

Binary PSO Variants for Feature Selection in Handwritten Signature Authentication

Emrah HANCER^{1,*}, Marina BARDAMOVA², Ilya HODASHINSKY²,
Konstantin SARIN², Artem SLEZKIN², Mikhail SVETLAKOV²

¹ *Department of Software Engineering, Mehmet Akif Ersoy University, Turkey*

² *Department of Security, Tomsk State University of Control Systems and Radioelectronics, Russia*
e-mail: ehancer@mehmetakif.edu.tr, 722bmb@gmail.com, hodashn@gmail.com,
sks@security.tomsk.ru, saotom724@gmail.com, svetlakov.m4@gmail.com

Received: February 2021; accepted: December 2021

Abstract. In this paper we propose modifications of the well-known algorithm of particle swarm optimization (PSO). These changes affect the mapping of the motion of particles from continuous space to binary space for searching in it, which is widely used to solve the problem of feature selection. The modified binary PSO variations were tested on the dataset SVC2004 dedicated to the problem of user authentication based on dynamic features of a handwritten signature. In the example of k-nearest neighbours (kNN), experiments were carried out to find the optimal subset of features. The search for the subset was considered as a multicriteria optimization problem, taking into account the accuracy of the model and the number of features.

Key words: feature selection, handwritten signature, particle swarm optimization, biometric authentication.

1. Introduction

A handwritten signature is one of the most commonly used instruments of person authenticating. There are two types of handwritten signature recognition: static (offline) and dynamic (online) (Mailah and Han, 2012). Static type consists only in processing a graphic image of the signature. Dynamic recognition involves the analysis of dynamic parameters, such as coordinates, pressure, pen speed, and the total time of the signature. Such dynamic parameters can't be determined with only the final image of the signature, so the task of falsification is much more complicated. Research has shown that dynamic signature recognition can significantly improve the accuracy of user identification (Linden *et al.*, 2018).

To obtain initial dynamic signature signals, various technical devices are used, for example, a graphic tablet (Nelson and Kishon, 1991) or a pen equipped with special sensors (Sakamoto *et al.*, 2001). Further, it is necessary to extract features from the received sig-

*Corresponding author.

nals. Features should make it possible to distinguish one user from another, taking into account the variability of the signature of the same user (Li and Jain, 2015).

Features can be extracted from the time domain, frequency domain or wavelet transform domain (Wu *et al.*, 2018). For feature extraction, algorithms such as Fourier transform (Dimauro, 2011) or discrete wavelet transform can be used (Fahmy, 2010). However, after extracting features, their number may be quite large. It negatively affects the operating time of the user authentication system. In addition, not all of the obtained features can be equally useful for the correct identification of the user, some of them can reduce the classification accuracy. To solve this problem, the feature selection is used.

The selection of the most informative features when constructing a machine learning model is not only a way to reduce the computational complexity of the model, but also to improve the forecast accuracy. This procedure is particularly important for data describing digitized signals, which generally contain a large number of variables with different informativeness. In particular, some of these variables have a crucial role to recognize the corresponding signals, while others may be useless or even may have a negative effect for the recognition process.

There are three groups of feature selection methods: filters, wrappers and embedded methods. While the last group is particular to specific decision algorithms, the rest are more general. Filters are applied during a stage of data preprocessing. They rely on analysis of the relationship between feature space and output variable. The criteria for assessing the relationship can be mutual information, chi-square test, Fisher's test, correlation coefficient and other characteristics. Disadvantages of this group of methods include the complexity of assessing a synergy between subsets of features and some separation of filters from decision algorithms.

In contrast, methods from the wrapper group function in conjunction with the decision algorithm. They select a subset of features that allows achieving the best value of the objective function of the model. As a rule, optimization algorithms are used as wrappers, and, above all, metaheuristics. However, not all metaheuristics in their original version are capable of working in a binary search space, such as, for example, the genetic algorithm. The development of effective variations of binary algorithms is an urgent task.

One of the most famous metaheuristics is the particle swarm algorithm (PSO). It is a stochastic search method based on the iteratively repeated interaction of a group of particles (Shi and Eberhart, 1999). Its advantages are simplicity and computational efficiency due to the small number of tunable parameters and low memory usage.

This article is devoted to the task of PSO binarization. Binarization is a modification of the algorithm to perform a search in binary space. The contributions of this article are as follows:

- We propose the following new methods for binarization of continuous metaheuristics: Local search, New fitness, The merge procedure, The hybrid method of the modified arithmetic operation and the merge procedure.
- We modify the continuous PSO with new binarization methods to select features of the user's classification based on a handwritten signature.

- We propose and demonstrate the effectiveness of our binary PSOs for classifying a user by handwritten signature. We will show that these new enhancements have greater authentication efficiency compared to standard binary PSO with transform functions.

The rest of the article is organized as follows. Section 2 reviews binarization methods in the literature. Section 3 describes the investigated modifications of the PSO algorithm for finding sub-optimal subsets of features. Section 4 describes the experiment. Section 5 contains a statistical comparison of PSO binarization methods for the problem of user authentication by handwritten signature. Finally, Section 6 contains an overview of the main results.

2. Literature Review

2.1. Metaheuristic Algorithm PSO

Each solution vector in the algorithm is a particle moving due to the combined effect of the force of inertia, memory and acceleration reported by the best particle. The coordinates of the i th particle at the time of iteration t are determined by the vector $\mathbf{x}_i(t) = (x_{i1}(t), x_{i2}(t), \dots, x_{iD}(t))$, where $i = 1, 2, \dots, N$, N is a particle count in a swarm. All particles store their best position $\mathbf{p}_i(t) = (p_{i1}(t), p_{i2}(t), \dots, p_{iD}(t))$ for the iterations they have passed. The velocity of a particle is set by a vector $\mathbf{v}_i(t) = (v_{i1}(t), v_{i2}(t), \dots, v_{iD}(t))$. The speed and position of the i th particle at $t + 1$ iteration are determined by the following expressions:

$$\mathbf{x}_i(t + 1) = \mathbf{x}_i(t) + \mathbf{v}_i(t + 1), \quad (1)$$

$$\begin{aligned} \mathbf{v}_i(t + 1) = & w \times \mathbf{v}_i(t) + c_1 \times \text{rand}_1 \times (\mathbf{p}_i(t) - \mathbf{x}_i(t)) \\ & + c_2 \times \text{rand}_2 \times (\mathbf{p}_g(t) - \mathbf{x}_i(t)), \end{aligned} \quad (2)$$

where c_1, c_2 are positive acceleration coefficients; $\mathbf{p}_i(t)$ is a vector of previous coordinates of the particle $\mathbf{x}_i(t)$ that had the best value of the fitness function, $\mathbf{p}_g(t)$ is a vector of coordinates of the particle that reached the best value of the fitness function in the entire swarm for t iterations; w is the empirical coefficient of inertia; rand_1 and rand_2 are random numbers from the interval $[0, 1]$. The swarm stops moving when at least one of the following conditions is met: the swarm has reached a state of equilibrium, an optimal solution has been found, or a specified number of iterations has been performed.

2.2. Feature Selection

Feature selection can be represented as an NP-hard binary optimization problem (Kohavi and John, 1997). The most optimal solution of this problem can be guaranteed to be found only by a brute-force search. However, brute-force approach is ineffective for solving this problem, since the size of the feature space in modern datasets can amount to several tens or even hundreds. Going through all possible combinations of such number of elements is

a very time- and resource-consuming task. The decision to this problem can be the use of metaheuristic algorithms that allow finding a suboptimal solution in an acceptable time.

Most metaheuristic algorithms are designed to operate in a continuous search space. However, binary search space may be considered as a more suitable way to deal with feature selection task due to its structure. Accordingly, in order to use metaheuristics for addressing the feature selection task in binary search space, their modification is required. The modification should allow the algorithm to function in a binary space, while maximally preserving the original idea of metaheuristics. There are many ways to binarize metaheuristic algorithms. The main ones will be considered below: the method of modified algebraic operations, the method of transformation functions, and the quantum method.

1) *Modified algebraic operations*: The idea of the method is that to convert the algorithm to the binary version, it is enough to put their logical analogues in accordance with the original mathematical operations of the formulas used. For example, to use disjunction (OR) instead of addition, conjunction (AND) instead of multiplication, exclusive or (XOR) instead of subtraction, and logical negation (NOT). Thus, the search space of the algorithm changes from continuous to binary, but the original idea of changing the position of particles is formally preserved.

However, there are two disadvantages that make it difficult to apply this method to some algorithms. The first is that the method is practically inapplicable if the algorithm uses any operations other than addition, subtraction, multiplication, and division. The second disadvantage is typical for all methods that require working with binary vectors. Many metaheuristic algorithms, in addition to operations directly on vectors, also use a variety of coefficients represented by real numbers. When switching to binary vectors, these coefficients must also be changed or abandoned, which leads to a change in the original idea of the algorithm and, accordingly, may affect the quality of its work. So, when modifying the PSO algorithm, the authors of Yuan *et al.* (2009) refused to use the inertia coefficient, and used randomly generated binary vectors as speed coefficients.

Despite the external similarity of the algorithm binarized by the method of modified algebraic operations with the original continuous algorithm, its results are not always satisfactory. In such cases, the authors of some works use only a part of logical operations or use their own modified versions instead of some “classic” logical operations. In Kiran and Gündüz (2012) when binarizing the artificial bee colony algorithm, the authors analysed the use of logical operations and came to the conclusion that it is necessary to use only two of them: XOR and NOT. In this case, the NOT operation was modified as follows: a random real number from 0 to 1 was generated, and if it was greater than 0.5, a logical negation was performed, otherwise the vector remained unchanged.

2) *Transfer functions*: Transformation functions are often used to switch from a continuous space to a binary one. They convert a real number into the range between zero and one. If the calculated function value is greater than a random number, then the corresponding element of the vector either takes the value of one, or is replaced with the opposite (Mirjalili and Lewis, 2013).

This method avoids the limitations of the previous method, since it does not require any changes to the original algorithm. However, this method has another complexity, which

consists in the problem of choosing a specific function as a transformation one. The research (Mirjalili and Lewis, 2013), which examined six transformation functions from two families (S-shaped and V-shaped), showed that the use of V-shaped functions significantly improves the performance of the PSO algorithm. In Souza *et al.* (2020), the PSO algorithm, binarized with a V-shaped transformation function, was used to select features when solving the problem of static verification of a handwritten signature. S-shaped and V-shaped transform functions have also been successfully applied to binarize gray wolf algorithm (Chantar *et al.*, 2020), bat swarm (Qasim and Algamal, 2020), dragonfly algorithm (Mafarja *et al.*, 2017), and butterfly swarm (Arora and Anand, 2019). In Ghosh *et al.* (2020), a new X-shaped transformation function, consisting of two S-shaped ones, was proposed and applied to binarize the Social Mimic algorithm. With this approach, one vector of real numbers is converted into two binary vectors, from which the best is selected.

3) *Quantum Method*: The idea of this method is to use vectors represented by Q-bits instead of binary vectors, and principles that apply to quantum mechanics, such as quantum superposition. In Hamed *et al.* (2009), it was shown that using the quantum method for binarization of the PSO algorithm allows increasing the speed of searching for the optimal set of features. In Barani *et al.* (2017), the authors also explored a quantum approach for binarization of the gravitational search algorithm when solving the feature selection task using the kNN classifier. The principle of superposition was used in the work, and a quantum rotation gate was applied to update the solutions. The researchers concluded that the proposed algorithm can significantly reduce the total number of features, while maintaining a comparable classification accuracy relative to other algorithms. The authors (Ashry *et al.*, 2020) compared binary gray wolf algorithms using the quantum method (BQI-GWO) and the S-shaped transformation function (BGWO). Comparison of the algorithms showed that the BQI-GWO makes it possible to exclude a larger number of features, while increasing the classification accuracy. Researchers in Zouache and Ben Abdelaziz (2018) proposed a hybrid swarm intelligence algorithm based on quantum computations and a combination of the firefly algorithm and PSO for feature selection. Quantum computations provided a good tradeoff between the intensification and diversification of search, while the combination of the firefly algorithm and PSO made it possible to efficiently investigate the generated subsets of features. To evaluate the relevance of these subsets, rough set theory was employed.

3. Formulation of the Problem and Algorithmic Framework

3.1. Formulation of the Problem

The set of instances is a list of objects of the type $x_i = \{x_{i1}, x_{i2}, \dots, x_{iD}\}$, where x_{ij} is a value of the j th feature of the i th object, $j \in \{1, 2, \dots, D\}$, D is a number of features, $i \in \{1, 2, \dots, N\}$, N is a number of objects. The task of classification is to match each object with the most suitable class from the set of all classes $c = \{c_1, c_2, \dots, c_K\}$, K is a

number of classes. Decision algorithms restore the dependence between input attributes and classes.

In this paper, k-nearest neighbour (KNN) is used as a decision algorithm, as one of the simplest and most frequently used classifiers. The algorithm has only one parameter k that indicates the number of neighbours used to define a class label for a new data instance. As a result of the absence of other parameters, the algorithm does not need a learning stage or setting of these parameters, which is expressed in its high speed.

The algorithm operation process can be described as follows:

1. To determine the value k .
2. Save all positions and class labels of all training data (**TR**).
3. For each new instance of data **X**:
 - 3.1 Calculate distances from **X** to each of the elements **TR**.
 - 3.2 Choose from **TR** k elements with the shortest distances up to **X**.
 - 3.3 Define a class c label that has a majority of the elements selected in Step 3.2.
 - 3.4 Assign the **X** to class c .

In this paper, the value of k was chosen to be 5, which is the most preferred value in the literature (Hancer, 2020, 2021).

The feature selection problem is formulated as follows: on the given set of features, find a feature subset that does not cause a significant decrease in classification accuracy, or even increases it, when the number of features is decreased. The solution is represented as a vector $s = \{s_1, s_2, \dots, s_D\}$, where $s_i = 0$ means that the i th feature is excluded from classification and $s_i = 1$ means that the i th feature is used by the classifier.

Since an algorithm that works as a wrapper has been chosen to select features, a fitness function that evaluates the quality of a subset of selected variables must be used. We have applied the following function:

$$Fit = \alpha \times Error + (1 - \alpha) \times F/D, \quad (3)$$

where *Error* is a classification error, F is a number of features in the evaluated subset, α is the priority coefficient between an error and the ratio of selected features, $\alpha \in [0, 1]$. The classification error is the number of incorrectly classified objects in relation to the total number of instances.

The task of the binary optimization algorithm is to search for the minimum of the given function.

3.2. Fuzzy Entropy and Mutual Information

Since first founded by Shannon (1948) to address communication, information theory has now been implemented to a variety of fields, from clustering, fuzzy logic to decision making, just to name a few. Specifically, information theory has been implemented to quantify information. The information quantity is described as the amount of information observed in an event (called uncertainty) computed using probabilistic measures. In addition to the probabilistic measures, the fuzziness measures are also very common as an

uncertainty measure among researchers. In particular, the fuzziness measures differ from the probabilistic measures by introducing vagueness uncertainties rather than randomness uncertainties.

The first fuzziness extension of the Shannon's entropy was first considered by De Luca and Termini (1972). Specifically, the fuzzy entropy was described on the basis of a membership function. Based on the axioms of De Luca and Termini, Al-Ani and Khushaba (2012) defined the membership of k th feature belonging to i th class as follows:

$$\mu_{ik} = \left(\frac{\|\hat{x}_i - x_k\|_\sigma}{r \pm \epsilon} \right)^{\frac{-2}{m-1}}, \quad (4)$$

where m is the fuzziness, σ is the standard deviation, r is the radius of the data such that $r = \max \|\hat{x}_i - x_k\|_\sigma$, \hat{x}_i is the mean of data samples belonging to i th class.

Based on the membership function defined by Eq. (4), the memberships of all data samples (with size NP) for each of fuzzy sets (with size C) along k th and l th features are respectively denoted as A and B . In other words, A and B are $C \times NP$ membership matrixes for k th and l th features, respectively. Based on the membership matrix, the fuzzy marginal uncertainty measure along each k th feature is defined as follows:

$$P(k) = \frac{\sqrt{AA'}}{NP}. \quad (5)$$

The fuzzy marginal entropy along each k th feature is then calculated as follows:

$$H(k) = - \sum \sum P(k) \log P(k). \quad (6)$$

The fuzzy joint entropy, called $H(k, l)$, between k th and l th features is calculated as follows:

$$H(k, l) = - \sum \sum \frac{\sqrt{AB'}}{NP} \log \frac{\sqrt{AB'}}{NP}. \quad (7)$$

Using the joint entropy, the fuzzy mutual information between k th and l th features can be calculated as follows:

$$I(k, l) = H(k) + H(l) - H(k, l). \quad (8)$$

Assuming S is a selected feature subset for a dataset. The intra-class feature dependencies (denoted as relevancy (Rel)) and the inter-feature dependencies (denoted as redundancy (Red)) are respectively defined as follows:

$$Rel(S) = \sum_{k \in S} I(k, y), \quad (9)$$

$$Red(S) = \sum_{k \in S} \sum_{l \in S} I(k, l), \quad (10)$$

where y denotes the output variable. Then, in terms of the intra-class and the inter-feature dependencies, the fuzzy version of MIFS (Battiti, 1994) along S can be calculated as follows:

$$FMIFS = \gamma \times Rel + (1 - \gamma) \times Red, \quad (11)$$

where γ is a priority coefficient between Rel and Red .

In case of fuzzy entropy measures, the fuzzy version of symmetry uncertainty (SU) (Witten and Frank, 2002) between the k th feature and the output variable can be calculated as follows:

$$FSU(k, y) = \frac{H(k) + H(k|y)}{H(k) + H(y)}, \quad (12)$$

where $H(k|y)$ represents the conditional entropy of the output variable given k th feature.

3.3. Proposed Binary PSO Variants

This subsection describes the investigated modifications of the PSO algorithm for finding suboptimal subsets of features. A population of randomly generated binary vectors s is given to the algorithm input. After passing a specified number of iterations, the PSO outputs a solution with the best value of the fitness function.

3.3.1. Standard Binary PSO with S-Shaped Transformation Function

This method uses an S-shaped transformation function to update elements of binary vectors. The transformation function takes as input the velocity of the j th element of the particle. We use the following standard expression:

$$F_S(v_j) = \frac{1}{1 + e^{-v_j}}, \quad (13)$$

where v_j is the velocity value of the particle element s_j , j is a position number of the element in the vector s . Based on the velocity, the function value is calculated, which is then converted to the binary equivalent according to a specific rule:

$$s_i = \begin{cases} 1, & rand(0, 1) < F_S(v_j), \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

Thus, the elements with the highest positive speed are more likely to be equal to zero.

V-shaped transformation functions are symmetric about the ordinate axis. In this paper, two such functions were tested. The first one is calculated using hyperbolic tangent:

$$F_{V_1}(v_j) = |\tanh(v_j)|. \quad (15)$$

The second function belongs to the same family:

$$F_{V_2}(v_j) = \left| v_j / \sqrt{1 + (v_j)^2} \right|. \quad (16)$$

To update elements in the binary feature vector, we will use the same rule:

$$\text{IF } \text{rand}(0, 1) < F_V(v_j), \quad \text{then } s_j = 1, \quad \text{else } s_j = 0,$$

where $F_V(v_j)$ is one of the two V-shaped transformation functions. In this case, a vector element that has gained high negative speed also has a high chance of becoming zero.

3.3.2. Local Search

Due to its promising performance with differential evolution for feature selection, the local search module (Hancer, 2019) is also adopted in the standard binary PSO with S-shaped transformation function to eliminate less informative features from the selected feature subset of the global best solution and add unselected informative features to the selected feature subset of the global best solution (p_g). Once all the positions are updated, the add/remove process is applied in a probabilistic manner. The add process is described as follows:

1. A set of unselected features in the data, which are not present in the selected subset, are ranked in descending order in terms of the fuzzy version of the SU criterion, defined by Eq. (10).
2. A predefined percentage of top features from the unselected feature subset are determined as candidate features for the selected feature subset.
3. The mean value of the fuzzy SU (called mean(FSU)) criterion within the selected feature subset is computed.
4. If the fuzzy SU score of each candidate feature is greater than mean(FSU), it is added to the selected feature subset.

The remove process is described as follows:

1. The features from the selected subset are ranked in terms of the fuzzy version of the MIFS criterion, defined by Eq. (9).
2. A predefined percentage of selected features from the selected feature subset which own lower fuzzy MIFS scores are determined as candidates for the elimination process.
3. A random number between 0 and 1 is generated for each candidate.
4. If the random number of each candidate is smaller than a user-specified threshold, it is eliminated from the selected feature subset.

After adding or removing features, the fitness value of the updated feature subset is calculated by Eq. (1). If the fitness value of the updated feature subset is better than that of the current selected feature subset, the positions of p_g are then updated based on the updated feature subset. This process is repeated for a smaller predefined number of iterations. Notice that the fuzzy SU and the fuzzy MIFS scores of each feature in the data are first calculated before the evolutionary process to avoid additional computational cost. More information concerning the local search module can be found in Hancer (2019).

3.3.3. New Fitness

The overall aim of a wrapper method is to minimize both the classification error rate and the number of features using a classifier. From an optimization perspective, a wrapper method tries to minimize the objective function, defined by Eq. (1). Different from wrappers, filters select a feature subset by evaluating the relationships between feature space and the output variable. To evaluate this relationship, information theoretic criteria may be treated as the most widely applied for feature selection among a variety of criteria. To utilize filter criteria in wrappers, researchers generally use the following cases:

1) Two-stage feature selection: In this case, a filter approach is initially carried out to reduce a feature space, and then a wrapper approach is performed on the reduced feature space. However, no interaction is built between the stages, i.e. it does not guarantee high-performing feature subsets.

2) Locally hybridized feature selection: In this case, a local elimination module based on a filter (wrapper) evaluation is adopted in a wrapper (filter) method to increase the effectiveness of the method in the feature space. Compared to the previous case, this case can help a feature selection method to obtain high-performing feature subsets. The aforementioned method, called local search, is an example of this case.

To bring the positive effects of both wrappers and filters, different from these cases, we develop a new fitness function which does not require any additional module or computations during the selection process. The newly developed function is adopted in the standard binary PSO with an S-shaped transformation function, defined as follows.

$$newFF = \alpha \times (Error \times Rel) + (1 - \alpha) \times (F/D \times Red), \quad (17)$$

where *Rel* and *Red* are defined by Eqs. (7) and (8).

3.3.4. The Merge Procedure

The essence of this operation is to save matching elements of two binary vectors and randomly select from non-matching elements. Getting the value of the d_j element of the resulting d vector from the corresponding elements of the a and b vectors can be described as follows:

$$merge(\mathbf{a}, \mathbf{b}) = \begin{cases} d_j = a_j = b_j, & \text{if } a_j = b_j, \\ d_j = a_j, & \text{if } a_j \neq b_j \text{ and } rand \leq 0.5, \\ d_j = b_j, & \text{if } a_j \neq b_j \text{ and } rand > 0.5, \end{cases} \quad (18)$$

where a_j is the j th element of the input vector \mathbf{a} , b_j is the j th element of the input vector \mathbf{b} , d_j is the j th element of the output vector \mathbf{d} , and $rand$ is a random real number from 0 to 1.

An example of how this operation works is shown as follows. Assume that $\mathbf{a} = [1, 0, 0, 1]$, $\mathbf{b} = [1, 0, 1, 0]$, $rand_1 = 0.38$ and $rand_2 = 0.67$. The first two elements of vectors \mathbf{a} and \mathbf{b} are the same, so $d_1 = 1$, $d_2 = 0$. To determine the values of the third and fourth elements, we generate two random numbers: $rand_1$ to determine d_3 and $rand_2$ to determine d_4 . Since $rand_1 < 0.5$, then $d_3 = a_3 = 0$. Similarly, since $rand_2 > 0.5$, then

$d_4 = b_4 = 0$. Thus, the resulting vector \mathbf{d} will be equal to $[1, 0, 0, 0]$. The essence of the binarization method in this case will be to replace all the original arithmetic operations with the merge operation.

3.3.5. The Hybrid Method of the Modified Arithmetic Operation and the Merge Procedure

This approach to binarization is a combination of the modified algebraic operations and the merge procedure. As in the method of modified algebraic operations, instead of multiplication you need to use conjunction (AND), instead of subtraction – exclusive or (XOR), but instead of addition – the merge procedure.

After the binarization of the PSO algorithm by this method, change of the particle position can be described by the following expressions:

$$s(t+1) = \text{merge}(s(t), v(t+1)), \quad (19)$$

$$v(t+1) = \text{merge}(v(t), \text{merge}(\text{randbinvector}_1 \wedge (\mathbf{p}(t) \oplus s(t)), \text{randbinvector}_2 \wedge (\mathbf{p}_g(t) \oplus s(t))), \quad (20)$$

where $\mathbf{p}(t)$ is a vector of previous coordinates of the particle $s(t)$ with the best value of the fitness function, $\mathbf{p}_g(t)$ is a vector of coordinates of the particle that reached the best fitness function in the entire swarm for t iterations; randbinvector_1 and randbinvector_2 are random binary vectors.

As already mentioned when describing the method of modified algebraic operations, the disadvantage of this method is the impossibility to use real coefficients. That is why in this case the coefficients w , c_1 , c_2 will not be used.

4. Description of the Experiment

4.1. Preprocessing

The graphics tablets produce signals of the X -axis and the Y -axis for the points where the user touches the tablet with a digital pen. These signals are the source of the pure digitized data. While writing a signature, it is possible to obtain signatures which differ in the case of size, orientation and deviation due to some factors (e.g. the limited space and the inclination of the tablet) even with the same person. Such differences may adversely affect the reliability of a verification system. Therefore, all signatures should be transformed in the case of position, rotation and size to decrease the differences in the signals. We follow the procedures described in Lam and Kamins (1989), Yanikoglu and Kholmatov (2009) to carry out the transformation process, defined as follows:

1. *Elimination of discontinuities*: It is possible to observe discontinuities (broken lines) in signals, if the user detaches the pen from the tablet while writing the signature. To recombine the broken lines in signals, two interpolation techniques can be used. If the space between broken lines is wide, the newton interpolation can be used; otherwise, the linear

interpolation can be applied. For more detailed information concerning the reconstruction of broken lines, please see Samet and Hancer (2012).

2. *Elimination of the signature inclination*: The angles of a signature written by the same user is possible to be varied. The signature inclination is reduced using an exponential function with respect to the X -axis, defined as in Yanikoglu and Kholmatov (2009).

3. *Normalization*: The user can write his or her signature in any area of the tablet. Accordingly, the positions of the signature may differ. To address this issue, normalization is applied by subtracting the mean value from each value of the signal.

4. *Scaling*: The same person can write his or her signature in different sizes. It is therefore necessary to scale their size at a reasonable rate.

4.2. Feature Extraction

A set of dynamic signature features described by authors of Fierrez-Aguilar *et al.* (2005) are used in this work. One hundred features were selected. The initial data for the feature extraction are x and y signals ($x_1, x_2, \dots, x_N; y_1, y_2, \dots, y_N$). Some of the features used require signals that characterize the velocity and acceleration of the digital pen. For the velocity signals, their absolute values and values for each coordinate are computed; for the acceleration signals, only their relative values are used. These signals can be obtained using the x and y signals. The absolute velocity signals at certain instants can be measured using the distance between two adjacent touch points of the pen. The absolute velocity signals are defined as follows:

$$v_i = \frac{\sqrt{(x_{i+1} - x_i)^2} + \sqrt{(y_{i+1} - y_i)^2}}{\Delta t},$$

where $i = 1, 2, \dots, N - 1$ and Δt is the duration of an instant. The velocity signals in the respective coordinates are computed as follows:

$$V_{x_i} = \frac{x_{i+1} - x_i}{\Delta t}, \quad v_{y_i} = \frac{y_{i+1} - y_i}{\Delta t},$$

where $i = 1, 2, \dots, N - 1$. The relative acceleration signals in the respective coordinates are computed as follows:

$$a_{x_i} = \frac{v_{x_{i+1}} - v_{x_i}}{\Delta t}, \quad v_{y_i} = \frac{v_{y_{i+1}} - v_{y_i}}{\Delta t},$$

where $i = 1, 2, \dots, N - 2$.

4.3. Dataset

The signature signals were taken from the SVC2004 dataset (Task 1) (Yeung *et al.*, 2004). There were 40 sets of signatures. Each set contained 20 genuine signatures from one signer and 20 skilled forgeries from at least four other signers. Each signer was asked to provide

Table 1
The brief notation of the tested modifications of the PSO algorithm.

Method	Designation
The S-shaped transformation function	<i>standard</i>
Previous method with new fitness function	<i>newFF</i>
New version of local search	<i>local</i>
The merge operation	<i>merge</i>
The hybrid method of the modified arithmetic and merging operation	<i>opmerge</i>
The FV1 transformation functions	<i>tanh</i>
The FV2 transformation function	<i>Vshaped</i>

20 genuine signatures. For privacy reasons, the signers were advised not to use their real signatures. Instead, each of them was asked to design a new signature and practice writing it so that this new signature remained relatively consistent over different signature instances, just like a real one. The signers were also reminded that consistency should not be limited to the spatial consistency in the signature shape and it should also include the temporal consistency of the dynamic features.

4.4. Parameter Settings

Table 1 shows the designations of the studied PSO modifications. The parameter settings for these PSO modifications are presented as follows. The population size is set to 50 and the maximum number of evaluations is chosen as 2500 for all the methods. The parameters of local are set to the default values defined in Hancer (2019). The following parameters were used in methods using transformation functions: positive acceleration coefficients c_1 and c_2 are equal to 1, the coefficient of inertia w is 0.5.

5. Experimental Results

The experiment was conducted in accordance with a 10-fold cross-validation scheme. The data was split into ten training and ten test samples. However, instances in different test samples are not repeated. Due to the stochasticity of the algorithm, each version was run 30 times, and then the results were averaged.

5.1. Classification Results

Table 2 presents the results of constructing classifiers using the binary PSO variants in terms of different α values. Four characteristics are given: overall accuracy, type I error (False Rejection Rate, FRR), type II error (False Acceptance Rate, FAR), and the number of selected features (Features). Type I error shows what percentage of legitimate users was mistaken by the classifier for an intruder. Type II error demonstrates the percentage of intruders who managed to deceive the classifier and pass their signature off as the signature of a legitimate user. Bold text in the table indicates the best results within a group with one value of the coefficient α in the fitness function.

Table 2
Averaged classification results after feature selection by the PSO algorithm.

Metric	Accuracy			
Coefficient	$\alpha = 1$	$\alpha = 0.9$	$\alpha = 0.7$	$\alpha = 0.5$
<i>standard</i>	88.34 ± 5.22	89.18 ± 5.57	88.75 ± 5.76	88.65 ± 5.69
<i>newFF</i>	87.69 ± 3.65	89.18 ± 3.69	89.93 ± 4.07	89.58 ± 4.77
<i>local</i>	88.29 ± 5.09	90.64 ± 5.54	90.37 ± 5.57	89.72 ± 5.94
<i>merge</i>	87.20 ± 6.83	87.62 ± 7.14	87.54 ± 7.17	87.68 ± 7.07
<i>opmerge</i>	89.82 ± 6.62	88.35 ± 7.94	88.49 ± 7.82	88.07 ± 8.07
<i>tanh</i>	88.38 ± 6.51	87.88 ± 7.73	87.99 ± 7.63	87.64 ± 8.10
<i>Vshaped</i>	88.80 ± 6.70	87.65 ± 7.91	87.49 ± 8.25	87.65 ± 8.05
Metric	FRR			
Coefficient	$\alpha = 1$	$\alpha = 0.9$	$\alpha = 0.7$	$\alpha = 0.5$
<i>standard</i>	2.84 ± 3.07	3.90 ± 4.64	4.02 ± 4.36	4.10 ± 4.71
<i>newFF</i>	2.28 ± 1.53	2.26 ± 1.65	2.31 ± 2.01	2.68 ± 2.87
<i>local</i>	2.92 ± 3.16	5.08 ± 5.23	5.43 ± 5.44	5.35 ± 5.63
<i>merge</i>	2.73 ± 3.62	3.82 ± 5.55	3.97 ± 5.52	3.73 ± 5.47
<i>opmerge</i>	4.35 ± 5.67	9.08 ± 9.16	9.05 ± 9.13	9.04 ± 8.93
<i>tanh</i>	3.08 ± 4.07	5.48 ± 7.36	5.37 ± 7.46	6.05 ± 8.21
<i>Vshaped</i>	3.47 ± 4.72	6.59 ± 8.49	6.67 ± 8.56	6.77 ± 8.54
Metric	FAR			
Coefficient	$\alpha = 1$	$\alpha = 0.9$	$\alpha = 0.7$	$\alpha = 0.5$
<i>standard</i>	20.48 ± 9.62	17.74 ± 9.88	18.49 ± 10.40	18.60 ± 10.21
<i>newFF</i>	22.35 ± 6.74	19.39 ± 6.73	17.83 ± 7.51	18.15 ± 8.64
<i>local</i>	20.49 ± 9.30	13.65 ± 9.08	13.82 ± 9.11	15.22 ± 9.96
<i>merge</i>	22.88 ± 12.74	20.94 ± 13.10	20.95 ± 13.21	20.91 ± 12.96
<i>opmerge</i>	16.00 ± 11.92	14.22 ± 12.40	13.97 ± 12.76	14.82 ± 12.86
<i>tanh</i>	20.16 ± 12.11	18.77 ± 13.37	18.65 ± 13.41	18.68 ± 13.79
<i>Vshaped</i>	18.93 ± 12.17	18.12 ± 13.45	18.35 ± 14.17	17.93 ± 13.43
Metric	Features			
Coefficient	$\alpha = 1$	$\alpha = 0.9$	$\alpha = 0.7$	$\alpha = 0.5$
<i>standard</i>	48.17 ± 4.99	28.97 ± 2.76	28.65 ± 2.72	26.27 ± 2.46
<i>newFF</i>	66.54 ± 3.75	49.25 ± 3.26	36.16 ± 2.81	29.12 ± 2.55
<i>local</i>	46.65 ± 5.28	9.19 ± 4.64	8.11 ± 4.45	6.95 ± 3.83
<i>merge</i>	55.64 ± 8.84	30.27 ± 4.35	30.29 ± 4.33	29.53 ± 4.24
<i>opmerge</i>	37.60 ± 10.99	1.89 ± 0.66	1.89 ± 0.64	1.87 ± 0.62
<i>tanh</i>	47.33 ± 9.98	16.07 ± 5.58	16.07 ± 5.58	12.95 ± 5.59
<i>Vshaped</i>	44.21 ± 10.35	10.10 ± 4.13	10.25 ± 4.06	9.57 ± 3.71

The presented results show that with a value of α less than one, the number of features in the set decreases with any method. In this case, the overall accuracy of the classification is either improving (methods *standardFF*, *newFF*, *local*, *merge*) or decreasing (methods *opmerge*, *tanh*, *Vshaped*). In general, the best ability to reduce features with a decrease in the coefficient α was demonstrated by the methods *opmerge*, *local* and *Vshaped*. It should be indicated that the classification model built on a smaller number of features can even provide a remarkable performance from the perspective of the other metrics. For instance, *opmerge* can achieve competitive verification results despite selecting only 2 features from 100 available features. For another instance, *local* shows the best verification performance

Table 3
Ranks of Friedman's two-way analyses for investigated metrics.

Metric	1 – Accuracy			
Coefficient	$\alpha = 1$	$\alpha = 0.9$	$\alpha = 0.7$	$\alpha = 0.5$
<i>standard</i>	4.01	3.05	2.67	2.91
<i>newFF</i>	4.54	3.34	2.64	2.67
<i>local</i>	3.97	2.55	3.61	3.61
<i>merge</i>	5.15	4.85	4.85	4.64
<i>opmerge</i>	2.97	4.22	4.32	4.11
<i>tanh</i>	3.9	4.9	4.74	4.91
<i>Vshaped</i>	3.45	5.09	5.16	5.14
Metric	FRR			
Coefficient	$\alpha = 1$	$\alpha = 0.9$	$\alpha = 0.7$	$\alpha = 0.5$
<i>standard</i>	3.79	4.11	4.21	4.38
<i>newFF</i>	2.35	1.48	1.46	1.78
<i>local</i>	3.64	3.3	2.96	2.94
<i>merge</i>	3.59	3.03	3.23	2.83
<i>opmerge</i>	5.74	5.69	5.85	5.53
<i>tanh</i>	4.13	4.73	4.63	4.98
<i>Vshaped</i>	4.78	5.68	5.66	5.59
Metric	FAR			
Coefficient	$\alpha = 1$	$\alpha = 0.9$	$\alpha = 0.7$	$\alpha = 0.5$
<i>standard</i>	4.18	2.43	2.36	2.75
<i>newFF</i>	4.71	4.24	3.8	3.78
<i>local</i>	4.04	3.48	4.4	4.23
<i>merge</i>	5.28	5.61	5.2	5.36
<i>opmerge</i>	2.63	3.04	3.11	3.06
<i>tanh</i>	3.91	4.58	4.6	4.5
<i>Vshaped</i>	3.26	4.64	4.53	4.33
Metric	FRR			
Coefficient	$\alpha = 1$	$\alpha = 0.9$	$\alpha = 0.7$	$\alpha = 0.5$
<i>standard</i>	3.4	2.38	2.18	2
<i>newFF</i>	6.4	7	6.9	6.3
<i>local</i>	4.35	5.25	5.18	5.13
<i>merge</i>	5.85	5.75	5.93	6.58
<i>opmerge</i>	2.15	1	1	1
<i>tanh</i>	3.33	3.98	4	4
<i>Vshaped</i>	2.53	2.65	2.83	3

for lower α cases by reducing the dimensionality at least ten times compared to the original feature set. In other words, the corresponding methods have shown good skills to detect noisy features or variables. It is therefore possible to build less complex and more generalized classification models that can obtain promising verification results.

5.2. Statistical Processing of Results

Table 3 summarizes the mean ranks calculated according to Friedman's criterion for different metrics and values of the coefficient α (at 40 observations and 6 degrees of free-

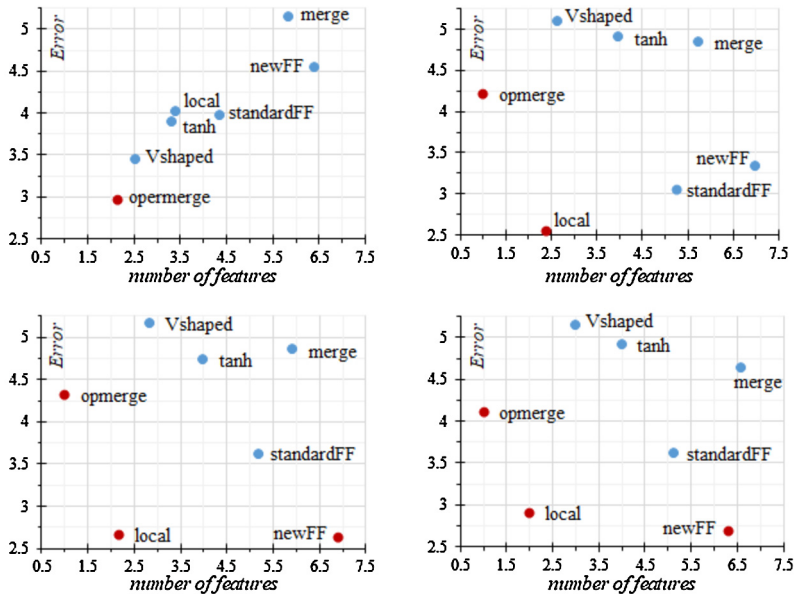


Fig. 1. Pareto front based on the ranks of the classification error and the number of features.

dom). The ranks for Accuracy and Errors (FAR, FRR) are multidirectional. To bring them to a single scale, we consider 1-Accuracy (called Error) instead of Accuracy. Since the p-value in all cases was equal to zero, which is less than the significance level (0.05), it can be concluded that there is a significant statistical difference in the results of the modification methods used.

Graphs have been created on the basis of the obtained average rank values in order to assess the Pareto front in terms of the ratio of metrics to the number of selected features. In the first group of graphs shown in Fig. 1, the inverse metric, classification error, was used instead of accuracy.

From the graphs obtained, it is clear that in all cases the use of the opmerge method provides the smallest set of features. With α coefficients equal to 0.7 and 0.5, PSO with the newFF fitness function allowed us to obtain a classifier with the lowest error. The best results in terms of the ratio of error and number of features were demonstrated by PSO with the local modification when the value of the coefficient α is less than one.

Fig. 2 shows plots with a Pareto front for the ratio of the type I error (FRR) and the number of selected features. The smallest value of the type I error for any value of the coefficient α was obtained using *newFF*, but this method coped worst of all with the selection of features.

Fig. 3 shows graphs with a Pareto front in terms of the ratio of type II error (FAR) and the number of selected features. *Opmerge* and *local* showed the lowest type II error values. For the first, this statement is valid for any value of the coefficient α , for the second, only for α less than 1.

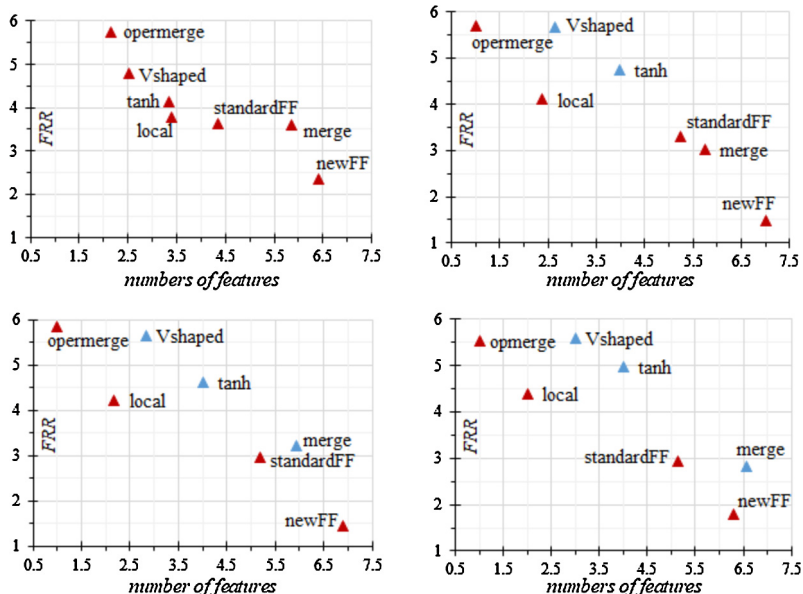


Fig. 2. Pareto front based on the ranks of the type I error and the number of features.

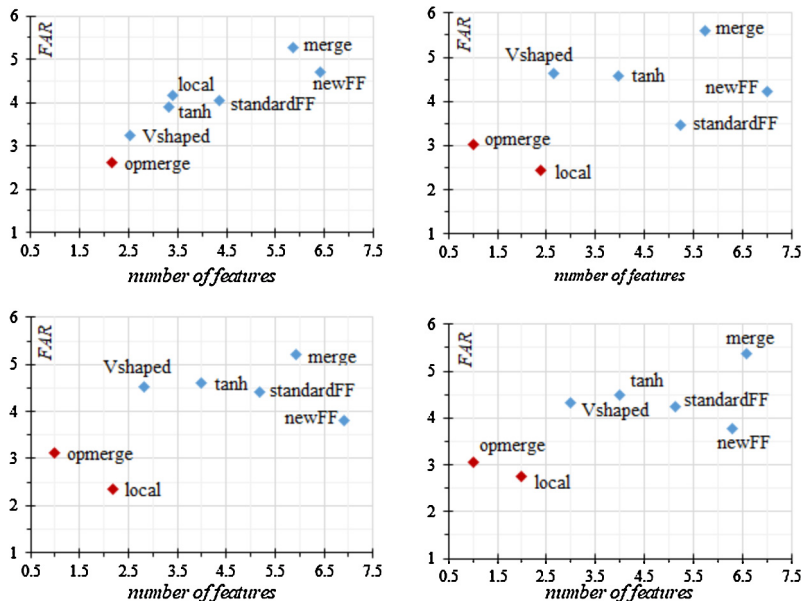


Fig. 3. Pareto front based on the ranks of the type II error and the number of features.

6. Conclusion

The work considered seven approaches for PSO modification when solving the task of feature selection for a handwritten signature authentication. The investigated methods made it possible in the considered set SVC2004, containing 100 input variables, to select subsets of size from 66.54 (using new fitness function with coefficient $\alpha = 1$) to 1.87 features (combination of the modified algebraic operations and the merge procedure with coefficient $\alpha = 0.5$). The best selection ability was demonstrated by the hybrid method, which combines the modified algebraic operations with the author's method of merging binary vectors. This method should be used when it is necessary to reduce the complexity of the authentication model as much as possible. If there is a requirement to ensure maximum availability for legal users, it is appropriate to use PSO with the new proposed fitness function, as this method demonstrates the smallest type I error. The smallest type II error, which indicates that the system is more secure from intruders, was shown by PSO with new local search. The same method made it possible to obtain a balanced result in terms of the ratio between the overall error and the number of features with the coefficient α equal to 0.9.

The determination of the α parameter may be a difficult and time-consuming process depending on a specified handwritten-signature problem. It would be better if the fitness functions which comprise of two conflicting objectives are reconsidered as a multi-objective optimization problem. This is because it is reported in the literature that it is possible to achieve higher learning performance by adopting feature selection in a multi-objective design. Concerning this subject, a possible arising problem is how a single solution will be selected from the Pareto Front. In the future, we will try to address this problem to reach higher verification performance.

References

- Al-Ani, A., Khushaba, R.N. (2012). A population based feature subset selection algorithm guided by fuzzy feature dependency. In: Hassanien, A.E., Salem, A.-B.M., Ramadan, R., Kim, T.-h. (Eds.), *Advanced Machine Learning Technologies and Applications*. Springer, Berlin, Heidelberg, pp. 430–438.
- Arora, S., Anand, P. (2019). Binary butterfly optimization approaches for feature selection. *Expert Systems with Applications*, 116, 147–160. <https://doi.org/10.1016/j.eswa.2018.08.051>.
- Ashry, A., Alrahmawy, M., Rashad, M. (2020). Enhanced quantum inspired grey wolf optimizer for feature selection. *International Journal of Intelligent Systems and Applications*, 12, 8–17. <https://doi.org/10.5815/ijisa.2020.03.02>.
- Barani, F., Mirhosseini, M., Nezamabadi-Pour, H. (2017). Application of binary quantum-inspired gravitational search algorithm in feature subset selection. *Applied Intelligence*, 47(2), 304–318. <https://doi.org/10.1007/s10489-017-0894-3>.
- Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4), 537–550. <https://doi.org/10.1109/72.298224>.
- Chantar, H., Mafarja, M., Alsawalqah, H. (2020). Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification. *Neural Computing and Applications*, 32, 12201–12220. <https://doi.org/10.1007/s00521-019-04368-6>.
- De Luca, A., Termini, S. (1972). A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory. *Information and Control*, 20(4), 301–312. [https://doi.org/10.1016/S0019-9958\(72\)90199-4](https://doi.org/10.1016/S0019-9958(72)90199-4).

- Dimauro, G. (2011). Fourier transform in numeral recognition and signature verification. In: Wang, P.S.P. (Ed.), *Pattern Recognition, Machine Intelligence and Biometrics*. Springer, Berlin, Heidelberg, pp. 823–857. https://doi.org/10.1007/978-3-642-22407-2_31.
- Fahmy, M.M.M. (2010). Online handwritten signature verification system based on DWT features extraction and neural network classification. *Ain Shams Engineering Journal*, 1(1), 59–70. <https://doi.org/10.1016/j.asej.2010.09.007>.
- Fierrez-Aguilar, J., Nanni, L., Lopez-Peñalba, J., Ortega-Garcia, J., Maltoni, D. (2005). An on-line signature verification system based on fusion of local and global information. In: Kanade, T., Jain, A., Ratha, N.K. (Eds.), *Audio- and Video-Based Biometric Person Authentication*. Springer, Berlin, Heidelberg, pp. 523–532.
- Ghosh, K.K., Singh, P.K., Hong, J., Geem, Z.W., Sarkar, R. (2020). Binary social mimic optimization algorithm with X-shaped transfer function for feature selection. *IEEE Access*, 8, 97890–97906. <https://doi.org/10.1109/ACCESS.2020.2996611>.
- Hamed, H.N.A., Kasabov, N., Shamsuddin, S.M. (2009). Integrated feature selection and parameter optimization for evolving spiking neural networks using quantum inspired particle swarm optimization. In: *International Conference of Soft Computing and Pattern Recognition*, pp. 695–698. <https://doi.org/10.1109/SoCPaR.2009.139>.
- Hancer, E. (2019). Differential evolution for feature selection: a fuzzy wrapper—filter approach. *Soft Computing*, 23(13), 5233–5248. <https://doi.org/10.1007/s00500-018-3545-7>.
- Hancer, E. (2020). New filter approaches for feature selection using differential evolution and fuzzy rough set theory. *Neural Computing and Applications*, 32, 2929–2944. <https://doi.org/10.1007/s00521-020-04744-7>.
- Hancer, E. (2021). An improved evolutionary wrapper-filter feature selection approach with a new initialisation scheme. *Machine Learning*, 1–24. <https://doi.org/10.1007/s10994-021-05990-z>.
- Kohavi, R., John, G.H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1), 273–324. [https://doi.org/10.1016/S0004-3702\(97\)00043-X](https://doi.org/10.1016/S0004-3702(97)00043-X).
- Kıran, M.S., Gündüz, M. (2012). XOR-based artificial bee colony algorithm for binary optimization. *Turkish Journal of Electrical Engineering and Computer Science*, 21, 2307–2328. <https://doi.org/10.3906/elk-1203-104>.
- Lam, C.F., Kamins, D. (1989). Signature recognition through spectral analysis. *Pattern Recognition*, 22(1), 39–44. [https://doi.org/10.1016/0031-3203\(89\)90036-8](https://doi.org/10.1016/0031-3203(89)90036-8).
- Li, S.Z., Jain, A. (2015). *Encyclopedia of Biometrics*. Springer Publishing Company, Incorporated, Boston, MA. 148997489X.
- Linden, J., Marquis, R., Bozza, S., Taroni, F. (2018). Dynamic signatures: a review of dynamic feature variation and forensic methodology. *Forensic Science International*, 291, 216–229. <https://doi.org/10.1016/j.forsciint.2018.08.021>.
- Mafarja, M.M., Eleyan, D., Jaber, I., Hammouri, A., Mirjalili, S. (2017). Binary dragonfly algorithm for feature selection. In: *International Conference on New Trends in Computing Sciences (ICTCS2017)*, pp. 12–17. <https://doi.org/10.1109/ICTCS.2017.43>.
- Mailah, M., Han, L. (2012). Biometric signature verification using pen position, time, velocity and pressure parameters. *Jurnal Teknologi*, 48(1). <https://doi.org/10.11113/jt.v48.218>.
- Mirjalili, S., Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, 9, 1–14. <https://doi.org/10.1016/j.swevo.2012.09.002>.
- Nelson, W., Kishon, E. (1991). Use of dynamic features for signature verification. In: *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 201–205. <https://doi.org/10.1109/ICSMC.1991.169685>.
- Qasim, O.S., Algamal, Z.Y. (2020). Feature selection using different transfer functions for binary bat algorithm. *International Journal of Mathematical, Engineering and Management Sciences*, 5(4), 697–706. <https://doi.org/10.33889/IJMEMS.2020.5.4.056>.
- Sakamoto, D., Morita, H., Ohishi, T., Komiya, Y., Matsumoto, T. (2001). On-line signature verification algorithm incorporating pen position, pen pressure and pen inclination trajectories. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*, Vol. 2, pp. 993–996. <https://doi.org/10.1109/ICASSP.2001.941084>.
- Samet, R., Hancer, E. (2012). A new approach to the reconstruction of contour lines extracted from topographic maps. *Journal of Visual Communication and Image Representation*, 23(4), 642–647. <https://doi.org/10.1016/j.jvcir.2012.02.005>.
- Shannon, C.E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3), 379–423. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>.

- Shi, Y., Eberhart, R.C. (1999). Empirical study of particle swarm optimization. In: *Proceedings of the 1999 Congress on Evolutionary Computation (CEC99)*, Vol. 3, pp. 1945–1950. <https://doi.org/10.1109/CEC.1999.785511>.
- Souza, V.L.F., Oliveira, A.L.L., Cruz, R.M.O., Sabourin, R. (2020). Improving BPSO-based feature selection applied to offline WI handwritten signature verification through overfitting control. In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, GECCO '20*. Association for Computing Machinery, New York, NY, USA, pp. 69–70. <https://doi.org/10.1145/3377929.3390038>.
- Witten, I.H., Frank, E. (2002). Data mining: practical machine learning tools and techniques with Java implementations. *SIGMOD Record*, 31(1), 76–77. <https://doi.org/10.1145/507338.507355>.
- Wu, G., Wang, J., Zhang, Y., Jiang, S. (2018). A continuous identity authentication scheme based on physiological and behavioral characteristics. *Sensors*, 18(1), 179. <https://doi.org/10.3390/s18010179>.
- Yanikoglu, B., Kholmatov, A. (2009). Online signature verification using fourier descriptors. *Journal on Advances in Signal Processing*, 2009, 260516. <https://doi.org/10.1155/2009/260516>.
- Yeung, D.-Y., Chang, H., Xiong, Y., George, S., Kashi, R., Matsumoto, T., Rigoll, G. (2004). SVC2004: first international signature verification competition. In: Zhang, D., Jain, A.K. (Eds.), *Biometric Authentication*. Springer, Berlin, Heidelberg, pp. 16–22.
- Yuan, X., Nie, H., Su, A., Wang, L., Yuan, Y. (2009). An improved binary particle swarm optimization for unit commitment problem. *Expert Systems with Applications*, 36(4), 8049–8055. <https://doi.org/10.1016/j.eswa.2008.10.047>.
- Zouache, D., Ben Abdelaziz, F. (2018). A cooperative swarm intelligence algorithm based on quantum-inspired and rough sets for feature selection. *Computers & Industrial Engineering*, 115, 26–36. <https://doi.org/10.1016/j.cie.2017.10.025>.

E. Hancer received the BS degree in the Department of Mathematics and Computer Science from Cankaya University, Turkey, in 2009, and the MS and PhD degrees in Computer Engineering from Erciyes University, Turkey, in 2012 and 2016, respectively. During his PhD, he worked as a visiting researcher at Victoria University of Wellington in New Zealand between 2014 and 2015 for 1 year. He is currently an associate professor at the Department of Software Engineering, Mehmet Akif Ersoy University. In 2020 and 2021, he was selected among the top 2% of the world according to the article titled “Updated science-wide author databases of standardized citation indicators” published in PLOS Biology by Dr. John PA Ioannidis, K.W. Boyack, and J. Baas from Stanford University. His research interests include image analysis, clustering, dimensionality reduction, evolutionary computation and IoT authentication. Dr. Hancer has been serving as a reviewer for several international journals and conferences in the field, such as *Applied Soft Computing*, *Pattern Recognition*, *IEEE CEC* and *IEEE SSCI*.

M. Bardamova is a junior researcher at the Laboratory of Collection, Analysis and Control of Biological Signals at Tomsk State University of Control Systems and Radioelectronics (TUSUR). She graduated from the Faculty of Security of the same university in 2017. Currently, she is a senior instructor of the Department of Complex Information Security of Computer Systems. Her main research interests include data mining, fuzzy modelling and machine learning. She has authored and co-authored more than 10 journal and conference articles.

I. Hodashinsky graduated from the Faculty of Control Systems, Novosibirsk Electrotechnical Institute, in 1975. He received the PhD degree in 1984, the Dr. Sc. degree in 2004 from Tomsk State University of Control Systems and Radioelectronics, Russia. He received the professor title at the 2011 and currently is a professor at Tomsk State University of Control Systems and Radioelectronics. His main research interests include the computational intelligence, fuzzy modelling, pattern recognition, knowledge discovery, and data mining. He is the author and co-author of over 170 journal and conference papers as well as technical articles. Prof. Hodashinsky is a member of IEEE, IEEE Computational Intelligence Society.

K. Sarin received Candidate of Sciences in Technology degree from Tomsk State University of Control Systems and Radioelectronics, Russian Federation, in 2016. He is currently an assistant professor at the Department of Complex Information Security of Computer Systems, Tomsk State University of Control Systems and Radioelectronics. His research interests include data mining, fuzzy systems, metaheuristic optimization and biometric authentication.

A. Slezkin is a certified specialist in the field of information security and automated systems (he received his specialist diploma in 2019 from Tomsk State University of Control Systems and Radioelectronics). He is also an experienced software developer, including software designed for auditing and information security. He is currently a postgraduate student at Tomsk State University of Control Systems and Radioelectronics and a junior researcher in the laboratory of extraction, analysis and management of biological signals at the Institute of Systems Integration and Security. His research interests include fuzzy systems, metaheuristic optimization algorithms, and feature selection.

M. Svetlakov is a research associate at the Department of Complex Information Security of Computer Systems, Tomsk State University of Control Systems and Radioelectronics. M. S. received the qualification of Information Security Specialist from Tomsk State University of Control Systems and Radioelectronics, Russia, in 2019. His research interests include fuzzy systems, clustering, metaheuristics optimization, biometrics.