local presented in Section 3 are compared to other kernels, namely:

1. Fisher-kernel based on HMM [9] (Pre-processing: rescaling, 7 hidden states),
2. DTW-kernel [8] (Pre-processing: amplitude normalization and resampling to person-specific fixed length, standard deviation σ of the Gaussian function: person-specific), and
3. Gauss-kernel [14] (Pre-processing: amplitude normalization and resampling to person-specific fixed length, standard deviation σ of the Gaussian function: person-specific).

For LCSS-global all signatures are rescaled to $[0,1]$ and ϵ is set to 0.1. For LCSS-local no pre-processing is applied. The parameters are set to $w = 20$, $\epsilon = 0.2$, $\varrho = 15$ and local rescaling to $[0,1]$ is performed. These settings have been found empirically.

4.2 Results

The primary objectives of our experiments are the evaluation of SVM with LCSS-global and LCSS-local applied to online signature verification on the basis of two

Table 1. Experiment 1: Comparison with 3 originals for testing

Kernel	Error in %		
	FRR	FAR	TER
LCSS-global	3.81	0.24	0.57
LCSS-local	3.92	0.01	0.37
DTW-kernel	4.66	0.66	1.02
Fisher-kernel	14.18	3.79	4.73
Gauss-kernel	35.34	0.10	3.30

Table 2. Experiment 2: Comparison with 13 originals for testing

Kernel	Error in %		
	FRR	FAR	TER
LCSS-global	3.72	0.06	1.16
LCSS-local	3.08	0.00	0.93
DTW-kernel	5.38	0.00	1.63
Fisher-kernel	19.23	4.00	8.60
Gauss-kernel	37.44	0.00	11.32

signature sets with two different numbers of originals used for testing (experiment 1 and 2) and the performance of the proposed methods with varying numbers of original signatures used for the training (experiment 3).

In experiment 1, a training set containing 7 original signatures and 35 random forgeries selected randomly from the set of the 70 remaining persons is created for every person. Additionally, 3 originals and 30 random forgeries are selected for testing purposes and not used for training. Each of the five kernels is applied to each of the 63 data sets. Due to the fact that signatures used for training and testing are selected randomly, this experiment as well as the two following experiments are repeated 5 times in order to get a statistically more reliable result. Table 1 shows the average false rejection rates (FRR), the average false acceptance rates (FAR), and the average total error rates (TER). LCSS-local, LCSS-global, and the DTW-kernel yield the best results, with LCSS-local providing the lowest total error rate (TER = 0.37%). Fisher-kernel and Gauss-kernel seem not to be appropriate.

After this first experiment, the methods are evaluated with a second set of signatures (experiment 2): Therefor, 12 persons are selected who provided at least 20 signatures. 7 originals and 35 random forgeries are chosen for training as in experiment 1, but here, 13 originals and 30 random forgeries are taken for testing (see Table 2 for results). Due to the larger number of originals used for testing, the results are worse than in the first experiment. Again, LCSS-local provides the best results (TER = 0.93%) followed by LCSS-global and DTW-kernel.

In experiment 3, the number of originals used for training is varied between 3 and 17 and 35 random forgeries are used. For testing, 3 originals and 30 random forgeries are selected. Since LCSS-based methods provided the best results in experiments 1 and 2, only LCSS-global and LCSS-local are considered (see Fig. 3 for results). As the number of originals increases with LCSS-global and LCSS-local, FRR is decreasing rapidly. But in turn, as the variance of a person's signatures used for training increases due to the larger number of originals, FAR increases for both methods, but even faster for LCSS-global. It can be concluded that a number of seven originals effects a suitable compromise between FRR and FAR.

In a nutshell: LCSS-global, LCSS-local, and DTW-kernel provide the best results, with LCSS-local ranking first in each experiment. Fisher-kernels yield very high error rates just as Gauss-kernels. For a larger set of persons and only

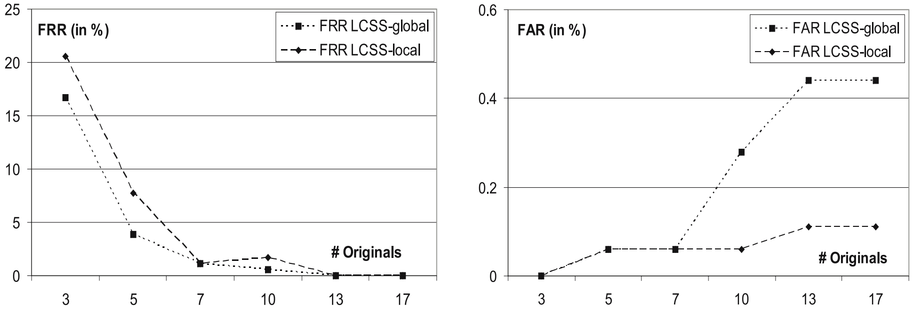

Fig. 3. Experiment 3: LCSS-global and LCSS-local with increasing number of originals used for training purposes

a few originals taken for the training of a reference model, LCSS-based methods are superior, whereas with an increasing amount of originals available for training purposes, LCSS-global and LCSS-local perform almost equally. But as the number of originals needed for very good authentication results is lowest with LCSS-local, it should be preferred.

5 Conclusion and Outlook

In this article, two new kernel functions (LCSS-global and LCSS-local) for SVM were proposed and applied to online signature verification. The experiments showed that these methods are capable to authenticate persons very reliable with LCSS-local providing the best results with only seven originals used for training purposes. It was also shown that in this particular application LCSS-local, LCSS-global, and DTW-kernel are superior to Gauss-kernel and Fisher-kernel.

Currently, we develop new time series segmentation and classification methods for online signature verification and intend to apply the new kernel functions to other time series classification tasks. To obtain a less biased evaluation of our methods, we also intend to use skilled forgeries. We also intend to improve our algorithms to reflect possible correlations between the individual dimensions of a multivariate time series and to compare the proposed kernels to a personalized version of the Fisher-kernel or related kernels (i.e., TOP-kernel [16]).

References

1. Thelusma F. and Muherjee S.: *Signature Verification*, Internal Report, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, URL: citeseer.ist.psu.edu/366946.html (06/03/2005)
2. Fuentes M., Garcia-Salicetti S., and Dorizzi B. *On-Line Signature Verification: Fusion of a Hidden Markov Model and a Neural Network via a Support Vector Machine* Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02), Ontario, Canada, 2002, pp. 253–258

3. Wan V.: *Speaker Verification using Support Vector Machines*, Ph.D. Thesis, University of Sheffield, GB, 2003
4. Rüping S.: *SVM Kernels for Time Series Analysis*, Tagungsband der GI-Workshop-Woche LLWA 01, Dortmund, 2001, pp. 43–50
5. Chakrabartty S. and Deng Y.: *Dynamic Time Alignment in Support Vector Machines for Recognition Systems*, Internal Report, The John Hopkins University, Baltimore, URL: bach.ece.jhu.edu/gert/courses/774/2001/dtw.pdf (06/03/2005)
6. Shimodaira H., Noma K., Nakai M., and Sagayama S.: *Dynamic Time-Alignment Kernel in Support Vector Machine*, Advances in Neural Information Processing 14 (NIPS 2001), 2001, pp. 921–928
7. Shimodaira H., Noma K., Nakai M., and Sagayama S.: *Support Vector Machine with Dynamic Time-Alignment Kernel for Speech Recognition*, Proceedings of the European Conference on Speech Communication and Technology (Eurospeech), Aalborg, 2001, vol. 3, pp. 1841–1844
8. Bahlmann C., Haasdonk B., and Burkhardt H.: *On-line Handwriting Recognition with Support Vector Machines – A Kernel Approach*, 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR '02), Ontario, 2002, pp. 49–54
9. Jaakkola T., Diekhans M., and Haussler D.: *Using the Fisher Kernel Method to Detect Remote Protein Homologies*, 7th International Conference on Intelligent Systems for Molecular Biology, Menlo Park, CA, 1999, pp. 149–158
10. Moreno P.J., Ho P.P., and Vasconcelos N.: *A Kullback-Leibler Divergence Based Kernel for SVM Classification in Multimedia Applications*, HP Laboratories Cambridge, Tech. Report HPL-2004-4, 2004
11. Das G., Gunopulos D. and Mannila H.: *Finding Similar Time Series*, 1st European Symposium on Principles of Data Mining and Knowledge Discovery, Trondheim, 1997, pp. 88–100
12. Bollobás B., Das G., Gunopulos D., and Mannila H.: *Time-Series Similarity Problems and Well-Separated Geometric Sets*, Proceedings of the 13th Annual ACM Symposium on Computational Geometry, Nice, 1997, pp. 454–456
13. Agrawal R., Lin K.-I., Sawhney H. S., and Shim K.: *Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases*, 21st International Conference on Very Large Databases, Zürich, 1995, pp. 490–501
14. Burges C. J. C.: *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery, vol. 2 (2), 1998, pp. 121–167
15. Hook C., Kempf J., and Scharfenberg, G.: *A Novel Digitizing Pen for the Analysis of Pen Pressure and Inclination in Handwriting Biometrics*, LNCS, Springer, Berlin, Heidelberg, New York, vol. 3087, 2004, pp. 283–294
16. Tsuda K., Kawanabe M., Rötsch G., Sonnenburg S. and Müller K.-R.: *A new discriminative kernel from probabilistic models*, Neural Computation, vol. 14 (10), MIT Press, Cambridge, MA, 2002, pp. 2397–2414