**Piotr BILSKI**

Warsaw University of Life Sciences, Warsaw University of Technology

# Automated selection of kernel parameters in diagnostics of analog systems

***Abstract***. *The paper presents an approach to automatically select the optimal parameters of the kernel functions used in the Support Vector Machines (SVM) approach for the diagnostic task. Various variants of the simulated annealing were implemented and verified in order to obtain the best diagnostic outcomes. The tested system was the fourth order lowpass filter, consisting of two Sallen-Key sections and nine diagnosable elements. The tests covered verification of simulated annealing parameters (starting temperature and annealing ratio) and various SVM kernels (with coding schemes) in the multiple faults detection and location task. The proposed method verified against the exhaustive search.*

***Streszczenie.*** *Artykuł przedstawia metodę automatycznego doboru optymalnych parametrów funkcji jądra wykorzystywanych przez maszyny wektorów podpierających w diagnostyce systemów analogowych. Różne warianty symulowanego wyżarzania zostały zaimplementowane w celu uzyskania jak najlepszych wyników diagnostycznych. Metoda została przetestowana na modelu filtru dolnoprzepustowego czwartego rzędu składającego się z dwóch sekcji Sallen-Key oraz dziewięciu elementów mogących być przyczyną uszkodzeń. Eksperymenty obejmowały dobór parametrów symulowanego wyżarzania (temperatura początkowa oraz szybkość schładzania) oraz jąder wektorów podpierających w detekcji i lokalizacji uszkodzeń. Opisana metoda została porównana z przeszukiwaniem wyczerpującym. (**Automatyczny dobór optymalnych parametrów funkcji jądra w diagnostyce systemów analogowych**)*

**Keywords:** diagnostics of analog systems, optimization, simulated annealing, support vector machines.
**Słowa kluczowe:** diagnostyka systemów analogowych, optymalizacja, symulowane wyżarzanie, maszyny wektorów podpierających.

## Introduction

Modern approaches in diagnostics of analog systems resort to the artificial intelligence (AI) methods that have to consider uncertainty conditions. Most of systems under test (SUT) are subject to noises of various nature, which make their monitoring and testing difficult. Also, accuracy of the measurement process is limited. The diagnostic task usually consists in determining the state of the SUT and, if needed, identifying values of elements responsible for its incorrect behaviour. Application of AI methods to this task requires knowledge about the SUT's work regime. The latter is usually obtained from training and testing data sets, which are created after simulating the SUT model.

Increasing computational power of the equipment used in laboratories and industrial conditions allows for introducing sophisticated methods and algorithms aimed at fault detection and classification. Currently three approaches are the most promising: fuzzy logic, rough sets and support vector machines. All are capable of working with systems in the uncertainty conditions, although there are significant differences between them. Fuzzy logic is a rule-based module, which does not have the ability to extract knowledge – it must be delivered by the expert or a machine learning method [1]. Rough sets are also rule-based and have the ability to find dependencies in data. Here, decision about the state of the system is made using the voting process [2]. Support Vector Machines (SVM) are kernel based approach, aiming at minimizing the classification error when the separation of different faults is impossible [3]. Here both data extraction and its usage for decision making are present and the main problem is the proper selection of the kernel and its parameters. The increasing popularity of SVM and their numerous applications in engineering tasks justify further research on their usefulness. As SVM are not rule-based, their structure and work regime are difficult to follow by the human observer. Currently attempts to combine presented approaches are made with promising results [4].

The paper presents the optimization procedure for the SVM parameters in the analog systems diagnostics. The manual selection of the kernel and its parameters is a difficult and time-consuming challenge. Therefore the automated selection and verification process can significantly facilitate the algorithm design and adjustment.

The structure of the paper is as follows. Firstly, aims of the AI-based diagnostics and its paradigms are presented. Then, the SVM approach, which was utilised in the presented research, is briefly introduced. As the SVM is in the general form prepared to distinguish between only two categories and presented research considers more faults, extending the method to cover multiple classes is also described here. Further, the approach for the automated selection of the kernel parameters (i.e. simulated annealing) is presented. It is confronted against the full search method that ensures finding optimal parameters. After the implemented methods are described, the tested object, i.e. the fifth order analog filter is shown. The experiments and verification of the proposed methods are then performed. Finally, conclusions and future prospects are presented.

## Diagnostic principles

The diagnostic procedure consists of three stages, which are performed in the on-line monitoring of working SUT. These are: detection (discovery of the incorrect SUT behaviour), identification (determining the element or elements responsible for the fault) and location (determining the value of the faulty elements). These stages facilitate repairs of the SUT (especially when it is expensive and cannot be replaced with another object – the case of electrical motor), or minimizing its incorrect behaviour [5].

To fulfil these requirements the SUT is excited (at the accessible nodes) with the selected signal and its responses at accessible or partially accessible nodes are measured. The main issue here is the type of the signal (for instance, sinusoid or *Sa(t)*) and the domain in which the SUT is analyzed (time, frequency, mixed, etc.). Also, extraction of characteristic traits (symptoms or stamps) from the responses is important. If designed correctly, this approach allows making decisions about the SUT state.

Introduction of AI methods requires preparing training and testing data sets. The former are used to extract the knowledge about the SUT behaviour in nominal (fault-free) and faulty states. The latter are used to verify diagnostic efficiency of the proposed approach. Both sets are generated during the simulation of the SUT's model. The procedure consists in inserting the fault into the model (setting the value of the selected element) and recording stamps from responses. This way the designer can decide

about the type of the faults (parametric or catastrophic) and range of the element values. In practical applications the approach is verified on the real objects, therefore the accuracy of the SUT model is the important problem. The typical form of the data set is the table, containing information extracted from the SUT's model. Every experiment (single simulation of the model, one of *n*) is stored in a row with *m* stamps. The row is supplemented with the information regarding the source of the fault and deviation of the responsible element's value from the nominal one. In the presented research it has the form of a fault code, consisting of the identifier of the faulty element ("0" for the fault-free state) and its deviation from the nominal value (which can be either positive or negative). The degree of deviation is represented by the natural number (1,2,3, etc.). For instance, "-21" indicates that the second element is the fault source, having the value smaller than the nominal one ("-1"). Similarly, "42" means the fourth element has value much greater that the nominal one. Changing the number of degrees of deviation from the nominal value determines the number of the fault codes in the data set. In this research for each element four codes were defined. For example, the second element can cause four fault states: "very small" (-22), "too small" (-21), "too large" (21), "very large" (22). The designer can increase or decrease the number of fault codes, according to his needs and expected resolution of the diagnostic module.

Verification of the method is done using the testing data set of the same form as the training data set. The classification performed by the method is compared to the fault codes stored in the last column of the set. In the presented research the following evaluation function was used:

$$f(T) = \frac{|T| - |T_-| - 0{,}5 \cdot |T_+|}{|T|} \tag{1}$$

where $|T|$ is the number of experiments in the testing data set, $|T_+|$ is the "soft" error of the diagnostic method (the faulty element was identified correctly, but the deviation from the nominal value was determined incorrectly, such as "-11" instead of "12") and $|T_-|$ is the "hard" error of the diagnostic method (the source of the fault was not determined correctly, such as "11" instead of "21"). The measure (1) is independent of the size of the data set and has values between 0 (none of the faults were detected correctly) and 1 (all faults were detected and located correctly). It was assumed that "hard" error is two times more important than the "soft" error (coefficient 0.5 in (1)).

**Application of Support Vector Machines in diagnostics**

The method selected for the diagnostic task is relatively new, but its usefulness was proven in multiple domains (such as electronics [7] and biomedicine). SVM are classifiers working in the high-dimensional feature (here stamps) space, which are trained using the quadratic programming [9]. Their main advantage is considering uncertainty and random errors in data. While traditional classifiers are aimed at dividing space of separable categories, SVM work with objects that are not linearly separable. The method is a maximal margin classifier as it aims to minimize the error of the categories separation. The main operation is transformation of the original feature space, in which the objects reside, into the new one, where they can be separated. It is possible because of the kernel functions. When the separation is not possible, the border between the categories is defined to minimize the classification error.

In the most realistic, nonlinear case, SVM are able to distinguish between two categories ("-1" and "+1") [8]:

$$\operatorname{sgn}(f(x)) \geq 1 \tag{2}$$

where the decision function is given by the equation:

$$f(x) = \sum_{i}^{m} \alpha_i \cdot K(x_i, x) + b \tag{3}$$

Here, $\alpha_i$ are variables related to weights of the hyperplane separating fault categories, $x$ is the vector of stamps (representing the single experiment), $K(x_i,x)$ is the kernel function and $m$ is the number of stamps. The learning of SVM consists in finding parameters $\alpha$ and $b$, ensuring the maximal separation of experiments, i.e. minimizing $||w||^2$ with respect to the condition (2).

There are multiple kernel functions, such as linear, radial basis function (RBF), or polynomial [9]. Most of them are parameterized. The proper parameter selection depends on the particular case and usually requires experienced designer and time-consuming trials. Although there are attempts to automatically select the kernel parameters, there is the need to search for the fastest and most accurate method. The presented research is focused on the RBF only because of its greatest efficiency, but for other kernel functions the procedure is the same.

The task of the diagnostic module goes beyond the binary separation (there are many fault sources), so multiple classifiers described by (1) are required. Their responses are combined to create a sequence of outputs identifying the fault. The required number of classifiers depends on the applied coding scheme. There are various schemes available (Error Correcting Output Coding, One vs All, One vs One, etc. [10]). The Minimum Output Coding was applied in this research. It ensures the minimum number of the classifiers required to identify all faults:

$$n_{SVM} = \lceil \log_2 n_c \rceil \tag{4}$$

Every classifier is responsible for the particular position in the binary code word. For instance, four faults require two classifiers, as presented in Table 1.

Table 1. Exemplary coding scheme for four fault codes

| Fault code | Classifier 1 | Classifier 2 |
|---|---|---|
| 0 | -1 | 1 |
| 11 | -1 | -1 |
| -11 | 1 | -1 |
| 21 | 1 | 1 |

The proposed method is aimed at detecting multiple faults (i.e. faults occurring at the same time), so for every faulty element a group of classifiers was created. Every group was aimed at detecting faults related only to the particular element (for example, "-12", "-11", "11" and "12"). All other codes (including the fault-free state) are treated as one category gathering remaining states of the SUT. This way the SVM classifiers form independent diagnostic modules, each distinguishing faults for one element. Each group consists of three classifiers (the minimal number required to cover five codes). Additional group, consisting of one classifier, is responsible for the fault-free state. Note that it is important to maximize efficiency of every classifier within a group as the fault code is detected correctly only if all classifiers indicate the correct binary category. Therefore the diagnostic efficiency of each group is usually smaller than partial efficiencies of classifiers inside the group (see results section). The described algorithms were implemented using the Matlab SVM toolbox [11].

**Automated selection of kernel parameters**

The procedure of designing the SVM classifier for the diagnostics of SUT is in Fig. 1. The diagnostic module is trained using the training set *R* with the selected *kernel* and its parameters *p*. The aim are optimal values of parameters (*optimal_p*), obtained for every group of classifiers, responsible for detecting faults of the particular element. Searching for the optimal parameters is then the optimization task, where the aim is to maximize the evaluation function – measure (1) by exploring solutions - kernel parameters. The procedure *select_param(p,N)* and the algorithm stopping criterion depend on the optimization method. The parameter *N* is the neighbourhood of the current solution. The variable *result* is calculated as in (1).

```
procedure design_SVM
input: R, T, kernel
output: optimal_p

for element=1 to N+1
    optimal_result=0
    optimal_p=0
    while not stopping_criterion
        p=select_param(p,N)
        weights := train_svm(R,kernel,p)
        result=test_svm(T,kernel,p, weights)
        if (result>optimal_result)
                optimal_p(element)=p
        end
    end
next element
```

Fig.1. Algorithm of automatic selection of optimal kernel parameters

The applied RBF kernel (5), has only one parameter, $\gamma$ (width of the RBF [11]), changed during the optimization.

$$K(x_i, x) = \exp(-\gamma \cdot \|x - x_i\|^2) \tag{5}$$

The most straightforward optimization method is the full search approach. It ensures finding the global optimum, as all solutions are checked. Unfortunately, verifying all of them usually takes a lot of time. Also, the RBF parameter is a real number, therefore there is the infinite number of possible solutions, making the true full search impossible. In such a case the value range is divided into intervals of equal length. The evaluation function is then calculated only in the middle (dashed arrow) or on the border (solid arrow) of each interval (Fig. 2a) [12].

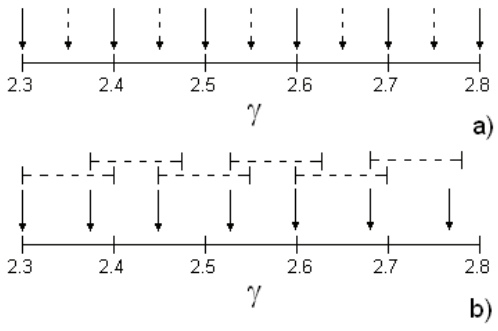


Fig.2. Full search method with the constant (a) and random step (b)

In the research the modified version of the full search was applied. Instead of changing the parameter's value with the constant step, the new value is randomly selected with the uniform distribution from the neighbourhood *N* of the current value (6). In the example from Fig. 2 *N=0.1*. This way the randomized full search was obtained. Its advantage is the ability to find the optimal parameter value, even if it is not among the values found with the constant step. The disadvantage is larger number of iterations (as the new values are generated with smaller step than in the traditional approach).

$$\gamma_{new} = \gamma_{old} + rnd(n) \tag{6}$$

For all classifiers the algorithm was run to obtain the optimal parameter values (three for each group). The exemplary search space is in Fig. 3. The parameters' range usually consists of flat intervals, where the evaluation function (1) has constant value and narrow areas of sharp changes of the function. For some classifiers the function is constant in the whole value range.

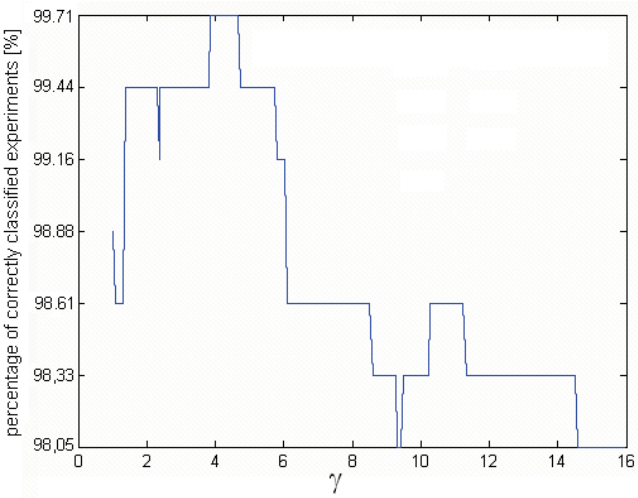


Fig.3. Exemplary parameter value range for the full search method

Because the full search approach is time-consuming, there is the need to look for faster alternatives that give similar diagnostic quality. There are multiple heuristic methods that allow finding sub-optimal solutions in a reasonable time. One of the most efficient is simulated annealing. Originating from metallurgy, it ensures high accuracy as is able to escape from local optima [12]. In each iteration the previous and the current solutions are compared and the worse one has the chance to be selected for the next iteration with probability proportional to the temperature (*Temp*), which is relatively high at the beginning and gets reduced in the process.

The algorithm of simulated annealing in training the SVM classifiers is in Fig. 4. The method exists in the uniform and non-uniform version. The difference between them is the number of iterations performed at the constant temperature $n_{it}$. The non-uniform version has $n_{it}=1$, while in the uniform version it varies.

The assumption is that the simulated annealing can produce parameters of the classifiers ensuring high diagnostic efficiency much faster than the full search. To correctly apply the algorithm from Fig. 4, the following characteristics were determined:

- optimal starting and stopping temperature, which determines the number of iterations of the algorithm and speed of decreasing the algorithm randomness,
- annealing scheme, which determines the speed and form of the temperature decrease,
- number of iterations in the constant temperature for the uniform version.

The annealing scheme is a factor determining how fast the temperature changes from $T_{start}$ to $T_{stop}$. Three schemes were selected for tests [9]: geometric (7), linear (8) and logarithmic (9).

(7) $$Temp = T_{start} \cdot 1/k$$

(8) $$Temp = T_{start} \cdot 1/\log(k)$$

(9) $$Temp = T_{start} \cdot 0.5^{k}$$

where $k$ is the number of iteration.

```
procedure simulated_annealing
input: T_start, T_stop, R, T, kernel
output: optimal_p

for element = 1 to N+1
    optimal_result := 0
    optimal_p := 0
    result := 0
    Temp := T_start
    while Temp > T_stop
        for k = 1 to n_it
            weights := train_svm(R,kernel,p)
            new_result := test_svm(T,kernel,p,weights)
            if (result>optimal_result)
                optimal_p := p
            end
            if (new_result>result)
                result := new_result
            elseif rnd < exp-(|new_result-result|/Temp)
                result := new_result
            end
        next k
    end
next element
```

Fig.4. Simulated annealing algorithm in the SVM training procedure

**System Under Test**

The SUT used for the verification of the approach is the fifth order bandpass filter. It is the typical analog circuit, currently produced as the integrated circuit. All its nodes were assumed partially accessible, while the excitation signal was inserted at node 1. All elements (resistors and capacitors) were considered potential fault sources. Their nominal values were: $R_1$=$R_2$ $R_3$=$R_4$=$R_5$=1 kΩ, $C_1$=16nF, $C_2$=19nF, $C_3$=13nF, $C_4$=51nF, $C_5$=51nF. The band of the nominal filter was between 100Hz and 10kHz. The model of the filter was simulated in the SIMULINK environment.

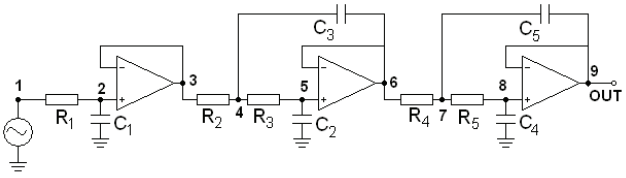


Fig.5. Scheme of the 5th order bandpass circuit

The SUT was simulated to obtain information about parametric faults ranging from 10% to 190% of the nominal value. The excitation signal was a sinusoid with 9kHz frequency (i.e. close to the border frequency of the filter) and 1V of amplitude. Resolution of the diagnostic module was set to four fault states (as explained in Diagnostic principles section). Two data sets were obtained, each with 180 experiments (rows) and 54 columns (stamps). The overall number of fault codes was 45.

**Experiments**

The full search method was first used to determine the optimal obtainable widths of RBFs. Optimization was executed for 31 classifiers in 11 groups – ten groups (of three classifiers) for each faulty element, the final one classifier responsible for the fault-free state. The range of the kernel parameter was between 0 and 16. The values ensuring optimal diagnostic efficiency are in Table 2. The optimal evaluation function values are in Table 3. Numbers in the latter are relative classification outcomes (in percent). The column "Ovr Clas" is for the classification outcome for the whole group. Values "1" in Table 2 mean that the parameter is constant in the whole range.

Table 2. Optimal kernel parameters for the full search method

| Element | Class.1 | Class. 2 | Class. 3 |
|---|---|---|---|
| R1 | 3.86 | 1 | 5.77 |
| R2 | 0.50 | 1.13 | 1.21 |
| R3 | 1 | 1 | 1 |
| R4 | 1 | 1 | 1.02 |
| R5 | 1 | 1 | 1 |
| C1 | 3.37 | 1 | 1 |
| C2 | 1 | 2.409 | 1 |
| C3 | 1 | 1 | 1.301 |
| C4 | 1 | 1.15 | 1 |
| C5 | 1 | 2.103 | 2.124 |
| Nominal | 3.036 | -- | -- |

Table 3. Optimal values of the evaluation function for the full search method

| Element | Class.1 | Class. 2 | Class. 3 | Ovr Clas |
|---|---|---|---|---|
| R1 | 99.72 | 98.89 | 97.22 | 96.67 |
| R2 | 99.17 | 99.44 | 98.89 | 98.33 |
| R3 | 99.44 | 98.89 | 98.61 | 98.33 |
| R4 | 99.44 | 99.17 | 98.89 | 98.61 |
| R5 | 99.17 | 99.44 | 98.89 | 98.61 |
| C1 | 97.78 | 98.06 | 97.22 | 96.11 |
| C2 | 100.00 | 99.17 | 97.22 | 96.94 |
| C3 | 100.00 | 98.61 | 98.06 | 97.22 |
| C4 | 99.44 | 99.44 | 98.61 | 97.78 |
| C5 | 99.44 | 100.00 | 98.89 | 98.33 |
| Nominal | 99.17 | -- | -- | 99.17 |

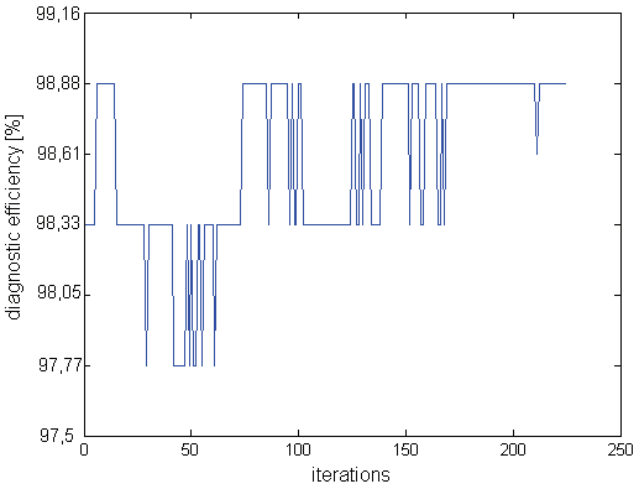


Fig.6. Exemplary result of the simulated annealing procedure

Results in Table 2 and Table 3 were treated as the reference outcomes for the simulated annealing. All experiments were performed on the computer with Core2Duo processor (1,66 GHz) and 2,5 GB of RAM.

The simulated annealing was performed for multiple configurations of the algorithm settings. The exemplary result of the optimization process is in Fig. 6. In most trials the first stage of the algorithm has multiple changes in the evaluation function value. Further the changes are rare and are located around the optimal or suboptimal value.

Results obtained with simulated annealing are similar to the ones from the full search method. The uniform version

of the algorithm works slower but gives usually better outcomes, therefore for the correctly selected parameters it is a better choice.

Table 4. Optimal values of the evaluation function for the simulated annealing method

| Element | Class.1 | Class. 2 | Class. 3 | Ovr Clas |
|---|---|---|---|---|
| R1 | 99.44 | 98.89 | 97.22 | 96.11 |
| R2 | 99.44 | 99.44 | 98.89 | 98.33 |
| R3 | 99.44 | 98.61 | 98.61 | 98.06 |
| R4 | 99.44 | 98.89 | 99.44 | 98.33 |
| R5 | 99.44 | 99.44 | 98.89 | 98.33 |
| C1 | 97.78 | 97.78 | 96.11 | 92.78 |
| C2 | 98.89 | 99.17 | 97.22 | 96.11 |
| C3 | 100.00 | 98.61 | 98.89 | 97.78 |
| C4 | 99.44 | 97.78 | 96.67 | 95.56 |
| C5 | 98.89 | 99.72 | 98.89 | 98.33 |
| Nominal | 98.61 | -- | -- | 98.61 |

Selection of temperatures determines the number of iterations of the algorithm and its stochastic behaviour. The lower the temperature, the smaller are fluctuations of the kernel parameter in the terminal stage of the operation. If $T_{start}$ is too large, the number of random changes in the kernel parameter value is significant. On the other hand, when $T_{stop}$ is too small, the algorithm works long without significant changes during the last iterations. The preferred terminal temperatures for $T_{start}$=10 and non-uniform version were:

- geometric scheme: $T_{stop}=0.02$, allowing for 8 iterations,
- linear scheme: $T_{stop}=0.25$, allowing for 39 iterations,
- logarithmic scheme: $T_{stop}=4.5$, allowing for 166 iterations.

The logarithmic annealing scheme is the least attractive as it causes very slow decrease of the probability of worse solution selection. Therefore through the whole operating time the latter is random (decreasing the probability below 0.5 takes over 400 iterations). The quickest is geometric scheme. It allows to quickly move from the probability close to 1 (where the algorithm is the most random) to the probability close to 0 (where the algorithm is deterministic). When the temperature is set to the values giving probabilities below 0.01 the algorithm can be stopped.

Selection of the neighbourhood range, from which the next solution is selected does not give significant change in the diagnostic results. Because of the form of the evaluation function, every neighbourhood above 2 ensures that the algorithm will be able to find the optimal kernel values.

Table 5. Execution times [s] of the kernel parameter optimization for the element R1

| Algorithm | Class.1 | Class. 2 | Class. 3 |
|---|---|---|---|
| Full search | 3460,1 | 3062,5 | 3045,2 |
| ANT10Geom | 297.17 | 332.75 | 303.01 |
| ANT20Geom | 314.32 | 397.9 | 345.8 |
| ANT30Geom | 342.60 | 442.45 | 398.33 |
| ANT10Linear | 1233.0 | 1388.4 | 1373.0 |
| AUT10Geom | 2223.9 | 2390.4 | 2231.8 |

The simulated annealing allowed maintaining high diagnostic efficiency (see Table 4) and saving time, comparing to the full search methods. Table 5 contains compared time measurements (in seconds) for various versions of the algorithm. The first column contains the configuration of the algorithm. The name of the algorithm contains the version of the annealing (U – uniform, N – non-uniform), starting temperature (T=10, 20 or 30) and annealing scheme (geometric or linear). The smaller starting temperature, the smaller is the number of iterations. Also, the non-uniform version of the algorithm is faster than uniform. For every value of temperature, only one iteration is performed here, while in the uniform version it was 25.

**Conclusions**

The proposed approach of automated kernel parameter selection for the SVM classifiers allows obtaining optimal or close to optimal solutions in the time much shorter than the full search method. Although the SVM structure (weights) is optimized for every parameter, its proper selection enables maximizing the diagnostic efficiency. The method is even more useful when the selected kernel has more than one parameter (for example sigmoid).

Optimal kernel parameters were found for the particular case, however some of the algorithm settings can be used for any SUT and kernel. These are starting and stopping temperatures and annealing scheme.

The future investigations should cover automatic selection of parameters for other kernels and applying other optimization methods, which can incorporate evolutionary algorithms, particle swarm optimization, etc. Also, other SUTs should be tested.

REFERENCES


[1] P. Bilski and J. Wojciechowski, "Automated Diagnostics of Analog Systems Using Fuzzy Logic Approach", IEEE Trans. Instr. and Meas., Dec. 2007, Vol. 56, Issue 6, pp. 2175-2185.

[2] G. Anderson, U. Zheng, R. Wyeth, A. Johnson, and J. Bissett, "A rough set/fuzzy logic based decision making system for medical applications", International Journal of General Systems, Vol. 29, No.6, 2000, pp. 879-96.

[3] V. Malathi and N.S. Marimuthu, "Support Vector Machine for Discrimination Between Fault and Magnetizing Inrush Current in Power Transformer," Journal of Computer Science 3 (11), 2007, pp. 894-897.

[4] Y. Wang, S. Wang, and K. K. Lai, "A New Fuzzy Support Vector Machine to Evaluate Credit Risk," IEEE Trans. Fuzzy Systems, Vol. 13, No. 6, Dec 2005, pp. 820-831.

[5] M. Catelani and A. Fort, "Soft Fault Detection and Isolation in Analog Circuits: Some Results and a Comparison Between a Fuzzy Approach and Radial Basis Function Networks," IEEE Trans. Instr. Meas. Vol. 51, No. 2, April 2002, pp. 196-202.

[6] R. Sałat and S. Osowski, "Analog Filter Diagnosis Using Support Vector Machine," Proc. ECCTD'03 Conf., Cracow, Poland, 2003, pp. III-421-424.

[7] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods," Cambridge University Press, 2000.

[8] C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining and Knowledge Discovery, No 2, 1998, pp. 121–167.

[9] K. Pelckmans, J.A.K. Suykens, T. Van Gestel, J. De Brabanter, L. Lukas, B. Hamers, B. De Moor, J. Vandewalle, "LS-SVMlab Toolbox User's Guide," online at http://www.esat.kuleuven.ac.be/sista/lssvmlab/

[10] S. Gunn, "Support Vector Machines for Classification and Regression," online at: http://www.isis.ecs.soton.ac.uk/resources/svminfo/

[11] Z. Michalewicz, D.B. Fogel, "How to Solve It: Modern Heuristics," Springer-Verlag, 2004.



***Authors**: dr inż. Piotr Bilski, Szkoła Główna Gospodarstwa Wiejskiego, Wydział Zastosowań Informatyki i Matematyki, ul. Nowoursynowska 159, 60-965 Poznań, E-mail: piotr_bilski@sggw.pl; Instytut Radioelektroniki, Politechnika Warszawska, ul. Nowowiejska 15/19, 00-650 Warszawa, E-mail: pbilski@ire.pw.edu.pl.*