Semi-supervised SVM with Fuzzy Controlled Cooperation of Biology Related Algorithms

Shakhnaz Akhmedova, Eugene Semenkin and Vladimir Stanovov
Department of System Analysis and Operations Research, Reshetnev Siberian State University of Science and Technology, "Krasnoyarskiy Rabochiy" avenue, 31, Krasnoyarsk, Russia

Keywords: Bio-inspired Algorithms, Fuzzy Controller, Support Vector Machines, Semi-Supervised Learning, Classification, Constrained Optimization.

Abstract: Due to its wide applicability, the problem of semi-supervised classification is attracting increasing attention in machine learning. Semi-Supervised Support Vector Machines (SVM) are based on applying the margin maximization principle to both labelled and unlabelled examples. A new collective bionic algorithm, namely fuzzy controlled cooperation of biology related algorithms (COBRA-f), which solves constrained optimization problems, has been developed for semi-supervised SVM design. Firstly, the experimental results obtained by the two types of fuzzy controlled COBRA are presented and compared and their usefulness is demonstrated. Then the performance and behaviour of proposed semi-supervised SVMs are studied under common experimental settings and their workability is established.

1 INTRODUCTION

One of the most important machine learning tasks is classification that consists in identifying to which of a set of categories a new instance belongs. If sufficient labelled training data are given, there exists a variety of techniques, for example, artificial neural networks (Bishop, 1996), fuzzy logic classifiers (Kuncheva, 2000) or Support Vector Machines (SVM) (Vapnik and Chervonenkis, 1974), to address such a task. However, labelled data are often rare in real-world applications. Therefore, recently semi-supervised learning has attracted increasing attention among researchers (Zhu and Goldberg, 2009).

In contrast to supervised methods, the latter class of techniques takes both labelled and unlabelled data into account to construct appropriate models. A well-known concept in this field is semi-supervised support vector machines (Bennett and Demiriz, 1999), which depict the direct extension of support vector machines to semi-supervised learning scenarios.

In this study semi-supervised SVMs generated by a new collective bionic optimization algorithm, namely fuzzy controlled cooperation of biology related algorithms or COBRA-f, are described. Initially, a meta-heuristic approach called Co-Operation of Biology Related Algorithms or COBRA (Akhmedova and Semenkin, 2013 (1)) was developed for solving unconstrained real-parameter optimization problems. Its basic idea consists in the cooperative work of different nature-inspired algorithms, which were chosen due to the similarity of their schemes. However, there are still various algorithms which can be used as components for COBRA as well as previously conducted experiments demonstrating that even the bionic algorithms already chosen can be combined in various ways.

Thus, to solve the described problem, in this work COBRA was modified by implementing controllers based on fuzzy logic (Lee, 1990). The aim of this was to determine in an automated way which bionic algorithm should be included in the co-operative work. The proposed modification also allows resources to be allocated properly while solving unconstrained optimization problems. And finally the obtained modification COBRA-f was adopted for solving constrained optimization problems.

Therefore, in this paper firstly a brief description of the semi-supervised SVM is presented. Then the COBRA meta-heuristic approach and the fuzzy controller are described. In the next section, the

Akhmedova, S., Semenkin, E. and Stanovov, V.
Semi-supervised SVM with Fuzzy Controlled Cooperation of Biology Related Algorithms.
DOI: 10.5220/0006417400640071
In *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2017) - Volume 1*, pages 64-71
ISBN: 978-989-758-263-9

experimental results obtained by two types of fuzzy controller are discussed. And after that the implementation of the best obtained fuzzy controlled COBRA was applied for solving constrained optimization problems as well as training the semi-supervised SVM. For experiments several datasets have been chosen, among which there are synthetic and real datasets. In particular, we have used a popular two moons problem, two datasets from the UCI repository (namely Breast Cancer Wisconsin (BCW) and Pima Indian Diabetes (PID)) with only the 10 labels used available, and the gas turbine dangerous vibrations detection problem. Finally, some conclusions are given in the last section.

2 SEMI-SUPERVISED SUPPORT VECTOR MACHINES

In Support Vector Machines (SVM), the intuition is to try to create a separating hyperplane between the instances from different classes (Vapnik and Chervonenkis, 1974). SVM is based on the maximization of the distance between the discriminating hyperplane and the closest examples. In other words since many choices could exist for the separating hyperplane, in order to generalize well on test data, the hyperplane with the largest margin has to be found.

Suppose $L = \{(x_1, y_1), \dots, (x_l, y_l)\}$, $x_i \in R^m$ is a training set with l examples (instances), each instance x_i has m attributes and is labelled as y_i, where $i = \overline{1, l}$. Let v be a hyper-plane going through the origin, δ be the margin and $w = \frac{v}{\delta}$. The margin maximizing hyperplane can be formulated as a constrained optimization problem in the following manner:

$$\frac{1}{2}\|w\|^2 \rightarrow \min \tag{1}$$

$$y_i(w \cdot x_i) \geq 1 \tag{2}$$

To solve the given optimization problem, the proposed fuzzy controlled cooperation of biology related algorithms or COBRA-f was used.

However, in this study semi-supervised SVMs were considered. Thus, given the additional set $U = \{x_{l+1}, \dots, x_{l+u}\}$ of unlabelled training patterns, semi-supervised support vector machines aim at finding an optimal prediction function for unseen data based on both the labelled and the unlabelled part of the data (Joachims, 1999). For unlabelled data, it is assumed that the true label is the one predicted by the model based on what side of the hyperplane the unlabelled point ends up being.

In this study, self-training was used to learn from the unlabelled data. Namely, the idea is to design the model with labelled data and then use the model's own predictions as labels for the unlabelled data to retrain a new model with the original labelled data and the newly labelled data and then iteratively repeat this process.

The problem with this method is that considering its own predictions as true labels can cause the model to drift away from the correct model if the predictions were wrong initially. The model would then continue to mislabel data and use it again and continue to drift away from where it should be. Therefore, to prevent this problem the technique described in (Ravi, 2014) was used. More specifically, the model's predictions were used to label the data only when there is a high confidence about the predictions.

The notion of confidence used for the SVM model is the distance from the found hyperplane. The larger the distance from the hyperplane, the higher the probability that the instance belongs to the corresponding side of the separating hyperplane.

Consequently, the following basic steps were performed:

- Train SVM on the labelled set L by the proposed meta-heuristic approach COBRA-f;
- Use obtained SVM to classify all unlabelled instances from U by checking the confidence criteria from (Ravi, 2014);
- Label instances from the set U if this is possible;
- Repeat from the first step.

Thus, the simplest semi-supervised learning method was used for examining the workability of COBRA-f.

3 CO-OPERATION OF BIOLOGY RELATED ALGORITHMS

The meta-heuristic approach called Co-Operation of Biology Related Algorithms or COBRA (Akhmedova and Semenkin, 2013) was developed based on five optimization methods, namely Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995), Wolf Pack Search (WPS) (Yang et al., 2007), the Firefly Algorithm (FFA) (Yang, 2009), the Cuckoo Search Algorithm (CSA) (Yang and Deb,

2009) and the Bat Algorithm (BA) (Yang, 2010) (hereinafter referred to as "component-algorithms"). Also, the Fish School Search (FSS) (Bastos and Lima, 2009) was later added as COBRA's component-algorithm.

The main reason for the development of a cooperative meta-heuristic was the inability to say which of the above-listed algorithms is the best one or which algorithm should be used for solving any given optimization problem (Akhmedova and Semenkin, 2013). Thus, the idea was to use the cooperation of these bionic algorithms instead of any attempts to understand which one is the best for the problem in hand.

The originally proposed approach consists in generating five populations, one population for each bionic algorithm (or generating six populations with the FSS algorithm added) which are then executed in parallel, cooperating with each other. The COBRA algorithm is a self-tuning meta-heuristic, so there is no need to choose the population size for each component-algorithm. The number of individuals in the population of each algorithm can increase or decrease depending on the fitness values: if the overall fitness value was not improved during a given number of iterations, then the size of each population increased, and vice versa, if the fitness value was constantly improved during a given number of iterations, then the size of each population decreased.

There is also one more rule for population size adjustment, whereby a population can "grow" by accepting individuals removed from other populations. The population "grows" only if its average fitness value is better than the average fitness value of all other populations. Therefore, the "winner algorithm" can be determined as an algorithm whose population has the best average fitness value. This can be done at every step. The described competition among component-algorithms allows the biggest population size to be allocated to the most appropriate bionic algorithm on the current generation.

The main goal of this communication between all populations is to bring up-to-date information on the best achievements to all component-algorithms and prevent their preliminary convergence to their own local optimum. "Communication" was determined in the following way: populations exchange individuals in such a way that a part of the worst individuals of each population is replaced by the best individuals of other populations. Thus, the group performance of all algorithms can be improved.

The performance of the COBRA algorithm was evaluated on a set of various benchmark problems and the experiments showed that COBRA works successfully and is reliable on different benchmarks (Akhmedova and Semenkin, 2013). Besides, the simulations showed that COBRA is superior to its component-algorithms when the dimension grows or when complicated problems are solved.

Then COBRA's modification for solving constrained optimization problems COBRA-c was developed (Akhmedova and Semenkin, 2013 (2)). Three constraint handling methods were used for this purpose: dynamic penalties (Eiben and Smith, 2003), Deb's rule (Deb, 2000) and the technique described in (Liang, Shang and Li, 2010). The method proposed in (Liang, Shang and Li, 2010) was implemented in the PSO-component of COBRA; at the same time other components were modified by implementing Deb's rule followed by calculating function values using dynamic penalties.

The performance of this modification was evaluated with a set of various test functions. It was established that COBRA-c works successfully and is sufficiently reliable. Finally, COBRA's modification outperforms all of its component-algorithms.

4 FUZZY CONTROLLER

The size control of the COBRA populations was performed by the fuzzy controller, which received algorithms' success rates as inputs, and returned the populations' size modification values. Overall, there were 7 input variables, i.e. one variable for each of COBRA's 6 component-algorithms, showing its success rate, plus the overall success rate of all components.

The success rate for all input variables except for the last one was evaluated as the best fitness value of its population. The last input variable was determined as the ratio of the number of iterations, during which the best-found fitness value was improved, to the given number of iterations, which was a constant period.

The number of outputs was equal to the number of components.

The fuzzy rules had the following form:

$$R_q\text{: IF } x_1 \text{ is } A_{q1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{qn} \text{ THEN } y_1 \text{ is } B_{q1} \text{ and } \ldots\ y_k \text{ is } B_{qk} \tag{3}$$

where R_q is the q-th fuzzy rule, $x = (x_1,...,x_n)$ are the input values (components' success rate) in n-dimensional space ($n = 7$ in this study),

$y = (y_1,...,y_k)$ is the set of outputs ($k = 6$), A_{qi} is the fuzzy set for the i-th input variable, B_{qj} is the fuzzy set for the j-th output variable. The Mamdani-type fuzzy inference with a centre of mass calculation was used as the defuzzification method.

For the purposes of this study two variants of the fuzzy controller, which differed in the number of terms for output variables, have been implemented. All inputs were values in the range $[0;1]$, so that the input fuzzy terms were equal for all variables. Also, 3 basis triangular fuzzy terms were used, and in addition the A_4 term combining A_2 and A_3, as well as the "Don't Care" condition (DC) have been included to decrease the number of rules. The term shapes are shown in Figure 1.

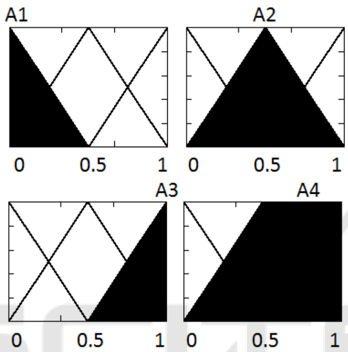

Figure 1: Fuzzy sets for inputs.

The first fuzzy controller's 3 fuzzy terms, which were used for the output, are demonstrated in Figure 2.

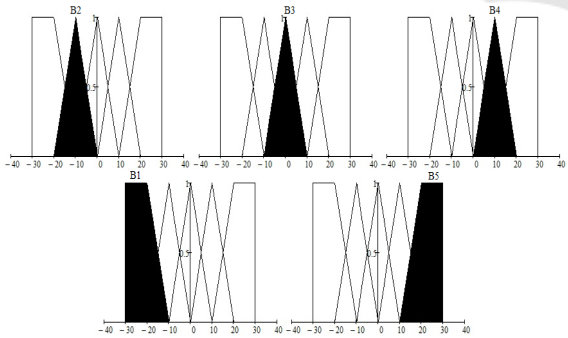

Figure 2: Fuzzy terms for all 6 outputs, first controller.

The adjustable parameters of the first fuzzy controller are the values encoding positions of output fuzzy terms, i.e. the position of central term and side terms. For the example shown in Figure 2, these values are -20, -10, 10 and 20, i.e. four values were encoded, so that the terms may appear to be non-symmetric after optimization.

A part of the rule base for the first controller is presented in Table 1.

Table 1: Part of the first controller's rule base.

№	IF			THEN	
1	X_1 is A_3	X_2-X_6 is A_4	X_7 is DC	Y_1 is B_5	Y_2-Y_6 is B_2
2	X_1 is A_2	X_2-X_6 is A_4	X_7 is DC	Y_2 is B_5	Y_2-Y_6 is B_2
3	X_1 is A_1	X_2-X_6 is A_4	X_7 is DC	Y_3 is B_5	Y_2-Y_6 is B_2
…					
19	X_1-X_6 is DC		X_7 is A_1	Y_1 is B_1	
20	X_1-X_6 is DC		X_7 is A_2	Y_1 is B_3	
21	X_1-X_6 is DC		X_7 is A_3	Y_1 is B_5	

The second controller had 7 terms for output variables instead of 5. Two additional terms were used for the last part of the rule base, which defined the influence of the overall success rate, i.e. the 7-th variable. The fuzzy terms for the second controller are shown in Figure 3.

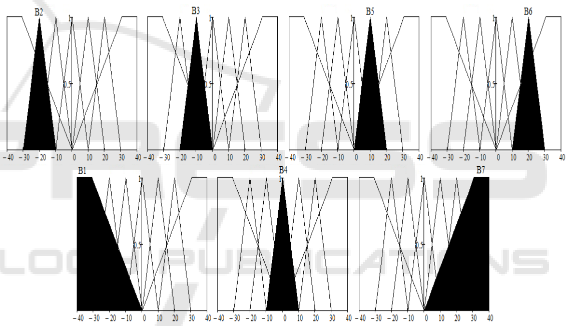

Figure 3: Fuzzy terms for all 6 outputs, second controller.

Terms 1 and 5 in the second controller have different shapes, which allows the size control of overall populations to be tuned more accurately.

The rule base is also different; it uses different terms for the output. A part of the rule base is presented in Table 2.

Table 2: Part of the second controller's rule base.

№	IF			THEN	
1	X_1 is A_3	X_2-X_6 is A_4	X_7 is DC	Y_1 is B_6	Y_2-Y_6 is B_3
2	X_1 is A_2	X_2-X_6 is A_4	X_7 is DC	Y_2 is B_6	Y_2-Y_6 is B_3
3	X_1 is A_1	X_2-X_6 is A_4	X_7 is DC	Y_3 is B_6	Y_2-Y_6 is B_3
…					
19	X_1-X_6 is DC		X_7 is A_1	Y_1 is B_1	
20	X_1-X_6 is DC		X_7 is A_2	Y_1 is B_4	
21	X_1-X_6 is DC		X_7 is A_3	Y_1 is B_7	

The second controller tends to be more flexible, although it requires the tuning of 6 parameters instead of 4 for the first one. For the case shown in Figure 3 the parameter values are -30, -20, -10, 10, 20, 30, but terms may end up non-symmetric afterwards.

5 EXPERIMENTAL RESULTS

In this section, the methodology employed in this study to validate the proposed approach is presented. The next sections describe the techniques used for comparison purposes, the benchmark functions and the statistical analysis.

5.1 Constrained Optimization Problems

In this study 6 benchmark problems taken from (Whitley, 1995) were used in experiments for comparing the constrained optimization algorithms. Optimal solutions for these problems are already known, thus the algorithm's reliability was estimated by the achieved error value.

The given benchmark functions were considered to evaluate the robustness of the fuzzy controlled COBRA, which was modified for solving constrained optimization problems in two ways:

- By using dynamic penalties (Eiben and Smith, 2003);
- By using Deb's rule (Deb, 2000).

Consequently, firstly test problems were used to determine the best parameters for the four types of fuzzy controllers:

- Controller with 4 parameters, constraint handling technique is dynamic penalties;
- Controller with 4 parameters, constraint handling technique is Deb's rule;
- Controller with 6 parameters, constraint handling technique is dynamic penalties;
- Controller with 6 parameters, constraint handling technique is Deb's rule.

The standard Particle Swarm Optimization algorithm was used for this purpose. Therefore, the individuals were each represented as parameters of the fuzzy controlled COBRA, namely the positions of the output fuzzy terms. The following objective function was optimized by the PSO algorithm:

$$F(x)=\frac{1}{6}\sum_{i=1}^{6}\frac{1}{T}\sum_{t=1}^{T}f_i^t(x), \quad (4)$$

where $T=10$ is the total number of program runs for each benchmark problem listed earlier. Thus, on each iteration all test problems were solved T times by a given fuzzy controlled COBRA and then the obtained results were averaged. Calculations were stopped on each program run if the number of function evaluations exceeded $10000D$. The population size for the PSO algorithm was equal to 50 and the number of iterations was equal to 100; calculations were stopped on the 100-th iteration for the PSO heuristic.

Accordingly, the following parameters for the fuzzy controllers were obtained:

- $[-9;-5;5;27]$;
- $[-33;-10;27;34]$;
- $[-14;0;0;1;24;29]$;
- $[-23;-6;-3;14;16;21]$;

On the following step, the obtained parameters were applied to the fuzzy controlled COBRA and it was tested on the mentioned benchmark functions. There were 51 program runs for each constrained optimization problem, and calculations were stopped if the number of function evaluations was equal to $10000D$. Also, for example, a change in population sizes was obtained while testing on the benchmark problems. This change for the third problem is presented in Figure 4.

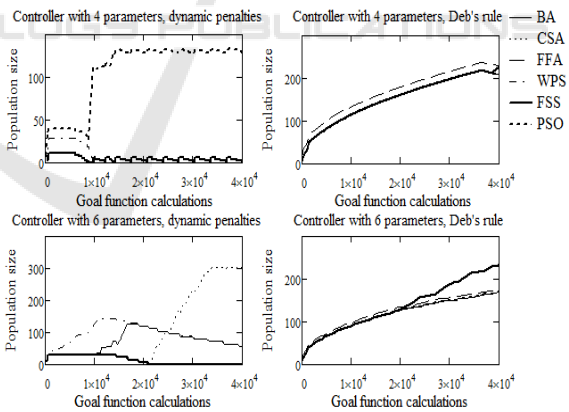

Figure 4: Change in population sizes.

Based on the results received for other functions, it was concluded that the algorithm with Deb's rule exhibits strange behaviour, i.e. it tends to increase the size of all populations, while dynamic penalties show more complicated cooperation. More specifically, for the second fuzzy controller with 6 variables, for example, CSA is one of the worst algorithms for the first 20000 calculations (it has around 3 points available), but after 20000 it rapidly increases its population size to 300 and more, because it shows a much better ability to optimize the function. At the same time, BA and WPS, who were winners at the first phase, gradually decrease their resource.

In Table 3 and Table 4 the results obtained by the fuzzy controlled COBRA, the controller of which has 4 parameters, are presented with the best parameters. The following notations are used: the best found function value (*Best*), the function value averaged by the number of program runs (*Mean*) and the standard deviation (*STD*).

Table 3: Results obtained by the fuzzy controlled COBRA (4 parameters) with dynamic penalties.

	Best	*Mean*	*STD*
1	0.000222883	0.000450223	0.000269495
2	4.44089e-016	1.20453e-006	5.73208e-006
3	0.0168729	0.0861003	0.106644
4	0.00246478	0.0270861	0.0240794
5	0.000349198	0.00205344	0.00184918
6	4.07214e-005	0.00012163	7.28906e-005

Table 4: Results obtained by the fuzzy controlled COBRA (4 parameters) with Deb's rule.

	Best	*Mean*	*STD*
1	0.000194127	6.48378	12.6967
2	1.37668e-014	0.0508509	0.153545
3	0.241445	1.59201	1.35446
4	0.093186	6.36309	6.95277
5	0.0646682	5.93762	7.14939
6	0.00463816	3.50586	19.3554

Table 5: Results obtained by the fuzzy controlled COBRA (6 parameters) with dynamic penalties.

	Best	*Mean*	*STD*
1	0.000222799	0.0023863	0.00386538
2	4.44089e-016	5.45027e-006	1.14566e-005
3	0.0265527	0.249473	0.645633
4	0.0114249	0.0379335	0.0643213
5	0.000159134	0.0380605	0.125444
6	0.000236098	0.000955171	0.00178648

Table 6: Results obtained by the fuzzy controlled COBRA (6 parameters) with Deb's rule.

	Best	*Mean*	*STD*
1	1.47682e-009	0.00234639	0.0054576
2	3.10862e-015	0.000577974	0.00155304
3	0.015536	0.055806	0.0984222
4	0.00559601	0.038134	0.113326
5	0.000160994	0.0656134	0.130952
6	6.15548e-006	0.000296266	0.000680345

In Table 5 and Table 6 the results obtained by the fuzzy controlled COBRA, the controller of which has 6 parameters, are presented with the best parameters. The same notations as in the previous tables are used.

For comparison, in Table 7 the results obtained by COBRA-c with six component-algorithms with the standard tuning method are given.

Table 7: Results obtained by COBRA-c with six component algorithms.

	Best	*Mean*	*STD*
1	2.54087e-005	0.00710629	0.0177543
2	2.08722e-014	0.000114461	5.15068e-005
3	6.39815e-005	0.0477759	0.0422946
4	0.0267919	0.0324922	0.00119963
5	0.000341288	0.0678906	0.0331402
6	1.21516e-005	0.000278707	0.000268217

Thus, the comparison demonstrates that the fuzzy controlled COBRA with dynamic penalties outperformed the same algorithms with Deb's rule. Aside from this, there is no significant difference between the results obtained by the fuzzy controlled COBRA with either 4 or 6 parameters. However, the 4-parameter fuzzy controlled COBRA with dynamic penalties also outperformed the COBRA with six components without a controller. Therefore, it can be used for solving the optimization problems instead of the given algorithm's versions.

5.2 Classification Performance

Several artificial and real-world data sets described in Table 8 were considered in this study, namely the well-known two-dimensional "Moons" data set and data sets for two medical diagnostic problems (Frank and Asuncion, 2010). Each data set instance was split into a labelled part and an unlabelled one, and the different ratios for the particular settings were used.

Table 8: Data sets considered in the experimental evaluation, each consisting of *n* patterns having *d* features.

Data Set	*n*	*d*
Moons	200	2
Breast Cancer Wisconsin	699	9
Pima Indians Diabetes	768	8

For the sake of exposition, firstly the well-known “Moons” data set was considered. This choice is conditioned by the fact that the given data set is a difficult training instance for semi-supervised support vector machines due to its non-linear structure. The “Moons” problem is a classical semi-supervised problem for testing algorithms. It consists of two groups of moon-like sets of points, which are easily recognized as two classes by a human, but represent significant difficulty for modern algorithms. In the conducted experiments only 2 labelled points for every class were known, and the rest of the points were classified using the semi-supervised SVM described above. The results obtained on the “Moons” problem are shown in Figure 5.

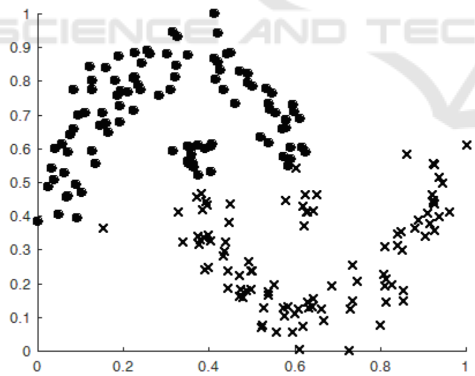

Figure 5: Semi-supervised classification of “Moons”.

As can be seen, the algorithm does not recognize all points correctly, i.e. it builds an almost linear classification. However, most of the points are in the right class.

Then two medical diagnostic problems, namely Breast Cancer Wisconsin (BCW) and Pima Indian Diabetes (PID), were solved. Both problems are binary classification tasks. For these data sets, 10 examples were randomly selected to be used as labelled examples, and the remaining instances were used as unlabelled data. The experiments are repeated 10 times and the average accuracies and standard deviations are recorded. The results are shown in Table 9. Alternative algorithms (linear SVMs) for comparison are taken from (Li and Zhou, 2011).

Table 9: Performance comparison of semi-supervised methods.

	BCW	*PID*
TSVM	89.2±8.6	63.4±7.6
S3VM-c	94.2±4.9	63.2±6.8
S3VM-p	93.9±4.9	65.6±4.8
S3VM-us	93.6±5.4	65.2±5.0
This study	95.5±1.8	69.3±1.5

The gas turbine dangerous vibrations problem includes 11 input variables, which are process parameters, potentially connected to the vibration level, and the output is the class number – dangerous/stable vibration level. The vibration signal is one of the most important diagnostic instruments when measuring the turbine wear.

For the experiments with this dataset, we have used 5%, 10% and 15% of the labelled data for training, while the rest of the training set was unlabelled. The total size of the dataset is 1000 instances, 900 were used for training, while 100 instances were left for a test set. In 3 experiments, the number of labelled examples was 45, 90 and 135 instances. The average classification quality on the test set obtained after 10 experiments is presented in Table 10.

Table 10: Performance comparison, gas turbine dataset.

Labelled	*COBRA Semi-supervised SVM*
5%	86.2±1.7
10%	87.8±0.4
15%	88.2±0.6

The classification quality is relatively high even with only 5% of labelled examples in the training set. This result provides the possibility to use a vast amount of available unlabelled data for model improvements in future.

Consequently, the inference should be drawn that the suggested algorithm successfully solved all the problems of designing semi-supervised SVM-based classifiers with competitive performance. Thus, the study results can be considered as confirming the reliability, workability and usefulness of the fuzzy controlled cooperative algorithm in solving real world optimization problems.

6 CONCLUSIONS

The problem of semi-supervised classification is important due to the fact that obtaining labelled examples is often very expensive. However, using this data during classification may be helpful. In this paper, the semi-supervised SVM was trained using a cooperative algorithm, whose components were automatically adjusted by a fuzzy controller. The fuzzy controller itself was tuned to deliver better results for constrained optimization problems. This tuning of the meta-heuristic allowed better results of SVM training to be achieved, compared to other studies. The proposed approach, combining biology-related algorithms and fuzzy controllers could be applied to other complex constrained optimization problems.

ACKNOWLEDGEMENTS

Research is performed with the support of the Ministry of Education and Science of Russian Federation within State Assignment project № 2.1680.2017/ПЧ.

REFERENCES

Akhmedova, Sh., Semenkin, E., 2013 (1). Co-Operation of Biology related Algorithms. In *IEEE Congress on Evolutionary Computations*. IEEE Publications.

Akhmedova, Sh., Semenkin, E., 2013 (2). *New optimization metaheuristic based on co-operation of biology related algorithms*, Vestnik. Bulletine of Siberian State Aerospace University. Vol. 4 (50).

Bastos, F. C., Lima, N. F., 2009. *Fish School Search: an overview*, Nature-Inspired Algorithms for Optimization. Series: Studies in Computational Intelligence. Vol. 193.

Bennett, K. P., Demiriz, A., 1999. Semi-supervised support vector machines, *Advances in Neural Information Processing Systems 11*.

Bishop, C. M., 1996. *Theoretical foundation of neural networks*. Technical report, Aston Univ., Neural computing research group, UK.

Deb, K., 2000. *An efficient constraint handling method for genetic algorithms*, Computer methods in applied mechanics and engineering. Vol. 186(2-4).

Eiben, A. E., Smith, J. E., 2003. *Introduction to evolutionary computation*, Springer. Berlin.

Frank, A., Asuncion, A., 2010. *UCI Machine Learning Repository*. Irvine, University of California, School of Information and Computer Science. http://archive.ics.uci.edu/ml

Joachims, T., 1999. Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning*.

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In *IEEE International Conference on Neural Networks.*

Kuncheva, L. I., 2000. *How Good Are Fuzzy If-Then Classifiers*, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics. Vol. 30, No. 4.

Lee, C.-C., 1990. *Fuzzy logic in control systems: fuzzy logic controller-parts 1 and 2*, IEEE Transactions on Systems, Man, and Cybernetics. Vol. 20, No. 2.

Li, Y. F., Zhou, Z. H., 2011. Improving Semi-Supervised Support Vector Machines Through Unlabeled Instances Selection. In *The Twenty Fifth AAAI Conference on Artificial Intelligence*.

Liang, J. J., Shang Z., Li, Z., 2010. Coevolutionary Comprehensive Learning Particle Swarm Optimizer. In *CEC'2010, Congress on Evolutionary Computation*. IEEE Publications.

Ravi, S., 2014. *Semi-supervised Learning in Support Vector Machines.* Project Report COS 521.

Vapnik, V., Chervonenkis, A., 1974. *Theory of Pattern Recognition*, Nauka. Moscow.

Whitley, D., 1995. Building Better Test Functions. In *The Sixth International Conference on Genetic Algorithms and their Applications*.

Yang, Ch., Tu, X., Chen, J., 2007. Algorithm of marriage in honey bees optimization based on the wolf pack search. In *International Conference on Intelligent Pervasive Computing*.

Yang, X. S., 2009 Firefly algorithms for multimodal optimization. In *The 5th Symposium on Stochastic Algorithms, Foundations and Applications*.

Yang, X. S., 2010. A new metaheuristic bat-inspired algorithm. *Nature Inspired Cooperative Strategies for Optimization*, Studies in Computational Intelligence. Vol. 284.

Yang, X. S., Deb, S., 2009. Cuckoo Search via Levy flights. In *World Congress on Nature & Biologically Inspired Computing*. IEEE Publications.

Zhu, X., Goldberg, A. B., 2009. *Introduction to Semi-Supervised Learning*, Morgan and Claypool.