Proceedings of the 2014 Federated Conference on
Computer Science and Information Systems pp. 1381–1390

DOI: 10.15439/2014F158
ACSIS, Vol. 2

Application of selected classification schemes for fault diagnosis of actuator systems

Mateusz Kalisch, Piotr Przystałka, Anna Timofiejczuk
Silesian University of Technology,
Institute of Fundamentals of Machinery Design,
18a Konarskiego Street,
44-100, Gliwice, Poland,
Telephone: +48 32 237 10 69
Email: {mateusz.kalisch, piotr.przystalka, anna.timofiejczuk}@polsl.pl

Abstract—**The paper presents the application of various classification schemes for actuator fault diagnosis in industrial systems. The main objective of this study is to compare either single or meta-classification strategies that can be successfully used as reasoning means in off-line as well as on-line diagnostic expert systems. The applied research was conducted on the assumption that only classic and well-practised classification methods would be adopted. The comparison study was carried out within the DAMADICS benchmark problem which provides a popular framework for confronting different approaches in the development of fault diagnosis systems.**

I. Introduction

THE INCREASING complexity of recent industrial objects makes the issue of fault diagnosis one of the most important directions of research in modern automatic control and robotics [1], [2], [3]. Technical systems and processes are required to be safely and reliably operated due to the protection of human life and health, the quality of the environment, as well as the economic interests. It is possible to specify numerous areas of interdependence of human and technical means, where safety plays a key role, such as aircraft, spaceship, automotive, power or chemical industry. The above mentioned factors cause that new developments in control theory such as passive and active fault-tolerant control approaches are more often applied in these areas of the industry [4], [5], [6]. A special attention is currently paid on the second type of the advanced control methodologies, where fault diagnosis methods hold a critical importance. The present state of the art in the field of fault diagnosis shows the really need for development of fault diagnosis expert systems. The goal is to elaborate general-purposes systems with multi-domain knowledge representations and multi-inference engines [7], [8], [9]. Generally, the fault diagnosis can be divided into three steps [10]: fault detection, fault isolation and fault identification. Moreover, each of them can be developed by means of model-free (based on data), model-based and knowledge-based approaches [4]. In this paper the first approach, where experimental data are exploited was discussed. In this kind of methods data that represents normal and faulty situations can be obtained from historical databases or from simulators as well as laboratory stands.

There are many types of classifiers available in the literature, as well as different concepts are introduced [11]. Examples are methods based on the similarity between objects in the feature space, probabilistic methods or methods based on black box models. Generally, the classification problems can be divided into two groups including approaches of the machine learning techniques: supervised learning and unsupervised learning. In the paper, the authors concentrated the attention only on methods belonging to the first group.

Currently, the information fusion and meta-classification problems are recognized as the most important directions of the research in the domain of supervised learning. The main idea in this approach is the application of simple classifiers working together to solve a problem with better results than it can by means of single one or more complicated classifiers. There are a lot of different kinds of information fusion methods, but the most popular are majority voting, weighted voting, boosting, AdaBoost [11]. On the other hand, meta-classifiers are very often used for the same reason that its efficiency is higher, than the efficiency of the best single classifier [12].

The current research trends in developing machine learning methods are focused on ideas of improving the general efficiency of different classification and meta-classification methods. The main directions are concentrated on optimization techniques which are used to tune relevant parameters of the classical methods, e.g. with the use of evolutionary and particle swarm algorithms [13], [14], [15], [16], [17], [18]. A number of results included in the works show the benefits of using these methods. In case of a task of fault detection and isolation the key features of the signals in time or frequency domains are most commonly used. Industrial actuators may be characterized by a very high complexity which affects the large number of measuring signals and their features. Therefore, another approach aimed at improving the efficiency of the classifier, and often also shortening the time of its learning, is to remove irrelevant variables [19]. There are various methods that can be used in this procedure, e.g. forward or backward selection methods, as well as elimination methods based on

978-83-60810-58-3/$25.00 © 2014, IEEE

statistical measures. Another group of methods are known as fusion methods such as bagging, boosting, and development of these concepts that is AdaBoost method [20], [21]. These methods are often more effective than simple classifiers but also show some drawbacks. Some advanced concepts were developed to take advantage of positive aspects of classic methods and to eliminate their limitations [22]. There are also attempts to connect together several different methods such as selection of relevant features and usage of boosting into one algorithm [23]. Such approach may lead to the final result that should be better than the results of the methods applied separately.

The paper is organized as follows. In Section I a brief introduction to the problem is given. Section II illustrates the issue of fault diagnosis using the model-free fault detection and isolation methodology. The next section deals with several classification schemes that can be applied to develop fault detection and isolation systems. A case study is included in Section IV. This example shows the comparison research of the classification schemes for creating fault diagnosis system of the benchmark actuator [24] which were elaborated on the basis of the activity of the DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems) Research Training Network funded by the European Commission. The last section is focused on conclusions.

II. Model-free fault detection and isolation

One of the most often used model-free fault detection and isolation methods is presented in Fig. 1. It can be seen, that faults are detected and distinguished using primary and redundant process variables. In this method two separated classifiers must be created. The first classifier uses the subset of process variables ($U' \cup Y'$) as its input and it is dedicated for generating diagnostic signals (S), whereas the second one has the same set of input variables but its task is to calculate a fault signature (F). This classifier is triggered in case when the diagnostic signal indicates a fault scenario.

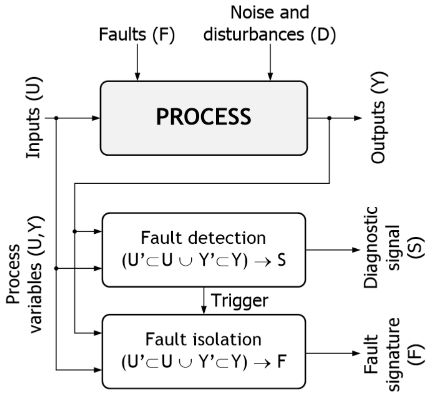


Fig. 1: A diagram of model-free fault detection and isolation

The algorithms corresponding to the diagram presented in Fig. 1 can be designed using different classification methods [10], [4], [25]. Generally, it is possible to apply so-called classical (e.g. decision trees, k-nearest neighbour, naive Bayes, etc.) or soft computing approaches (e.g. neural networks, bayesian networks, fuzzy systems, neuro-fuzzy systems, etc.). The paper deals with the first group of the methods only.

III. Model-free fault diagnosis using different classification schemes

In the next part of the article, model-free fault detection and isolation approaches with the use of different classification schemes were described. As it was mentioned above, these kinds of methods require data (process variables) corresponding to regular (fault-free) and faulty states of the system. In this section, different variants of three basic concepts with a single classifier, meta-classifier and bank of classifiers were applied in order to provide the fault detection and isolation system that is directly based on the process variables.

A. Fault detection

The first concept of the fault detection was presented in Fig. 2 and was elaborated basing on a single classifier, which returns a diagnostic signal corresponding to fault or faultless states of the device. In this method, the process variables were converted by a moving window in order to compute scalar features of the measuring signals. These values were used as input of a single classifier, which generates directly the diagnostic signal. The second detection method was presented in Fig. 3. In this approach a series of two-state classifiers was applied and their task was to determine the degree of the belief for fault detection. The level of belief about faults occurring was a numerical value from 0 to 1. The signal values returned from each classifier were connected to the meta-classifier as its input. The features of the process variables were also connected to the meta-classifier, as the additional input.

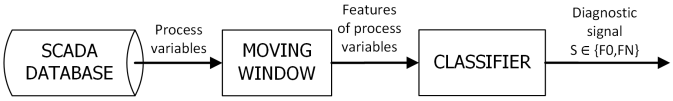


Fig. 2: A scheme of fault detection using the global classifier

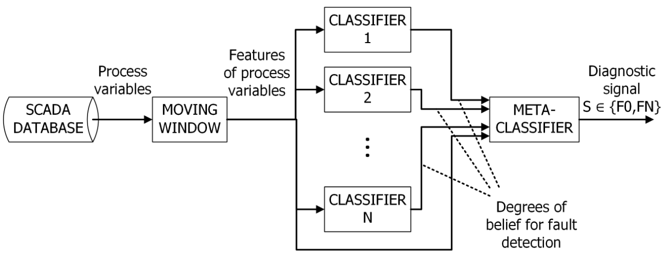


Fig. 3: A scheme of fault detection using the set of various classifiers and meta-classifier

The result of both methods was a diagnostic signal, which indicated fault occurrence. When a classifier or a meta-classifier detects a fault, the second part of the fault diagnosis system was run in order to isolate the faults.

B. Fault isolation

The first method of fault isolation was comparable to the method that was proposed for the fault detection. It was presented in Fig. 4. As one could see it was a single global classifier. Its task was to determine a type of the fault. Similarly to the previous method, in this case the process variables were calculated in the moving window to obtain scalar features of the measuring signals. The preprocessed signals were connected to the input of a global classifier. This classifier returns a fault signature.

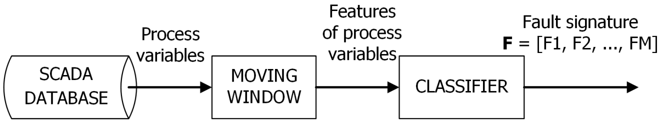


Fig. 4: A scheme of fault isolation using the global classifier

The next fault isolation scheme was presented in Fig. 5. In this approach a set of classifiers of different types was used in order to calculate the degrees of beliefs that were related to fault signatures. These values were given to the input of the meta-classifier and the final decision (fault signature) was obtained.

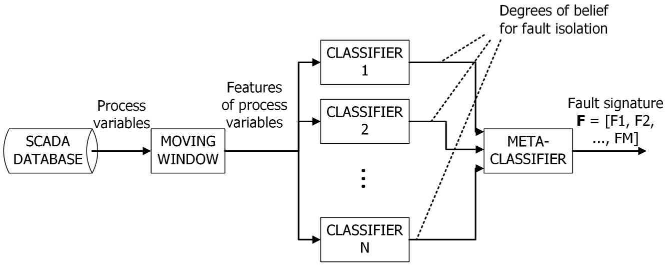


Fig. 5: A scheme of fault isolation using the set of different classifiers and meta-classifier

The last concept of fault isolation was shown in Fig. 6. The main idea was based on a bank of classifiers that were used to calculate degrees of beliefs for specific faults and unknown states of a device. In this case, M single classifiers must be created for M faulty states. Each classifier was dedicated for one state only (it was used for detection one fault solely). In the next step, all available variables (features of the process variables and outputs from base classifiers) are linked to a single dataset. The prepared signals were sent to the input of the meta-classifier which was employed to return the final decision.

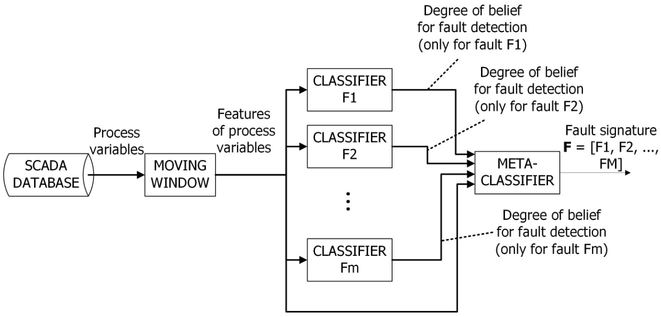

Degree of belief for fault detection (only for fault F1)
Degree of belief for fault detection (only for fault F2)
Degree of belief for fault detection (only for fault Fm)
CLASSIFIER F1
CLASSIFIER F2
CLASSIFIER Fm
Features of process variables
Process variables
SCADA DATABASE
MOVING WINDOW
META-CLASSIFIER
Fault signature
F = [F1, F2, ..., FM]

Fig. 6: A scheme of fault isolation using the set of local classifiers (fault detectors) and meta-classifier

C. Used classifiers

The schemes of fault detection and isolation presented in Sections III.A and III.B can be elaborated with use of basic classification methods. The classification problem is possible to be solved using many known approaches, however, in this research the following methods were applied:

- k - nearest neighbour,
- naive Bayes,
- decision tree,
- rules inductions.

Each of these classifiers returns a label of a chosen class and the degrees of belief for all predicted classes. The best solution is when one of the class is characterised by the belief level equal to 1 and the rest of them are equal to 0. It gives us 100% certainty that a new element should be classified as this particular class. In the next subsections a more precise description of the selected methods was given.

1) k - Nearest neighbour: This is one of the simplest classification techniques. The class label assigned to an example is based on the similarity of this example to one or more prototypes. Typically, the similarity is defined in a geometrical sense using a certain distance. The smaller distance between new object and classified element means the higher similarity between new element and the class represented by the known object. The classifier looks for one nearest neighbour, then it is called 1NN or search for more nearest neighbours, then decision is made by voting. In kNN method, different types of distance measures can be used, e.g. Euclidean (1), Manhattan (2) or Chebyshev (3) distances [26].

$$D(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{1}$$

$$D(x,y) = \sum_{i=1}^{n}|x_i - y_i| \tag{2}$$

$$D(x,y) = \max_{i=1:n}(|x_i - y_i|) \tag{3}$$

The level of confidence about a result of classification depends on a value of k parameter. For k equal to 1, the confidence level for the classification result is always equal to 1. For k greater then 1, when the final result is determined by majority voting, the confidence level depends on the number and types of classified elements in the considered group [27].

2) Naive Bayes [28]: This is a simple probabilistic classification method which is based on bayesian theory. However the naive Bayes classifier considers each of existing features independently.

$$P(d_i|V_1,\cdots,V_n)=\frac{P(V_1,\cdots,V_n|d_i)P(d_i)}{P(V_1,\cdots,V_n)} \qquad (4)$$

Taking into account this assumption, the bayesian equation (4) can be transformed to (5), where the denominator of the equation is replaced by a constant C and the conditional probability is calculated by the multiplication.

$$P(d_i|V_1,\cdots,V_n)=C\cdot P(V_1|d_i)\cdot ...\cdot P(V_n|d_i)\cdot P(d_i) \quad (5)$$

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

3) Decision tree: This is the classifier based on the tree-like graph created by nodes and connections between them, where the end nodes are called *leafs* and the rest of them 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 [29], [27]. The confidence levels about the classification results are calculated separately for all leafs of the tree during the learning process. Sometimes, when learning data is very complex, the results of the decision tree may be uncertain since some of the leafs may be connected to more then one class. The class which is described by more elements then others (in 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 probability for each class in the leaf.

4) Rules induction: The method is based on Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [30] algorithm. The confidence level is calculated in the same way as in the decision tree method.

IV. Verification studies

The proposed schemes of fault detection and isolation were implemented using RapidMiner software. It is an open source software created for solving data mining problems. The verification studies were conducted on data generated using the DAMADICS simulator [31] in order to investigate selected classification schemes. The research problem was actuator fault diagnosis.

A. Benchmark problem

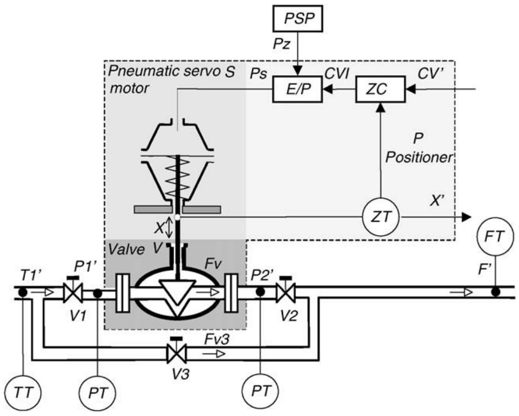


Fig. 7: Structure of benchmark actuator system [24]

DAMADICS was elaborated for scientists and engineers to simplify the process of evaluating and comparing different methods of fault detection and isolation for industrial systems. In the literature there were available several papers where case study results deal with this problem were presented [32], [33], [34]. The numeric model is used to simulate an electro-pneumatic valve (Fig. 7) which is a part of the production line in Lublin sugar factory in Poland. The model was created in MatLAB/Simulink®software and was on a careful study of the physical phenomena that gave the origin to faults in the actuator system. This simulator generated the following signals of the process variables:

- CV - process control external signal,
- P1 - inlet pressures on valve,
- P2 - outlet pressures on valve,
- X - valve plug displacement,
- F - main pipeline flow rate,
- T - liquid temperature,
- f - standard diagnostic signal.

All of these signals are normalized to the range between 0 and 1. The DAMADICS simulator allows to choose only one from nineteen available faults (due to this, only scenarios with single faults were taken into account). A part of them is considered only as incipient faults or as abrupt faults (there are three sizes of abrupt faults: small, medium and big) and some of them as both. In this paper the authors decided to investigate only abrupt faults, such as:

- f1 - valve clogging,
- f2 - valve or valve seat sedimentation,
- f7 - medium evaporation or critical flow,
- f8 - twisted servo-motor stem,
- f10 - servomotor diaphragm perforation,
- f11 - servomotor spring fault,
- f12 - electro-pneumatic transducer fault,
- f13 - stem displacement sensor fault,
- f14 - pressure sensor fault,
- f15 - positioner spring fault,
- f16 - positioner supply pressure drop,
- f17 - unexpected pressure change across valve,
- f18 - fully or partly opened bypass valves,
- f19 - flow rate sensor fault.

The list does not have some faults, because the incipient faults such as f3 - *Valve or valve seat erosion* or f4 - *Increase of valve friction* were not considered. The verification tests were performed basing on the process variables generated by the DAMADICS simulator for fault-free and faulty scenarios.

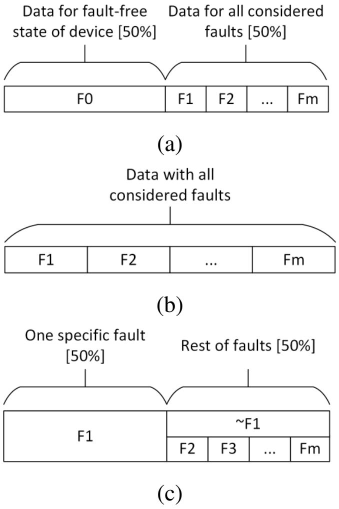

Data for fault-free state of device [50%]
Data for all considered faults [50%]
F0
F1
F2
...
Fm
(a)
Data with all considered faults
F1
F2
...
Fm
(b)
One specific fault [50%]
Rest of faults [50%]
F1
~F1
F2
F3
...
Fm
(c)

Fig. 8: Examples of data groups for specific methods

B. Data preparation

Collections of data for the training, testing and verification of classifiers were prepared in such a way that the results of classifiers were very similar to classifiers working in a real environment. The process of learning (training and testing) and verification for the applied classifiers was described in this section.

1) Data for fault detection classifiers: Data preparation is a very important part of the classifier learning process. The dataset should be divided into two equal parts, where the first part describes correctly working device (fault-free state) and the second part corresponds to situation when fault occurs (Fig. 8a). It was important to divide the prepared data again to two separated groups (learning group and verification group). For the meta-classifier the number of groups was extended to four, because the first and the second groups were used in the learning and verification process for the base classifier. The other two groups were used for a meta-classifier. In this approach, the size of the dataset for each group was equal 10000 samples, where 5000 samples were prepared from data without faults and rest of them contained data with all considered faults.

2) Data for fault isolation classifiers: The data prepared for the first two fault isolation methods (Fig. 2, 3) consists of characteristic process values for all chosen faults (Fig. 8b). The number of elements for each fault was the same for all sets. For learning and verification process four independent groups of data were prepared (like in fault detection methods, two for base classifiers and two for meta-classifier). The dataset for a single fault for one group contains approximately 600 samples, while the full dataset size is about 8000 samples. The third method of fault isolation requires a different type of data. The initial classifier needs data, where a half of the elements describes an actuator device working with one specific fault and the rest of the elements describe the device working with the other faults (Fig. 8c). In this case, a classifier can generate a two-state signal where the first state defines one specific fault and the other ones are correlated with unknown faults. The size of the dataset in this approach is similar as in case for the method of fault detection. The size of the dataset for the considered fault is equal to 5000 samples and it is equal to the rest of a dataset which contains samples corresponding to other faults.

C. Statistical analysis

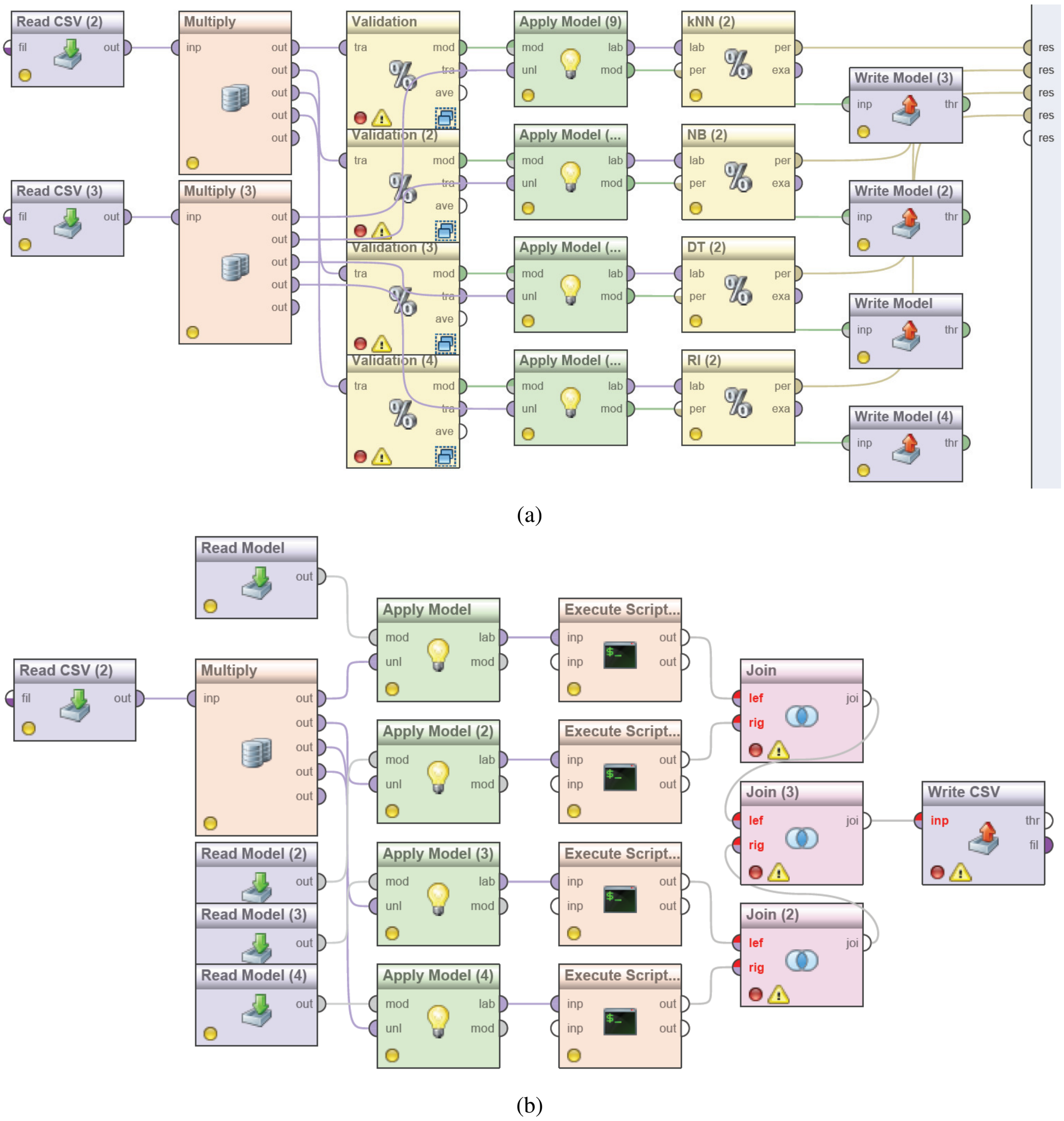
Linear correlation and mutual information analysis were used for choosing relevant process variables and a proper value of the width of a moving window function. In the analysis, all of available process variables were compared between themselves for different device status (e.g. device without faults and with a chosen fault). The results of these tests showed very strong correlations between states F8, F14 and F0. A group of useful process signals was prepared on the basis of results of these tests. Most of the process signals had very difficult character for model-free fault detection and isolation methods. Therefore, the authors decided to apply scalar features of the process variables. Among all available functions in RapidMiner software, four of them were tested: moving average, moving median, maximal value, minimal value.

The scalar features were computed using a moving window of 100 samples width. Such the width value was assumed on the basis of frequency of the harmonic control signal of the valve which was equal to $0,01$ Hz. The authors also studied other values of the window width, however, expected effects in increasing of the efficiencies of the models were not observed. In Fig. 9 exemplary process variables and their features as the time function are plotted. Figures 9a, 9b and 9c show time series of measured signals X, F, P2 for fault F17, whereas figure 9d presents the change in the temperature signal T1 as a result of the fault F7. The sudden change of this signal and its scalar features can be observed at around 800 second when the fault F7 starts affecting the process.

D. Classification schemes implementation

The RapidMiner®software allows to create data mining processes with the use of a visual programming language. This tool gives the opportunity for developing different classification schemes using so-called drag and drop methodology. In this way the classification processes can be viewed as dataflow graphs (Fig. 10).

Fig. 10a presents the scheme of learning and verification processes using four different classifiers. In the first step, the data is red from CSV files by means of *Read CSV* blocks. Next, the learning dataset from the first step is sent to validation blocks, where the learning and evaluation processes for each classifiers is run. The output of these blocks is the ready-to-use classifier, which is applied in *Apply model* block using another dataset read in *Read CSV (2)* block. In the *Performance* block, the process of classifier evaluation is again carried out, but the data are completely different than those in the learning process. *Write Model* blocks are



(a)

(b)

Fig. 10: Selected examples of RapidMiner processes

used in order to write the models into files, because these classifiers are necessary during the process of meta-classifier learning. This scheme can be employed for creating single and meta classifiers. Although, there is a small difference between these two cases in the manner of pre-processing input data. Figure 10b illustrates the process that is applied to generate either learning or verification datasets for meta-classifiers. The data for meta-classifiers is forwarded to their inputs. These samples are totally different from those used at the learning and verification stage of the base classifiers. *Read Model* blocks are employed for reading models of base classifiers. The results of these classifiers are degrees of belief for faultless and faulty states of the actuator. In order to use the data in the next part of the classification process, the pre-processing operations must be done in the *Execute Script* blocks. In the last phase of the process relevant information generated by base classifiers is put together in the form of a coherent dataset which is written in the other file.

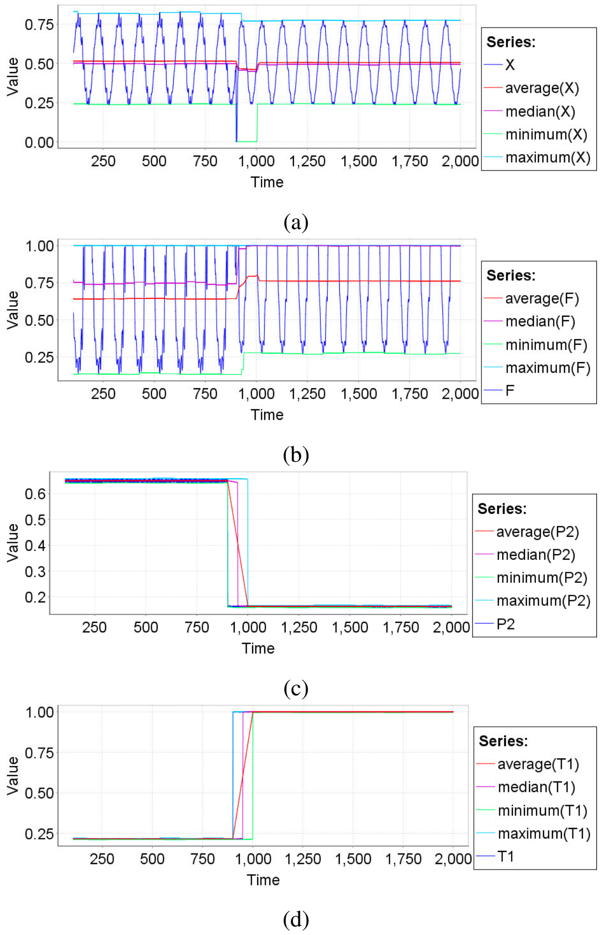
E. Results of verification studies

The learning process for the whole set of applied classifiers was conducted using the X-validation method. At firs the case study tests were carried out to choose the most important parameters for X-validation method. The described schemes of fault detection and isolation were examined with all types of classifiers.

1) Results of fault detection: In the first concept of fault detection (see Fig. 2) four single classifiers were compared. In each table the following notation was assumed: kNN - k-nearest neighbours, NB - Naive Bayes, DT - Decision Tree, RI - Rule Induction. The letter M before each label (kNN, NB, etc.) means meta-version of a classifier, for example, the

TABLE I: Results of fault detection for global classifiers and meta-classifiers

	All	F0	F1	F2	F7	F8	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19
kNN	**0,869**	0,987	**0,752**													
			0,791	1,000	1,000	0,005	0,946	1,000	0,889	1,000	0,003	0,771	0,451	1,000	1,000	1,000
NB	**0,857**	1,000	**0,717**													
			0,667	1,000	1,000	0,000	0,637	1,000	1,000	1,000	0,000	1,000	0,333	1,000	1,000	1,000
DT	**0,864**	0,947	**0,783**													
			0,836	1,000	1,000	0,007	0,991	1,000	1,000	1,000	0,007	1,000	0,429	1,000	1,000	1,000
RI	**0,864**	0,958	**0,772**													
			0,964	1,000	0,787	0,029	1,000	1,000	1,000	1,000	0,030	1,000	0,602	1,000	1,000	1,000
MkNN	**0,867**	0,959	**0,776**													
			1,000	1,000	1,000	0,007	1,000	1,000	1,000	1,000	0,007	1,000	0,450	1,000	1,000	1,000
MNB	**0,876**	0,989	**0,764**													
			1,000	1,000	1,000	0,000	1,000	1,000	1,000	1,000	0,000	1,000	0,334	1,000	1,000	1,000
MDT	**0,871**	0,935	**0,809**													
			1,000	1,000	1,000	0,086	1,000	1,000	1,000	1,000	0,086	1,000	0,667	1,000	1,000	1,000
MRI	**0,869**	0,952	**0,788**													
			1,000	1,000	1,000	0,086	1,000	1,000	1,000	1,000	0,086	1,000	0,425	1,000	1,000	1,000



(a)

(b)

(c)

(d)

Fig. 9: Examples of process variables and their features

label MNB denotes a meta-classifier which is based on a naive Bayes classifier.

Table I shows results of two concepts of fault detection realized by schema presented in Fig. 2 and 3. The effectiveness of the classifiers is given in columns. The column indicated as “All” includes the general efficiency calculated on the basis of the confusion matrix which was generated after the classifier verification process. The next column (F0) includes the efficiency obtained for faultless states. The rest of the columns (F1-F19) show the efficiency of fault detection for all considered faults, separately. Above these columns the general result of the efficiency of fault detection is presented. Rows from 1 to 4 show results for single classifiers, whereas the next four rows show results for considered meta-classifiers.

2) Results of fault isolation: The results of comparison of the first fault isolation method (Fig. 4) are included in Tab. II. The same types of classifiers were used. The column indicated as “All” includes values of the global efficiency for single classifiers of different types. The rest of the columns show information about the efficiency of fault isolation for each scenario.

The second method presented in Fig. 5 use four classifiers as in the previous method but the outputs of these classifiers are connected to a meta-classifier. The results obtained for the meta-classifier are compared in Tab. III. Tab. IV presents the general efficiency of single classifiers (rows from 1 to 4) and meta-classifiers (rows from 5 to 8) for the fault isolation process.

TABLE IV: Comparison between all methods of fault detection for the learning and verification stages

	Learning and testing stage	Verification stage
kNN	0,988	0,844
NB	0,722	0,661
DT	0,176	0,177
RI	0,989	0,808
MkNN	0,990	0,837
MNB	0,860	0,835
MDT	0,890	0,777
MRI	0,982	0,812

The last method of fault isolation (Fig. 6) is based on series of single classifiers, where each classifier is used for detecting a single fault. The first task of the verification process was to choose a single classifier (from four available) for the fault

TABLE II: Results of fault isolation for global classifiers

	All	F1	F2	F7	F8	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19
kNN	**0,844**	**1,000**	**1,000**	**1,000**	**0,516**	**1,000**	**1,000**	**1,000**	**0,876**	0,007	**1,000**	0,829	**1,000**	**1,000**	**1,000**
NB	0,661	0,667	**1,000**	**1,000**	0,347	0,333	**1,000**	0,669	0,153	**0,608**	**1,000**	0,383	**1,000**	**1,000**	**1,000**
DT	0,177	**1,000**	0,000	**1,000**	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
RI	0,808	0,885	**1,000**	**1,000**	0,443	0,947	0,754	**1,000**	0,822	0,003	**1,000**	**0,917**	**1,000**	**1,000**	0,893

TABLE III: Results of fault isolation for meta-classifiers

	All	F1	F2	F7	F8	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19
MkNN	**0,835**	**1,000**	**1,000**	**1,000**	0,672	**1,000**	**1,000**	**1,000**	**1,000**	0,086	**1,000**	**0,708**	**1,000**	**0,668**	**1,000**
MNB	0,802	**1,000**	**1,000**	**1,000**	**0,729**	**1,000**	**1,000**	**1,000**	**1,000**	0,253	**1,000**	0,362	**1,000**	0,416	**1,000**
MDT	0,777	**1,000**	**1,000**	**1,000**	0,175	**1,000**	**1,000**	**1,000**	0,540	**1,000**	**1,000**	0,391	**1,000**	0,369	**1,000**
MRI	0,812	**1,000**	**1,000**	**1,000**	0,621	**1,000**	**1,000**	**1,000**	**1,000**	0,339	**1,000**	0,578	**1,000**	0,326	**1,000**

detection purpose. To solve this problem the authors tested all classifiers for all available faults. The results are presented in Tab. V. The values included in the table present the general efficiency of each classifier. The bolded values are related to the classifiers which were chosen as the basic classifiers for the meta-classifier.

In the next step of the method the meta-classifier is used. Its inputs are connected to the outputs of basic classifiers (the degrees of the belief for single fault detection). The main task of this meta-classifier is to compute the final result. Table VI presents results of different types of classifiers which are presented in the same form as in the second method of fault isolation (Tab. III). In the first column indicated as "All" there are included values of the general efficiency of the meta-classifiers. In the next columns the efficiency values of single fault isolation obtained by means of meta-classifiers are included.

3) Discussion of the results: The comparison of two schemes of the fault detection (Fig. 2 and 3) as well as different types of classifiers (kNN, NB, DT, RI) showed that results which were obtained using these schemes were very similar to each other. Moreover, the comparison between the same classifiers and other methods showed that the meta-classifier was characterized by the highest fault detection efficiency at the expense of the fault-free state. Both methods revealed the some problems with detection of faults F8 and F14, because these faults were very strong correlated with faultless state. The best meta-classifier returned more accurate results then the best single classifier, but the difference between them was too small to conclude that this meta-classifier could be better than the second one. The last three methods concerned fault isolation without taking into account a faultless scenario. In the first method (Fig. 4) the global single classifier was used in order to qualify the current fault class of the device. The results of tested classifiers were different, e.g. the global efficiency of DT was 0.177 and the second one (sorted by their efficiency values) was 0.661. Decision tree was able to recognize fault F7. However, value 1.000 for fault F1 indicated 100% efficiency of detection fault F1, but on the other hand the rest of faults were also recognized by this classifier as fault F1. The next classifier (NB) had better results than a decision tree algorithm. The last two classifiers showed a similar global efficiency. The k-Nearest Neighbour classifier (kNN) provided the best basic efficiency for most of single faults. The exception was fault F16 which was better recognized by the rule induction classifier (RI) and fault F14 which was better detected by the Naive Bayes (NB). The results obtained for the second scheme of fault isolation (Fig. 5), which was based on the meta-classifier were more similar to each other than in the first case (Fig. 4). There was a group of faults which were very easy to isolate by all meta-classifiers (F1, F2, F7, F10, F11, F12, F15, F17, F19). The general level of the efficiency in this method was increased significantly in some cases. But the comparison between the best single classifier and the best meta-classifier showed that in the second case, the global efficiency was worse. The more careful analysis of the results for these two classifiers showed that most of the best meta-classifier results were better than the results of the best single classifier. Some exceptions were faults F16 and F18. In the second case deterioration was very sizeable and it was the main reason of the worse result for this classifier. The third scheme of fault isolation (Fig. 6) was divided into two parts. The first part was dealt with the selection of the basic classifiers, applied to isolate a single fault. After the analysis of the results presented in Table V the authors nominated the classifiers for single fault detection. These classifiers were chosen on the basis of general results. In case more than one classifier had the same efficiency value (more classifiers with the efficiency equal to $1,000$ for the fault at the same time) the authors pointed out a classifier with more stable results in the time domain. The previous analysis showed that this type of classifiers carried with it the increase of the efficiency of the meta-classifier. In the second part different types of meta-classifiers were compared. The final results of this method showed that the difference between the efficiency of specific fault isolation for a single meta-classifier was sizeable. Also the difference for the global efficiency among all met-classifiers was sizeable. The best result was obtained for the decision tree (0.840) and it was comparable with a case of the best efficiency level for the single global classifier (k-nearest neighbours: 0.844) in the first method (Fig. 4). The meta-classifier based on the naive Bayes method demonstrated the smallest efficiency for this approach.

TABLE V: Comparison results of base classifiers for fault isolation of single faults

	F1	F2	F7	F8	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19
kNN	**1,000**	**1,000**	**1,000**	**0,930**	**0,992**	0,999	0,986	**0,981**	0,899	**1,000**	**0,908**	**1,000**	**1,000**	**1,000**
NB	0,643	1,000	1,000	0,904	0,696	0,974	0,894	0,934	0,904	1,000	0,709	1,000	0,934	0,845
DT	1,000	0,987	1,000	0,880	0,889	**0,999**	0,987	0,913	0,743	1,000	0,853	1,000	0,960	0,987
RI	0,947	1,000	1,000	0,879	0,954	0,987	**0,999**	0,949	**0,905**	1,000	0,875	1,000	0,973	1,000

TABLE VI: Results of fault isolation for meta-classifiers with a bank of classifiers for isolating single faults

	All	F1	F2	F7	F8	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19
MkNN	0,824	**1,000**	**1,000**	**1,000**	0,517	0,870	**1,000**	**1,000**	**1,000**	0,010	**1,000**	**0,825**	**1,000**	**0,777**	**1,000**
MNB	0,359	**1,000**	**1,000**	**1,000**	**0,578**	**1,000**	0,472	0,000	0,000	0,000	**1,000**	0,000	**1,000**	0,000	0,000
MDT	**0,840**	**1,000**	**1,000**	**1,000**	0,328	0,889	**1,000**	**1,000**	0,889	**0,847**	0,000	0,459	0,000	**0,777**	**1,000**
MRI	0,748	**1,000**	**1,000**	**1,000**	0,718	0,718	0,914	0,000	**1,000**	0,086	**1,000**	0,595	**1,000**	**0,777**	**1,000**

V. Conclusion

In the paper the application of selected classification schemes for fault diagnosis of the actuator systems was presented. The main purpose of the paper was to compare single and meta-classification strategies that could be successfully used as reasoning approaches in off-line as well as on-line diagnostic expert systems. The research was realized basing on the classical and well-practised classification methods. The comparison study was carried out within the DAMADICS benchmark problem. The classification schemes were implemented in RapidMiner software which is a well-known open source system for data mining and knowledge discovery. The particular results of the fault detection study showed that for simple industrial actuators it is possible to apply simple classification schemes without the necessity of using more advanced methods which are based on meta-classifiers. Significant differences can be observed in case of the results that are related to fault isolation schemes. The best evaluation results obtained from the three classification methods are ranged from 0.835 to 0.844. It should be stated that it is possible to observe some important differences in outcomes obtained using simple classification methods in the first fault isolation scheme (see Fig. 4) and similar results in the second one (see Fig. 5). The third concept (see Fig. 6) leads to the varied results of the classification process. The merits in the case of using meta-classifier (the second method applied according to Fig. 5) can be seen for several faults, especially when compared this to the best single classifier. The last scheme (see Fig. 6) is the most complicated and there is the need to test various classifiers and to have additional learning datasets. Moreover, in this scheme the general efficiency of fault isolation is close to the result achieved by means of the single classifier.

In this study, the authors used a confusion matrix in order to evaluate fault diagnosis systems that were created applying different classification schemes. Nevertheless, this measure can be directly compared with false and true detection/isolation rates proposed by the authors of the DAMADICS simulator [24]. The results of fault detection and isolation using single or meta-classification strategies that were achieved in this study are comparable to even more advanced methods described in the literature [35], [36]. Furthermore, in this study the whole set of potential faults were investigated, whereas in the related papers only selected states were taken into consideration.

Overall, the application of single or meta-classification strategies allows to create effective as well as relatively less-complicated computational fault detection and isolation systems that can be successfully employed for on-line and off-line fault diagnosis of industrial actuators.

Acknowledgement

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.

[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.

[4] J. Korbicz, J. M. Kościelny, Z. Kowalczuk, and Cholewa, W. (Eds.), *Fault diagnosis. Models, artificial intelligence, applications.* Springer Berlin/Heidelberg, 2004. [Online]. Available: http://dx.doi.org/10.1017/S0263574704241133

[5] R. Isermann, "Model-based fault detection and diagnosis - status and applications," *Annual Reviews in Control*, vol. 29, no. 1, pp. 71–85, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.arcontrol.2004.12.002

[6] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroświecki, *Diagnosis and Fault-Tolerant Control.* Springer-Verlag Berlin Heidelberg, 2006.

[7] W. Moczulski, "Inductive acquisition of diagnostic knowledge for states tree with complex structure," *Mech. Syst. Signal Process.*, vol. 15, no. 4, pp. 813–825, 2001. [Online]. Available: http://dx.doi.org/10.1006/mssp.2001.1389

[8] W. Moczulski, *Diagnostyka techniczna. Metody pozyskiwania wiedzy.* Gliwice: Wydawnictwo Politechniki Śląskiej, 2002.

[9] W. Cholewa, "Real-time diagnostic expert systems," *CAMES*, vol. 9, no. 1, pp. 21–40, 2002.

[10] R. Isermann, *Fault-Diagnosis Systems. An Introduction from Fault Detection to Fault Tolerance.* Springer, 2006.

[11] L. Kuncheva, *Combining Pattern Classifier: Methods and Algorithms.* New Jersey: Wiley-Interscience, 2004.

[12] L. Lam, "Classifier combinations: Implementations and theoretical issue," *Lecture Notes in Computer Science*, vol. 1857, pp. 77–86, 2000. [Online]. Available: http://dx.doi.org/10.1007/3-540-45014-9_7

[13] M. Namdari, H. Jazayeri-Rad, and S.-J. Hashemi, "Process fault diagnosis using support vector machines with a genetic algorithm based parameter tuning," *Journal of Automation and Control*, vol. 2, no. 1, pp. 1–7, 2014.

[14] B.-S. Yang, X. Di, and T. Han, "Random forests classifier for machine fault diagnosis," *Journal of Mechanical Science and Technology*, vol. 22, pp. 1716–1725, 2008. [Online]. Available: http://dx.doi.org/10.1007/s12206-008-0603-6

[15] S. Pöyhönen, "Support vector machines in fault diagnostics of electrical motors," Helsinki University of Technology Control Engineering Laboratory, Tech. Rep., 2002.

[16] Q. Wu and Z. Ni, "Car assembly line fault diagnosis based on triangular fuzzy support vector classifier machine and particle swarm optimization," *Expert Systems with Application*, vol. 38, pp. 4727–4733, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2010.08.099

[17] T. Bhadra, S. Bandyopadhyay, and U. Maulik, "Differential evolution based optimization of svm parameters for meta classifier design," *Procedia Technology*, vol. 4, pp. 50–57, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.protcy.2012.05.006

[18] R. Cretulescu, D. Morariu, M. Breazu, and L. Vintan, "Weights space exploration using genetic algorithms for meta-classifier in text document classification," *Studies in Informatics and Control*, vol. 21, no. 2, pp. 147–154, 2012.

[19] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[20] T. Jingyuan, S. Yibing, Z. Longfu, and Z. Wei, "Analog circuit fault diagnosis using adaboost and svm," in *Communications, Circuits and Systems*, 2008, pp. 1184–1187. [Online]. Available: http://dx.doi.org/10.1109/ICCCAS.2008.4657978

[21] P. Yao, Z. Liu, Z. Wang, and S. Bu, "Fault signal classification using adaptive boosting algorithm," *Elektronika ir Elektrotechnika*, vol. 18, no. 8, 2012. [Online]. Available: http://dx.doi.org/10.5755/j01.eee.18.8.2635

[22] M. Woźniak, *Metody fuzji informacji dla komputerowych systemów rozpoznawania*. Oficyna Wydawnicza Politechniki Wroc|awskiej, 2006.

[23] K. Kerdprasop and N. Kerdprasop, "Feature selection and boosting techniques to improve fault detection accuracy in the semiconductor manufacturing process," in *Proceedings of the International MultiConference of Engineers and Computer Scientist*, 2011.

[24] M. Bartyś, R. Patton, M. Syfert, Salvador de las Heras, and J. Quevedo, "Introduction to the DAMADICS actuator FDI benchmark study," *Control Engineering Practice*, vol. 14, no. 6, pp. 577–596, June 2006. [Online]. Available: http://dx.doi.org/10.1016/j.conengprac.2005.06.015

[25] R. Patton, F. Uppal, and C. Lopez-Toribio, "Soft computing approaches to fault diagnosis for dynamic systems: A survey," in *IFAC Symposium SAFEPROCESS*, June 2000, pp. 298–311.

[26] H. B. Kekre, T. K. Sarode, and J. K. Save, "Gender classification of human faces using class based pca," *International Journal of Scientific and Research Publications*, vol. 4, no. 2, 2014.

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

[28] P. Cichosz, *Systemy uczące się*. Warszawa: WNT, 2000.

[29] 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

[30] W. W. Cohen, "Fast effective rule induction," in *Twelfth International Conference on Machine Learning*, 1995.

[31] M. Bartyś and M. Syfert. Using damadics actuator benchmark library (dablib). [Online]. Available: http://diag.mchtr.pw.edu.pl/damadics/

[32] V. Puig, M. Witczak, F. Nejjari, J. Quevedo, and J. Korbicz, "A GMDH neural network-based approach to passive robust fault detection using a constraint satisfaction backward test," *Engineering Applications of Artificial Inteligence*, vol. 20, pp. 886–897, 2007. [Online]. Available: http://dx.doi.org/10.1016/j.engappai.2006.12.005

[33] M. Mrugalski, M. Witczak, and J. Korbicz, "Confidence estimation of the multi-layer perceptron and its application in fault detection systems," *Engineering Applications of Artificial Inteligence*, vol. 21, pp. 895–906, 2008. [Online]. Available: http://dx.doi.org/10.1016/j.engappai.2007.09.008

[34] J. Korbicz and M. Kowal, "Neuro-fuzzy networks and their application to fault detection of dynamical systems," *Engineering Applications of Artificial Inteligence*, vol. 20, pp. 609–617, 2007. [Online]. Available: http://dx.doi.org/10.1016/j.engappai.2006.11.009

[35] J. Calado, J. Sá da Costa, M. Bartyś, and J. Korbicz, "FDI approach to the damadics benchmark problem based on qualitative reasoning coupled with fuzzy neural networks," *Control Engineering Practice*, vol. 14, pp. 685–698, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.conengprac.2005.03.025

[36] F. Previdi and T. Parisini, "Model-free actuator fault detection using a spectral estimation approach: the case of the damadics benchmark problem," *Control Engineering Practice*, vol. 14, pp. 635–644, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.conengprac.2005.04.001