Signature Verification with Dynamic RBF Networks and Time Series Motifs

Christian Gruber, Michael Coduro, Bernhard Sick

Signature Verification with Dynamic RBF Networks and Time Series Motifs

Christian Gruber, Michael Coduro, Bernhard Sick
Institute of Computer Architectures, University of Passau, Germany
{gruberc,coduro,sick}@fmi.uni-passau.de

Abstract

This article presents a novel classification algorithm for (multivariate) time series. In a first step, so-called time series motifs, which represent characteristic subsequences of the time series, are extracted using extreme points. In a second step, the extracted motifs are used to train a dynamic radial basis function network (DRBF). Compared to a standard radial basis function network, this DRBF has the advantage, that not only similar motifs of the same class are detected but also sequences of these motifs. For performance evaluation, the proposed classification algorithm is applied to online signature verification. Our experiments show, that the presented DRBF based on time series motifs is capable of a very reliable authentication with an equal error rate of about 1.5%.

Keywords: signature verification, dynamic radial basis function network, motif, time series, biometrics

1 Introduction

Authentication of a person's identity is a tedium task. Often, this is done by means of signatures. However, in most cases a signature is compared to only one reference signature with the naked eye. Since authentication by signature is more widely accepted than any other technique (e.g., fingerprint or iris scan), an electronic signature verification system that ensures a high level of security must be developed. Typeface-based techniques (so-called *off-line signature verification*) can easily be outsmarted. Biometric signature verification systems that are based on the dynamics of a person's signature and not on its image (so-called *online signature verification*) are considerably better for a reliable authentication.

In the last years, time series data mining became a very active research topic, but methods based on motifs are rather new in this area. A motif can be defined as a characteristic pattern (i.e., subsequence), which is reoccurring within one and/or in different similar time series (see Fig. 1). In the context of online signature verification, a motif corresponds to a specific movement of the hand and/or the arm of the signing person, which is reoccurring similarly in every signature of this person and is, therefore, characteristic to this person.

But this definition of a motif and its application to online signature verification raises two questions: How can

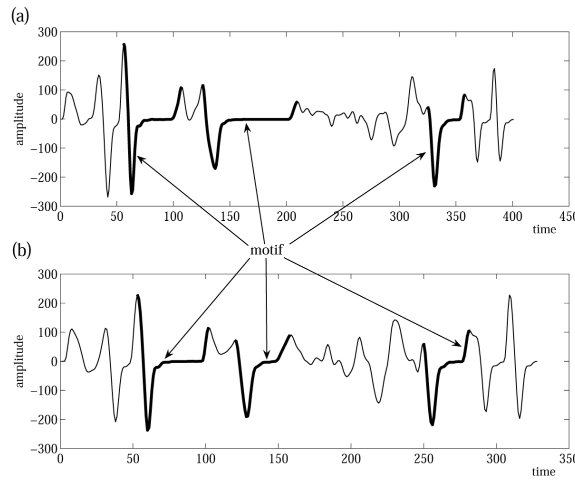

Figure 1. Examples for Motifs in Time Series

the similarity of two motifs be measured and how can the similarity of the temporal order of the motifs be assessed? Both problems can be solved by a specifically designed dynamic radial basis function network (DRBF), which provides several advantages such as very good classification performance and a fast training phase as no gradient-based algorithm is used.

The remainder of the article is organized as follows: In Section 2, some related work on motif-based time series data mining and online signature verification is presented. In Section 3.1, a distance measure for the computation of the similarity of two motifs is introduced. The algorithm for the extraction of motifs from time series is explained in Section 3.2, the classification of these motifs with dynamic RBF networks is set out in Section 3.3. Experimental results are presented in Section 4. Finally, Section 5 summarizes the major findings.

2 Related Work

Only a few researchers dealt with the extraction of motifs from time series so far, but as [5] states, motif extracted from time series can be used for a large field of applications. [14, 17] extracted motifs for various time series data mining applications. [20] computed motifs for multivariate time series, representing the 3-dimensional movement of body parts, to get information about repeated motion sequences of humans. For sensor data of a robot (sonar, camera, etc.), [16] used a motif-based approach to discover decision rules for the robot's actions. [9] used time series motifs to derive associative rules.

Motif based classification algorithms were not applied to online signature verification so far, but methods based on probabilistic models, such as Hidden Markov Models or Gaussian Mixture Models, are closely related to motif based algorithms, as they are also looking for characteristic, reoccurring patterns in sequences. Such models are widely used for online signature verification and provide very good authentication rates (e.g., [10, 19]). Other very popular algorithms for online signature verification are based on sophisticated distance measures for time series, such as Dynamic Time Warping [15] or Longest Common Subsequences [11].

3 Motif Based Dynamic RBF Network

In this section, the novel time series classification algorithm is described. The algorithm includes two major steps. First, the extraction of time series motifs (Section 3.2) and second, the classification of these motifs with a DRBF (see Section 3.3). But first, an appropriate distance measure for time series motifs is needed (Section 3.1).

3.1 Constrained Distance Measure

A key problem in time series data mining is the definition of significant distance measures. Often, simple methods, such as the Euclidian distance, or more sophisticated distance algorithms, such as dynamic time warping (DTW), are used. The Euclidian distance as well as DTW are very sensitive to varying offsets and different ranges. Therefore, to determine the similarity of two time series, they have to be normalized, e.g. with respect to a reference interval (e.g., $[0, 1]$) or to a given mean and variance (e.g., $mean = 0$ and $variance = 1$). But, with these transformations based on user-defined parameters, similarities within the time series are not reflected optimally. Sometimes, two time series are aligned by linear mapping functions, where the optimal parameters for scaling and translation are found heuristically [2] or analytically [6]. But without constraints on parameter ranges (particularly to avoid negative scaling parameters), such methods possibly transform dissimilar time series into similar ones. The method proposed below computes (with regard to linear transformations, the Euclidean distance, and the given constraints) the optimal distance of two time series analytically and constrains the scaling and translation parameters to a user-specified range.

We are given two univariate time series $T = (t_1, \dots, t_N)$ and $Q = (q_1, \dots, q_N)$ of equal length N. The minimal distance $d_{min}(T, Q)$ of T and Q with constraints is given by

$$d_{min}(T,Q) = \min_{\substack{a \in [a_{min}, a_{max}] \\ \wedge\, b \in [b_{min}, b_{max}]}} \sum_{i=1}^{N} (a \cdot t_i + b - q_i)^2. \tag{1}$$

To find a global optimum, d_{min} is defined as a convex quadratic optimization function with constraints. Generally, a convex quadratic optimization problem f can be defined by

$$f(\mathbf{x}) = \min_{\mathbf{x}} \mathbf{x}^T \mathbf{C} \mathbf{x} + \mathbf{p}^T \mathbf{x}, \tag{2}$$

with $\mathbf{x} \in \mathbb{R}^n$, a symmetric $(n \times n)$-Matrix $\mathbf{C}$ and an $(n \times 1)$ vector $\mathbf{p}$. If $\mathbf{C}$ is positive semidefinite, f is convex. If $\mathbf{C}$ is positive definite, f is strictly convex. The aim of quadratic optimization with constraints is to minimize f with regard to constraints posed on $\mathbf{x}$ (i.e., $\mathbf{x} \in M \subset \mathbb{R}^n$). If Eq. 1 is transferred to Eq. 2, $\mathbf{C}$, $\mathbf{p}$ and $\mathbf{x}$ are given by

$$\mathbf{C} = \begin{pmatrix} \sum_{i=1}^N t_i^2 & \sum_{i=1}^N t_i \\ \sum_{i=1}^N t_i & N \end{pmatrix}, \tag{3}$$

$$\mathbf{p} = \begin{pmatrix} -2 \cdot \sum_{i=1}^N (t_i \cdot q_i) \\ -2 \cdot \sum_{i=1}^N q_i \end{pmatrix}, \text{ and} \tag{4}$$

$$\mathbf{x} = \begin{pmatrix} a \\ b \end{pmatrix}, \tag{5}$$

with constraints $a \in [a_{min}, a_{max}],\ b \in [b_{min}, b_{max}]$. It can be proven, that $\mathbf{C}$ is positive definite if the time series T is not constant (see [8] for details). Therefore, the constrained minimal distance of T and Q can be computed using a standard method. Here, the algorithm of Hildreth and d'Esopo (see [3] for details) is used.

With this constrained distance measure, the dissimilarity of two time series T and Q can be computed by

$$D^{ab}(T, Q) = \min(d_{min}(T, Q), d_{min}(Q, T)). \tag{6}$$

As this dissimilarity measure depends on the lengths of the two time series, the normalized version

$$D^{ab}_{norm}(T,\, Q) = \frac{1}{N} D^{ab}(T, Q) \tag{7}$$

is used below.

It should be noted that D^{ab}_{norm} is not a metric as the triangle inequality is not satisfied.

For online signature verification, multivariate time series must be processed and the dissimilarity measure (Eq. 7) has to be extended appropriately. We are given two multivariate time series $\mathbf{T} = (\mathbf{t}_1, \dots, \mathbf{t}_N)$ and $\mathbf{Q} = (\mathbf{q}_1, \dots \mathbf{q}_N)$ of equal length N with $\mathbf{t}_i, \mathbf{q}_i \in \mathbb{R}^n$. The multivariate time series are transformed into univariate time series by

$$T' = (t_1^1, \dots, t_N^1, \dots, t_1^n, \dots, t_N^n) \tag{8}$$

with length $N \cdot n$ (and Q' analogously). The dissimilarity is then given by

$$D^{ab}_{norm}(\mathbf{T},\, \mathbf{Q}) = D^{ab}_{norm}(T',\, Q'). \tag{9}$$

Typically, two time series do not have equal lengths. To apply the introduced dissimilarity measure to such time series, it has to be extended once again. We are given two univariate time series $T = (T_1, \dots, T_N)$ and $Q = (Q_1, \dots, Q_M)$ with $M > N$. Then, the longer time series Q is aligned to T by resampling Q to N equidistant data points by means of a spline interpolation technique. The dissimilarity of T and Q is then given by

$$D^{ab}_{res}(T,\, Q) = D^{ab}_{norm}(T,\, Q_{resampled}). \tag{10}$$

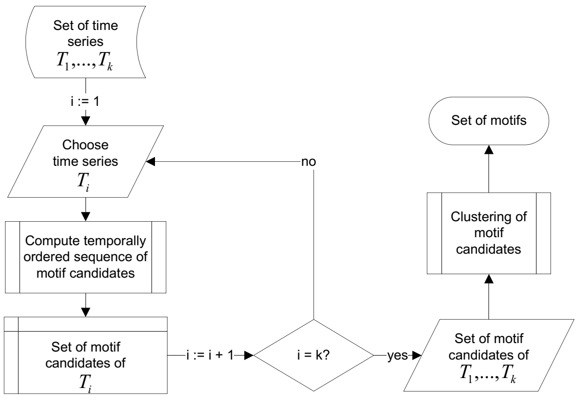

Figure 2. Extraction of Time Series Motifs

3.2 Time Series Motifs

In the first step of the proposed time series classification method, motifs have to be extracted from time series (see Fig. 2).

In a nutshell, this algorithm works as follows: Given a set of time series $T_1, \dots, T_k$, for every time series of this set a temporally ordered sequence of possible motifs (so-called motif candidates) is extracted. Afterwards, similar motifs of one time series are grouped by means of a clustering method (i.e., motifs reoccurring in a time series are only considered once) in order to determine a user-defined number of characteristic motifs given by cluster prototypes.

3.2.1 Motif Candidates of a Time Series

To extract a temporally ordered sequence of motif candidates, extreme points of a time series have to be found. As a time series recorded by a technical device contains many extreme points (e.g., due to noise) the significant ones have to be found.

A univariate time series T has a *significant minimum* at position m with $1 < m < N$, if $(t_i, \dots, t_j)$ with $1 \leq i < j \leq N$ in T exists, such that t_m is the minimum of all points of this subsequence and $t_i \geq \mathtt{R} \cdot t_m, t_j \geq \mathtt{R} \cdot t_m$ with user-defined $\mathtt{R} \geq 1$. A *significant maximum* is existent at position m with $1 < m < N$, if a subsequence $(t_i, \dots, t_j)$ with $1 \leq i < j \leq N$ in T exists, such that t_m is the maximum of all points of this subsequence and $t_m \geq \mathtt{R} \cdot t_i$, $t_m \geq \mathtt{R} \cdot t_j$ with user-defined $\mathtt{R} \geq 1$.

Fig. 3 illustrates the definition of significant minima (a) and maxima (b).

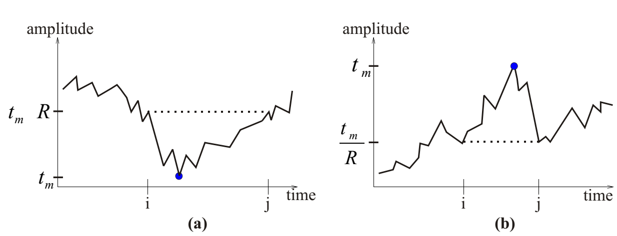

Figure 3. Illustration of Significant Extreme Points: (a) Minimum, (b) Maximum

Starting at the beginning of a time series T all significant minima and maxima of the time series are computed (see [18] for details). The result of this step is a sequence of extreme points $EP = (ep_1, \dots, ep_l)$. With these significant extreme points a set of motif candidates can be extracted. The method for the computation of this motif candidate sequence is set out in Algorithm 1.

```
function getMotifCandidateSequence (T)
    N = length (T);
    EP = findSignificantExtremePoints
    (T,R);
    maxLength = MAX_MOTIF_LENGTH ;
    for i = 1 to (length (EP)−2) do
        motifCandidate = getSubsequence (T, ep_i,
        ep_{i+2});
        if length (motifCandidate)> maxLength
        then
            addMotifCandidate (resample
            (motifCandidate, maxLength));
        else
            addMotifCandidate
            (motifCandidate);
        end
    end
end
```

Algorithm 1: Extraction of a Set of Motif Candidates

In Algorithm 1, all significant extreme points $EP(T)$ of a time series T are extracted first. Then, all motif candidates are computed iteratively. A motif candidate $MC_i(T),\ i = 1, \dots, l-2$ is represented by that subsequence of T that is bounded by the two extreme points ep_i and ep_{i+2}. Additionally, a motif candidate is resampled using a spline interpolation method to a user-defined MAX_MOTIF_LENGTH, if the length of this candidate exceeds this maximal user-defined length. This parameter is especially useful, as the dissimilarity measure described in Section 3.1 is used later to determine the distance of two motifs. As the resampling used within this distance measure only aligns the longer motif to the shorter one, which can still be longer than MAX_MOTIF_LENGTH and provides only global temporal compression and dilation, the restriction of the length of a motif candidate is reasonable. But, with an increasing motif length, local temporal compressions and dilations, which have a negative impact on the accuracy of the proposed distance measure, accumulate with an increasing probability (e.g., due to different writing speeds within corresponding parts of signatures of a person).

The result of this Algorithm 1 is a sequence of motif candidates $MCS(T) = (MC_1(T), \dots, MC_{l-2}(T))$ of an univariate time series $T = (t_1, \dots, t_N)$ with $MC_i(T) = (t_{ep_i}, \dots, t_{ep_{i+2}}),\ i = 1, \dots, l-2$.

3.2.2 Clustering of Motif Candidates

Having extracted the motif candidates from all time series $T_1, \dots T_k$, a large set of motif candidates $MCS = \{MCS(T_1), \dots, MCS(T_k)\}$ is obtained. As some motif candidates reoccur within the same time series or even within several similar time series, not all of these similar motifs are needed for classification. Similar motif candidates can be combined and be replaced by one prototype. This is achieved by clustering the motif candidates. Here, we use a hierarchical bottom-up algorithm (see Algorithm 2 for a generic version and [21] for details).

function `getHierarchicalClustering` (MCS, u, d)
    $\mathbf{C}$ = `getInitialClustering` (MCS);
    while `size` ($\mathbf{C}$) $> u$ **do**
        [$\mathbf{C}_i$, $\mathbf{C}_j$]=`getMostSimilarClusters` ();
        `addCluster` ($\mathbf{C}$, `mergeClusters` ($\mathbf{C}_i$, $\mathbf{C}_j$));
        `removeCluster` ($\mathbf{C}$, $\mathbf{C}_i$);
        `removeCluster` ($\mathbf{C}$, $\mathbf{C}_j$);
    end
    return $\mathbf{C}$;
end

Algorithm 2: Clustering Algorithm

For this clustering, the data to be clustered (MCS), the desired number of clusters u, and a distance function d have to be provided. First, an initial partition $\mathbf{C}$ is determined (i.e., every single motif represents a cluster, $|\mathbf{C}| = |MCS|$). Then, the following steps are repeated until the number of clusters equals u. In the first step, the two most similar clusters are determined. Therefor, the distance of two clusters is computed by the average linkage criterion

$$D^{cluster}(\mathbf{C_i}, \mathbf{C_j}) = \frac{1}{NM} \sum_{\substack{1 \le i \le N,\\ 1 \le j \le M}} D^{ab}_{res}(MC_i, MC_j) \tag{11}$$

with motif candidates MC_i $(i = 1 \ldots N)$ and MC_j $(j = 1 \ldots M)$ denoting the members of the two clusters $\mathbf{C_i}$ $(|\mathbf{C_i}| = N)$ and $\mathbf{C_j}$ $(|\mathbf{C_j}| = M)$, respectively.

As prototype $MC_{\mathbf{C}_i}$ of a cluster $\mathbf{C}_i$ the median center is used which is given by the following property

$$\begin{aligned}\forall MC_j \in \mathbf{C}_i :\ & med\left(D^{ab}_{res}(MC_{\mathbf{C}_i}, MC_i) | MC_i \in \mathbf{C}_i\right) \\ & \le med\left(D^{ab}_{res}(MC_j, MC_i) | MC_i \in \mathbf{C}_i\right). \end{aligned} \tag{12}$$

The two most similar clusters are then merged into one, a new prototype for this cluster is computed, and the two most similar clusters are removed from the set of clusters.

The result of this clustering of MCS is a set of motifs MS with $|MS| = u$ consisting of the prototypes of the u clusters $\mathbf{C}_1, \ldots, \mathbf{C}_u$.

3.3 Dynamic RBF networks

RBF networks are widely used for various pattern recognition problems [1]. In general, RBF networks realize nonlinear functions $f : \mathbb{R}^n \to \mathbb{R}^s$. Here, we only need networks with one output neuron which realize a nonlinear function

$$f(\mathbf{x}) = \sum_{j=1}^{u} w_j \phi_j(\mathbf{x}) \tag{13}$$

with $\mathbf{x} \in \mathbb{R}^n$, w_j denoting the weights of the network and $\phi_j(\mathbf{x})$ being a radial basis function, such as the Gaussian function defined by

$$\phi_j(\mathbf{x}) = e^{-\frac{\left\|(\mathbf{x}-\mathbf{c}_j)^2\right\|}{\sigma_j^2}}. \tag{14}$$

$\mathbf{c}_j$ represents the prototype of the Gaussian function and σ_j its standard deviation. The Gaussian function provides a nonlinear transformation of samples into the so-called feature space. When used for classification, these networks aim at separating different classes in the feature space by means of hyperplanes (linear separation). That is, if they were applied to time series classification task (e.g., online signature verification) temporal patterns within the time series, such as characteristic sequences of motifs in the signature of a person, are not considered. But these networks can be extended to show dynamic instead of static system behavior (so-called Dynamic RBF networks). Therefor, the products $w_j \phi_j(\mathbf{x})$ are replaced by FIR filters (finite impulse response filters) [7]. Given a multivariate time series $\mathbf{T} = (\mathbf{t}_1, \ldots, \mathbf{t}_N)$, such Dynamic RBF networks (DRBF) are defined by

$$f(\mathbf{t}_{pos}) = \sum_{j=1}^{u} \sum_{i=0}^{del-1} w_j^{(i)} \cdot \phi_j(\mathbf{t}_{pos-i}) \tag{15}$$

with $pos \in \{del, \ldots, N\}$ and maximum delay $del \in \mathbb{N}$.

If such a DRBF is used for the classification of a time series $\mathbf{T}$, from which a sequence of motif candidates $MCS(\mathbf{T})$ has been extracted, every motif candidate of the sequence is compared to the prototypes of all radial basis functions, determining the similarity of the motif candidates to the stored motifs (i.e., prototypes that have been determined by means of training data as described before). Then, the temporal relationship of these similarities is evaluated by the FIR filters. Mathematically, the sequence of motif candidates $MCS(\mathbf{T})$ of a time series $\mathbf{T}$ is processed by a DRBF with

$$f(MC_{pos}(\mathbf{T})) = \sum_{j=1}^{u} \sum_{i=0}^{del-1} w_j^{(i)} \cdot \phi_j^{D^{ab}_{res}}(MC_{pos-i}(\mathbf{T})) \tag{16}$$

with

$$\phi_j^{D^{ab}_{res}}(MC_{pos-i}(\mathbf{T})) = e^{-\omega \cdot \frac{D^{ab}_{res}(MC_{\mathbf{C}_j}, MC_{pos-i}(\mathbf{T}))}{\sigma_j^2}} \tag{17}$$

and $\omega \in \mathbb{R}^+$.

We are given a binary time series classification task $\mathcal{L} = \{\mathbf{T}_i, y_i\}$, $i = 1, \ldots, k$ with $\mathbf{T}_i = (\mathbf{t}_1, \ldots, \mathbf{t}_{N_i})$, $N_i \in \mathbb{N}$ and $y_i \in \{+1, -1\}$. For the training of the DRBF, a two-stage method similar to the one described in [4] is used. In the first stage a sequence of motif candidates is extracted from every time series $\mathbf{T}_i$, which is then used to determine the prototypes and the radii of the radial basis functions. Therefor, the described clustering method is used with a number of $u = |MCS| \cdot \mathtt{NMF}$ ($\mathtt{NMF} \in]0, 1]$) clusters according to the number of radial basis functions to determine the prototypes $MC_{\mathbf{C}_j}$. Furthermore, with this clustering the radii σ_j are computed by the maximal distance of a cluster prototype to all other members of the same cluster $\mathbf{C}_j$.

In stage two, the coefficients $w_j^{(i)}$ are computed by solving a linear least-squares problem. The coefficients are then optimal (in a least-squares sense) with respect to the selected prototypes and radii. Details concerning the solution of this problem can be found in [4].

Finally, a threshold has to be computed that can be used to assign a class label $+1$ or -1 to a given time series. Therefor, the means of the output sequences of the $MCS(\mathbf{T}_i)$ of all $\mathbf{T}_i$ in the training data are determined by

$$\mu_{\mathbf{T}_i} = \frac{1}{|MCS(\mathbf{T}_i)| - del + 1} \sum_{pos=del}^{|MCS(\mathbf{T}_i)|} f(MC_{pos}(\mathbf{T}_i)). \tag{18}$$

The threshold θ is then defined by

$$\theta = \max_{\forall i: y_i = -1} \mu_{\mathbf{T}_i} + \left[\min_{\forall i: y_i = +1} \mu_{\mathbf{T}_i} - \max_{\forall i: y_i = -1} \mu_{\mathbf{T}_i} \right] \cdot \mathtt{thf}, \tag{19}$$

with a user specified threshold factor $\mathtt{thf} \in \mathbb{R}^+$.

After the training, a test time series $\mathbf{T} = (\mathbf{t}_1, \ldots, \mathbf{t}_N)$ with a sequence of motif candidates $MCS(\mathbf{T})$ is classified by $+1$ if $\mu_{\mathbf{T}} \geq \theta$ and -1 otherwise.

4 Experimental Results

The Biometric Smart Pen BiSP (see [13] 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 angles of the pen at a sampling rate of 500 Hz (see Fig. 4 for a sample signature recorded with the BiSP).

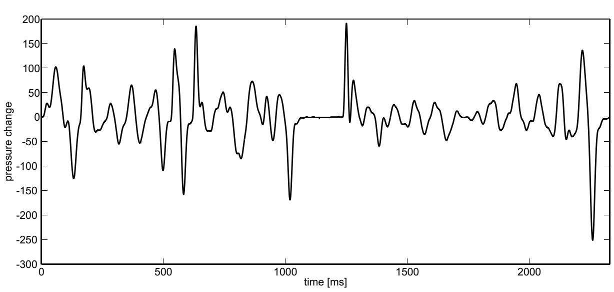

Figure 4. Pressure Change of a Sample Signature

For our experiments, signatures of 65 persons have been recorded with the BiSP. These 65 persons can be divided into two groups. The 29 persons of group A signed 12 times in two sessions on different days (6 signatures per session). The remaining 36 persons of group B signed 6 times in only one single session.

For every person of group A a reference model has been created with the algorithm presented in Section 3. The user-defined parameters used for model creation are set out in Table 1. These parameters have been found empirically. The person-specific constraints a_{min}, a_{max} and b_{min}, b_{max} of the distance measure D^{ab}_{res} are computed heuristically using the training set $\mathcal{L}$ (see [8] for details).

For training purposes, a varying number of originals (3 to 7) and random forgeries (35 to 60) is used. As random

Table 1. User-defined parameters for model creation

parameter	value
`R`	1.09
`MAX_MOTIF_LENGTH`	100
del	4
ω	0.1
`NMF`	0.2

forgeries, randomly selected signatures of the remaining 64 persons are taken. For testing, all left genuine signatures of the trained person (i.e., 9 to 5 originals) and 2 randomly selected signatures of each of the remaining 64 persons (i.e., 128 random forgeries) are taken. It should be noticed, that the signature sets used for training and testing are disjoint and this experiment was repeated 10 times to obtain statistically significant results.

Table 2. EERs with different numbers of originals and random forgeries used for training

# forgeries \ # originals	3	4	5	6	7
35	3.44%	3.07%	2.44%	2.19%	2.13%
40	3.56%	2.55%	2.32%	2.12%	1.75%
45	2.94%	2.44%	2.32%	1.82%	1.75%
50	3.19%	2.25%	1.93%	1.69%	1.93%
55	2.69%	2.19%	1.94%	1.69%	1.25%
60	2.75%	2.38%	1.56%	1.45%	1.32%

The resulting equal error rates (EER) of this experiment are set out in Table 2. It can be seen, that the EER is dropping with an increasing number of genuine signatures used for training. This means, that the verification of a person's identity becomes more reliable if more genuine signatures are provided for model creation. Another interesting aspect is the decrease of the EER if the number of random forgeries used for training is increased. A large number of random forgeries is easier to obtain than a large number of genuine signatures, because signatures of other persons stored in the database can be used as random forgeries. This means, that a more reliable verification model can be created just using more random forgeries. Table 2 shows for example, that with 5 genuine signatures and 35 random forgeries, an EER of 2.44% is achieved. If the number of random forgeries is increased to 60, the EER is dropping from 2.44% to 1.56%.

In a nutshell, a number of 5 genuine signatures and 60 random forgeries used for training seems to be a good compromise between a reliable authentication and a reasonable effort for the registering person.

Fig. 5 shows the false rejection rate (FRR) and the false acceptance rate (FAR) depending on the threshold factor `thf`. If one is looking for a more secure system, `thf` has to be increased. If the verification system should be more comfortable for the users, a lower FRR has to be provided, which can be achieved by decreasing `thf`.

5 Conclusion and Outlook

In this article, a new time series classification method based on time series motifs and dynamic radial basis function networks has been presented. The new algorithm has been applied to online signature verification. The experi-

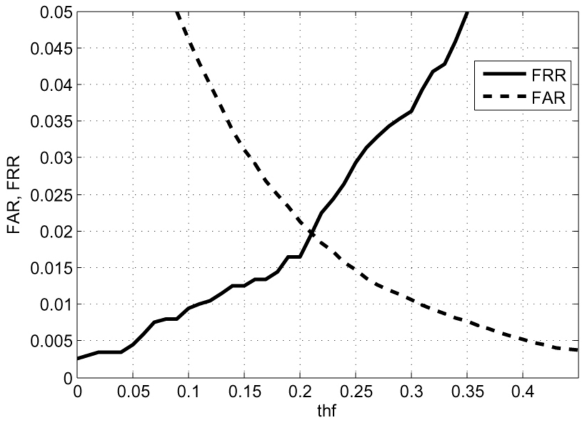

Figure 5. FAR and FRR depending on `thf`

ments showed, that the new algorithm is capable of a reliable authentication with equal error rates of 1.32% depending on the number of originals and random forgeries used for training.

Compared to our previous approach [11] to online signature verification based on Support Vector Machines (SVM), the method proposed here yields slightly higher error rates with 7 genuine signatures used for training. However, the presented algorithm yields much better authentication rates with a number of originals less than 7. Even with 3 genuine signatures used for training, the EER is still at an acceptable level.

In our future work, we want to evaluate the proposed method with skilled forgeries as well. Furthermore, we are integrating the presented algorithm into a software framework for online signature verification [12] and evaluate its performance in an ensemble of different classification methods. We also will apply the proposed algorithm to other time series classification tasks.

References

[1] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1994.

[2] B. Bollobás, G. Das, D. Gunopulos, and H. Mannila, "Time-series similarity problems and well-separated geometric sets," in *Proc. of the 13th SCG*, Nice, 1997, pp. 454–456.

[3] I. Bronstein and K. Semendjajew, *Taschenbuch der Mathematik*. B.G. Teubner, 1991.

[4] O. Buchtala, A. Hofmann, and B. Sick, "Fast and efficient training of RBF networks," in *LNCS*, vol. 2714. Springer, 2003, pp. 43–51.

[5] B. Chiu, E. Keogh, and S. Lonardi, "Probabilistic discovery of time series motifs," in *Proc. of the 9th International Conference on Knowledge Discovery and Data Mining*, Washington, 2003, pp. 493–498.

[6] K. K. W. Chu and M. H. Wong, "Fast time-series searching with scaling and shifting," in *Proc. of the 18th PODS*, Philadelphia, 1999, pp. 237–248.

[7] I. B. Ciocoiu, "Time series analysis using RBF networks with FIR/IIR synapses," in *Neurocomputing*, vol. 20, 1998, pp. 57–66.

[8] M. Coduro, "Online-Unterschriftenverifikation auf der Basis von Motiven aus Zeitreihen," Master's thesis, University of Passau, 2006.

[9] G. Das, K.-I. Lin, H. Mannila, G. Renganathan, and P. Smyth, "Rule discovery from time series," in *Proc. of the KDD-98*, New York City, 1998, pp. 16–22.

[10] J. Dolfing, E. Aarts, and J. Van Oosterhout, "On-line signature verification with hidden markov models," in *Proc. of the 14th ICPR*, Brisbane, 1998, pp. 1309–1312.

[11] C. Gruber, T. Gruber, and B. Sick, "Online signature verification with new time series kernels for support vector machines," in *LNCS*, vol. 3832. Springer, 2006, pp. 500–508.

[12] C. Gruber, C. Hook, J. Kempf, G. Scharfenberg, and B. Sick, "A flexible architecture for online signature verification based on a novel biometric pen," in *Proc. of the SMCals '06*, Logan, USA, 2006, pp. 110–115.

[13] C. Hook, J. Kempf, and G. Scharfenberg, "A novel digitizing pen for the analysis of pen pressure and inclination in handwriting biometrics," in *LNCS*, vol. 3087. Springer, 2004, pp. 283–294.

[14] J. Lin, E. Keogh, S. Lonardi, and P. Patel, "Finding motifs in time series," in *Proc. of the 2nd WTDM*, Edmonton, 2002, pp. 53–68.

[15] R. Martens and L. Claesen, "Dynamic programming optimisation for on-line signature verification," in *Proc. of the 4th ICDAR*, Ulm, 1997, pp. 653–656.

[16] T. Oates, M. D. Schmill, and P. R. Cohen, "A method for clustering the experiences of a mobile robot that accords with human judgments," in *Proc. of the 17th AAAI and 12th IAAI*, Austin, 2000, pp. 846–851.

[17] P. Pranav, E. Keogh, J. Lin, and S. Lonardi, "Mining motifs in massive time series databases," in *Proc. of the ICDM '02*, Maebashi City, 2002, pp. 370–377.

[18] K. B. Pratt and E. Fink, "Search for patterns in compressed time series," *International Journal of Image and Graphics*, vol. 2, no. 1, pp. 89–106, 2002.

[19] J. Richiardi and A. Drygajlo, "Gaussian mixture models for on-line signature verification," in *Proc. of the WBMA '03*, Berkley, 2003, pp. 115–122.

[20] Y. Tanaka, K. Iwamoto, and K. Uehara, "Discovery of time series motif from multi-dimensional data based on mdl principle," in *Machine Learning*. Springer, 2005, vol. 58, pp. 269–300.

[21] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Academic Press, 1999.