NOVEL MODIFICATIONS OF THE MULTI-OBJECTIVE GENETIC ALGORITHM FOR SVM CLASSIFIER[1]

Liliya Demidova, Maxim Egin, Roman Tishkin

Ryazan State Radio Engineering University, Moscow Technological Institute
demidova.liliya@gmail.com, eginmm@gmail.com, roman.tishkin@mail.ru
Russian Federation

Abstract: In this paper we suggest new modifications of the multiobjective genetic algorithm – NSGA-II, which are aimed to improve quality classification ratings in the SVM classifier development. The proposed modifications of genetic operators in NSGA-II, autotuning algorithm in particularly allows to minimize the dependence of the NSGA-II result from the parameters' values which are set manually. The experimental results obtained on the basis of the model and real datasets of loan scoring, medical diagnostics, psychognosis, etc. confirm the efficiency of the proposed algorithms and the practicability of further research in this field.

Key words: support vector machine, genetic algorithm, NSGA-II, Pareto-front, autotuning algorithm

1. INTRODUCTION

The machine learning algorithms can be divided into the supervised and unsupervised learning algorithms. Unsupervised algorithms are applied when the labels of data are unavailable. Otherwise, the supervised learning is used. In the unsupervised learning, the problem is to find the structure like clusters in the unlabeled dataset. The supervised learning uses the training dataset to construct classifier, which can be used to classify new data.

The Support Vector Machine (SVM) algorithm is the supervised machine learning algorithm. This algorithm is one of the boundary classification algorithms [1, 2]. Nowadays, it is used to solve different classification problems in various applications with great success.

The SVM classifier is used for training, testing, and classification. Satisfactory quality of training and testing allows using the SVM classifier to classify new objects. The SVM classifiers are well-known for their excellent performance in the sphere of the statistical classification. Choosing optimal parameters for the SMV classifier is a very signifi-

[1] The paper has been presented on the 6th International Workshop on Mathematical Models and their Applications (IWMMA'2017), 13-15.11.2017, Krasnoyarsk, Russian Federation and it is not published in the Conference Proceedings.

The reported study was funded by RFBR according to the research project № 17-29-02198.

cant problem. It is necessary to find the kernel function type, values of the kernel function parameters and value of the regularization parameter [1, 2]. It is difficult to provide implementing of high-accuracy data classification with the use of the SVM classifier without adequate solution to this problem.

In the simplest case solution to this problem can be found by a search of the kernel function types, values of the kernel function parameters and the value of the regularization parameter that demands significant computational expenses. To estimate the classification quality, the following classification indicators can be used [3]: overall accuracy, completeness, sensitivity, specificity, F-measure, etc. All of them should be maximized. Usually, one of them is chosen as the main classification quality indicator, which describes the quality of the SVM classifier in the process of development [4, 5]. Other indicators don't participate in the SVM classifier assessment and don't influence the choice of the best SVM classifier. But sometimes, it is necessary to consider two or more classification quality indicators simultaneously to choose the best SVM classifier. In this case, the set of the most significant classification indicators must be chosen. These classification quality indicators should be maximized simultaneously. In addition, it is actually to minimize the support vectors number that allows minimizing the time expenditures on the SVM classifier development and classification of new data and decreasing the risk of retraining.

Usually, developing the SVM classifiers requires working with complex multiextreme functions and multi-parameter objective functions. Gradient methods are not suitable for the search of the optimum of such objective function. But the search algorithms of stochastic optimization, such as the genetic algorithm [6], the particle swarm algorithm [3], the clonal selection algorithm [7], etc., are being used. Each of the optimal decisions is carried out at once in the space of possible decisions. These nature inspired swarm optimization algorithms are very well suited for the distributed architecture and handling of high volume unstructured data.

Various well proved approaches can be applied for the solution of the problem of the simultaneous accounting of two or more classification quality indicators. In particular, the approach, based on the several multiobjective optimization algorithms, including, the evolutionary algorithms, is being actively developed. In recent years a number of the multiobjective evolutionary algorithms (MOEA) have been suggested [8 – 11]. The main reason for this is their ability to find the multiple Pareto-optimal solutions in one single simulation run. These algorithms work with the population of solutions. Therefore, the primary attention is to be paid to maintaining the diversity and spread of solutions. Such MOEAs provide the solution of the account's problem of the several objective functions (quality indicators) at the analysis of various applied problems. The multiobjective genetic algorithms (MOGA) [8] are one of the best known multiobjective optimization algorithms.

It is necessary to mention the multiobjective clonal selection algorithms (MOCSA) [7], which are less designed and, in the majority, borrow the principles of the multiobjective optimization underlain in the genetic algorithms. The possibility of this borrowing can be explained with many similar mechanisms of the evolutionary process realization in the MOGA and MOCSA. The analysis of merits and demerits of the MOEAs shows that such the MOGAs as the NSGA-II and the NSGA-III are significantly better than others because they can successfully solve more difficult problems of the multiobjective optimization [8 – 10]. However, the NSGA-III requires the additional user knowledge in determining parameter settings [9].

Due to the aforesaid we suggest to use NSGA-II in the SVM classifier development for the purpose of the simultaneous accounting of several quality indicators. Both the autotuning and the hybridization problems of the NSGA-II, which are being considered in this article, are relevant.

The offered modifications of NSGA-II are supposed to allow, firstly, the reduction of dependence of the results of the work of the NSGA-II on such parameters whose values are more efficiently to change during the algorithm work based on the success of these values specified during the initialization. The values of these parameters depend on the features of the problem being solved, therefore it is expedient to introduce their autotuning. Secondly, the offered hybrid algorithm of post-processing of the work's results of the NSGA-II which are presented by the Pareto-fronts of decisions will allow both to improve some decisions of the front, and to reduce the front size that will facilitate the choice of the decision which is most suitable to statements of the problem.

The rest of this paper is structured as follows. Section "SVM classifiers" presents the main stages of the SVM classifier development. Section "NSGA-II" presents the main ideas of the NSGA-II. Section "Novel modifications of the NSGA-II" details the proposed novel modifications of the NSGA-II. Firstly, it describes the autotuning algorithm for probability and distribution indexes for crossover and mutation. Secondly, it explains ideas of the hybrid algorithm, which allows minimizing the Pareto-front's size. Experimental results follow in Section "Results". Finally, conclusions are drawn in Section "Conclusions".

2. RELATED WORK

2.1. SVM classifiers

Let the dataset be a set in the form of $\{(z_1, y_1),...,(z_s, y_s)\}$, in which each object $z_i \in Z$ ($i = \overline{1,s}$; s is the number of objects) is assigned to a number $y_i \in Y = \{+1; -1\}$ having a value of +1 or −1 depending on the class of the object z_i. It is assumed that the i-th object is mapped to q-dimensional vector of numerical values of characteristics $z_i = (z_i^1,...,z_i^q)$, where z_i^l is the numeric value of the l-th characteristic for the i-th object ($i = \overline{1,s}$; $l = \overline{1,q}$).

It is necessary to use the special function $\kappa(z_i, z_\tau)$, which is called the kernel, to build the classifier $F: Z \to Y$, which compares the class to the number from the set $Y = \{+1; -1\}$ or some object from the set Z. The separating hyperplane for the objects from the training set can be represented by equation $< w, z > + b = 0$, where w is a vector-perpendicular to the separating hyperplane; b is a bias; $< w, z >$ is a scalar product of vectors w and z [1 – 3]. The condition $-1 < < w, z > + b < 1$ specifies a strip that separates the classes. The wider the strip, the more confidently we can classify objects. The objects closest to the separating hyperplane, are exactly on the bounders of the strip.

Finding the separating hyperplane is basically the dual problem of searching a saddle point of the Lagrange function, which reduces to the problem of quadratic programming, containing only dual variables [1, 2]:

$$\begin{cases} -L(\lambda) = -\sum_{i=1}^{S} \lambda_i + \\ \quad + \frac{1}{2} \cdot \sum_{i=1}^{S} \sum_{\tau=1}^{S} \lambda_i \cdot \lambda_\tau \cdot y_i \cdot y_\tau \cdot \kappa(z_i, z_\tau) \to \min\limits_{\lambda}, \\ \quad \sum_{i=1}^{S} \lambda_i \cdot y_i = 0, \\ \quad 0 \le \lambda_i \le C, \ i = \overline{1,S}, \end{cases} \tag{1}$$

where λ_i is a dual variable; z_i is the object of the training set; y_i is a number (+1 or −1), which characterize the class of the object z_i from the experimental data set; $\kappa(z_i, z_\tau)$ is a kernel function; C is a regularization parameter ($C > 0$); S is the number of objects in the dataset; $i = \overline{1,S}$.

In training of the SVM classifier it is necessary to determine the kernel function type $\kappa(z_i, z_\tau)$, values of the kernel parameters and value of the regularization parameter C, which allows finding the compromise between maximizing of the gap separating the classes and minimizing of the total error. The linear, polynomial, radial basis or sigmoid function can be used as the kernel function $\kappa(z_i, z_\tau)$. In this research the radial basis function (Gaussian function) is used as the kernel function type: $\kappa(z_i, z_\tau) = exp(- < z_i - z_\tau, z_i - z_\tau > /(2 \cdot \sigma^2))$, where $< z_i - z_\tau, z_i - z_\tau >$ is a scalar product; σ [$\sigma > 0$ (by default $\sigma^2 = 1$)].

As a result of the training, the classification function is determined in the following form [1, 2]:

$$f(z) = \sum_{i=1}^{S} \lambda_i \cdot y_i \cdot \kappa(z_i, z_\tau) + b. \tag{2}$$

The classification decision, associating the object z to the class −1 or +1, is adopted in accordance with the rule:

$$F(z) = sign(f(z)) = sign(\sum_{i=1}^{S} \lambda_i \cdot y_i \cdot \kappa(z_i, z_\tau) + b). \tag{3}$$

The SVM classifier training results in determining the support vectors. Using the evolutionary algorithm, for example, the genetic algorithm (GA), provides better quality of classification by choosing the values of the kernel function parameters and the regularization parameter value. The single-objective genetic algorithm (SOGA) works with the single classification quality indicator which is used as the objective function. In particular, overall accuracy, completeness, sensitivity, specificity, F-measure can be used in the role of the objective function of the SOGA. If it is necessary to optimize two or more classification quality indicators simultaneously, the multi-objective genetic algorithm can be used.

2.2. NSGA-II

The NSGA-II (Non-dominated Sorting Genetic Algorithm II) [10] is one of the most prominent algorithms applied for multi-objective problems. The NSGA-II has the following three features. It uses an elitist principle, i.e. the elites of a population are given the opportunity to be carried to the next generation. Is uses an explicit diversity preserving

mechanism on the basis of the crowding distance. It emphasizes the non-dominated solutions. This algorithm implements three innovations: a fast nondominated sorting procedure, a fast crowded distance estimation procedure, and a simple crowded comparison operator. The main ideas of the NSGA-II are described in [10].

In this work the Matlab implementation of the NSGA-II was used [12]. The main steps of this implementation can be described as follows.

Step 1. Population initialization. The population with P chromosomes is initialized based on the problem range and constraints.

Step 2. Non-dominated sorting. The initialized population is sorted based on non-domination with application of the fast sorting algorithm [8]. This algorithm utilizes the information about the set that an individual dominate and number of individuals that dominate the individual.

Step 3. Crowding distance calculation. All the individuals in the population are assigned a crowding distance value. The crowding-distance computation requires sorting the population according to each j-th objective function value ($j=\overline{1,M}$) in ascending order of magnitude. Then, for each objective function, the boundary solutions (solutions with smallest and largest function values) are assigned an infinite distance value. All other intermediate solutions are assigned a distance value equal to the absolute normalized difference in the function values of two adjacent solutions. The overall crowding-distance value is calculated as the sum of individual distance values corresponding to each objective function [10].

Step 4. Selection. The selection is carried out using a crowded-comparison-operator (π_n). The crowded-comparison operator guides the selection process toward a uniformly spread-out Pareto optimal front. Every i-th individual in the population has two attributes: non-domination rank i_{rank}; crowding distance $i_{distance}$. Individuals in the u-th front will have the rank $i_{rank}=u$. The partial order for the i-th and l-th individuals is define as $i\ \pi_n\ l$ if ($i_{rank}<l_{rank}$) or (($i_{rank}=l_{rank}$) and ($i_{distance}>l_{distance}$)) [10]. If two solutions belong to the same u-th front, the solution which is located in a lesser crowded region is better.

The individuals are selected by using a binary tournament selection with crowed-comparison-operator.

Step 5. Genetic operators' application. The children are created on the basis of the simulated binary crossover [13, 14] and the polynomial mutation [14].

5.1. Application of the simulated binary crossover [13, 14]. The simulated binary crossover simulates the binary crossover observed in nature. The children are created according to formulas:

$$c_{1,k}=\frac{1}{2}\cdot[(1-\beta_k)\cdot p_{1,k}+(1+\beta_k)\cdot p_{2,k}]; \quad (4)$$

$$c_{2,k}=\frac{1}{2}\cdot[(1+\beta_k)\cdot p_{1,k}+(1-\beta_k)\cdot p_{2,k}], \quad (5)$$

where $c_{i,k}$ is the *i*-th child with *k*-th component; $p_{i,k}$ is the selected parent; β_k is a sample from a random number generated having the density

$$p(\beta_k) = \begin{cases} \frac{1}{2} \cdot (\eta_c + 1) \cdot \beta_k^{\eta_c}, \ if \ 0 \le \beta_k \le 1, \\ \frac{1}{2} \cdot (\eta_c + 1) \cdot \frac{1}{\beta_k^{\eta_c + 2}}, \ if \ \beta_k > 1, \end{cases}; \quad (6)$$

$$\beta_k = \begin{cases} (2 \cdot r_k)^{\frac{1}{\eta_c + 1}}, \ if \ r_k \le 0.5, \\ \left(\frac{1}{2 \cdot (1 - r_k)} \right)^{\frac{1}{\eta_c + 1}}, \ if \ r_k > 0.5, \end{cases}; \quad (7)$$

r_k is the uniformly sampled random number in [0, 1]; η_c is the crossover distribution index.

The crossover distribution determines how well spread the children will be from their parents.

5.2. Application of the polynomial mutation [14]. The child is created according to formula:

$$c_k = p_k + (p_k^u - p_k^l) \cdot \delta_k, \quad (8)$$

where c_k is the child; p_k is the parent; p_k^u being the upper bound on the parent component; p_k^l is the lower bound on the parent component; δ_k is small variation which is calculated from a polynomial distribution by using formula

$$\delta_k = \begin{cases} (2 \cdot r_k)^{\frac{1}{\eta_m + 1}} - 1, \ if \ r_k < 0.5, \\ 1 - [2 \cdot (1 - r_k)]^{\frac{1}{\eta_m + 1}}, \ if \ r_k \ge 0.5, \end{cases}, \quad (9)$$

where r_k is the uniformly sampled random number in [0, 1]; η_m is the mutation distribution index.

Step 6. Recombination and selection. The offspring population is combined with the current generation population and selection is performed to set the individuals of the next generation. Elitism is ensured, since all the previous and current best individuals are added in the population. Population is sorted based on non-domination. The new generation is filled by each front subsequently until the population size exceeds the current population size. If by adding all the individuals in the u -th front the population size exceeds P then individuals in the u -th front are selected based on their crowding distance in the descending order until the population size is P .

The steps 3 – 6 should be repeated to generate the subsequent generations.

In the case of the SVM classifier development all chromosomes are coded by the values of the classifier model parameters: the value of the regularization parameter and the value of the Gaussian kernel function parameter.

3. NOVEL MODIFICATIONS OF THE NSGA-II

It is offered to modify the NSGA-II by the additional algorithm which performs the autotuning of such parameters of the NSGA-II as the probability and distribution indexes for crossover and mutation. These parameters are responsible for the value of the chil-

drens' chromosome deviation from the parents' ones. This autotuning algorithm means the search of the most effective values of the probability and distribution indexes during the algorithm's work.

Let the number of the analyzed values of the crossover probability p_c is m and the number of the analyzed values of the distribution indexes η is n ($\eta = \eta_c = \eta_m$). Then the number of the sets of these values is calculated as $\Psi = n \cdot m$. The choice between crossover and mutation is carried out according to the rule: if $r < p_c$, where r the uniformly sampled random number in [0, 1], then the crossover is applied, else the mutation is applied [12]. Within the suggested autotuning algorithm at the Step 4 of the NSGA-II all parents of the first generation are divided into Ψ equal groups. Each group represents a certain set of values of the crossover probability and distribution indexes for the crossover and mutation. In view of the division into groups, it's worthwhile to set the population size divisible by $2 \cdot \Psi$. As the result of the genetic operators' work at the Step 5 of the NSGA-II the children population is created, in which every child has the same group as its' parent. At the end of the Step 6 of the NSGA-II the individuals can be divided into 3 types:

– the individuals which are considered as the parents and belong to one of the Ψ groups;

– the individuals which are considered as the children and belong to one of the Ψ groups;

– the individuals which didn't participate in the crossing and thus don't belong to any of the groups.

It is suggested to estimate the success of each group (and its' set of the parameters' values) according to the number of children of this group in the result population.

The group that has the minimum ratio of the number of children to the size of group is considered as "unsuccessful" and the size of such group is reduced by one. The size of group that has the maximum ratio of the number of children to the size of groups is increased by one. Such group with the increased size is considered as "successful". If there are several "unsuccessful" groups than only one of them is selected randomly for the purpose of decreasing its' size. Similarly, if there are several "successful" groups, than only one of them is selected randomly for the purpose of increasing its' size. The use of the ratio of the number of children to the size of group in the result population as the indicator of success or unsuccess of group is caused by the fact that the increase of this ratio shows the effectiveness of the genetic operators with certain values of parameters.

The suggested rule of the group's size changing allows to avoid early rapid and random removal of some groups from the autotuning process.

If the group doesn't have any child in the result population or its' size reaches the minimum size, which is equal to 1, then the group and its' parameters set will be removed from the autotuning process. The minimum size of group is selected according to the minimum number of parents for the crossover operator that equals 2. If some "unsuccessful" group is removed, the size of the "successful" group is increased such way that total size of groups is equal to the population size P.

Deb K. and Goel T. suggest using the hybrid algorithm to approximate the found Pareto fronts obtained as a result of the NSGA-II's work, to true [11].

This algorithm can be described by the following step.

Step 1. To use the local search for the weighted sum of the optimized classification quality indicators in the district of the found decision.

Step 2. To make the substitution in the Pareto-front of decisions and conduct the check on the non-dominance, if the best decision is obtained.

Step 3. To generate the clustering of the values of the classification quality indicators of the Pareto-front to minimize its size.

During the hybrid algorithm's work the classification quality indicators are used as the objective functions $f_j(x)$, where $j=\overline{1,M}$; M is the objective functions' number; decision x is the cortege (C,σ). Firstly, for each decision x in the optimized set of the classification quality indicators the weights values of the criterion functions for formation of the weighed sum are calculated:

$$w_j^x = \frac{(f_j^{\max} - f_j(x))/(f_j^{\max} - f_j^{\min})}{\sum_{k=1}^{M}\left(f_k^{\max} - f_k(x)\right)/(f_k^{\max} - f_k^{\min})}, \quad (10)$$

where $f_j^{\max}$ is the maximum value of $f_j(x)$; $f_j^{\min}$ is the minimum value of $f_j(x)$; $j=\overline{1,M}$.

Then, new criterion function for transfer to the local search algorithm is formed:

$$F(x) = \sum_{j=1}^{M} w_j^x \cdot f_j(x). \quad (11)$$

In this research the pattern search [15] algorithm is used as the local search algorithm. This algorithm is realized by the built-in function patternsearch.m of the Matlab programming language.

The nondominability test is similar to the same test in the NSGA-II and allows defining the solutions which are belong to the first front (values of which are not dominant) and to eliminate ones which are belong to other fronts.

The third step of the hybrid algorithm allows minimizing the decisions number in the Pareto-front by means of the clustering algorithm. The main objective of the Pareto-front's clustering is the reduction of the decisions number in the Pareto-front, because from the practical point of view researchers are interested in some concrete decisions while the population size (and the Pareto-front's size) can reach over 50 decisions. In [13] the agglomerative clustering approach for reducing the size of the obtained nondominated set of solutions is suggested.

In this research the fuzzy c-mean algorithm (FCM algorithm) [16] is used as the clustering algorithm. The choice of this algorithm is caused by the fact that it allows object to belongs to several clusters simultaneously unlike from other clustering algorithms. As a result, it is possible to receive more exact clustering results. As the optimum number of clusters is unknown in advance, it is necessary to apply repeatedly the FCM algorithm at the different number of clusters, and also at the different initial splittings into clusters to choose the optimum number of clusters, to which the minimum value of the objective function of the algorithm corresponds. The required splitting into clusters corresponds to the optimum number of clusters.

4. EXPERIMENTAL RESULTS

The program implementing the offered modifications of the NSGA-II as the input data accepts: the dataset which is used for the SVM classifier development, the set of the optimized classification quality indicators, the parameters values of the NSGA-II. The structure containing the Pareto fronts of decisions for each set of the optimized classification quality indicators is the result of the program work.

The operability analysis of the developed program has been made on the real and artificially created datasets. In particular, the following datasets have been used: Heart Diseases Data (Heart; https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/heart/); Breast Cancer Data (WDBC; https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/); German credit data (German; https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/ german/), the artificially created data MOTP12 (MOTP12; http://machinelearning.ru/wiki/ images/b/b2/MOTP12_ svm_example.rar).

The analysis consists in comparison of the results received during the work of the standard NSGA-II, and the results of the NSGA-II with the offered modifications. To each of the mentioned-above dataset the following algorithms were applied: the algorithm that implements the basic NSGA-II ("NSGA-II" in Table 1), and the algorithm that implements the NSGA-II with the suggested modifications ("NSGA-II+Autotune+Hybrid" in Table 1). For the "NSGA-II" the following values genetic operators' parameters were used: the distribution indexes for crossover and mutation were equal to $\eta = \eta_c = \eta_m = 20$, the crossover probability was equal to $p_c = 0.9$. For the "NSGA-II+Autotune+Hybrid" the distribution indexes η for crossover and mutation belonged to the set $\{10, 20, 30\}$, the crossover probability p_c belonged to the set $\{0.9, 0.8, 0.7\}$. The parameters for both cases have the following values: the population size is equals to 180, the maximum number of generations is equals to 200, the search spaces of the parameters' values for the regularization parameter and the kernel function parameter are defined as $[10^{-5}, 200]$, the kernel function is the radial basis function.

The experimental results are shown in Table 1 in the form of the individual's values of errors on the training and test sets of the optimized set of the classification quality indicators which is the best for these indicators. The chosen individual has the minimum number of errors on the training and test sets in the whole front. The best fronts for both variants (with modifications and without modifications) have been chosen as follows: the best set of the optimized classification quality indicators on the basis of the number of errors on the training and test set have been defined (for the same dataset the sets of the optimized classification quality indicators can differ). Besides, the best set of the classification quality indicators is characterized by the "Search time" of the Pareto front. Also, the corresponding number of the best generation is defined. The exceeding of the maximum number of stall generations for which the objective functions' values (the classification quality indicators' values) don't change is taken as the stop criteria. In this research the maximum number of stall generations is set to 5.

Table 1. The experimental results

Dataset	Number of objects		Algorithm type	Parameters' values		The best set of the classification quality indicators for the number of errors and the support vectors				Search time, sec	Number of the best generation
	Class "+1"	Class "–1"		η	p_c	Optimized set	Train errors	Test errors	SV		
Heart	150	120	NSGA-II	20	0.9	Ac+F	0/230	6/40	186/ 230	94	14
			NSGA-II +Autotun +Hybrid	30	0.7	Ac+P	0/230	6/40	186/ 230	109	15
WDBC	212	357	NSGA-II	20	0.9	Ac+F	0/485	4/84	56/ 485	101	14
			NSGA-II +Autotun +Hybrid	10	0.9	Ac+Se	0/485	2/84	72/ 485	112	17
German	195	205	NSGA-II	20	0.9	Ac+P	0/850	41/150	698/ 850	1769	90
			NSGA-II +Autotun +Hybrid	30	0.8	Ac+P	0/850	38/150	577/ 850	1331	80
MOTP12	700	300	NSGA-II	20	0.9	Ac+Se	0/341	7/59	310/ 341	190	25
			NSGA-II +Autotun +Hybrid	30	0.8	Ac+Se	0/341	4/59	228/ 341	130	20

Usually, the suggested autotuning algorithm increases the time expenditures on the solution search. But it selects the most fitting set of the parameters' values at once. Therefore, this algorithm allows to refuse the multiple launch of the basic NSGA-II in order to select the best set of the parameters' values. In general, the autotuning algorithm requires one execution of the NSGA-II instead of Ψ executions of the basic NSGA-II. For example, for the WDBC dataset the application of the basic NSGA-II will take approximately 900 seconds (for 9 runs of the basic NSGA-II algorithm) to analyze 9 sets of the parameters' values, while the autotuning algorithm with the NSGA-II takes 112 seconds for the search of the required parameters' values.

The Table 1 shows that in most cases the search time of the best set of the classification quality indicators on the basis of the "NSGA-II+Autotune+Hybryd" is slightly greater than the search time on the basis of the "NSGA-II". Also, the Table 1 shows that the solutions of the modified NSGA-II algorithm are better than the solutions of the basic NSGA-II: the number of the training and test errors is less; also, the number of the support vectors (SV) is less.

The hybrid modification on the basis of the FCM algorithm allows to reduce the number of solutions to 3 – 6 and simplify the choice of the best solution.

For the best sets of the classification quality indicators (Table 1) the following values of the distribution indexes η have been recognized as the effective: 30 for the Heart, German and MOTP12 datasets, 10 for the WDBC dataset. Also, the following values of the crossover probability p_c have been recognized as the effective: 0.7 for the Heart dataset, 0.8 for the WDBC dataset, 0.9 for both German and MOTP12 datasets.

The use of the offered modifications didn't affect the results of the NSGA-II implementation for the Heart dataset, that is possible for the following reasons: firstly, it is impossible to use the hybrid because of the diversity lack in the values of the objective functions in the population, secondly, the used splitting into the training and test sets doesn't involve the situation where the potential support vectors enter into the test set. The use of the offered modifications for the WDBC dataset has reduced the number of errors on the training and test sets, but, at the same time, the number of the support vectors has increased that demonstrates the deterioration in the generalizing ability of the SVM classifier or shows the possible retraining. The use of the offered modifications for the German dataset has reduced the number of errors on the test set and the number of the support vectors. The use of the offered modifications for the MOTP12 dataset has reduced the number of errors on the training and test sets. Also, the number of the support vectors has decreased.

5. CONCLUSIONS

The efficiency of the suggested approaches has been confirmed by the results of the experimental studies. The offered modifications can improve the results of the work of the basic NSGA-II in the context of solving the problem of the search of the optimal parameters values of the SVM classifier on the sets of the classification quality indicators which should be optimized. In particular, the parameters' autotuning of the algorithm will save the user from the need of the repeated start of the program at the different parameters values. The hybrid, in its' turn, will allow to bring closer the decision to true by the local search method and, also, to reduce the front size, that can simplify the choice of the best solution for the user. Further researches will be devoted to the development of recommendations on application of the SVM classifiers based on the suggested modifications of the NSGA-II for the solution of the practical problems.

REFERENCES

[1] Vapnik V., *Statistical Learning Theory,* New York: John Wiley&Sons, 1998, 732 p.

[2] Gunn S.R., *Technical report on "Support Vector Machines for Classification and Regression"*, University of Southhampton.

[3] Demidova L., Nikulchev E., Sokolova Yu., The SVM Classifier Based on the Modified Particle Swarm Optimization, *International Journal of Advanced Computer Science and Applications (IJACSA)*, 2016, Vol. 7, No. 2, pp. 16–24.

[4] Demidova L., Sokolova Yu., Development of the SVM Classifier Ensemble for the Classification Accuracy Increase, *6th Seminar on Industrial Control Systems: Analysis, Modeling and Computation (ITM Web of Conferences)*, 2016, Vol. 6.

[5] Demidova L., Klyueva I., Sokolova Yu., Stepanov N., Tyart N., Intellectual Approaches to Improvement of the Classification Decisions Quality on the Base of the SVM Classifier, *Procedia Computer Science*, 2017, Vol. 103, pp. 222–230.

[6] Sopov E., Semenkin E., Automated Synthesis of Selection Operators in Genetic Algorithms using Genetic Programming, *International Journal on Information Technologies and Security*, No. 4 (vol. 9), 2017, pp. 13–24.

[7] Astakhova N.N., Demidova L.A., Kuzovnikov A.V., Tishkin R.V., Multiobjective clonal selection algorithm for solving forecasting problems, *IOP Conference Series: Materials Science and Engineering*, 2017, 173.

[8] Deb K., Jain H., An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints, *IEEE Transactions on Evolutionary Computation*, 2014, Vol. 18, pp. 577 – 601.

[9] Deb K., Pratap A., Agarwal S., Meyarivan T., A Fast Elitist Multiobjective Genetic Algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 2, 2002, pp. 182 – 197.

[10] Deb K., Goel T., A Hybrid Multi-Objective Evolutionary Approach to Engineering Shape Design, *Lecture Notes in Computer Science*, 2001, Vol. 1993, pp. 385-399.

[11] Brester C., Ryzhikov I., Semenkin E., Restart Operator for Multi-Objective Genetic Algorithms: Implementation, Choice of Control Parameters and Ways of Improvement, *International Journal on Information Technologies and Security*, 4 (vol. 9), 2017, pp. 25–36.

[12] Seshardi A., Multi-objective optimization using evolutionary algorithm. http://read.pudn.com/downloads71/sourcecode/speech/254461/254461/MOEA-NSGA-II/NSGA_2.pdf

[13] Beyer H.-G., Deb K., On Self-Adaptive Features in Real-Parameter Evolutionary Algorithm, *IEEE Transactions on Evolutionary Computation*, 5, 2001, no. 3, pp. 250 –270.

[14] Deb K., Agarwal R.B., Simulated Binary Crossover for Continuous Search Space, *Complex Systems*, 9, 1995, pp. 115 – 148.

[15] Hooke R., Jeeves T.A., "Direct search" solution of numerical and statistical problems, *Journal of the Association for Computing Machinery (ACM)*, 1961, 8 (2), pp. 212–229.

[16] James C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981, 272 p.

Information about the authors:

Liliya Demidova – Dr. Sc. (technical sciences), full professor, full professor of Ryazan State Radio Engineering University and Moscow Technological Institute. Areas of scientific research are data mining, machine learning, evolutionary computation, remote sensing. Phone: +79106412403. demidova.liliya@gmail.com

Maxim Egin – Student (bachelor) of Ryazan State Radio Engineering University. Areas of scientific research are data mining, machine learning, evolutionary computation. Phone: +79106304425. eginmm@gmail.com

Roman Tishkin – Ph. D. (technical sciences), associate professor, associate professor of Ryazan State Radio Engineering University. Areas of scientific research are data mining, machine learning, remote sensing. Phone: +79065454925. roman.tishkin@mail.ru

Manuscript received on 28 January 2018