

Automatically Generated Data Mining Tools for Complex System Operator's Condition Detection Using Non-Contact Vital Sensing

Shakhnaz AKHMEDOVA^{†a)}, Vladimir STANOVOV[†], Sophia VISHNEVSKAYA[†], Chiori MIYAJIMA^{††},
and Yukihiro KAMIYA^{††}, *Nonmembers*

SUMMARY This study is focused on the automated detection of a complex system operator's condition. For example, in this study a person's reaction while listening to music (or not listening at all) was determined. For this purpose various well-known data mining tools as well as ones developed by authors were used. To be more specific, the following techniques were developed and applied for the mentioned problems: artificial neural networks and fuzzy rule-based classifiers. The neural networks were generated by two modifications of the Differential Evolution algorithm based on the NSGA and MOEA/D schemes, proposed for solving multi-objective optimization problems. Fuzzy logic systems were generated by the population-based algorithm called Co-Operation of Biology Related Algorithms or COBRA. However, firstly each person's state was monitored. Thus, databases for problems described in this study were obtained by using non-contact Doppler sensors. Experimental results demonstrated that automatically generated neural networks and fuzzy rule-based classifiers can properly determine the human condition and reaction. Besides, proposed approaches outperformed alternative data mining tools. However, it was established that fuzzy rule-based classifiers are more accurate and interpretable than neural networks. Thus, they can be used for solving more complex problems related to the automated detection of an operator's condition.

key words: classification, fuzzy logic, neural networks, non-contact vital sensing

1. Introduction

The importance of data analysis and machine learning techniques is increasing due to the coming of the Internet of Things (IoT) era [1]. These two research areas are expected to play a main role towards the realization of attractive concepts such as Industry 4.0 or Society 5.0 [2]. Therefore, new ideas regarding data mining tools and their applications are proposed regularly.

Automated detection of human condition or reaction to something at a given moment or period is one of the most interesting and the hardest problems related to the mentioned concepts [3]. Such problems can be formulated for various complex systems with human operators, for example, in medicine operators are elderly people in nursing houses or mentally ill people.

First of all, before the automated detection of an operator's condition, the person's state should be monitored.

This can be done by standard sensors. However, for higher efficiency the non-contact vital sensing using the Doppler sensors introduced in [4] was applied for human condition monitoring. It was done because of their low computational complexity, which allows implementing them with small-scale processors so that the battery life can be extended, and it is one of the most important factors in the IoT context.

In this study the conducted experiments are related to the human condition (or reaction) while listening (and not listening at all) to music. Measurements obtained by the non-contact vital sensing for each person that participated in the experiments were formalized and pre-processed. Thus, two classification problems were formulated.

To solve the mentioned classification problems artificial neural networks (ANN) [5] and fuzzy systems [6] were used. In this study two modifications of the Differential Evolution algorithm (DE) [7], developed for solving multi-objective optimization problems, were proposed for ANN automated design. Thus, the idea was to generate ANN with a relatively simple structure which would effectively solve a given classification problem. Also a population-based optimization method called Co-Operation of Biology Related Algorithms or shortly COBRA [8] was applied for the automated generation of fuzzy rule-based classifiers [9].

Other data mining tools were used to solve classification problems related to human condition detection: support vector machines (SVM) [10], k nearest neighbours (k -NN) [11], decision trees (DT) [12], the Hybrid Evolutionary Fuzzy Classification Algorithm (HEFCA) [13] and standard artificial neural networks are among them. Comparison of the obtained results by all mentioned classifiers is demonstrated.

Thus, in this paper firstly a brief description of non-contact vital sensing using Doppler sensors is given. Then the considered classification problems are presented. After that the proposed approaches are introduced. In the next section, the experimental results obtained by different data mining tools are discussed, and finally, some conclusions are given in the last section.

2. Non-Contact Vital Sensing

Nowadays, there are various technologies that can be used for condition monitoring of operators of different complex systems. A Doppler sensor [4] is a device that allows us to realize a non-contact vital sensing, namely the vital sensing

Manuscript received October 28, 2020.

Manuscript publicized December 24, 2020.

[†]The authors are with Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, Russian Federation.

^{††}The authors are with Aichi Prefectural University, Nakaguteshi, 480-1342 Japan.

a) E-mail: shahnaz@inbox.ru

DOI: 10.1587/transcom.2020HMI0001

without needing electrodes to be put on a body surface. It enables us to measure heartbeats, breathing and body motion without causing any stress which would affect the measurement [4]. This technology is expected to enhance the performance of machine learning techniques used for the condition detection of the corresponding complex systems' operators and to improve their management.

The principle of the Doppler sensor can be described as follows. The transmitter embedded in the Doppler sensor radiates an electromagnetic wave and the reflected waves yielded by human bodies are gathered by the receiver. Thus, the receiver detects the reflected electromagnetic wave whose frequency is fluctuated by the Doppler Effect caused by the movement of the body surface. Therefore, the respiration can be extracted by digital signal processing of the frequency deviation.

It should be noted that if there are multiple human bodies, there is also a need to separate their heartbeats. However, in reality there are hardly any conventional works on the non-contact monitoring of heartbeats capable of coping with multiple bodies. Therefore, a new scheme to cope with multiple bodies was proposed in [4].

Later the pre-processing technique, namely Accumulation for Real-time Serial-to-parallel converter (ARS) [14], was introduced as a simple parameter estimation method for the heartbeats and respirations. This paper proposes to use ARS as a pre-processing method for signals received by non-contact Doppler sensors.

3. Problem Statement

In this study, as an example, human condition and reaction while listening and not listening to music were determined. For this purpose, firstly ten people of different gender and age were asked to participate in experiments. Their condition was monitored by non-contact vital sensing using Doppler sensors over three stages:

- listening to music that a participant admitted to like in three different time periods;
- listening to music that a participant admitted to dislike in three different time periods;
- not listening to music at all in two different time periods.

Conditions and parameters of the conducted experiments for respiration monitoring using a Doppler sensor can be briefly described as follows. Firstly, participants seated in front of a desk were monitored by a Doppler sensor installed 30 cm away from their chests (Fig. 1).

The Doppler sensor module was equipped with two output ports, called *I*- and *Q*-channels. The outputs from the two ports were amplified by two amplifiers so that the voltages are sampled by a data logger (Fig. 2).

The model numbers and specifications of the equipment are listed in the Table 1.

Received signals were pre-processed by using the ARS technique mentioned in the previous section. Thus, eight

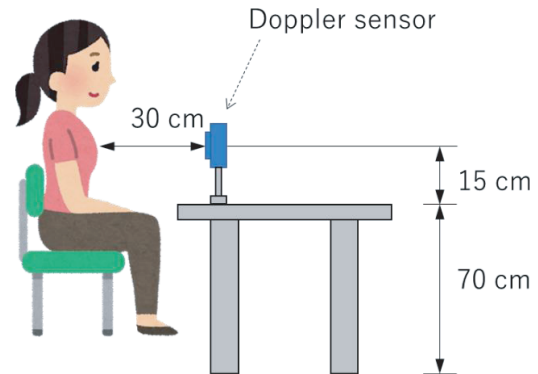


Fig. 1 Settings of the experiment.

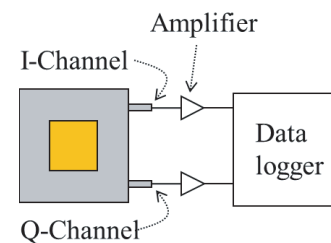


Fig. 2 Configuration of the data acquisition system.

Table 1 Specifications of equipment.

Data logger	GL-900 (GRAPHTECH)
Sampling	100 Hz, 16 bits
Doppler sensor module	NJR4262 (New Japan Radio) Frequency 24 GHz

data sequences were obtained for each person, who participated in the experiments. Every data sequence pre-processed by ARS technique consists of the following four real-valued attributes (or features in other words): the deviation of the respiratory rate, voltage, the average value of the respiratory rate and the variance of the respiratory rate.

The obtained data were then also normalized; and the following two classification problems were formulated:

- the problem, which was referred to as “listened”, where each data sequence was labelled as “1” if the participant listened to the music and “0” otherwise;
- the problem, which was referred to as “liked”, where each data sequence was labelled as “1” if the participant liked the music and “0” otherwise.

The database for the classification problem “listened” consists of 80 instances: 60 records of the class “listened to music” and 20 records for the class “did not listen to music”. Therefore, the first dataset was imbalanced and harder to classify.

The second classification problem, “liked”, can be considered as an opinion mining problem, because opinion mining problems are the problems of determining the judgment of a speaker about a particular topic. The dataset for that problem contains 60 instances, of which half are from the class “liked music” and half are from the class “did not like music”.

Table 2 Description of the formulated classification problems.

Problem	Classes	Dataset sizes
1	1 – listened to music	1: 60 instances 0: 20 instances
	0 – did not listen to music	whole set: 80 instances
2	1 – liked music	1: 30 instances 0: 30 instances
	0 – did not like music	whole set: 60 instances

The datasets for both classification problems are presented in Table 2.

4. Proposed Data Mining Tools

4.1 Automatically Generated Neural Networks

The tuning of artificial neural network structure (the number of hidden layers, the number of neurons on each layer and activation functions for these neurons) as well as its weight coefficients is considered as the solving of multi-objective unconstrained optimization problem with binary and real-valued variables. The type of variables depends on the representation of the ANN structure and coefficients.

First of all the maximum number of hidden layers was equal to ML and the maximum number of neurons on each hidden layer was set to MN, so the maximum number of neurons was equal to $ML \times MN$ in this study.

Each node was represented by a binary string of the length 4. If the string consisted of zeros (“0000”) then this node did not exist in the ANN. So, whole structure of the neural network was represented by $MN \times ML \times 4$ binary variables, where each $MN \times 4$ variables represented one hidden layer. The number of inputs depended on the problem in hand. ANN had one output layer.

The list of 15 activation functions [15] (linear, exponential, sigmoid, hyperbolic tangent and others) was used for nodes. To determine the activation function for a given node the integer that corresponded to its binary string was calculated. For example, if a neuron was represented by the binary string “0110”, then the integer was $0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 = 6$ and for this neuron the sixth activation function from the list mentioned above was used.

The total number of weight (NW) coefficients was calculated in the following way:

$$NW = (NI \times ML) + (ML - 1) \times MN + NO, \quad (1)$$

where NI is the number of inputs, ML is the maximum number of hidden layers, MN is the maximum number of neurons on each layer, NO is the number of neurons on the last layer.

Thus, for each classification problem a population of N individuals, which represented N different neural networks, were generated. To be more specific, each individual is a vector of $(MN \times ML \times 4 + NW)$ binary (network's structure) and real-valued (weight coefficients) variables. Weight coefficients represented by the real-valued variables were in

the range $[-1, 1]$. If the node did not exist, then weight coefficients related to it were set to 0.

Two objective functions were defined: the first one was the classification error and the second one was the total number of network's neurons. The idea was to minimize both of these functions, thus to find simple but effective networks.

To solve the above-mentioned multi-objective optimization problem two modifications of the DE [7] algorithm were proposed. These modifications were based on two well-known schemes: Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D) [16] and Non-dominated Sorting Genetic Algorithm (NSGA) [17]. Proposed modifications were called DE+MOEA/D and DE+NSGA respectively [18].

Both proposed algorithms for multi-objective optimization use the Pareto optimality theory, so a set of different networks (non-dominated solutions) were obtained for every classification problem solved. It should be noted that at the end of the optimization process each network's weight coefficients were additionally adjusted by the standard DE algorithm.

Thus, the aforementioned set of found networks was considered as an ensemble. The final decision regarding the class assignment was made in the following way:

- class with the biggest number of “votes” from members was chosen;
- if the number of “votes” was divided into equal parts, then the network with the highest confidence level assigned the class.

So both classification problems described in the previous section were solved by using the DE+NSGA and DE+MOEA/D approaches.

4.2 Fuzzy Rule-Based Classifiers

Currently various algorithms for solving classification problems are being developed. Researchers frequently use classifiers based on fuzzy logic (fuzzy rule-based classifiers) for categorization. There are various works in which this method has been used and it has been established that generally it is efficient and works successfully.

Fuzzy rule-based classifiers (FRBC) can be described as follows. Firstly, let $L = \{l_1, \dots, l_c\}$ be a set of class labels and $x = [x_1, \dots, x_n]^T$ be a vector in R^n describing an object. Each component of vector x expresses the value of a feature, thus a classifier is any mapping $C: R^n \rightarrow L$. A classifier is considered as a black box at the input of which x is submitted and at the output the values of c functions $f_1(x), \dots, f_c(x)$, which express the support for the respective classes, are obtained. The maximum membership rule assigns x to the class with the highest support.

In FRBC the features are associated with linguistic labels. Fuzzy systems are meant to be transparent models implementing logical reasoning, presumably understandable to the end-user of the system. A class of such systems employs if-then rules and an inference mechanism which, ideally,

should correspond to the expert knowledge and decision-making process for a given problem.

Thus solving classification problems, using fuzzy systems, requires two problems to be solved: rule base selection and membership function tuning. These problems can be considered as optimization tasks: the selection of the classifier rule base can be described as an optimization problem with binary variables and the parameter tuning of membership functions as an optimization problem with real-valued variables.

In this study there are three Gaussian membership functions for each feature or attribute of a given input vector with two parameters each: the mean value a and the variance σ . So there are 2 parameters for each function and therefore $6n$ real-valued parameters that have to be tuned.

As a result each data feature or attribute is represented by 2 bits: “00” means that the feature is not used in a given rule, “01” means that for a given feature the 1st membership function is used, “10” means that the feature uses the 2nd membership function and “11” means that the feature uses the 3rd membership function. Let m be the number of rules and consider classification problems with 2 classes (1 bit for class label): each rule base can be presented by a binary string with the length which is equal to $(2n + 1) \times m$.

In this study the meta-heuristic approach COBRA [8] and its modification for solving optimization problems with binary variables were applied for the design of fuzzy rule-based classifiers. Consequently the binary version of COBRA was used for finding the best rule base and the original COBRA was used for adjustment of the membership function parameters for every rule base.

5. Experimental Results

5.1 Experimental Settings

In this study the maximum number of ANN hidden layers (ML) was equal to 5; the maximum number of neurons on each layer (MN) was set to 5. The number of inputs, NI, was equal to 4 for both classification problems. Thus, the number of variables was equal to 225 (100 binary variables and 125 real-valued variables) as for the problem “listened”, so for the problem “liked”.

The population size for both algorithms (DE+NSGA and DE+MOEA/D) N was equal to 50. The maximum number of function evaluations to generate networks’ structures by the mentioned modifications was equal to 100000, while the same amount of function evaluations were also used by the standard DE for the final weight coefficients adjustment.

DE parameters F and CR were set to 0.4 and 0.6 respectively and finally for the DE+MOEA/D modification the number of indices of the nearest neighbours T was equal to $0.2 \times N$.

From the viewpoint of optimization, fuzzy rule-based classifiers for these problems had 90 binary variables for the rule base and 24 real-valued variables for the membership function parameters. For the final parameter adjustment

of membership functions the maximum number of function evaluations was equal to 15000, and the maximum number of rules m was equal to 10. If there were several identical rules then only one of them was left in database.

Additionally several classification methods were tested, including some popular state-of-the-art classification methods implemented in RapidMiner 5.3 software and HEFCA approach. In RapidMiner, the following classification methods were applied: SVM, k -NN, DT and standard ANN.

The k -NN used Euclidian distance between objects, and no weighting was applied. The k parameter was tuned with a grid search within the range [1, 21]. SVM used a linear kernel function; the tolerance for misclassification was tuned with a quadratic grid search within the range [0.001, 1]. The DT method used a gain ratio as the main criterion for the split, the maximal depth was set to 20, the minimal gain to 0.1, the minimal size for the split was 4, and the minimal leaf size was tuned in the range [1, 20]. For ANNs, normalization and shuffling were used, the momentum was set to 0.2, and the learning rate was defined by a grid search within the range [0.001, 1] with a quadratic grid.

The HEFCA method was developed to solve complex classification problems with high class imbalance rates, and creates a fuzzy rule base where the number of rules is defined automatically [13]. The rules could contain one of 15 fuzzy sets for input variables, including division into two, three, four and five triangular terms and the “Don’t Care” condition [13]. The rule base was evolved by specific crossover and mutation operators. The new rules were generated from the instances of the training sample for initialization and mutation operators.

The HEFCA method also included an instance selection mechanism that creates a subsample from the original sample in which the distribution of instances belonging to each class is as uniform as possible. The instances which are classified correctly had a smaller chance of being chosen into the subsample, while those which were misclassified received a larger chance.

5.2 Numerical Results

Each dataset was divided into two sets: train (70% of instances) and test (30% of instances) sets. Additionally, the train and test sets were randomly generated in such a way that both classes were represented in them.

For all the above-mentioned data mining tools, 20 program runs were executed. For the k -NN method, k was equal to 7 for the first classification problem and 5 for the second problem. The HEFCA approach was applied for these classification problems both with the instance selection mechanism and without it.

After 20 program runs of each given algorithm, the average portion of correctly classified instances from test sets (%) was calculated. In addition, the F -score value with parameter $\beta = 1$ was used for evaluating the obtained results for each class respectively. The F -score depends on the

“precision” (pr) and “recall” (rc):

$$F\text{-score} = F_1/F_2, \quad (2)$$

where $F_1 = (\beta_2 + 1) \times pr \times rc$ and $F_2 = \beta_2 \times (pr + rc)$.

The classification “precision” for each class is calculated as the number of correctly classified instances for a given class divided by the number of all instances which the algorithm has assigned for this class. “Recall” is the number of correctly classified instances for a given class divided by the number of instances that should have been in this class.

Results obtained by different algorithms are presented in Table 3 and Table 4 for the first and the second classification problems respectively. It should be noted that in these tables, the best values of the $F\text{-score}$ criteria are demonstrated.

Here is the ensemble obtained for the first classification problem “listened” by the DE+NSGA approach (3 neural networks, each network has 5 hidden layers with 12, 8 and 8 neurons respectively).

Table 3 Results obtained for the first problem.

Algorithm	$F\text{-score}$		Accuracy, %
	“listened”	“didn’t listen”	
$k\text{-NN}$	86.57	30.77	77.50
SVM	85.71	0.00	75.00
DT	85.71	0.00	75.00
ANN (standard)	85.71	0.00	75.00
HEFCA (with instance selection)	74.34	38.30	63.80
HEFCA (without instance selection)	84.06	0.00	72.50
ANN+DE+NSGA	64.28	73.68	75.80
ANN+DE+MOEA/D	93.33	14.84	75.80
FRBC+COBRA	75.00	93.75	80.50

Table 4 Results obtained for the second problem.

Algorithm	$F\text{-score}$		Accuracy, %
	“liked”	“didn’t like”	
$k\text{-NN}$	66.67	63.16	65.00
SVM	48.89	69.33	61.67
DT	62.50	25.00	50.00
ANN (standard)	60.71	65.62	63.33
HEFCA (with instance selection)	58.62	61.29	60.00
HEFCA (without instance selection)	57.14	62.50	60.00
ANN+DE+NSGA	44.44	51.85	72.22
ANN+DE+MOEA/D	59.26	37.04	72.22
FRBC+COBRA	75.00	62.50	70.66

- The first network’s structure: the first layer is (1110 0100), neurons with the 14th and 4th activation functions; the second layer (0010 1001 1011), neurons with the 2nd, 9th and 11th activation functions; the third layer is (1010), neuron with the 10th activation function; the fourth layer is (1111 0100 0110), neurons with the 15th, 4th and 6th activation functions; the fifth layer is (0111 1000 1110), neurons with the 7th, 8th and 14th activation functions.
- The second network’s structure: the first layer is (0100 0100), neurons with the 4th activation function; the second layer (1000), neuron with the 8th; the third layer is (1010), neuron with the 10th activation function; the fourth layer is (0110 0100), neurons with the 6th and 4th activation functions; the fifth layer is (0101 1110), neurons with the 5th and 14th activation functions.
- The third network’s structure: the first layer is (0100 0100), neurons with the 4th activation function; the second layer (1000), neuron with the 8th activation function; the third layer is (1010), neuron with the 10th activation function; the fourth layer is (0111 0100), neurons with the 7th and 4th activation functions; the fifth layer is (0101 1110), neurons with the 5th and 14th activation functions.

And for the same classification problem “listened” only one artificial neural network was found by the DE+MOEA/D approach (5 hidden layers with 15 neurons). It can be described as follows:

- the first layer is (1100 1010 1010 1010 0100), neurons with the 12th, 10th and 4th activation functions;
- the second layer (0011 1110 0110), neurons with the 3rd, 14th and 6th activation functions;
- the third layer is (1011), neuron with the 11th activation function;
- the fourth layer is (1011 1001 0101), neurons with the 11th, 9th and 5th activation functions;
- the fifth layer is (0111 1101 1001), neurons with the 7th, 13th and 9th activation functions.

However, for the second classification problem “liked” the best results were achieved not by ensembles but by single neural networks generated by using the DE+NSGA and DE+MOEA/D approaches.

Here is the artificial neural network obtained for the problem “liked” by the DE+NSGA approach (5 hidden layers with 19 neurons):

- the first layer is (1000 0011 1000), neurons with the 8th and 3rd activation functions;
- the second layer (1111 1101 1101 1111), neurons with the 15th and 13th activation functions;
- the third layer is (0110 1000 1001 0011), neurons with the 16th, 8th, 9th and 3rd activation functions;
- the fourth layer is (1110 0110 0010 1100 1100), neurons with the 13th, 6th, 3rd and 12th activation functions;

Table 5 Example of the rule base for the first problem.

Feat. 1	Feat. 2	Feat. 3	Feat. 4	Assigned class
2	1	2	2	listened
3	1	2	1	didn't listen
3	1	3	1	didn't listen
3	2	2	1	listened
1	1	1	1	didn't listen
DC	1	2	3	didn't listen
2	3	3	1	didn't listen
DC	DC	3	3	listened
3	2	3	1	listened
3	1	3	1	listened

Table 6 Example of the rule base for the second problem.

Feat. 1	Feat. 2	Feat. 3	Feat. 4	Assigned class
DC	1	1	1	didn't like
DC	3	1	3	liked
3	1	DC	1	didn't like
3	2	DC	1	liked
3	3	1	2	liked
DC	DC	3	1	liked
3	2	3	3	liked
DC	2	1	1	didn't like
2	3	1	1	didn't like
2	3	1	3	didn't like

- the fifth layer is (1100 0110 1001), neurons with the 12th, 6th and 9th activation functions.

And the structure of the one artificial neural network found for the problem “liked” by the DE+MOEA/D approach (5 hidden layers with 18 neurons) can be described as follows:

- the first layer is (1010 1110 1010), neurons with the 10th and 12th activation functions;
- the second layer (0100 1110 1010), neurons with the 4th, 14th and 10th activation functions;
- the third layer is (1111 0100), neurons with the 15th and 4th activation functions;
- the fourth layer is (1100 1010 1011 1000 1010), neurons with the 12th, 10th, 11th and 8th activation functions;
- the fifth layer is (1011 1110 0001 0100 1001), neurons with the 11th, 14th, 1st, 4th and 9th activation functions.

Examples of the rule base for the problems “listened” and “liked” obtained during one of the program runs are presented in Table 5 and Table 6 respectively. The presented rule bases are typical for the solved problems. The following denotations are used: DC—feature does not appear in a given rule, 1, 2 or 3—the first, the second or the third membership function for a given feature is used, and the class identifier is given in the last column.

Let us also consider the problem “liked” as an example to demonstrate the interpretability of the obtained results. An instance for this problem was described by a class identifier and four attributes values generated by non-contact vital sensing using Doppler sensors.

Figures 3, 4, 5 and 6 show the membership functions of the rules presented in Table 6 for the features of the second classification problem. Thus, for the first feature it could be observed that the algorithm has placed fuzzy terms close to zero (“low”), in the middle (“average”), and one wide term

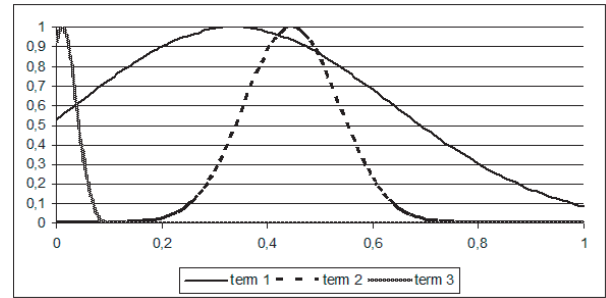


Fig. 3 The membership functions for the Feat.1 (“liked” problem).

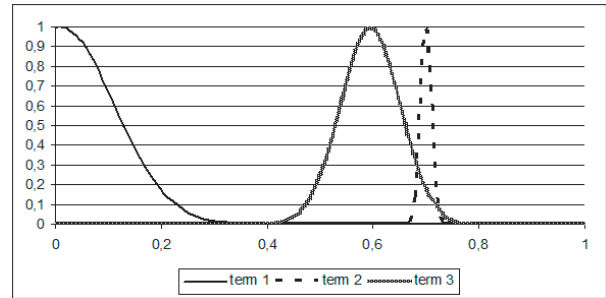


Fig. 4 The membership functions for the Feat.2 (“liked” problem).

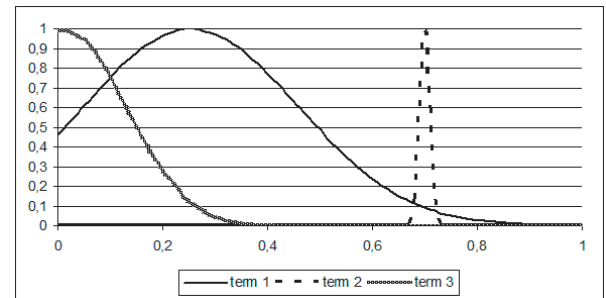


Fig. 5 The membership functions for the Feat.3 (“liked” problem).

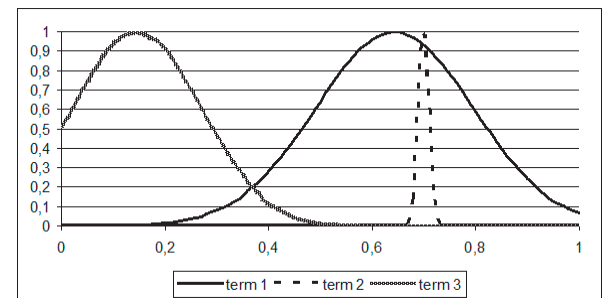


Fig. 6 The membership functions for the Feat.4 (“liked” problem).

covering the whole variable space, which may be interpreted as the “Don’t Care” condition.

For the second feature, the first term again describes low values, the second—high values around 0.7, and the third—average values. Here also there is a gap between terms around 0.4. It is likely that the values in this range do not matter for the final decision.

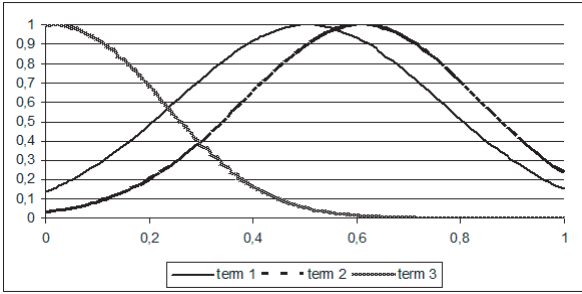


Fig. 7 The membership functions for the Feat.1 (“listened” problem).

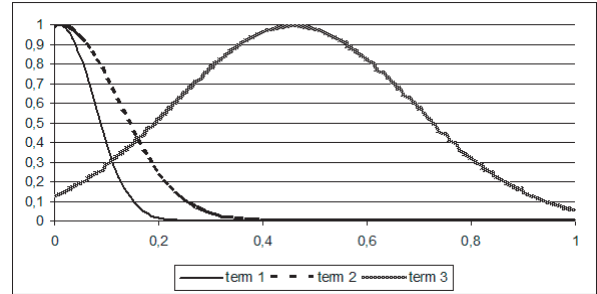


Fig. 9 The membership functions for the Feat.3 (“listened” problem).

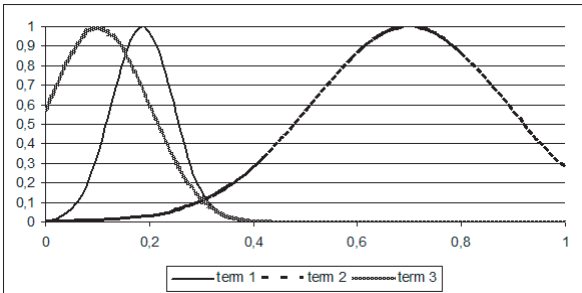


Fig. 8 The membership functions for the Feat.2 (“listened” problem).

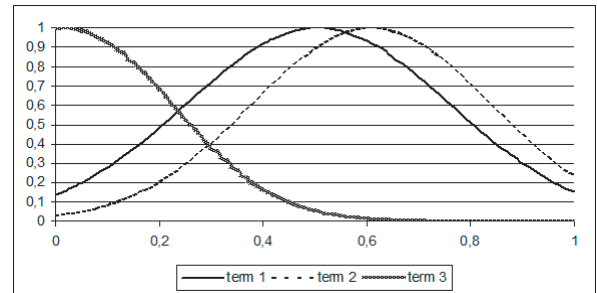


Fig. 10 The membership functions for the Feat.4 (“listened” problem).

The third feature has a similar structure, but here two wide terms are placed close to 0 and 0.3, covering most of the values except very high ones.

The fourth feature is of particular interest, as here we may observe two terms representing “low” and “high” with an overlap close to 0.4. Taking into consideration the fact that the fourth feature is present in all rules in the rule base from Table 6, it may be concluded that it has particular significance for the classification problem.

Furthermore, for the first attribute the rule base contains only the terms 2 and 3, but not term 1, which is a very wide term and probably does not have any descriptive power. It could be noted that the class is correlated with a fourth variable, i.e. for term 1 (“high”) the class is mostly “didn’t like”, while for term 3 (“low”) the “liked” class may be seen in several cases.

In addition, Fig. 7, 8, 9 and 10 demonstrate examples of the membership functions of rules obtained for the first classification problem (“listened”) during one of the program runs.

The experiments have shown that the neural networks generated by the DE+NSGA approach were able to find better results averagely than the DE+MOEA/D technique, besides it worked faster, which is important in case of such kind of problems. Therefore, the DE+NSGA outperformed DE+MOEA/D during experiments. However, overall fuzzy rule-based classifiers demonstrated better results in regards of F-score criteria for each class. Besides they have advantage comparing to neural networks, namely their work can be easily interpreted. Also it should be noted that proposed approaches outperformed other standard methods used in experiments.

The computer processing complexity of the proposed approaches is larger than for the mentioned classical methods like *k*-NN, support vector machines, neural networks or decision trees implemented in RapidMiner system. However, this difference exists only during training procedure. For example, the training of Fuzzy Rule Based Classifiers by COBRA takes around one hour. On the other hand, the application of the trained neural nets and fuzzy classifiers to new data takes very small time (unlike, for example, the *k*-NN method), which is important for real-world operation.

Thus, it was established that the FRBC+COBRA technique is more useful for such problems (with databases formed by non-contact vital sensing using Doppler sensors). Therefore, they can be used for the automated detection of a complex system operator’s condition.

6. Conclusions

In this study, non-contact vital sensing using Doppler sensors was considered as a technique for the statement of the problem related to the detection of a complex system operator’s condition. Non-contact vital sensing was used for data retrieval and the forming of the database for such problems. Namely, in this study two detection problems were defined as classification problems. Besides, one of them can be considered as an opinion mining problem. These problems were solved by various state-of-the-art classification methods as well as ones based on fuzzy logic and neural networks. It was established that automatically generated FRBC demonstrate the best results for detection problems defined as described. Thus, the proposed approach in the problem statement combined with FRBC can be used for

other problems concerning the detection of a complex system operator's condition.

Acknowledgments

This work was supported by the Ministry of Science and Higher Education of the Russian Federation within limits of state contract No. FEFE-2020-0013.

References

- [1] M.S. Mahdavejad, M. Rezvan, M. Barekatin, P. Adibi, P. Barnaghi, and A.P. Sheth, "Machine learning for internet of things data analysis: A survey," *Digital Communications and Networks*, vol.4, no.3, pp.161–175, 2018.
- [2] J.C. Kabugo, S.-L. Jämsä-Jounela, R. Schiemann, and C. Binder, "Industry 4.0 based process data analytics platform: A waste-to-energy plant case study," *International Journal of Electrical Power & Energy Systems*, vol.115, 2020.
- [3] B.J. Crowley-Koch and R.V. Houten, "Automated measurement in applied behavior analysis: A review," *Behav. Intervent.*, vol.28, no.3, pp.225–240, 2013.
- [4] Y. Kamiya, "A simple non-contact vital sensing method using Doppler sensors applicable to multiple targets," *Proc. 39th Annual Int'l Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC 2017)*, pp.2838–2842, 2017.
- [5] T.G. Dietterich, "Machine learning research: Four current directions," *AI Magazine*, vol.18, pp.97–136, 1997.
- [6] P. Angelov, X. Zhou, and F. Klawonn, "Evolving fuzzy rule-based classifiers," *Proc. IEEE Symp. on Computational Intelligence in Image and Signal Processing*, pp.220–225, 2007.
- [7] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol.11, no.4, pp.341–359, 1997.
- [8] Sh. Akhmedova and E. Semenkin, "Co-operation of biology related algorithms," *Proc. IEEE Congress on Evolutionary Computation (CEC 2013)*, pp.2207–2214, 2013.
- [9] Sh. Akhmedova, E. Semenkin, and V. Stanovov, "Fuzzy rule-based classifier design with co-operative bionic algorithm for opinion mining problems," *Proc. 13th Int'l Conf. on Informatics in Control, Automation and Robotics (ICINCO 2016)*, pp.68–74, 2016.
- [10] V. Vapnik and A. Chervonenkis, *Theory of Pattern Recognition*, Nauka, Moscow, 1974.
- [11] N.S. Altman, "An introduction to kernel and nearest-neighbor non-parametric regression," *The American Statistician*, vol.46, no.3, pp.175–185, 1992.
- [12] J.R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol.1, pp.81–106, 1986.
- [13] V. Stanovov, E. Semenkin, and O. Semenkina, "Self-configuring hybrid evolutionary algorithm for fuzzy imbalanced classification with adaptive instance selection," *Journal of Artificial Intelligence and Soft Computing Research*, vol.6, no.3, pp.173–188, 2016.
- [14] Y. Kamiya, "A simple parameter estimation method for periodic signals applicable to vital sensing using Doppler sensors," *SICE Journal of Control, Measurement, and System Integration*, vol.10, no.5, pp.378–384, 2017.
- [15] Sh. Akhmedova and E. Semenkin, "Co-operation of biology related algorithms meta-heuristic in ANN-based classifiers design," *Proc. Congress on Evolutionary Computations of the IEEE World Congress on Computational Intelligence (CEC WCCI 2014)*, pp.867–873, 2014.
- [16] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol.11, no.6, pp.712–731, 2007.
- [17] N. Srinivas and K. Deb, "Multi-objective function optimization using nondominated sorting genetic algorithms," *Evol. Comput.*, vol.2,

no.3, pp.221–248, 1994.

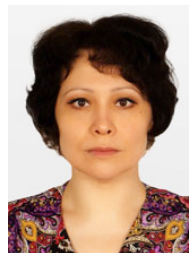
- [18] D.A. Erokhin and Sh.A. Akhmedova, "The development and investigation of the efficiency of the differential evolution algorithm for solving multi-objective optimization problems," *Siberian Journal of Science and Technology*, vol.20, no.2, pp.134–143, 2019.



Shakhnaz Akhmedova received the B.S. degree in Applied Mathematics and Computer Science from Siberian Federal University in 2012, then the M.S. degree in System Analysis and Control from Reshetnev Siberian State Aerospace University in 2014. She received her Ph.D. degree in Computer Science in 2016. She is now associate professor in the Reshetnev Siberian State University of Science and Technology.



Vladimir Stanovov received his B.S. degree in 2012, M.S. degree in 2014 from Siberian State Aerospace University. He received his Ph.D. in 2016 in Computer science. He is now associate professor at the department of higher mathematics, Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, Russia.



Sophia Vishnevskaya received her B.S. degree in Physics from Krasnoyarsk State University in 1991, then the M.S. degree in System Analysis and Control from Reshetnev Siberian State University of Science and Technology in 2017. She received her Ph.D. in 1997 in Computer science. She is now associate professor and chairman of the department of higher mathematics, Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, Russia.



Chiori Miyajima pursued her Master's degree in the Graduate school of Aichi Prefectural University from 2018 to 2019. Currently, she is now with Denso Corp., Japan. Her interests included vital sensing and data analysis.



Yukihiro Kamiya received M.S. and Ph.D. degrees in information engineering and information electronics from Nagoya University in 1994 and 2000, respectively. He joined Koku-sai Denshin Denwa (KDD) Co., Ltd. in 1994. From October 1998 to February 2001, he was a researcher at ATR Adaptive Communications Research Laboratories. He was a Post-doctoral Research Fellow at Ecole Nationale Supérieure d'Ingenieurs de Constructions Aeronautiques (ENSICA), Toulouse, France in 2001, then he

was with PCOM:13 ApS, Aalborg, Denmark as a project manager till 2003. He is currently an associate professor at Aichi Prefectural University. His interests include digital signal processing applicable IoT systems. He is a member of IEEE.