DOI: 10.15439/2015F292
ACSIS, Vol. 5

Supervised Context Classification Methods for an Industrial Machinery

Mateusz Kalisch
Silesian University of Technology
Institute of Fundamentals of Machinery Design
ul. Konarskiego 18a, 44-100 Gliwice, Poland
Email: mateusz.kalisch@polsl.pl

***Abstract*—The paper describes a method of supervised context classification for an industrial machinery. The main objective of this study is to compare single and ensemble classifiers in order to classify groups of contexts which are based on an operating state of the device. The applied research was conducted with the assumption that only classic and well-practised classification methods would be adopted. The comparison study was carried out using real data recorded from an industrial machinery working underground in a mine in Poland. The achieved results confirm the effectiveness of the proposed approach and also show its limitations.**

I. Introduction

THE INCREASING complexity of recent industrial objects causes that fault diagnosis is one of the most important directions of research in the fields of robotics and modern automatic controls [1], [2], [3]. There are a lot of areas where technical systems and processes are required to be safely and reliably operated, such as aircraft, spaceships, automotive or the mining industry. A large majority of the methods implemented for fault detection and isolation are based on simple approaches [4], because these are easy to implement and fast, but the final result can be unsatisfactory because of limitations e.g. too slow system reactions. More complex solutions can be used to achieve better results for more difficult cases but it can be impossible for even an expert to build this kind of the system. One solution is to create a fault detection and isolation system based on the classifiers which are used to prepare and classify datasets, but it is difficult to extract real data fragments connected with a faulty state, and as a consequence, the training data are not good enough for the classifiers. Another solution is context based reasoning [5]. A system based on this approach can be focused on the list of the contexts which are connected with e.g. working conditions of an examined device. Simpler models of the classifiers and more efficient results of the fault detection and isolation process can be considered as advantages of the system based on the context. However there are some problems connected with this approach like when and what kind of context occurs in a specific period of time and how to use context based approach in the fault detection and isolation process.

The rest of the paper is organized as follows. In Section 2 the context based approach with regards to machine learning is described. The next section includes a detailed description of the proposed method. In particular, there are investigations of the classification methods. Section 4 contains a case study description and the more interesting results of the verification experiments. The last section is devoted to concluding remarks and suggestions for future work.

II. Context in Machine Learning

In a classification task, it is possible to distinguish three types of features: primary, contextual and irrelevant [6]. Primary features are useful for classification, without regard to the other features. The irrelevant features are not useful for classification, either when combined with the other features or when they are considered alone. Contextual features cannot be used directly by a classifier, but can be useful when they are combined with other features. The primary features can be also divided into context-sensitive and context-insensitive features. In the case of a machine diagnosis, the context variable could be connected with a number of factors e.g. weather conditions. In another paper [5], the author used contextual variables such as humidity, barometric pressure and external temperature for a gas turbine engine diagnosis. Speech recognition is another example of an area which can use contextual features to improve the efficiency of classification [7]. A speaker's sex, nationality or age may have a strong influence on the relevance of various features but without the primary features the contextual features are useless for these methods. Another type of contextual variable is unknown context which can be identified from data by means of a method based on an evolutionary algorithm [8], [9]. In the final implementation, the context can be acquired directly from the data base or distinguished from the data by the classifier.

The contextual variable is a continuous or discrete variable connected with a specific object. In the case of a discrete contextual variable, the contextual value is equal to one of all the available contextual variants describing this variable. The contextual variant can be obtained from a continuous contextual variable by using the classifier. In the case of an expert system, the context may be connected with text information, where the first part of the message is connected to the contextual variable and the object related to this variable. The second part of the message is connected to the contextual variant. An example of a contextual message could be the contextual variable which refers to wind velocity. The first part of the message connected with the variable might be *Wind*

978-83-60810-66-8/$25.00©2015, IEEE

velocity is and the second part (connected with the variant) might be *too low* or *too high*.

In the literature some of the concepts for the usage of the context with machine learning algorithms are described [10], [11]. Peter Turney in [6], [5] described five strategies which show how context can be used: *Contextual normalization*, *Contextual expansion*, *Contextual classifier selection*, *Contextual classification adjustment* and *Contextual weighting*. The extraction of the context during a reasoning process is one part of the context based method of the classification. The context can be available as an additional variable in the dataset or can be hidden in the data. In the second approach, it is necessary to implement an algorithm which can extract this context from the data in the dataset. The context can occur as single context but also as group of contexts, where each context from the group can also occur independently.

III. Methods of context classification

In the next part of the article, the author describes two methods of classification for groups of contexts. Each context can be described as a binary variable whose value is equal to 0 or 1. All contexts can be connected together and be presented as a decimal value obtained from the binary representation of all contexts (e.g. six binary contexts connected together can be presented as a binary value 010010, which is equal to 16 in decimal notation). It is possible to define a list of all the available combinations of contexts in the group and to create a list of all possible decimal values. This approach (Figure 1) lets us create only one multi-class classifier and the final result of the classification can be decoded to a binary representation to see the state of each binary context in the group.

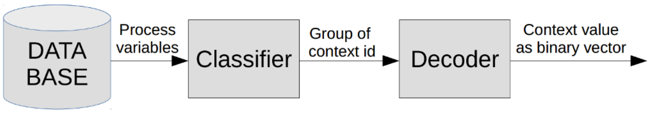

Fig. 1. A scheme for context classification using a single classifier

The advantage of the first method is that the result of the classification can be connected with only one of the known combination of the contexts, because all possible combinations are defined in the training data. The second method (Figure 2) is based on a bank of the binary classifiers where each of them is trained to detect different contexts. When six contexts are available during the reasoning process, it is necessary to implement six binary classifiers in the scheme.

In both methods, each context in the group can be used independently for fault detection and an isolation system. However in the second method, there is a possibility of reaching a result which is not correct. A more detailed description of this problem is presented in the next section of the article.

A. Used classifiers

In this paper the author compares four different classifiers based on various approaches: *Bayesian Network*, *Naive Bayes*, *Decision Tree* and *Artificial Neural Network*. Each of these classifiers returns a label for a chosen class and a degree of belief for all predicted classes. The best result occurs when one of the classes is characterised by the belief level equal to 1 and the rest of them are equal to 0. This gives us a 100% certainty that a new element should be classified as belonging to this particular class. In the next subsections, more precise descriptions of the selected methods are given.

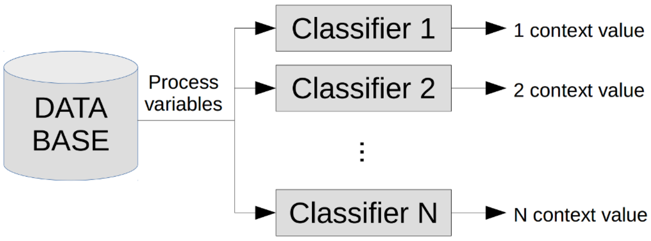

Fig. 2. A scheme of context classification using group of binary classifiers

1) Bayesian Network: Bayesian Network, also called *Belief Network* or *Casual Network*, is a graphical model for representing the conditional independences between a set of random variables. Each node in the network represents a variable [12], [13]. Each connection between the nodes is represented by the Bayesian equation 1 where $P(d_i|V_1, \cdots, V_n)$ is often known as the posterior probability of d_i given $V_1, \cdots, V_n$. $P(V_1, \cdots, V_n|d_i)P(d_i)$ is often referred to as the likelihood of d_i given $V_1, \cdots, V_n$, $P(d_i)$ is the prior or marginal probability of d_i and $P(V_1, \cdots, V_n)$ is a normalizing term.

$$P(d_i|V_1, \cdots, V_n) = \frac{P(V_1, \cdots, V_n|d_i)P(d_i)}{P(V_1, \cdots, V_n)} \tag{1}$$

2) Naive Bayes: Naive Bayes is a simple probabilistic classification method which is based on Bayesian theory. However, the Naive Bayes classifier considers each of the existing features independently. Taking into account this assumption, the Bayesian equation (1) can be transformed to (2), where the denominator of the equation is replaced by a constant C and the conditional probability is calculated by the multiplication.

$$P(d_i|V_1, \cdots, V_n) = C \cdot P(V_1|d_i) \cdot ... \cdot P(V_n|d_i) \cdot P(d_i) \tag{2}$$

The degrees of beliefs for the classification results are equal to the probability values obtained from the Bayesian equation.

3) Decision Tree: A Decision Tree is a classifier based on a tree-like graph created by nodes and the connections between them, where each end node is called a *leaf* and the rest of the nodes have conditions. The result of a decision tree application depends on a chosen leaf. In the algorithm, different split evaluation criteria (e.g. ratio gain in C4.5; information gain in ID3; the Gini impurity measure in CART; etc.) can be used [14], [15]. The confidence levels for the classification results are calculated separately for all leaves of the tree during the learning process. Sometimes, when the learning data is very complex, the results of the decision tree may be uncertain since some of the leaves may be connected to more than one class. The class which is described by more elements then others (in a specific leaf) is chosen as the main class for this leaf. The ratio between the number of elements for available classes is used to calculate the of probability for each class for the leaf.

4) Artificial Neural Network: The classifier is a feedforward neural model in which multiple layers of neurons with nonlinear activation functions allow the network to learn nonlinear or linear relationships between input and output vectors [16]. In this paper, a multiple-layer network is used which consists of three layers including n^1 neurons in the input layer, with n^2 and n^3 neurons in the first and the second hidden layers, respectively. In this case, the neural computation can be represented by the following equation:

$$y = \mathbf{LW}^3\mathbf{f}^2\left(\mathbf{LW}^2\mathbf{f}^1\left(\mathbf{LW}^1\mathbf{u}+\mathbf{b}^1\right)+\mathbf{b}^2\right)+b^3 \qquad (3)$$

where $\mathbf{LW}^{\{1,2,3\}}$ correspond to the weight matrices of the input layer and the first/second hidden layer, $\mathbf{b}^{\{1,2,3\}}$ are vectors of the biases, $\mathbf{u}$ is the input signal, and $\mathbf{f}^{\{1,2\}}$ are nonlinear transform operators consisting of tangensoidal activation functions.

IV. Case Study

A longwall shearer working in a coal mine in Poland is the subject of this study. Longwall mining is underground mining where a long wall of materials is removed in a single slice. The longwall mining method extracts ore along a straight front having a large longitudinal extension. The mining technology involves a longwall shearer, a machine 15 metres long, and weighting 100 tonnes, that has picks attached to two drums which rotate atat a speed of $3040 rev/min$. A longwall face is the mined area from which the materials are extracted. The shearer removes coal by traversing the face at approximately 25 minutes intervals. Traditionally, longwall mining equipment is controlled manually, and the face is aligned in a straight line [17], [18].

A. Data analysis

Available datasets consist of 36 signals including values of the currents, oil and water pressures, temperatures and rotational speeds of the left and right drums of the longwall shearer. Redundant signals were removed from the dataset after a statistical analysis and the final number of signals was reduced to 21. One of the variables was the operational state which contained information about the current state of the longwall shearer. Information for this variable is represented as binary value and each bit is connected with a specific state. The available dataset covering a few days was divided into the smaller datasets connected with single days. In each dataset the author calculated the number of empty rows and the rows containing data. The results of this calculation are presented in Table I.

TABLE I
Relation between the number of rows containing data and the size of the full dataset

Date	Rows containing data	Total number of rows	Ratio
18th October	64783	106385	61%
19th October	72337	102470	71%
20th October	79719	105248	76%
21st October	77200	103746	74%
22nd October	44746	67080	66%

It can be seen that all the datasets contained empty rows (with no values) and the size of these gaps was between 24% and 39% of all data.

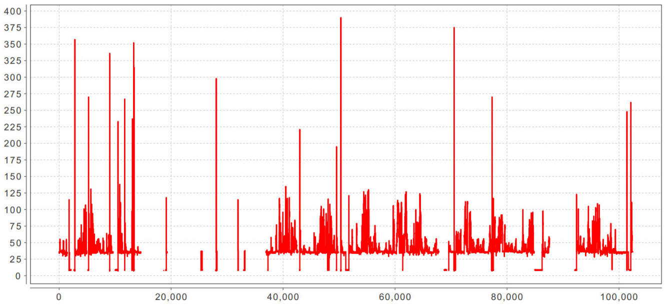

Fig. 3. Average current value of the drive engine

Figure 3 shows that gaps are placed in different fragments of the dataset and the lengths of the fragments of empty data and those filled with data are various. For the dataset from 19th October it is possible to distinguish 28 fragments filled the with the continuous data. Table II presents how many of these fragments filled with data lasted for specific periods of time.

TABLE II
Relation between the number of fragments of useful continuous data and their duration

Duration	Number of fragments
0 - 1 min	2
1 - 10 min	12
10 - 60 min	7
1 - 3 h	7

The author considered only the 7 datasets with the largest number of the samples. They contained in sequence 5031, 5362, 6461, 7351, 7680, 9937 and 10998 samples, so the duration of the series was between one hour and seven minutes and about two hours and thirty minutes. The higher number of the samples can delivered more samples connected with each operational state, providing more opportunities for the classifiers trained on this data to work properly.

B. Operating states of the considered device

In the article operating states of the longwall shearer are considered as contexts described in the first part of the article. There are six operating states represented by binary value (0 or 1):

1) Breakdown,
2) Warning,
3) Operation of drives,
4) Drives turned off,
5) Drive to the left,
6) Drive to the right.

In the study presented in this article the operating states of the longwall shearer were recorded in the dataset by a monitoring system. Sometimes this kind of information is not recorded in the data base and it is necessary to discover it or to add additional information as defined by an expert. The operating state is available in the dataset as a decimal value and it is important to convert it to a binary representation to extract information about each operating state. Table III presents the list of the considered states and the possible combinations of them. The first row shows lists the states, where each state can be equal to 0 or 1. The first column presents a list of all possible combinations of the states represented by decimal values (4, 5, 8, 20, 36, 38) and their binary representations are presented in the central area of the table. The numerical values in the first row of the table correspond to the labels of the operating states in the list presented above.

TABLE III
BINARY REPRESENTATION OF ALL CONSIDERED COMBINATIONS OF THE OPERATING STATES

	1	2	3	4	5	6
4	0	0	1	0	0	0
5	1	0	1	0	0	0
8	0	0	0	1	0	0
20	0	0	1	0	1	0
36	0	0	1	0	0	1
38	0	1	1	0	0	1

Some of the combinations of the states are not correct and they cannot be considered as possible combinations of the states, e.g. it is not possible to set the bit numbers 5 and 6 to 1 at the same time, because bit 5 is connected with the task *Drive to the left* and bit 6 is connected with the task *Drive to the right*. As it is impossible to move the machine in both directions at the same time, but it is possible to stop the machine, then bit 5 and 6 are equal to 0.

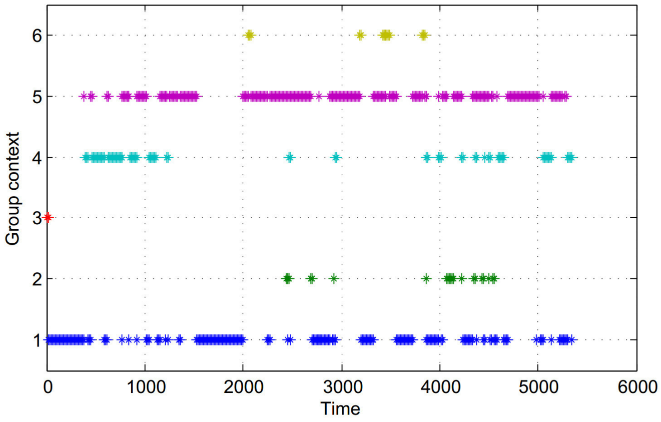

Fig. 4. Occurrence of possible groups of operating states in one dataset

Figure 4 presents the occurrence of the context groups in the fragment of the dataset where each context group id is connected with the following combinations of states:

1) *Operation of drives*,
2) *Breakdown* and *Operation of drives*,
3) *Drives turned off*,
4) *Operation of drives* and *Drive to the left*,
5) *Operation of drives* and *Drive to the right*,
6) *Warning*, *Operation of drives* and *Drive to the right*.

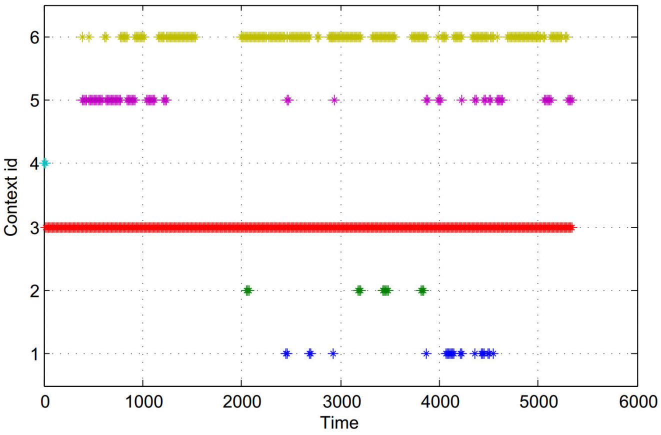

Fig. 5. Occurrence of considered operating states in the fragment of the dataset

Figure 5 shows the places were the specific states occurred in one of the considered datasets. It can be seen that the number of rows of data connected with each state is very various. ID values presented on the Y axis are connected with the list of considered operating states presented at the beginning of this section.

C. Results

The author used seven different parts of datasets, all of them recorded on the 19th of October. The author compared the data using four various classifiers and two methods of classification. Each classifier in the first method (Figure 1) and all classifiers in the second method (Figure 2) were trained on one dataset and tested by the rest of them. Two measurements were considered to evaluate the effectiveness of the classification: accuracy and recall. Accuracy was the basic evaluation method of classification but its result may be not fully reliable in the cases where the data was not well balanced. The second measurement was used to reduce the influence of various numbers of states in the considered datasets.

To keep all results fully consistent, the final result of the second method (Figure 1) was considered correct only if all results of the binary classifiers (each classifier is connected with different state) were correct. Even if only one classifier made a mistake, the final result was treated as incorrect. This solution is fully comparable with first method, where the final result is presented as a group of the contexts.

Table IV shows the accuracy of all classifiers used in the two considered methods. The classifiers are presented in the table by their short names (DT - Decision Tree; NB - Naive Bayes; NN - Neural Network; BN - Bayesian Network). Each column is connected with a different training dataset and the values in the cells of the table show the average value of the test cases. It is clear that the bank of binary classifiers reached a much better result than the single multi-class classifier.

TABLE IV
ACCURACY RESULTS FOR ALL CLASSIFIERS AND METHODS

Single classifier							
Dataset id	1	2	3	4	5	6	7
DT	78.1	42.8	68.9	57.7	63.6	76.6	65.4
NB	30.4	45.3	55.2	48.2	72.0	56.7	37.6
NN	51.4	53.6	70, 2	65.2	75.6	75.5	51.3
BN	51.9	27.6	33.1	37.3	46.0	48.5	51.5

Bank of binary classifiers							
Dataset id	1	2	3	4	5	6	7
DT	89.4	86.8	89.9	90.8	91.0	88.5	89.4
NB	76.9	80.1	87.2	79.8	88.3	85.9	83.3
NN	83.3	86.7	92.6	82.9	92.4	92.1	88.8
BN	77.9	60.0	58.2	75.7	80.0	70.4	78.0

Accuracy is based on a ratio between all correctly classified rows and the number of all rows in the dataset, and it does not take into account the distinctness of each class in the testing dataset. The result based on the accuracy can not be used only as a measurement of the efficiency of the classification because of the unbalanced test datasets. The second measurement used in this test is mean recall. Recall was calculated for each class separately. The recall value is presented as a ratio of the number of all rows of data with a correctly predicted class to all data rows connected with a specific class. The final result is obtained as the average value of all recall values calculated for all available classes.

TABLE V
MEAN RECALL VALUES FOR ALL CLASSIFIERS AND METHODS

Single classifier							
Dataset id	1	2	3	4	5	6	7
DT	68.3	46.6	57.7	51.3	55.3	64.9	61.1
NB	27.0	29.7	35.8	29.2	46.8	35.3	32.1
NN	36.1	31.6	55.1	36.6	47.2	37.9	44.5
BN	35.8	27.4	27.9	35.8	37.9	38.9	43.0

Bank of binary classifiers							
Dataset id	1	2	3	4	5	6	7
DT	82.4	73.2	79.8	84.3	79.4	81.8	82.5
NB	58.1	61.1	63.9	59.0	65.9	64.1	58.2
NN	64.6	65.9	64.2	60.1	73.4	57.2	55.3
BN	65.2	53.1	60.1	66.9	66.8	67.8	66.1

The method based on the bank of binary classifiers reached much better results than the single classifiers. It can be seen that the values for mean recall are worse than the results for accuracy. This proves that the classifiers trained on unbalanced datasets were not evaluated properly by the accuracy measurement. The results for accuracy show that only two classifiers in each column reached the best results interchangeably: Decision Tree and Neural Network. The rest of the classifiers almost always reached worse results than the two mentioned above. The mean recall value shows that the algorithm based on a Decision Tree is able to work more properly with the unbalanced data, and in all columns it reached the best result. The classifier based on a Neural Network had a tendency to ignore classes with smaller numbers of examples.

TABLE VI
ACCURACY OF CLASSIFICATION OF EACH STATE OBTAINED BY DECISION TREE

	1	2	3	4	5	6
1	99.81	100.00	100.00	100.00	98.69	96.91
2	90.81	98.59	99.94	99.94	65.50	73.37
3	91.50	95.75	99.97	99.97	48.45	77.60
4	96.87	98.67	99.95	99.95	52.84	83.17
5	97.04	98.59	99.71	99.71	71.09	79.44
6	99.42	98.16	100.00	100.00	30.14	81.99
7	96.82	98.89	98.63	98.63	65.01	75.22

Table VI shows the accuracy result for the classification of each operating state (columns 1 to 6) for all available datasets (rows 1 to 7) by the bank of the binary classifiers based on a Decision Tree. Each value shows the result of one classifier from the bank, e.g. the value in the third row and second column (89.67) presents the primary accuracy result obtained by the binary classifier (based on a Decision Tree) whose task was to distinguish the operating state called *Warning* (the label of column 2). The label of each row indicates the dataset which was used during the verification test. It can be seen that the accuracy value for the first four states is high but for states 5 and 6 it is lower (except for the first row, because the classifier used in this example was trained by the first dataset).

TABLE VII
RECALL OF THE CLASSIFICATION OF EACH STATE OBTAINED BY A DECISION TREE

	1	2	3	4	5	6
1	97.42	100.00	100.00	100.00	95.98	96.72
2	73.59	99.29	97.92	97.92	76.02	72.10
3	50.40	89.67	99.99	99.99	62.51	77.00
4	56.08	94.92	93.18	93.18	66.33	80.30
5	50.50	83.83	95.15	95.15	77.75	76.95
6	58.11	75.11	100.00	100.00	53.32	82.11
7	48.84	77.29	91.48	91.48	61.15	71.93

The results for a recall of the same situation (Table VII) shows which states are more difficult to isolate and which are not. The classifier reached a high level of efficiency for the 3th and 4th state (*Operation of drives* and *Drives turned off*). The efficiency for the 2nd state (*Warning*) was a little bit lower. The classifier had some problems with the identification of the 5th and 6th states (*Drive to the left* and *Drive to the right*) and the reason for this problems could be the lack of clear information about the direction of movement of the longwall shearer in the dataset. There is no signal which could clearly show the direction of the movement. The classifier reached the worst results for the 1st state (*Breakdown*) because the number of examples connected with this state was very small and the classifier was not able to distinguish this state properly in the test dataset.

V. CONCLUSIONS

It is possible to use different methods of classification to implement basic schemes of context identification. The author was able to increase the efficiency of classification by implementing groups of binary classifiers, instead of using a single multi-class classifier. The final decision of the presented schemes can be used in fault detection and isolation models implemented in an expert system.

The main advantage of the first method (Figure 1) of context classification is its simplicity. It requires only one multi-class classifier, and its result is always connected with a correct combination of states. But the classifier used in this method always reached a worse result than the classifiers used in the second method (Figure 2). Additionally the first scheme needed classified dataset which contained all possible combinations of states and sometimes it is impossible to prepare this kind of training dataset because some of the combinations might not occur in the recorded data. In the second method, it is not necessary to create a dataset with all possible combinations of contexts because each context is classified separately. It is therefore easier to prepare the appropriate training data. The scheme of this method is more complex, but the classification results are significantly more accurate than the results of the first method (Figure 1). Nonetheless, the results of this scheme cannot be fully correct because of the possibility of impossible combinations of contexts as a result of the classification (e.g. the longwall shearer moving in both directions at the same time).

A. Future work

The next step in future research will be connected with other methods of context classification based on ensemble classifiers and meta-classification. Another step is the implementation of the described and future methods inside a fault detection and isolation system, in order to increase the quality of the system in comparison to a solution working without context. It is necessary to see how strong the influence of the context classifier is on the final result of the diagnosis system, because low efficiency of the context classifier could be a reason for high uncertainty levels of the final decision used in the fault detection and isolation system.

Acknowledgment

The research presented in the paper was partially financed by the National Centre of Research and Development (Poland) within the frame of the project titled „Zintegrowany, szkieletowy system wspomagania decyzji dla systemów monitorowania procesów, urządzeń i zagrożeń" (in Polish) carried out in the path B of Applied Research Programme - grant No. PBS2/B9/20/2013. The part of the research was also financed from the statutory funds of the Institute of Fundamentals of Machinery Design.

References

[1] F. Caccavale and L. Villani, *Fault Diagnosis and Fault Tolerance for Mechatronic Systems: Recent Advances*, ser. Springer Tracts in Advanced Robotics. Springer Berlin/Heidelberg, 2003. [Online]. Available: http://dx.doi.org/10.1007/3-540-45737-2

[2] J. M. Kościelny, *Diagnostyka zautomatyzowanych procesów przemysłowych*. Warszawa: Akademicka Oficyna Wydawnicza EXIT, 2001.

[3] R. J. Patton, P. M. Frank, and R. N. Clark, *Issues of Fault Diagnosis for Dynamic Systems*. Springer-Verlag Berlin and Heidelberg, 2000. [Online]. Available: http://dx.doi.org/10.1007/978-1-4471-3644-6

[4] P. F. Odgaard and J. Stoustrup, "Results of a Wind Turbine FDI Competition," in *Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, A. Zaragoza, Ed., Aug. 2012, pp. 102–107. [Online]. Available: http://dx.doi.org/10.3182/20120829-3-mx-2028.00015

[5] P. D. Turney, "Exploiting context when learning to classify," in *Proceedings of the European Conference on Machine Learning*, ser. ECML '93. London, UK, UK: Springer-Verlag, 1993. ISBN 3-540-56602-3 pp. 402–407. [Online]. Available: http://dl.acm.org/citation.cfm?id=645323.649588

[6] P. Turney, "The management of context-sensitive features: A review of strategies," in *Proceedings of the ICML-96 Workshop on Learning in Context-Sensitive Domains*, 1996, pp. 60–65. [Online]. Available: http://dx.doi.org/10.1.1.51.3784

[7] G. Widmer, "Tracking context changes through meta-learning," in *Machine Learning*. Kluwer Academic Publisher, 1996, pp. 259–286. [Online]. Available: http://dx.doi.org/10.1023/A:1007365809034

[8] A. Timofiejczuk, "Context based diagnostics of rotating machinery," in *7th International Conference on Acoustical and Vibratory Surveillance Methods and Diagnostic Techniques (Surveillance 7), 29-30 October 2013, Chartres, France*, 2013, pp. 1–9.

[9] ——, "Identification of diagnostic rules with the application of an evolutionary algorithm," *Maintenance and Reliability*, no. 1, pp. 11–15, 2008.

[10] J. A. Jakubczyc, "Contextual classifier ensembles." in *BIS*, ser. Lecture Notes in Computer Science, W. Abramowicz, Ed., vol. 4439. Springer, 2007, pp. 562–569. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-72035-5_44

[11] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin markov networks." in *CVPR*. IEEE, 2009, pp. 975–982. [Online]. Available: http://dx.doi.org/10.1109/CVPRW.2009.5206590

[12] Z. Ghahramani, "Hidden markov models." River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2002, ch. An Introduction to Hidden Markov Models and Bayesian Networks, pp. 9–42.

[13] R. Daly, Q. Shen, and J. S. Aitken, "Learning bayesian networks: approaches and issues." *Knowledge Eng. Review*, vol. 26, no. 2, pp. 99–157, 2011. [Online]. Available: http://dx.doi.org/10.1017/S0269888910000251

[14] A. Lile, "Analyzing e-learning systems using educational data mining techniques," *Mediterranean Journal of Social Sciences*, vol. 2, no. 3, pp. 403–419, 2011. [Online]. Available: http://dx.doi.org/10.5901/mjss.2011.v2n3p403

[15] F. Akthar and C. Hahne, *RapidMiner 5, Operator Reference*. www.rapid-i.com, 2012.

[16] S. Haykin, *Neural Networks: A Comprehensive Foundation*, ser. 2nd edition. Prentice Hall International, 1999.

[17] A. Oksan Altun, I. Yilmaz, and M. Yildirim, "A short review on the surficial impacts of underground mining." *Scientific Research and Essays*, vol. 21, no. 5, pp. 3206–3212, 2010.

[18] G. A. Einicke, J. C. Ralston, C. O. Hargrave, D. C. Reid, and D. W. Hainsworth, "Longwall mining automation an application of minimum-variance smoothing." *IEEE Control Systems*, vol. 28, no. 6, pp. 28–37, 2008. [Online]. Available: http://dx.doi.org/10.1109/MCS.2008.929281