JAISCR, 2020, Vol. 10, No. 3, pp. 173 – 187
10.2478/jaiscr-2020-0012

AN ALGORITHM FOR THE EVOLUTIONARY-FUZZY GENERATION OF ON-LINE SIGNATURE HYBRID DESCRIPTORS

Marcin Zalasiński[1,*], Krzysztof Cpałka[1], Łukasz Laskowski[2], Donald C. Wunsch II[3], Krzysztof Przybyszewski[4]

[1] *Czestochowa University of Technology, Department of Computational Intelligence, Częstochowa, Poland*

[2] *Polish Academy of Sciences, Institute of Nuclear Physics, Krakow, Poland*

[3] *Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA*

[4] *Information Technology Institute, University of Social Sciences, Łódź, Poland and Clark University, Worcester, MA 01610, USA*

[*] *E-mail: marcin.zalasinski@pcz.pl*

Submitted: 3rd October 2019; Accepted: 6th April 2020

Abstract

In biometrics, methods which are able to precisely adapt to the biometric features of users are much sought after. They use various methods of artificial intelligence, in particular methods from the group of soft computing. In this paper, we focus on on-line signature verification. Such signatures are complex objects described not only by the shape but also by the dynamics of the signing process. In standard devices used for signature acquisition (with an LCD touch screen) this dynamics may include pen velocity, but sometimes other types of signals are also available, e.g. pen pressure on the screen surface (e.g. in graphic tablets), the angle between the pen and the screen surface, etc. The precision of the on-line signature dynamics processing has been a motivational springboard for developing methods that use signature partitioning. Partitioning uses a well-known principle of decomposing the problem into smaller ones. In this paper, we propose a new partitioning algorithm that uses capabilities of the algorithms based on populations and fuzzy systems. Evolutionary-fuzzy partitioning eliminates the need to average dynamic waveforms in created partitions because it replaces them. Evolutionary separation of partitions results in a better matching of partitions with reference signatures, eliminates disproportions between the number of points describing dynamics in partitions, eliminates the impact of random values, separates partitions related to the signing stage and its dynamics (e.g. *high* and *low* velocity of signing, where *high* and *low* are imprecise-fuzzy concepts). The operation of the presented algorithm has been tested using the well-known BioSecure DS2 database of real dynamic signatures.

Keywords: biometrics, on-line signature, dynamic signature, dynamic signature verification, evolutionary-fuzzy signature partitioning, horizontal and vertical partitioning.

1 Introduction

The security of information systems is associated with the security of the methods used in an identity verification of their users. Such verification may be based on the analysis of biometric features. Behavioral features describing human behavior are difficult to forge. The on-line signature takes an important place in the group of these characteristics [20] because it is not controversial and is a commonly acceptable way of identity verification. However, as a result, on-line signatures can be exposed to numerous falsification attempts. Therefore, we should aim at high precision in the analysis of waveforms describing the dynamics of the signature. Such dynamics describes individual habits of users very well. Therefore, effective methods for on-line signature verification are still being sought. For this purpose, artificial intelligence methods are used, especially those from the soft computing group [4, 10, 39, 41, 43, 44, 46]. Methods used in the on-line signature verification can also be used in the analysis of other behavioral attributes because they can be encoded in a similar way in which signatures are.

1.1 Methods presented in the literature

The precision of on-line signature processing has acted as a motivational springboard for developing methods that use the well-known principle of problem decomposition. In the context of on-line signature verification, this approach is known in the literature as the regional approach [20]. Dynamic signature partitioning is consistent with this approach.

In [16] signatures are segmented into strokes and for each of them, a reliability measure is computed on the basis of the feature values which belong to the current stroke. In [21] a stroke-based algorithm that splits the velocity signal into three bands was presented. This approach assumes that high and low-velocity bands of the signal are unstable, whereas the medium-velocity band is useable for discrimination purposes. In [12] a signature verification using Hidden Markov Models is presented. In [17] the authors present an approach in which decomposition of the signature shape is performed on the basis of the pressure and velocity profiles. Pressure and velocity are partitioned into high and low regions and their underlying horizontal and vertical trajectories are extracted on the basis of these subsets. In the classification phase, only the most stable partition is used. Classification is performed using a decision boundary determined in the two-dimensional space. In [9] and [26] a multi-section vector quantization algorithm for on-line signature recognition is presented. This method is an improved version of the classical vector quantization. In [7] presented is an approach based on co-called hybrid partitions created as intersections of separated horizontal and vertical sections which have a fixed size. The signature is classified using a flexible-fuzzy system. [38] presents a description of the method which selects the most characteristic partitions in the context of each individual separately.

In this work, we propose a new partitioning algorithm that uses the capabilities of fuzzy systems [3] and population-based algorithms [28]. Population-based algorithms are currently often used in real applications as the main and supporting methods alike [2, 24, 34, 43]. The main feature of the presented algorithm is that it eliminates the need to average dynamic waveforms in created partitions because it actually replaces them.

1.2 Method presented in this paper

A summary of the main features of on-line signature verification methods representing a regional approach is summarized in Table 1. The evolutionary-fuzzy method of signature partitioning presented in this paper is distinguished by the following main features:

– It makes partitions better suited to reference signatures. Evolutionary fitting is not based on averaging of signal groups as in the methods previously presented. Signal averaging is correct but it hinders processing of outlier values. Evolutionary mapping of signature points to hybrid-type partitions has not been considered in the literature before.

– It eliminates disproportions between the number of points in separated partitions. This approach can be conveniently implemented in evolutionary partitioning and is associated with an appropriate definition of the evaluation function of the individuals in a population. Elimination of disproportion in the number of points in hybrid partitions has not been previously considered in the literature.

Table 1. Main characteristics of the algorithms for the on-line signature verification based on the regional approach (**f1** - Does the method divide the signature into parts in order to increase the efficiency of signature verification accuracy? **f2** - Does the method focus on fast performance? **f3** - Does the method evaluate signature stability of in selected parts of the signature? **f4** - Does the method take into account the hierarchy of selected parts of the signature in the classification process? **f5** - Is the way of classification interpretable?) **f6** - Is signature partitioning supported by a population-based algorithm?

Characteristics of the method	**f1**	**f2**	**f3**	**f4**	**f5**	**f6**
Huang and Hong [16]	**yes**	no	**yes**	**yes**	no	no
Khan et al. [21]	**yes**	no	**yes**	no	no	no
Fierrez et al. [12]	**yes**	no	no	no	no	no
Ibrahim et al. [17]	**yes**	no	**yes**	no	no	no
Faúndez-Zanuy and Pascual-Gaspar [9]	**yes**	**yes**	no	no	no	no
Pascual-Gaspar et al. [26]	**yes**	**yes**	no	no	no	no
Cpałka and Zalasiński [7]	**yes**	no	**yes**	**yes**	**yes**	no
Zalasiński and Cpałka [38]	**yes**	no	**yes**	**yes**	**yes**	no
our method	**yes**	no	**yes**	**yes**	**yes**	**yes**

The method presented in this work is also characterized by other features. They can be summarized as follows:

– It creates signature partitions with the following interpretation: high and low velocity at the initial, middle and final moments of the signing process, high and low pen pressure at the initial, middle and final moments of the signing process [6, 8, 40, 42].

– It determines the value of the weight of importance for each created partition [29, 31]. Partitions' weights are used in the verification phase of test signatures.

– It uses the fuzzy sets and systems theory in assessment of imprecise similarity of test signatures to reference signatures [5, 30, 37].

– It is based on four types of signals: shape signals (x and y), pen pressure signal on the surface of a graphic tablet z, and pen velocity signal v.

This paper is organized into 4 Sections. Section 2 contains a description of the components of the presented algorithm. In Section 3 a detailed description of the algorithm is shown. The simulation results are presented in Section 4 while the conclusions are drawn in Section 5.

2 General description of the components of the presented algorithm

The presented algorithm for the evolutionary generation of hybrid descriptors of the on-line signature uses the flexible Mamdani-type fuzzy system (Section 2.1) and the differential evolution algorithm (Section 2.2).

2.1 Fuzzy system

The information on the implementation of individual components of the Mamdani-type flexible fuzzy system (FS) [28] is presented in Sections 2.1.1-2.1.3.

2.1.1 Notation of fuzzy sets

In this paper, we use Gaussian fuzzy sets. We have adopted a notation that allows maintaining the value of the Gaussian membership function above or below a certain limit value of linguistic variable $\overline{xA}$ at a constant level 1

$$\mu_A\left(x,\overline{xA},\overline{\sigma A},pL,pR\right) = \max\left\{\begin{array}{c}\operatorname{sgn}\left(-x+\overline{xA}\right)-(1-pL),\\ \exp\left(-\left(\frac{x-\overline{xA}}{\overline{\sigma A}}\right)^2\right),\\ \operatorname{sgn}\left(+x-\overline{xA}\right)-(1-pR)\end{array}\right\}, \tag{1}$$

where $\left\{\overline{xA},\overline{\sigma A}\right\}$ are the center and width of the Gaussian function, $\operatorname{sgn}(\cdot)$ is the signum function, and $\{pL,pR\}$ are the saturation parameters of the function (see Figure 1).

2.1.2 Notation of fuzzy rules

The FS presented herein uses *nRules* fuzzy rules $\{rule_1, rule_2, \ldots rule_{nRules}\}$. Each $rule_m$ has the following form

$$
rule_m : \left[\begin{array}{c}
\textbf{IF } x_{p=1,r=1} \textbf{ IS } A_{i,p=1,r=1,m} \\
\textbf{WITH } w_{i,p=1,r=1} \textbf{ AND } \ldots \\
x_{p=1,r=R} \textbf{ IS } A_{i,p=1,r=R,m} \\
\textbf{WITH } w_{i,p=1,r=R} \textbf{ AND } \ldots \\
x_{p=P,r=1} \textbf{ IS } A_{i,p=P,r=1,m} \\
\textbf{WITH } w_{i,p=P,r=1} \textbf{ AND } \ldots \\
x_{p=P,r=R} \textbf{ IS } A_{i,p=P,r=R,m} \\
\textbf{WITH } w_{i,p=P,r=R} \textbf{ AND} \\
\textbf{THEN } y \textbf{ IS } B_m
\end{array} \right], \quad (2)
$$

where $x_{p,r}$ are input linguistic variables, y is the output linguistic variable, $A_{i,p,r,m}$ are the input fuzzy sets of rule m of user i, B_m is the output fuzzy set of rule m of user i, $w_{i,p,r}$ are the weights of input fuzzy sets of user i ($w_{i,p,r} \in \langle 0,1 \rangle$) (shared by the rules). In the standard notation of the rule base, components $w_{i,p,r}$ are not used. Their use introduces a hierarchy of importance - we dealt with these issues in our previous works [3].

2.1.3 Notation of the fuzzy system

When singleton-type fuzzification and centre of area defuzzification are used [28], output signal $\bar{y}$ of the FS based on one output and rules (2) has the following form

$$
\bar{y} = \frac{\sum\limits_{r=1}^{nRules} \overline{xB}_r \cdot \mathop{S}\limits_{m=1}^{nRules} \left(T\left(\tau_m, \mu_{B_m}\left(\begin{array}{c} \overline{xB}_r, \overline{xB}_m, \\ \overline{\sigma B}, pL_m, \\ pR_m \end{array} \right) \right) \right)}{\sum\limits_{r=1}^{nRules} \mathop{S}\limits_{m=1}^{nRules} \left(T\left(\tau_m, \mu_{B_m}\left(\begin{array}{c} \overline{xB}_r, \overline{xB}_m, \\ \overline{\sigma B}, pL_m, \\ pR_m \end{array} \right) \right) \right)}, \quad (3)
$$

where τ_m is the activation level of rule $rule_m$ determined as follows

$$
\tau_m = \mathop{T^*}\limits_{\substack{p=1,\\ r=1}}^{\substack{p=P,\\ r=R}} \left(\begin{array}{c} \mu_{A_{i,p,r,m}} \left(\begin{array}{c} \bar{x}_{p,r}, \\ \overline{xA}_{i,p,r,m}, \\ \overline{\sigma A}_{i,p,r}, \\ pL_m, pR_m \end{array} \right); \\ w_{i,p,r} \end{array} \right). \quad (4)
$$

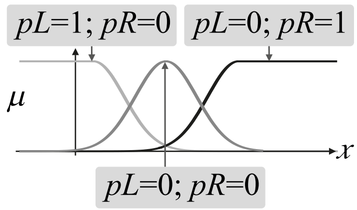

Figure 1. Interpretation of parameters $\{pL, pR\}$ of Gaussian membership function (1) adopted in this paper.

In formulas (3) and (4) the following notation has been used: $\{\bar{x}_1, \bar{x}_2, \ldots \bar{x}_n\}$ are the input signals of the system, $\{\mu_{A_{p,m}}(\cdot), \mu_{B_m}(\cdot)\}$ are the membership functions of the fuzzy sets of form (1), $T(\cdot)$ is a t-norm [28] being the inference operator, $T^*(\cdot)$ is a t-norm with weights of arguments [28] being aggregation operators of the predecessors of rules (2), $S(\cdot)$ is a t-conorm being the aggregation operator of the fuzzy conclusions from rules (2). Operator $T^*(\cdot)$ with the weights of arguments was presented in [29] in order to include the importance of predecessors in the rule base. The relationship between the triangular norms and their variants with the weights of arguments is as follows

$$
\left\{ \begin{array}{l} T^*(\mathbf{a}; \mathbf{w}) = \mathop{T}\limits_{i=1}^{n} \left(S\left(a_i, 1-w_i\right) \right) \\ S^*(\mathbf{a}; \mathbf{w}) = \mathop{S}\limits_{i=1}^{n} \left(T\left(a_i, w_i\right) \right), \end{array} \right. \quad (5)
$$

where $a_i \in \langle 0,1 \rangle$ ($i = 1, 2, \ldots n$) are the arguments of operators of form (5), and $w_i \in \langle 0,1 \rangle$ are the weights of these arguments. It is easy to see that if in dependencies (5) we use e.g. algebraic triangular norms, then they take the following detailed form

$$
\left\{ \begin{array}{l} T^*\{\mathbf{a}; \mathbf{w}\} = \prod\limits_{i=1}^{n} \left(1 + (a_i - 1) \cdot w_i\right) \\ S^*\{\mathbf{a}; \mathbf{w}\} = 1 - \prod\limits_{i=1}^{n} \left(1 - a_i \cdot w_i\right). \end{array} \right. \quad (6)
$$

More details about the operators of form (5) and their applications in the FSs can be found in our previous works [3, 29, 30].

2.2 Differential evolution algorithm

In the considered algorithm, the differential evolution (DE, [14]) algorithm was used as an example of the method based on a population. This is a well-known metaheuristic optimization method which in the following steps of the evolution process improves solutions encoded in individuals of the population (so-called agents). Due to the simplicity of implementation and high efficiency, the DE has undergone many modifications [32] and can be used in many interesting applications [18, 23, 35, 45].

The way in which the DE works is in accordance with Algorithm 2. It implements basic mutation scheme DE/rand/1 and basic crossover scheme DE/x/y/bin (see Algorithm 2). Scheme DE/rand/1 consists in drawing three individuals from the population and adding to the first of them (the base vector) a scaled difference between the other two. Parameter $F \in \langle 0,2\rangle$ is a scaling factor. Scheme DE/x/y/bin indicates a cross-breeding and consists in random determination for each component of the modified individual whether it is to come from that individual or from its variant changed by mutation (marked with the index '). Parameter $C_r \in \langle 0,1\rangle$ is the crossover probability. If $C_r = 1$, then the mutated individual replaces the one which has been the subject of the crossover.

3 Detailed description of the algorithm

The on-line signature partitioning algorithm presented in this paper uses the notation described in Section 3.1. It works in two modes, i.e. the learning mode (Section 3.2) and the test mode (Section 3.3).

3.1 Adopted notation

The algorithm presented in this paper has been divided into eight parts: Algorithm 1 - Algorithm 8. It uses the following variables:

- $\mathbf{x}_{i,j=jBase} = [x_{i,j=jBase,1},\ldots,x_{i,j=jBase,K_i}]$ and $\mathbf{y}_{i,j=jBase} = [y_{i,j=jBase,1},\ldots,y_{i,j=jBase,K_i}]$ - normalized trajectories describing the shape of the base reference signature of user i.

- $\mathbf{v}_{i,j=jBase} = [v_{i,j=jBase,1},\ldots,v_{i,j=jBase,K_i}]$ and $\mathbf{z}_{i,j=jBase} = [z_{i,j=jBase,1},\ldots,z_{i,j=jBase,K_i}]$ - normalized trajectories describing the dynamics of the base reference signature of user i (pen velocity v and pen pressure z).

- $\mathbf{X}_i^{\{v\}} = [\mathbf{x}_{i,1}^{\{v\}},\ldots,\mathbf{x}_{i,J}^{\{v\}}]$ ($\mathbf{x}_{i,j}^{\{v\}} = [x_{i,j,1}^{\{v\}},\ldots,x_{i,j,K_i}^{\{v\}}]$), $\mathbf{Y}_i^{\{v\}}$, $\mathbf{X}_i^{\{z\}}$, and $\mathbf{Y}_i^{\{z\}}$ - trajectories describing the shape of the reference signatures of user i normalized on the basis of his/her base signature *jBase* (signals $\mathbf{x}_{i,j=jBase}$, $\mathbf{y}_{i,j=jBase}$, $\mathbf{v}_{i,j=jBase}$, and $\mathbf{z}_{i,j=jBase}$).

- $\mathbf{xtst}_i^{\{v\}} = [xtst_{i,1}^{\{v\}},\ldots,xtst_{i,K_i}^{\{v\}}]$, $\mathbf{ytst}_i^{\{v\}}$, $\mathbf{xtst}_i^{\{z\}}$, and $\mathbf{ytst}_i^{\{z\}}$ - trajectories describing the shape of the test signature of the user claiming to be user i normalized on the basis of base signature *jBase* of user i (signals $\mathbf{x}_{i,j=jBase}$, $\mathbf{y}_{i,j=jBase}$, $\mathbf{v}_{i,j=jBase}$, and $\mathbf{z}_{i,j=jBase}$).

- $\mathbf{pv}_i = [pv_{i,1},\ldots,pv_{i,K_i}]$ - evolutionarily selected indicators of the membership of the shape trajectory points to vertical sections.

- $\mathbf{ph}_i^{\{s\}} = [ph_{i,1}^{\{s\}},\ldots,ph_{i,K_i}^{\{s\}}]$ - evolutionarily selected indicators of the membership of the shape trajectory points to horizontal sections.

- $\mathbf{kv}_i = [kv_{i,1},\ldots,kv_{i,P}]$ - the number of points in vertical sections.

- $\mathbf{Kc}_i^{\{s\}} = [\mathbf{kc}_{i,1}^{\{s\}},\ldots,\mathbf{kc}_{i,P}^{\{s\}}]$ ($\mathbf{kc}_{i,p}^{\{s\}} = [kc_{i,p,r=1}^{\{s\}},\ldots,kc_{i,p,r=R}^{\{s\}}]$) - the number of points in the partitions created from the intersection of P vertical sections and R horizontal sections.

- $\mathbf{TC}_i^{\{s,a\}} = [\mathbf{Tc}_{i,k=1}^{\{s,a\}},\ldots,\mathbf{Tc}_{i,k=K_i}^{\{s,a\}}]$ ($\mathbf{Tc}_{i,k}^{\{s,a\}} = [\mathbf{tc}_{i,k,p=1}^{\{s,a\}},\ldots,\mathbf{tc}_{i,k,p=P}^{\{s,a\}}]$, where $\mathbf{tc}_{i,k,p}^{\{s,a\}} = [tc_{i,k,p,r=1}^{\{s,a\}},\ldots,tc_{i,k,p,r=R}^{\{s,a\}}]$) - templates of the reference signatures' shapes determined for the shape trajectories and normalized on the basis of the signals describing the dynamics of the reference base signature of user i.

- $\mathbf{W}_i^{\{s,a\}} = [\mathbf{w}_{i,1}^{\{s,a\}},\ldots,\mathbf{w}_{i,P}^{\{s,a\}}]$ ($\mathbf{w}_{i,p}^{\{s,a\}} = [w_{i,p,r=1}^{\{s,a\}},\ldots,w_{i,p,r=R}^{\{s,a\}}]$) - weights of the partitions for P vertical sections and R horizontal sections.

- $d_{j,p,r}$, $dmax$, $avgd$, and $sd_{p,r}$ - temporary variables: the descriptors of reference signature j in the partitions, the boundaries of the conclusion of the reference signatures in the partitions, average values of the boundaries of conclusion of the reference signatures in the partitions, and values of the standard deviation of the reference signatures descriptors in the partitions.

- $\mathbf{Dtst}^{\{s,a\}} = \{\mathbf{dtst}^{\{s,a\}}_{p=1}, \ldots, \mathbf{dtst}^{\{s,a\}}_{p=P}\}$ ($\mathbf{dtst}^{\{s,a\}}_{p} = \{dtst^{\{s,a\}}_{p,r=1}, \ldots, dtst^{\{s,a\}}_{p,r=R}\}$) - descriptors of the test signature in the partitions (the distance between the test signature and the signatures' templates).
- $\overline{\mathbf{xA}}^{\{s,a\}}_{i,p,r} = [\overline{xA}^{\{s,a\}}_{i,p,r,m=1}, \ldots, \overline{xA}^{\{s,a\}}_{i,p,r,m=nRules}]$ - parameters describing the centres of the Gaussian input fuzzy sets of the system for assessing the similarity of the signatures of a user claiming to be user i to user i signatures' templates.
- $\overline{\sigma A}^{\{s,a\}}_{i,p,r}$ - parameters describing the widths of the Gaussian input fuzzy sets of the system for assessing the similarity of the signatures of a user claiming to be user i to user i signatures' templates.
- $\overline{\mathbf{xB}} = [\overline{xB}_{m=1}, \ldots, \overline{xB}_{m=nRules}]$ - parameters describing the centres of the Gaussian output fuzzy sets.
- $\overline{\sigma B}$ - parameter describing the width of the Gaussian output fuzzy sets.
- $\mathbf{pL} = [pL_1, \ldots, pL_{nRules}]$ and $\mathbf{pR} = [pR_1, \ldots, pR_{nRules}]$ - parameters forcing the infinity of the Gaussian function (left-sided and right-sided) (see Figure 1).

Algorithm 1 Learning phase for user i

1: acquire $J \geq 1$ reference signatures represented by the shape and dynamics signals
2: get the parameter describing the tolerance of the verification process $\delta_i > 0$
3: determine the base signature (determine $jBase \in [1,J]$) represented by reference signals $\mathbf{v}_{i,j=jBase}$ and $\mathbf{z}_{i,j=jBase}$
4: normalize the shape and length of J reference signatures of user i on the basis of his/her base signature $jBase$ (signals $\mathbf{x}_{i,j=jBase}$, $\mathbf{y}_{i,j=jBase}$, $\mathbf{v}_{i,j=jBase}$ and $\mathbf{z}_{i,j=jBase}$) - determine $\mathbf{X}^{\{v\}}_i$, $\mathbf{Y}^{\{v\}}_i$, $\mathbf{X}^{\{z\}}_i$ and $\mathbf{Y}^{\{z\}}_i$
5: perform the vertical and horizontal evolutionary partitioning of base signature $jBase$ for P vertical sections and $R = 2$ horizontal sections (Algorithm 2)
6: store in the database the parameters needed to verify the test signatures of user i (e.g. from individual **XBest**) (Algorithm 7)

Table 2. List of the formulas used in Algorithm 4.

Item no.	Equation
1.	$pv_{i,k} := \begin{cases} 1 \text{ for } 0 < k \leq \mathrm{int}(divpv_1) \\ 2 \text{ for } \mathrm{int}(divpv_1) < k \leq \mathrm{int}(divpv_2) \\ \vdots \\ P \text{ for } \mathrm{int}(divpv_{P-1}) < k \leq K_i \end{cases}$
2.	$ph^{\{s\}}_{i,k} := \begin{cases} 1 \text{ for } k \in \left(0, \mathrm{int}\left(divph^{\{s\}}_1\right) \right\rangle \\ 2 \text{ for } k \in \left(\begin{array}{c} \mathrm{int}\left(divph^{\{s\}}_1\right), \\ \mathrm{int}\left(divph^{\{s\}}_2\right) \end{array} \right\rangle \\ \vdots \\ R \text{ for } k \in \left(\mathrm{int}\left(divph^{\{s\}}_{R-1}\right), K_i \right\rangle \end{cases}$
3.	$avgw = 1 - \frac{\sum\limits_{p=1}^{P}\sum\limits_{r=1}^{R} \left(\begin{array}{c} w^{\{v,x\}}_{i,p,r} + w^{\{v,y\}}_{i,p,r} + \\ + w^{\{z,x\}}_{i,p,r} + w^{\{z,y\}}_{i,p,r} \end{array} \right)}{4 \cdot P \cdot R}$
4.	$avgkv = \frac{1}{P} \cdot \sum\limits_{p=1}^{P} kv_{i,p}$
5.	$sdkv = \sqrt{\frac{1}{P-1} \cdot \sum\limits_{p=1}^{P} \left(avgkv - kv_{i,p}\right)^2}$
6.	$avgkc = \frac{\sum\limits_{p=1}^{P}\sum\limits_{r=1}^{R} \left(kc^{\{v\}}_{i,p,r} + kc^{\{z\}}_{i,p,r}\right)}{2 \cdot P \cdot R}$
7.	$sdkc = \sqrt{\frac{\sum\limits_{p=1}^{P}\sum\limits_{r=1}^{R} \left(\begin{array}{c} \left(avgkc - kc^{\{v\}}_{i,p,r}\right)^2 + \\ + \left(avgkc - kc^{\{z\}}_{i,p,r}\right)^2 \end{array} \right)}{2 \cdot P \cdot R}}$
8.	$ff(\mathbf{X}_{ch}) = T^* \left(\begin{array}{c} avgw, \mu_z(sdkv), \\ \mu_z(avgkc); \\ wavgw, wsdkv, \\ wavgkc \end{array} \right)$

Algorithm 2 Differential evolution algorithm for determining partitions in the reference signatures of user i

1: randomly initialize population $\mathbf{Pop} = [\mathbf{X}_{ch=1}, \mathbf{X}_{ch=2}, \ldots, \mathbf{X}_{ch=nPop}]$ of individuals $\mathbf{X}_{ch}$ of form (7)
2: **for** $step := 1$ **to** $nSteps$ **do**
3:     **for** $ch := 1$ **to** $nInd$ **do**
4:         draw $ch1 \in \left\{ \begin{array}{c} \{1, 2, \ldots, nInd\} \backslash \\ \{ch\} \end{array} \right\}$
5:         draw $ch2 \in \left\{ \begin{array}{c} \{1, 2, \ldots, nInd\} \backslash \\ \{ch, ch1\} \end{array} \right\}$

draw $ch3 \in \left\{ \begin{array}{c} \{1,2,\ldots,nInd\}\backslash \\ \{ch,ch1,ch2\} \end{array} \right\}$
draw $gRand \in \{1,2,\ldots,nDim\}$
for $g := 1$ **to** $nDim$ **do**
$X'_{ch,g} := \left\{ \begin{array}{l} X_{ch1,g} + F \cdot \left(X_{ch2,g} - X_{ch3,g} \right) \\ \text{for rand}\,(0,1) \leq C_r \\ X_{ch,g} \text{ otherwise} \end{array} \right.$
end for g
repair population **Pop**
determine $ffX := \mathrm{ff}\left(\mathbf{X}_{ch}\right)$ ▷ Alg. 4
determine $ffX' := \mathrm{ff}\left(\mathbf{X}'_{ch}\right)$ ▷ Alg. 4
if $\left(ffX > ffX'\right)$ **then**
$ffX := ffX'$; $\mathbf{X}_{ch} := \mathbf{X}'_{ch}$
end if
if $\left(\begin{array}{l} ffXBest > ffX' \\ \textbf{or } (step \cdot ch = 1) \end{array} \right)$ **then**
$ffXBest := ffX'$; $\mathbf{XBest} := \mathbf{X}'_{ch}$
end if
end for ch
end for $step$

Algorithm 3 Algorithm of random initialization of population $\mathbf{Pop} = [\mathbf{X}_{ch=1}, \mathbf{X}_{ch=2}, \ldots, \mathbf{X}_{ch=nPop}]$ of individuals $\mathbf{X}_{ch}$ of form (7) of user i

for $ch := 1$ **to** $nPop$ **do**
for $p := 1$ **to** $P-1$ **do**
$divpv_p := \text{rand}\left(\begin{array}{c} \left(1 + p \cdot \frac{K_i - 1}{P}\right) - \frac{K_i - 1}{3 \cdot P}, \\ \left(1 + p \cdot \frac{K_i - 1}{P}\right) + \frac{K_i - 1}{3 \cdot P} \end{array} \right)$
end for p
for each s **in** $\{v,z\}$ **do**
for $r := 1$ **to** $R-1$ **do**
$\min_r^{\{s\}} := \min_{\substack{(k=1,2,\ldots,K_i) \wedge \\ (pv_{i,k}=r)}} \left(s_{i,j,k}\right)$
$\max_r^{\{s\}} := \max_{\substack{(k=1,2,\ldots,K_i) \wedge \\ (pv_{i,k}=r)}} \left(s_{i,j,k}\right)$
$divph_r^{\{s\}} := \text{rand}\left(\begin{array}{c} \min_r^{\{s\}} + r \cdot \frac{\max_r^{\{s\}} - \min_r^{\{s\}}}{R} + \\ - \frac{\max_r^{\{s\}} - \min_r^{\{s\}}}{3 \cdot R}, \\ \min^{\{s\}} + r \cdot \frac{\max_r^{\{s\}} - \min_r^{\{s\}}}{R} + \\ - \frac{\max_r^{\{s\}} - \min_r^{\{s\}}}{3 \cdot R} \end{array} \right)$
end for p
end for s
$\mathbf{X}_{ch} = \left\{ \begin{array}{c} divpv_{p=1}, \ldots, divpv_{p=P-1}, \\ divph_{r=1}^{\{s=v\}}, \ldots, divph_{r=R-1}^{\{s=v\}}, \\ divph_{r=1}^{\{s=z\}}, \ldots, divph_{r=R-1}^{\{s=z\}} \end{array} \right\} = \left\{X_{ch,1}, \ldots, X_{ch,nDim}\right\},$
end for ch

Algorithm 4 Determination of the evaluation function value of the DE algorithm for individual $\mathbf{X}_{ch}$

divpv $:= X_{ch}\{\mathbf{divpv}\}$ ▷ load from $\mathbf{X}_{ch}$
for $k := 1$ **to** K_i **do**
compute $pv_{i,k}$ ▷ Tab. 2, eq. 1
$kv_{i,p=pv_{i,k}} += 1$
end for k
for each s **in** $\{v,z\}$ **do**
$\mathbf{divph}^{\{s\}} := X_{ch}\left\{\mathbf{divph}^{\{s\}}\right\}$ ▷ load from $\mathbf{X}_{ch}$
for $k := 1$ **to** K_i **do**
compute $ph_{i,k}^{\{s\}}$ ▷ Tab. 2, eq. 2
$kc^{\{s\}}_{i,p=pv_{i,k},r=ph^{\{s\}}_{i,k}} += 1$
end for k
end for s
determine shape templates for J reference signatures (Algorithm 5)
determine parameters of the system for evaluation of similarity of the signatures of the user claiming to be user i to the templates of user i signatures (Algorithm 6)
compute $avgw$ ▷ Tab. 2, eq. 3
compute $avgkv$ and $sdkv$ ▷ Tab. 2, eq. 4 and 5
compute $avgkc$ and $sdkc$ ▷ Tab. 2, eq. 6 and 7
compute $ff\left(\mathbf{X}_{ch}\right)$ ▷ Tab. 2, eq. 8

Algorithm 5 Determination of the templates of the shapes of the reference signatures

for each s **in** $\{v,z\}$ **do**
for each a **in** $\{x,y\}$ **do**
for $k := 1$ **to** K_i **do**
$tc^{\{s,a\}}_{i,k,p=pv_{i,k},r=ph^{\{s\}}_{i,k}} := \frac{1}{J} \cdot \sum_{j:=1}^{J} a^{\{s\}}_{i,j,k}$
end for k
end for a
end for s

Algorithm 6 Determination of the system parameters for evaluation of the similarity of the signatures of the user claiming to be user i to the templates of user i signatures

for each s **in** $\{v, z\}$ **do**
  for each a **in** $\{x, y\}$ **do**
    $\mathbf{D} := \mathbf{0}$
    for $k := 1$ **to** K_i **do**
      for $j := 1$ **to** J **do**
        $d_{j,p=pv_{i,k},r=ph_{i,k}^{\{s\}}} += \left(a_{i,j,k}^{\{s\}} - tc_{i,k,p=pv_{i,k},r=ph_{i,k}^{\{s\}}}^{\{s,a\}}\right)^2$
      end for j
    end for k
    for $p := 1$ **to** P **do**
      for $r := 1$ **to** R **do**
        for $j := 1$ **to** J **do**
          $d_{j,p,r} := \sqrt{d_{j,p,r}}$
        end for j
        compute $dmax$ ▷ Tab. 3, eq. 1
        compute $avgd$ ▷ Tab. 3, eq. 2
        compute $sd_{p,r}$ ▷ Tab. 3, eq. 3
        compute $w_{i,p,r}^{\{s,a\}}$ ▷ Tab. 3, eq. 4
        for $m := 1$ **to** $nRules$ **do**
          compute $\overline{xA}_{i,p,r,m}^{\{s,a\}}$ ▷ Tab. 3, eq. 5
        end for m
        compute $\overline{\sigma A}_{i,p,r}^{\{s,a\}}$ ▷ Tab. 3, eq. 6
      end for r
    end for p
  end for a
end for s
for $m := 1$ **to** $nRules$ **do**
  compute pL_m ▷ Tab. 3, eq. 7
  compute pR_m ▷ Tab. 3, eq. 8
  compute $\overline{xB}_m$ ▷ Tab. 3, eq. 9
end for m
compute $\overline{\sigma B}$ ▷ Tab. 3, eq. 10

Algorithm 7 Saving in the database the parameters needed to verify the test signatures of user i

store in the db: $\mathbf{x}_{i,j=jBase}$, $\mathbf{y}_{i,j=jBase}$, $\mathbf{v}_{i,j=jBase}$ and $\mathbf{z}_{i,j=jBase}$ (reference signals of base signature $jBase$)
store in the db: $\mathbf{pv}_i$, $\mathbf{kv}_i$
store in the db: $\overline{\mathbf{xB}}$, $\overline{\sigma B}$, $\mathbf{pL}$, and $\mathbf{pR}$
for each s **in** $\{v, z\}$ **do**
  store in the db: $\mathbf{ph}_i^{\{s\}}$, $\mathbf{Kc}_i^{\{s\}}$
  for each a **in** $\{x, y\}$ **do**
    store in the db: $\mathbf{TC}_i^{\{s,a\}}$, $\mathbf{W}_i^{\{s,a\}}$
    for $p := 1$ **to** P **do**
      for $r := 1$ **to** R **do**
        store in the db: $\overline{\mathbf{xA}}_{i,p,r}^{\{s,a\}}$, $\overline{\sigma A}_{i,p,r}^{\{s,a\}}$
      end for r
    end for p
  end for a
end for s

Algorithm 8 Verification phase of the signature of the user claiming to be user i

get a test signature and index of user i
load from the database the parameters needed for verification of the test signatures of user i
normalize the shape and length of the test signature on the basis of base signature $jBase$ of user i - use $\mathbf{a}_{i,j=jBase}$, $\mathbf{s}_{i,j=jBase}$, and determine $\mathbf{atst}_i^{\{s\}}$
verify the signature of the user claiming to be user i represented by $\mathbf{atst}_i^{\{s\}}$ (Algorithm 9)

Algorithm 9 Verification of the signature represented by trajectories $\mathbf{xtst}_i^{\{v\}}$, $\mathbf{ytst}_i^{\{v\}}$, $\mathbf{xtst}_i^{\{z\}}$ and $\mathbf{ytst}_i^{\{z\}}$

for each s **in** $\{v, z\}$ **do**
  for each a **in** $\{x, y\}$ **do**
    $\mathbf{Dtst}^{\{s,a\}} := \mathbf{0}$
    for $k := 1$ **to** K_i **do**
      $dtst_{p=pv_{i,k},r=ph_{i,k}^{\{s\}}}^{\{s,a\}} += \left(atst_{i,k}^{\{s\}} - tc_{i,k,p=pv_{i,k},r=ph_{i,k}^{\{s\}}}^{\{s,a\}}\right)^2$
    end for k
    for $p := 1$ **to** P **do**
      for $r := 1$ **to** R **do**
        $dtst_{p,r}^{\{s,a\}} := \sqrt{dtst_{p,r}^{\{s,a\}}}$
      end for r
    end for p
  end for a
end for s

14: determine signal $\bar{y}$ of system (3) for $\{\mathbf{Dtst}^{\{v,x\}}, \mathbf{Dtst}^{\{v,y\}}, \mathbf{Dtst}^{\{z,x\}}, \mathbf{Dtst}^{\{z,y\}}\}$
15: **if** $\bar{y} > cth$ **then**
16: the test signature was created by user i
17: **else**
18: the test signature was not created by user i
19: **end if**

Table 3. List of the formulas used in Algorithm 6.

Item no.	Equation
1.	$dmax := \delta_i \cdot \max\limits_{j=1,2,\ldots,J} \left\{ d_{j,p,r} \right\}$
2.	$avgd := \frac{\sum_{j=1}^{J} d_{j,p,r}}{J}$
3.	$sd_{p,r} := \frac{1}{\sqrt{J}} \cdot \sum\limits_{j=1}^{J} \sqrt{\left(avgd - d_{j,p,r} \right)^2}$
4.	$w_{i,p,r}^{\{s,a\}} := \left(1 - \frac{sd_{p,r} \cdot avgd}{\max\limits_{\substack{p2:=1,2,\ldots,P,\\ r2:=1,2,\ldots,R}} (sd_{p2,r2} \cdot avgd)} \right)$
5.	$\overline{xA}_{i,p,r,m}^{\{s,a\}} := \frac{(m-1) \cdot dmax}{nRules-1}$
6.	$\overline{\sigma A}_{i,p,r}^{\{s,a\}} := \frac{\overline{xA}_{i,p,r,2}^{\{s,a\}} - \overline{xA}_{i,p,r,1}^{\{s,a\}}}{2 \cdot \sqrt{\log\left(\frac{1}{0.5}\right)}}$
7.	$pL_m := \begin{cases} 1 \text{ for } m = 1 \\ 0 \text{ otherwise} \end{cases}$
8.	$pR_m := \begin{cases} 1 \text{ for } m = nRules \\ 0 \text{ otherwise} \end{cases}$
9.	$\overline{xB}_m := \frac{m-1}{nRules-1}$
10.	$\overline{\sigma B} := \frac{\overline{xB}_2 - \overline{xB}_1}{2 \cdot \sqrt{\log\left(\frac{1}{0.5}\right)}}$

3.2 Learning phase

At the beginning of the learning phase user i creates J reference signatures (Algorithm 1, line 1) and the parameter describing the tolerance of the verification process is acquired (Algorithm 1, line 2). The use of parameter δ_i allows us, among others, to adjust the algorithm to specific fields of application and take into account the trend of changes occurring over time in the way in which the user signs.

Next, the base signature with index $jBase$ is selected from all reference signatures (Algorithm 1, line 3). It is one of the reference signatures acquired in the acquisition phase. The distance between the trajectories describing this signature and the trajectories of other reference signatures is the lowest taking into account the adopted distance measure (e.g. the Euclidean one). The other reference signatures in the learning phase (Algorithm 1, line 4) and the test signatures in the test phase (Algorithm 8, line 3) are matched to it in the standard signature normalization procedure [17, 19, 22, 25]. It uses, among others, the Dynamic Time Warping algorithm [1, 11, 33].

Following the normalization, the evolutionary partitioning of base signature $jBase$ is performed. This process is executed for the defined and equal for all users number of vertical sections $P \in [1,3]$ and the number of horizontal sections $R \in [1,2]$ (Algorithm 1, line 5 and Algorithm 2). The upper values of P and R have been limited because increasing these values causes an excessive decomposition of the signatures and reduces the ability to interpret the created partitions.

After the partitioning, the parameters of user i needed in the verification phase are stored in the database (Algorithm 7). Interpretation of these parameters is presented in Section 3.1.

3.2.1 Initialization of the population

The procedure of evolutionary partitioning starts with a proper initialization of a population (Algorithm 3). It was assumed that each individual in population $\mathbf{X}_{ch}$ ($ch = 1,2,\ldots Npop$, $Npop$ is the number of individuals in the population) has the following structure:

$$\mathbf{X}_{ch} = \left\{ \begin{array}{c} divpv_{p=1}, \ldots, divpv_{p=P-1}, \\ divph_{r=1}^{\{s=v\}}, \ldots, divph_{r=R-1}^{\{s=v\}}, \\ divph_{r=1}^{\{s=z\}}, \ldots, divph_{r=R-1}^{\{s=z\}} \end{array} \right\} = \left\{ X_{ch,1}, \ldots, X_{ch,nDim} \right\}, \tag{7}$$

where $nDim = 2 \cdot (P-1) \cdot (R-1)$ is the length of individual $\mathbf{X}_{ch}$, $divpv_p$ ($p = 1,2,\ldots P-1$) are the boundaries of vertical sections, $divph_r^{\{s\}}$ ($r = 1,2,\ldots R-1$) are the boundaries of horizontal sections. Generation of components $\mathbf{X}_{ch}$ and $divph_r^{\{s\}}$ has to be performed in accordance with the following conditions

$$\begin{cases} divpv_p < divpv_{p+1} \\ \text{for } p = 1, 2, \ldots, P-2 \\ divph_r^{\{s\}} < divph_{r+1}^{\{s\}} \\ \text{for } r = 1, 2, \ldots, R-2. \end{cases} \tag{8}$$

Taking into account conditions (8), initialization of **Pop** performs drawing of $divpv_p$ and $divph_r^{\{s\}}$ in the subsequent sections (Algorithm 3, lines 3 and 7-9).

3.2.2 Evolution of the population

Evolution of the population is performed after the initialization procedure according to the DE method. It runs in a typical way, i.e. the crossover (Algorithm 2, lines 4-7 and 9) and the mutation (Algorithm 2, line 9) are performed, the population is repaired (taking into account conditions (8), Algorithm 2, line 11), individual $\mathbf{X}_{ch}$ (Algorithm 2, line 12) and its modified version $\mathbf{X}'_{ch}$ (Algorithm 2, line 13) are evaluated, **Pop** is updated (Algorithm 2, lines 14-16), and the best solution found **XBest** is updated (Algorithm 2, lines 17-19).

3.2.3 Evaluation of the population

In Algorithm 2 the procedure for determining the evaluation function (Algorithm 4) is very important. It starts with loading the division points of time-domain **divpv** (Algorithm 4, line 1) and update of the indicators of membership of the shape trajectory points to the vertical sections (Algorithm 4, lines 3 and 4). Next, an analogous process is performed for the horizontal sections (Algorithm 4, lines 6-12). With the information about the partitions, the templates of the signature shape (Algorithm 4, line 13) can be determined in the partitions. The FS parameters (3) for evaluating the signatures' similarity (Algorithm 4, line 14; Section 3.2.4) can also be computed.

An important stage of Algorithm 4 is to determine the components of the evaluation function (Algorithm 4, lines 15-17), which are as follows:

– Component *avgw* determines the stability of the reference signatures of user *i*. It is determined on the basis of the weights of the partitions. The weights which are more stable have a higher value. In such partitions, the way in which the user signs is more stable. Component *avgw* is the negated average weight value.

– Component *sdkv* is used to evaluate uniformity in the allocation of the signature trajectory points to the vertical sections. It is expressed by the standard deviation from the arithmetic mean *avgkv*.

– Component *sdkc* is used to evaluate uniformity in the allocation of signature trajectory points to vertical sections. It is expressed by the standard deviation from the arithmetic mean *avgkc*.

The last step in Algorithm 4 consists in determining the value of fitness function $ff(\mathbf{X}_{ch})$ for individual $\mathbf{X}_{ch}$. For this reason, we use operator (6) in which weights $\{wavgw, wsdkv, wavgkc\}$ correspond to $\{avgw, sdkv, sdkc\}$ components. The weights are the algorithm parameters. Standard deviations of components $\{avgw, sdkv, sdkc\}$ can be greater than 1, so they are normalized to the range $(0, 1)$ by the sigmoid membership function

$$\mu(x; par_1, par_2, par_3, par_4) = par_4 + \frac{par_3}{1+\exp(-(par_1 \cdot x - par_2))}, \tag{9}$$

where the parameters have the following values in the simulations: $par_1 = 10$, $par_2 = 5$, $par_3 = 1$, and $par_4 = 0$. The purpose of the DE is to minimize the value of the evaluation function.

3.2.4 Determination of the FS parameters for assessing the similarity of the signatures

The purpose of the FS is a fuzzy assessment of the differences between the reference signatures' templates and the signatures (represented by the trajectories) which are the subject of the verification process. The FS performs verification independently at the level of each partition (Algorithm 9, line 5), so formula (2) takes the following form

$$rule_m: \left[\begin{array}{c} \mathbf{IF}\ x^{\{s,a\}}_{p=1,r=1}\ \mathbf{IS}\ A^{\{s,a\}}_{i,p=1,r=1,m} \\ \mathbf{WITH}\ w^{\{s,a\}}_{i,p=1,r=1}\ \mathbf{AND}\ \ldots \\ x^{\{s,a\}}_{p=1,r=R}\ \mathbf{IS}\ A^{\{s,a\}}_{i,p=1,r=R,m} \\ \mathbf{WITH}\ w^{\{s,a\}}_{i,p=1,r=R}\ \mathbf{AND}\ \ldots \\ x^{\{s,a\}}_{p=P,r=1}\ \mathbf{IS}\ A^{\{s,a\}}_{i,p=P,r=1,m} \\ \mathbf{WITH}\ w^{\{s,a\}}_{i,p=P,r=1}\ \mathbf{AND}\ \ldots \\ x^{\{s,a\}}_{p=P,r=R}\ \mathbf{IS}\ A^{\{s,a\}}_{i,p=P,r=R,m} \\ \mathbf{WITH}\ w^{\{s,a\}}_{i,p=P,r=R}\ \mathbf{AND} \\ \mathbf{THEN}\ y\ \mathbf{IS}\ B_m \end{array}\right]. \tag{10}$$

Weights of partitions $w^{\{s,a\}}_{i,p=1,r=1}$ are determined in Algorithm 6 in lines 3-17. In the following part of Algorithm 6 parameters of input and output fuzzy sets of form (1) of user i (Algorithm 6, lines 18-31) are computed. As for replacing the notation of the rules of form (2) by formula (10), formula (4) has the following form

$$\tau_m = \underset{\substack{p=1,\\ r=1}}{\overset{\substack{p=P,\\ r=R}}{T^*}} \left(\begin{array}{c} \mu_{A^{\{s=v,a=x\}}_{i,p,r,m}} \left(\begin{array}{c} \bar{x}^{\{s=v,a=x\}}_{p,r}, \\ \overline{xA}^{\{s=v,a=x\}}_{i,p,r,m}, \\ \overline{\sigma A}^{\{s=v,a=x\}}_{i,p,r}, \\ pL_m, pR_m \end{array} \right) \ldots \\ \mu_{A^{\{s=z,a=y\}}_{i,p,r,m}} \left(\begin{array}{c} \bar{x}^{\{s=z,a=y\}}_{p,r}, \\ \overline{xA}^{\{s=z,a=y\}}_{i,p,r,m}, \\ \overline{\sigma A}^{\{s=z,a=y\}}_{i,p,r}, \\ pL_m, pR_m \end{array} \right); \\ w^{\{s=v,a=x\}}_{i,p,r} \ldots w^{\{s=z,a=y\}}_{i,p,r} \end{array} \right). \tag{11}$$

3.3 Test phase

At the beginning of the test phase, one test signature of the user and the information about his/her alleged identity are acquired (Algorithm 8, line 1). This user will be further marked with index i. Next, the parameters of the reference signatures of user i are loaded from the database (Algorithm 8, line 2). The reading from the database is analogous to Algorithm 7 implemented for writing.

Later in the verification phase, the shape and length of the test signature is normalized on the basis of the base signature of user i (Algorithm 8, line 3). It is performed as in the training phase. After this step, the test signature is represented by a set of normalized trajectories: $\mathbf{xtst}^{\{v\}}_i$, $\mathbf{ytst}^{\{v\}}_i$, $\mathbf{xtst}^{\{z\}}_i$, and $\mathbf{ytst}^{\{z\}}_i$.

After the normalization, the key step in the verification process is performed. It is defined as Algorithm 9. In the first phase of the algorithm, normalized distances between the test signature trajectories and the templates of the reference signatures of user i are determined (Algorithm 9, lines 1-13). Matrices of distances $\mathbf{Dtst}^{\{s,a\}}$ are the input signals of the system used for assessing the similarity of signatures (3). They are used to determine the response of the system (Algorithm 9, line 14). The value of the response is used for the signature verification in the test phase (Algorithm 9, lines 15-19). In the verification coefficient $cth \in \langle 0,1 \rangle$ is used. Its value is common to all biometric system users and usually close to 0.5 (this is the value which we adopted in the simulations). Using this factor allows us to eliminate disproportions between FAR and FRR errors (see e.g. [36]) and adapt the system to the expectations arising from the area of its application.

4 Simulation results

We implemented and tested the presented algorithm for the evolutionary generation of the on-line signature hybrid descriptors in C#.NET language. The simulations were performed for $P \in \{2,3,4\}$ vertical sections. We assumed that the identity verification carried out with the use of hybrid partitions created by a population-based algorithm should work better than in the case of our previous method [7], which creates fixed-size hybrid partitions.

The simulations were performed using the BioSecure dynamic signature database DS2 [13], which contains signatures of 210 users acquired in two sessions with the use of a graphics tablet. Each session contains 15 genuine signatures and 10 skilled forgeries per person. In the learning phase, we used 5 randomly selected genuine signatures of each signer. During the test phase we used 10 genuine signatures and 10 so-called skilled forgeries [20] of each signer.

Table 4. Accuracy of the method using an algorithm for the evolutionary generation of the on-line signature hybrid descriptors.

Number of vertical sections P	**Average FAR**	**Average FRR**	**Average error**
2	**2.72 %**	**2.62 %**	**2.67 %**
3	3.36 %	3.30 %	3.33 %
4	2.72 %	2.62 %	2.67 %

Table 5. Comparison of the accuracy of the method using an algorithm for the evolutionary generation of the on-line signature hybrid descriptors to other methods used for the dynamic signature verification using BioSecure database.

Method	**Average FAR**	**Average FRR**	**Average error**
Methods of other authors [15]	-	-	3.48 %-30.13 %
Method using fixed hybrid partitions [7]	3.36 %	3.30 %	3.33 %
Our method	**2.72 %**	**2.62 %**	**2.67 %**

We adopted the following values of the Differential Evolution algorithm parameters in the simulations: (a) the number of chromosomes in the population $Npop = 100$, (b) the value of parameter F [27] of the DE algorithm is 0.5, (c) the value of parameter CR [27] of the DE algorithm is 0.9, (d) the value of weight $wavgw$ is 0.7, (e) the value of weight $wsdkv$ is 0.2, (f) the value of weight $wavgkc$ is 0.2, and (g) $nSteps$ is 100.

The simulations were repeated 5 times in accordance with the standard cross-validation procedure. The results of the simulations are presented in Table 4 in the form of FAR (False Acceptance Rate), FRR (False Rejection Rate) and EER (Equal Error Rate) coefficients which are used in the literature to evaluate the effectiveness of biometric methods [20]. Moreover, Table 5 contains the results obtained by our method in comparison to the methods of other authors.

One can see that the method for the on-line signature verification based on the evolutionary generation of signature hybrid descriptors achieved the highest accuracy when two vertical sections associated with the time moments of signing are used. Moreover, increasing the number of vertical sections causes a deterioration of verification results.

The results of the method presented in this paper are also better than the results of the method for the on-line signature verification using hybrid partitioning with a fixed size of created partitions presented in [7]. It means that the assumptions adopted in this paper are correct and the evolutionary generation of partitions increases the efficiency of the system used for the signature verification. Moreover, the accuracy of our method is good in comparison to the results achieved by other methods presented in [15].

5 Conclusions

In this paper, we have presented a new evolutionary-fuzzy algorithm for the generation of the on-line signature hybrid descriptors. It uses a Differential Evolution algorithm for creating hybrid partitions of the on-line signature, which are the most characteristic in the context of the individual. It is realized by selecting boundary points of the partitions which are used to divide signals describing the dynamics of the signature into some (certain) regions. The simulations performed using the BioSecure dynamic signature database confirmed that the partitions of the signature created by our method are more characteristic of the individual because the effectiveness of the identity verification process executed on the basis of the descriptors created in the partitions selected by our method is relatively high.

Acknowledgment

The authors would like to thank the associate editor and the reviewers for their helpful comments.

Donald C. Wunch II would like to thank for the support from the Mary K. Finley Missouri Endowment and the Missouri S&T Intelligent Systems Center.

This paper was financed under the program of the Minister of Science and Higher Education under the name 'Regional Initiative of Excellence' in the years 2019-2022, project number 020/RID/2018/19 with the amount of financing PLN 12 000 000.

References

[1] Z. Bankó, A. János, Correlation based dynamic time warping of multivariate time series, Expert Systems with Applications, 39, 12814–12823, 2012.

[2] Ł. Bartczuk, A. Przybył, K. Cpałka, A new approach to nonlinear modelling of dynamic systems based on fuzzy rules, International Journal of Applied Mathematics and Computer Science (AMCS), 26, 3, 603-621, 2016.

[3] K. Cpałka, Design of Interpretable Fuzzy Systems. Springer, Cham, 2017.

[4] K. Cpałka, K. Łapa, A. Przybył, M. Zalasiński, A new method for designing neuro-fuzzy systems for nonlinear modelling with interpretability aspects, Neurocomputing, 135, 203–217, 2014.

[5] K. Cpałka, O. Rebrova, R. Nowicki, L. Rutkowski, On Design of Flexible Neuro-Fuzzy Systems for Nonlinear Modelling, International Journal of General Systems, 42, 706–720, 2013.

[6] K. Cpałka, M. Zalasiński, On-line signature verification using vertical signature partitioning, Expert Systems with Applications, 41, 4170–4180, 2014.

[7] K. Cpałka, M. Zalasiński, and L. Rutkowski, A new algorithm for identity verification based on the analysis of a handwritten dynamic signature. Applied Soft Computing, 43, 47–56, 2016.

[8] K. Cpałka, M. Zalasiński, L. Rutkowski, New method for the on-line signature verification based on horizontal partitioning, Pattern Recognition, 47, 2652–2661, 2014.

[9] M. Faúndez-Zanuy, J.M. Pascual-Gaspar, Efficient on-line signature recognition based on multi-section vector quantization, Formal Pattern Analysis & Applications, 14, 37–45, 2011.

[10] Md.M. Ferdaus, S.G. Anavatti, M.A. Garratt, M. Pratama, Development of C-Means Clustering Based Adaptive Fuzzy Controller for a Flapping Wing Micro Air Vehicle, Journal of Artificial Intelligence and Soft Computing Research, 9, 99–109, 2019.

[11] J. Fernandez de Canetea, A. Garcia-Cerezoa, I. Garcia-Morala, P. Del Saza, E. Ochoa, Correlation based dynamic time warping of multivariate time series, Expert Systems with Applications, 40, 5648–5660, 2013.

[12] J. Fierrez, J. Ortega-Garcia, D. Ramos, J. Gonzalez-Rodriguez, HMM–based on-line signature verification: Feature extraction and signature modeling, Pattern Recognition Letters, 28, 2325–2334, 2007.

[13] Homepage of Association BioSecure. [Online] Available from: http://biosecure.it-sudparis.eu [Accessed: 21 February 2020].

[14] S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art. IEEE transactions on evolutionary computation, 15(1), 4-31, 2010.

[15] N. Houmani, A. Mayoue, S. Garcia-Salicetti, B. Dorizzi, M.I. Khalil, M.N. Moustafa, H. Abbas, D. Muramatsu, B. Yanikoglu, A. Kholmatov, M. Martinez-Diaz, J. Fierrez, J. Ortega-Garcia, J. Roure Alcobe, J. Fabregas, M. Faundez-Zanuy, J.M. Pascual-Gaspar, V. Cardenoso-Payo, C. Vivaracho-Pascual, BioSecure signature evaluation campaign (BSEC'2009): Evaluating online signature algorithms depending on the quality of signatures, Pattern Recognition, 45, 993-1003, 2012.

[16] K. Huang, Y. Hong, Stability and style-variation modeling for on–line signature verification, Pattern Recognition, 36, 2253–2270, 2003.

[17] M.T. Ibrahim, M.A. Khan, K.S. Alimgeer, M.K. Khan, I.A. Taj, L. Guan, Velocity and pressure-based partitions of horizontal and vertical trajectories for on-line signature verification, Pattern Recognition, 43, 2817–2832, 2010.

[18] S. Ikeda, E. Ooka, Application of differential evolution-based constrained optimization methods to district energy optimization and comparison with dynamic programming, Applied Energy, 254, 113670, 2019.

[19] A.K. Jain, F.D. Griess, S.D. Connell, On-line signature verification, Pattern Recognition, 35, 2963–2972, 2002.

[20] A.K. Jain, A. Ross, Introduction to Biometrics, In: A.K. Jain, P. Flynn, A.A. Ross (Eds.), Handbook of Biometrics, Springer, Berlin, Heidelberg, 2008.

[21] M.A.U. Khan, M.K. Khan, M.A. Khan, Velocity-image model for online signature verification, IEEE Trans. Image Process, 15, 3540–3549, 2006.

[22] H. Lei, V. Govindaraju, A comparative study on the consistency of features in on-line signature verification, Pattern Recognition Letters, 26, 2483–2489, 2005.

[23] S. Li, W. Gong, X. Yan, Ch. Hu, D. Bai, L. Wang, Parameter estimation of photovoltaic models with memetic adaptive differential evolution, Solar Energy, 15, 465–474, 2019.

[24] K. Łapa, K. Cpałka, L. Wang, New method for design of fuzzy systems for nonlinear modelling using different criteria of interpretability, Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, 8467, Springer, 217-232, 2014.

[25] Ch. O'Reilly, R. Plamondon, Development of a Sigma–Lognormal representation for on-line signatures, Pattern Recognition, 42, 3324–3337, 2009.

[26] J.M. Pascual–Gaspar, M. Faúndez–Zanuy, C. Vivaracho, Fast on-line signature recognition based on VQ with time modelling, Engineering Applications of Artificial Intelligence, 24, 368–377, 2011.

[27] M.E.H. Pedersen, Good parameters for differential evolution. Hvass Laboratories Technical Report, HL1002, 2010.

[28] L. Rutkowski, Computational Intelligence, Springer, Berlin, Heidelberg, 2008.

[29] L. Rutkowski, K. Cpałka, Flexible neuro-fuzzy systems, IEEE Trans. Neural Networks, 14, 554–574, 2003.

[30] L. Rutkowski, K. Cpałka, Designing and learning of adjustable quasi triangular norms with applications to neuro-fuzzy systems, IEEE Trans. Fuzzy Systems, 13, 140–151, 2005.

[31] L. Rutkowski, A. Przybył, K. Cpałka, Novel Online Speed Profile Generation for Industrial Machine Tool Based on Flexible Neuro-Fuzzy Approximation, IEEE Trans. Industrial Electronics, 59, 1238–1247, 2012.

[32] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, Journal of Global Optimization, 11, 341–359, 1997.

[33] I. Svalina, V. Galzina, R. Lujić, G. Šimunović, Correlation based dynamic time warping of multivariate time series, Expert Systems with Applications, 40, 6055–6063, 2013.

[34] J. Szczypta, A. Przybył, K. Cpałka, Some aspects of evolutionary designing optimal controllers, Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, 7895, Springer, 91-100, 2013.

[35] R. Xu, J. Xu, D.C. Wunsch II, A Comparison Study of Validity Indices on Swarm-Intelligence-Based Clustering, IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, vol. 42, no. 4, 1243–1256, 2012.

[36] D.Y. Yeung, SVC2004: First International Signature Verification Competition, Proc. of ICBA, Springer LNCS-3072, 16–22, 2004.

[37] L.A. Zadeh, Fuzzy Sets, Information and Control, 8, 338–353, 1965.

[38] M. Zalasiński and K. Cpałka, A new method for signature verification based on selection of the most important partitions of the dynamic signature. Neurocomputing, 289, 13-22, 2018.

[39] M. Zalasiński, K. Cpałka, New algorithm for online signature verification using characteristic hybrid partitions, Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology – ISAT 2015 – Part IV, Advances in Intelligent Systems and Computing, 432, Springer, 147-157, 2016.

[40] M. Zalasiński, K. Cpałka, New Approach for the On-Line Signature Verification Based on Method of Horizontal Partitioning, Lecture Notes In Artificial Intelligence, 7895, 342–350, 2013.

[41] M. Zalasiński, K. Cpałka, Novel algorithm for the on-line signature verification using selected discretization points groups, Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, 7894, Springer, 493-502, 2013.

[42] M. Zalasiński, K. Cpałka, Novel algorithm for the on-line signature verification, Lecture Notes in Artificial Intelligence 7268, 362–367, 2012.

[43] M. Zalasiński, K. Cpałka, Y. Hayashi, New fast algorithm for the dynamic signature verification using global features values, Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, 9120, Springer, 175-188, 2015.

[44] M. Zalasiński, K. Cpałka, E. Rakus-Andersson, An idea of the dynamic signature verification based on a hybrid approach, Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, 9693, Springer, 232-246, 2016.

[45] G. Zhang, K. Xing, Differential evolution metaheuristics for distributed limited-buffer flowshop scheduling with makespan criterion, Computers & Operations Research, 108, 33–43, 2019.

[46] Y. Zhao, Q. Liu, A Continuous-Time Distributed Algorithm for Solving a Class of Decomposable Nonconvex Quadratic Programming, Journal of Artificial Intelligence and Soft Computing Research, 8, 283–291, 2018.

Marcin Zalasiński received the M.Sc. and Ph.D. degrees from the Częstochowa University of Technology, Częstochowa, Poland, in 2009 and 2014, respectively. He is currently an Associate Professor with the Department of Computer Engineering. Dr. Zalasiński has authored over 20 publications. His current research interests include computational intelligence, biometrics, and expert systems.

Krzysztof Cpalka received the M.Sc. and Ph.D. degrees from the Częstochowa University of Technology, Częstochowa, Poland, in 1997 and 2002, respectively. He is currently a Professor with the Department of Computer Engineering. Prof. Cpałka has authored over 100 publications, including several papers in various series of IEEE Transactions. His current research interests include computational intelligence methods and their applications. He was a recipient of the IEEE Transactions on Neural Networks Outstanding Paper Award in 2005.

Łukasz Laskowski received Ms.C. and Ph.D. degrees from the Częstochowa University of Technology, Częstochowa, Poland, in 2004 and 2009, respectively. He is currently an Associate Professor with the Department of Molecular Engineering and Nanoelectronics at Institut of Nuclear Physics of the Polish Academy of Sciences. Dr. Laskowski has authored over 30 publications. His current research interests include artificial neural networks, nanoelectronics, molecular engineering, and nanotechnology.

Donald C. Wunsch II, received the B.S. degree in applied mathematics from the University of New Mexico, Albuquerque, and the M.S. degree in applied mathematics and the Ph.D. degree in electrical engineering from the University of Washington, Seattle.

Since 1999, he has been with Missouri University of Science & Technology, Rolla, where he is currently the Mary K. Finley Missouri Distinguished Professor of Computer Engineering. His prior positions were Associate Professor and Director of the Applied Computational Intelligence Laboratory, Texas Tech University, Lubbock; Senior Principal Scientist at Boeing; Consultant for Rockwell International; and Technician for International Laser Systems. His key research contributions are clustering, adaptive resonance and reinforcement learning architectures, hardware and applications, neuro-fuzzy regression, traveling salesman problem heuristics, robotic swarms, and bioinformatics. He has produced 16 Ph.D. recipients in computer engineering, electrical engineering, and computer science, has attracted over $8 million in sponsored research, and has over 300 publications including nine books. His research has been cited over 2500 times. Dr. Wunsch was a fellow and the past President of the International Neural Network Society (INNS) and has been an INNS senior fellow since 2007. He served as General Chair of the International Joint Conference on Neural Networks and on several boards, including the St. Patrick's School Board, the IEEE Neural Networks Council, the INNS, and the University of Missouri Bioinformatics Consortium.

Krzysztof Przybyszewski is a professor at the University of Social Sciences in Łódź. His adventure with applied computer science began in the 1980s with a simulation of non-quantum collective processes (the subject of a Ph.D. dissertation). At present, he is involved in research and applications of various artificial intelligence technologies and soft computing methods in selected IT problems (in particular, in expert systems supporting the management of education quality in universities - the use of fuzzy numbers and sets). As a deputy dean at the University of Social Sciences, he is the designer and organizer of the on-Computer Science Faculty education program. He is the author of over 80 publications in the field of computer science and IT applications.