

Actuator Fault Diagnosis Using Single and Meta-Classification Strategies

Mateusz Kalisch^(✉), Piotr Przystalka, and Anna Timofiejczuk

Institute of Fundamentals of Machinery Design, Silesian University of Technology,
18a Konarskiego Street, 44-100 Gliwice, Poland
mateusz.kalisch@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 the diagnostic expert system that is realized within the frame of the DISESOR project. The applied research was conducted on the assumption that classic as well as soft computing 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.

1 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 [7, 24, 32]. 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, for instance in aircraft, spaceship, automotive, power or mining 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 [5, 17, 22]. 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 [9, 28, 36]. Generally, the fault diagnosis can be divided into three steps [18]: 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 [22]. 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. This data is then used to create state classifiers and meta-classifiers.

The main goal of this paper is to compare different classification strategies that can be successfully used as reasoning means in the diagnostic expert system. The development of the diagnostic expert system shell with multi-domain knowledge representations and multi-inference engines is realized within the frame of the DISESOR project. The DISESOR is an acronym of the decision support system designed for fault diagnosis of machinery and other equipment operating in underground mines as well as for monitoring potential threats that can occur in such kind of industry. The DISESOR system can be used for different purposes, e.g. to assess seismic hazard probabilities in the area of the coal mine, to forecast dangerous increase in the methane concentration in the mine shafts, to detect and localize endogenous fires, and also to conduct fault diagnostics of machines working in such environment. This study shows the comparison research of the classification schemes for creating fault diagnosis system of the benchmark actuator [2] which was 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 current paper is a continuation of the research work presented in [20]. The authors taken into account the majority of reviewers' comments and also proposed a new approach for searching proper values of relevant parameters of classifiers used to fault diagnosis. The examined methods are planned to be used for designing the engine of the DISESOR system.

2 Single and Meta-Classification Strategies

There are many types of classifiers available in the literature, as well as different concepts of using them are introduced [25]. Some examples are methods based on the similarity between objects in the feature space, probabilistic methods or methods which are based on black box models. Generally, the classification problems can be divided into two groups including approaches of supervised and unsupervised machine learning techniques. 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 be done 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, and AdaBoost [25]. On the other hand, meta-classifiers are very often used for the same reason that means its efficiency is often higher, than the efficiency of the best single classifier [26].

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 most important investigations can be found for instance in [4, 12, 30, 33, 38, 39]. The main directions presented in these studies 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. A number of results included in the related 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 shortening the time of its learning, is to remove irrelevant variables [14]. 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 statistical measures. Another group of methods stands fusion methods such as bagging, boosting, and the development of these concepts that is AdaBoost method [19, 40]. These methods are often more effective than simple classifiers but also show some drawbacks. The advanced concepts were developed to take merits and positive aspects of classic methods and to eliminate their limitations [37]. There are also attempts to connect together several different methods such as selection of relevant features and usage of boosting into one algorithm [21]. Such approach may lead to the final result that should be better than the results of the methods applied separately.

3 Model-Free Fault Diagnosis Using Different Classification Schemes

The idea of the well practised model-free fault detection and isolation method 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. The proposed method can be viewed as the extension of the most often used model-free fault diagnosis approach, cf. Korbicz et al. (2004), Fig. 1.7 at p. 22 [22]. The novelty in this study depends on that single and also meta-classifiers are automatically tuned in order to obtain the maximum accuracy of fault diagnosis.

Fault detection and isolation algorithms corresponding to the diagram presented in Fig. 1 can be designed using different classification methods [18, 22, 31]. 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 either classic or soft computing methods. In the next part of the article,

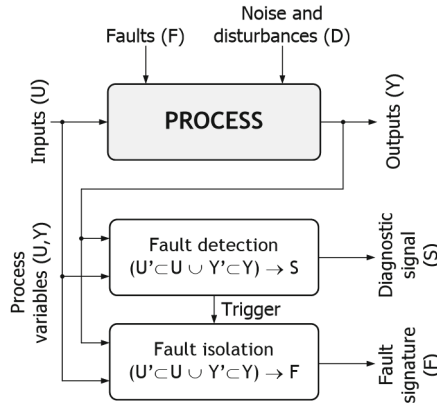


Fig. 1. A diagram of model-free fault detection and isolation

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

3.1 Fault Detection Schemas

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

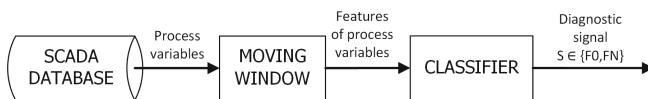


Fig. 2. A scheme of fault detection using the global classifier

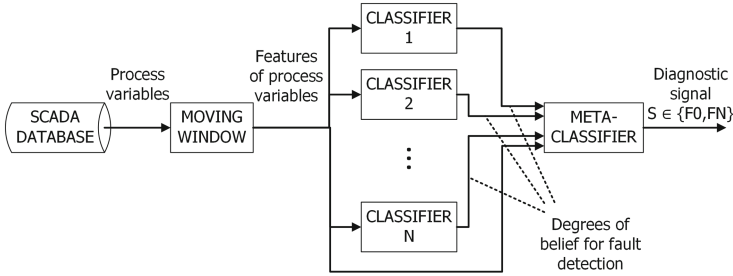


Fig. 3. A scheme of fault detection using the set of various classifiers and meta-classifier

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

3.2 Fault Isolation Schemes

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

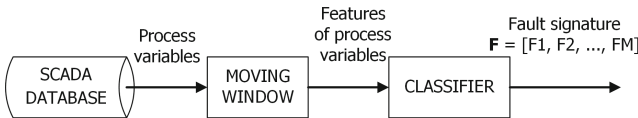


Fig. 4. A scheme of fault isolation using the global classifier

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

The last concept of fault isolation is shown in Fig. 6. The main idea is based on a bank of classifiers that are 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 is dedicated for one state only (it is 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 are sent to the input of the meta-classifier which is employed to return the final decision.

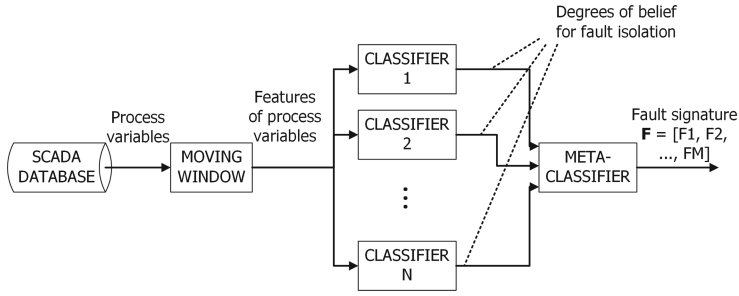


Fig. 5. A scheme of fault isolation using the set of different classifiers and meta-classifier

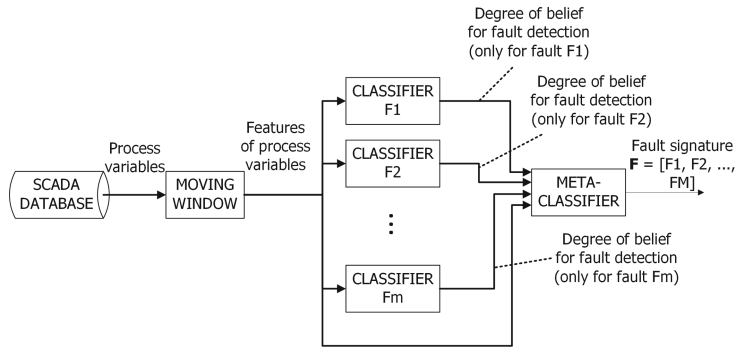


Fig. 6. A scheme of fault isolation using the set of local classifiers (fault detectors) and meta-classifier

The engines of fault detection and isolation schemes presented above can be elaborated with the use of well practised classification methods. The classification problem is possible to be solved using many known approaches, however, in this research the following methods are applied: k-nearest neighbour [1], naive Bayes [10], decision tree [6,27], rules induction [11], neural networks [13,15] and support vector machine [16]. Each of these classifiers returns a label of a chosen class and the degrees of belief for all predicted classes. The best solution is pointed at the moment 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.

4 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 [3] in order to investigate selected classification

schemes. This simulator was elaborated in collaboration of 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 are available several papers where case study results deal with this problem are presented, see e.g. [23,29,35]. 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.

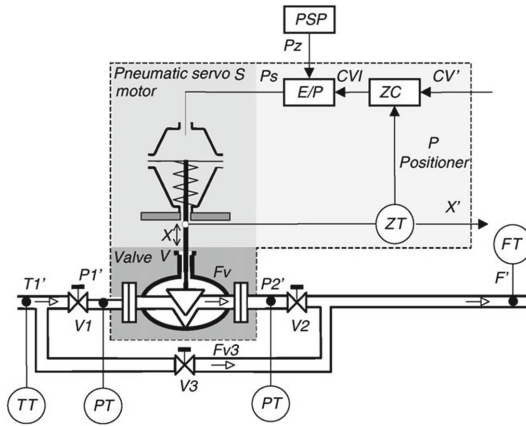


Fig. 7. Structure of benchmark actuator system [2]

The presented model was created and tuned in MATLAB/Simulink® software taking into account the physical phenomena related to the origin of faults in the real actuator system. This simulator was used to generate 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 - fault indicator. All of these signals were normalized to the range between 0 and 1.

The DAMADICS simulator allows to choose only one from nineteen available faults - 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. Moreover, only scenarios with single faults were taken into account. The list does not include 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.

4.1 Data Preparation

Collections of data for the training, test 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. Data preparation is a very important part of the classifier learning process. The dataset should be divided into two et last equal parts, where the first part describes correctly working device (fault-free state) and the second part corresponds to situation when fault occurs. It was important to divide the prepared data again into 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 group 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 to 25200 samples, where 12600 samples were prepared from data without faults and rest of them contained data with all considered faults.

The data prepared for the first two fault isolation methods consists of characteristic process values for all chosen faults. 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 900 samples, while the full dataset size is equal to 12600 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. 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 3900 samples and it is equal to the rest of a dataset which contains samples corresponding to other faults.

4.2 Statistical Analysis and Features Computation

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 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. The authors employed a few well known features often used in fault detection and isolation processes such as average, maximum and minimum values, standard deviation, root mean square, shape factor (a factor determines the shape of the periodic signal), kurtosis (the measure for evaluating the shape of the probability distribution), energy, skewness (the measure of the asymmetry of the probability distribution) and entropy. 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.

4.3 Results of Verification Studies

The learning process for the whole set of applied classifiers was conducted using the X-validation method. To show the efficiency of a classification process, the authors used three measures corresponding to accuracy, sensitivity and precision. The accuracy presents the proportion of correct guesses which can be used directly as efficiency of classification because all datasets were very good balanced (the number of samples for each class was equal). The sensitivity corresponds to the proportion of samples in actual class which are correctly identified. The precision denotes the probability that an example assigned to specific class should be really connected with this class. The sensitivity and precision were calculated for each class, whereas the accuracy were calculated as one value for the whole confusion matrix. In order to have study results more clearly in each table within this section the following notation was assumed: kNN - k-nearest neighbours, NB - Naive Bayes, DT - Decision Tree, RI - Rule Induction, NN - Neural Net, SVM - Support Vector Machine. The prefix letter M placed before each label of the classifier means the meta-version of this classifier, for example, the label MNB denotes the meta-classifier which is based on a naive Bayes classifier.

Moreover, the authors decided to apply the parameter optimization operator included in Rapidminer software in order to tune behavioural parameters of the investigated classifiers. The described schemes of fault detection and isolation were examined taking into consideration all types of classifiers. The optimization operator for each variant of a classifier was based on the evolutionary computation. The standard evolutionary algorithm was used [13]. The maximal number of generations was set to 30 and the population size was equal to 5. The tournament method was used to select parents for creating new generations (the tournament fraction was equal to 0.25). Reproduction was realized based on Gaussian mutation function and crossover operator. The crossover probability was equal to 0.9. The optimization operator maximized accuracy value of chosen classifier. In the case of Naive Bayes classifier the optimize selection operator was only applied to choose the most relevant attributes. The features of this algorithm were similar to the previous one. The k-nearest neighbours classifier was optimized by searching a proper value of the parameter k . In case of decision tree following values of parameters were chosen: the minimal leaf size, the maximal

depth of tree, the minimal gain which affects the ability of the splitting of leaf as well as the confidence used for the pessimistic error calculation of pruning. The rule induction was optimized in the context of the sample ratio that specifies the sample ratio of training data used for growing and pruning, the pureness corresponds to the desired pureness and the minimal prune benefit. In the case of a neural net classifier the number of layers and the number of neurons in each layer were tested and chosen by the authors based on heuristics known from the literature. The other parameters of neural classifiers were optimized by the operator: the number of training cycles, the learning rate determines how much the weights are adjusted at each step, the momentum coefficient smooths optimization directions and the error epsilon as training error threshold. The SVM based classifier has many behavioural parameters and therefore the authors decided to use the methodology described in [16] and focused their attention only on γ and C parameters of the radial base function. The more detailed description of relevant parameters of classifiers and the optimization operator can be found in [1]. The minimum and maximum values of the optimized parameters were defined on the basis of the information contained in the literature.

Results of Fault Detection. In the first concept of fault detection (Fig. 2) six single classifiers were compared. Table 1 shows the accuracy and sensitivity of two concepts of fault detection realized relating to the schemes presented in Figs. 2 and 3. The sensitivity values of classifiers are 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 sensitivity obtained for faultless states. The rest of the columns (F1–F19) show the sensitivity of fault detection for all considered faults, separately. Above these columns the general result of the sensitivity of fault detection is presented. Rows from 1 to 6 show results for single classifiers, whereas the next six rows show results for considered meta-classifiers. The second table (Table 2) shows the precision of fault detection for each classifier and class (faultless state and failure state). All precision values are less than 1,00 which means that there always exist some samples which are assigned to another state than it should.

In first fault detection scheme the accuracy of most used classifiers is high (above 0,93). The accuracy of neural net and naive bayes based classifiers is a little bit lower than other ones but it is still in the acceptable range. The sensitivity of faultless state detection is very close to 1,00, which means that almost all samples corresponding to faultless state were classified correctly to this class. It is easy to distinguish faults which are low-correlated with faultless state because the sensitivity in this case is equal to 1,00. Faults highly-correlated with faultless state have very low sensitivity value, sometimes even equal to zero. The precision values presented in the Table 2 shows proportion between the number of samples which belong to this class and the number of all samples assigned by classifier to the specific class. In the second scheme of fault detection meta-classifiers were used. All six classifiers trained in the previous process based on the first scheme

Table 1. The accuracy and sensitivity 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	0,933	0,978	0,887													
			0,93	1,00	1,00	0,67	0,84	0,84	0,67	1,00	0,67	1,00	0,78	1,00	1,00	0,99
NB	0,878	0,992	0,764													
			1,00	1,00	1,00	0,00	0,65	1,00	0,71	1,00	0,00	1,00	0,33	1,00	1,00	1,00
DT	0,960	0,976	0,946													
			1,00	1,00	1,00	0,65	0,98	1,00	1,00	1,00	0,67	1,00	0,94	1,00	1,00	1,00
RI	0,957	0,973	0,939													
			1,00	1,00	1,00	0,67	0,95	1,00	0,98	1,00	0,72	1,00	0,81	1,00	1,00	1,00
NN	0,829	0,977	0,681													
			0,75	1,00	1,00	0,24	0,49	0,17	0,25	1,00	0,24	1,00	0,50	1,00	1,00	0,89
SVM	0,938	0,982	0,894													
			0,96	1,00	1,00	0,67	0,86	0,95	0,67	1,00	0,67	1,00	0,78	1,00	1,00	0,96
MkNN	0,862	0,931	0,793													
			1,00	1,00	1,00	0,01	0,84	1,00	0,92	1,00	0,01	1,00	0,34	1,00	1,00	1,00
MNB	0,889	0,980	0,799													
			1,00	1,00	1,00	0,00	0,87	1,00	0,99	1,00	0,00	1,00	0,33	1,00	1,00	1,00
MDT	0,883	0,971	0,794													
			1,00	1,00	1,00	0,00	0,83	1,00	0,95	1,00	0,00	1,00	0,33	1,00	1,00	1,00
MRI	0,888	0,974	0,803													
			1,00	1,00	1,00	0,00	0,85	1,00	1,00	1,00	0,00	1,00	0,39	1,00	1,00	1,00
MNN	0,875	0,926	0,824													
			1,00	1,00	1,00	0,15	0,86	1,00	0,93	1,00	0,15	1,00	0,45	1,00	1,00	1,00
MSVM	0,773	0,959	0,587													
			0,69	0,89	0,75	0,00	0,57	0,87	0,63	0,39	0,00	0,88	0,23	0,87	0,69	0,76

(Fig. 3) were used to calculate degrees of belief of fault detection. The obtained data were merged with base classifier’s input data. The final results of classification for the second scheme of fault detection is presented in Tables 1 and 2. In most cases (except classifier based on neural net and naive Bayes) the general accuracy, sensitivity and precision of classification in the second scheme are lower or similar to analogous measures in the first scheme. The meta-classifier based on neural net provided the higher accuracy, sensitivity and precision than neural net used as a simple classifier. It means that additional variables (degrees of beliefs of base classifiers) can be successfully applied to improve the efficiency of classification.

Table 2. The precision of fault detection for global classifiers and meta-classifiers

	kNN	NB	DT	RI	NN	SVM	MkNN	MNB	MDT	MRI	MNN	MSVM
F0	0,90	0,81	0,95	0,94	0,75	0,90	0,82	0,83	0,83	0,83	0,84	0,70
F1–F19	0,98	0,99	0,97	0,97	0,97	0,98	0,92	0,98	0,97	0,97	0,92	0,93

Results of Fault Isolation. The last three methods concerned fault isolation without taking into account a faultless scenario. The results of comparison of the first fault isolation method (Fig. 4) are included in Tables 3 and 4. The column indicated as “All” in Table 3 includes values of the accuracy for single classifiers of different types. The rest of the columns show information deals with the sensitivity of fault isolation for each scenario.

Table 3. Accuracy and sensitivity of fault isolation for global classifiers

	All	F1	F2	F7	F8	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19
kNN	0,73	0,84	1,00	1,00	0,35	0,76	0,72	0,51	0,81	0,05	1,00	0,48	1,00	1,00	0,67
NB	0,77	0,67	1,00	1,00	0,58	0,63	1,00	1,00	0,58	0,40	1,00	0,28	1,00	0,91	0,67
DT	0,07	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RI	0,78	1,00	1,00	1,00	0,59	0,67	1,00	0,98	0,61	0,00	1,00	0,73	1,00	0,63	0,67
NN	0,77	1,00	1,00	1,00	0,00	0,62	0,96	1,00	0,81	0,00	1,00	0,65	1,00	1,00	0,67
SVM	0,67	0,48	0,96	1,00	0,09	0,78	0,95	0,79	0,64	0,00	0,98	0,50	0,91	0,63	0,67

Table 4 presents the precision of classification for each class and classifier. The results of tested classifiers (Table 3) were varied, e.g. the accuracy of decision tree was 0.07 and the second one (sorted by their accuracy values) was 0.67. The sensitivity of decision tree for fault 1 is equal to 1,00 but the precision is equal to 0,07. It means that a classifier is able to classify correctly samples related to fault 1 but unfortunately the rest of samples connected with other faults are also classified as fault 1. This kind of classifier is not useful. The accuracy of the next group of classifiers are better and rule induction based ones reach the best result equal to 0,78. In the sensitivity table (Table 2) all classifiers (except decision tree) reach very similar results for most of the classes. The sensitivity value is more varied for classes which are not easily distinguishable e.g. classification sensitivity for fault 16 is the range from 0,00 to 0,73 and each classifier reach a different value. The precision values (Table 4) are correlated with the sensitivity value. For faults which are easy to classify the precision is equal or close to 1,00, for more difficult faults to recognize the precision values are varied.

Table 4. The precision of fault isolation for global classifiers

	F1	F2	F7	F8	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19
kNN	0,99	1,00	1,00	0,27	0,61	0,50	0,47	0,99	0,15	1,00	0,61	0,75	0,84	1,00
NB	1,00	1,00	1,00	0,34	0,69	0,86	0,97	0,87	0,35	1,00	0,73	0,75	0,68	1,00
DT	0,07	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RI	0,75	1,00	1,00	0,47	1,00	0,88	0,95	0,58	0,00	1,00	0,37	1,00	0,74	1,00
NN	1,00	1,00	1,00	0,05	0,94	1,00	0,57	0,71	0,00	1,00	0,26	1,00	0,84	0,99
SVM	0,99	1,00	0,34	0,39	0,72	0,92	0,72	0,72	0,04	1,00	0,26	1,00	0,99	0,95

The second method presented in Fig. 5 uses six 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 Tables 5 and 6. 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). The accuracy of classification (Table 5) in the second scheme is much better than in the first approach of fault isolation. Meta-classifiers used only degrees of beliefs from base classifiers and they were able to significantly increase the general efficiency. The most of precision values (Table 5) for each class are very similar. The similarity between sensitivity and precision tables (Table 6) is very high but there is one exception. The sensitivity of fault 12 is less than 1,00 but the precision for almost all classifiers is equal to 1,00. It means that not all samples connected with fault 12 are recognized but one can be sure that all recognized samples are really connected with this fault.

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 six available) for the fault detection purpose. To solve this problem the authors tested all classifiers for all available faults. The results are presented in Table 7. The values included in the table present the general accuracy of each classifier. The bold values are related to the classifiers which were chosen as the basic classifiers for the meta-classifier.

Table 5. The accuracy and sensitivity of fault isolation for meta-classifiers

	All	F1	F2	F7	F8	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19
MkNN	0,86	1,00	1,00	1,00	0,63	0,98	0,99	0,88	0,98	0,11	1,00	0,53	1,00	1,00	1,00
MNB	0,84	0,97	1,00	1,00	0,14	0,91	0,99	0,78	0,97	0,65	1,00	0,49	1,00	0,91	1,00
MDT	0,84	0,99	1,00	1,00	0,00	0,84	0,99	0,74	0,97	0,84	1,00	0,43	1,00	1,00	1,00
MRI	0,85	0,99	1,00	1,00	0,70	0,99	1,00	0,74	0,98	0,14	1,00	0,40	1,00	1,00	1,00
MNN	0,87	1,00	1,00	1,00	1,00	0,95	1,00	0,88	0,97	0,00	1,00	0,41	1,00	1,00	1,00
MSVM	0,78	0,94	1,00	1,00	0,00	0,89	0,99	0,81	0,93	0,00	1,00	0,99	1,00	0,38	1,00

Table 6. The precision of fault isolation for meta-classifiers

	F1	F2	F7	F8	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19
MkNN	1,00	1,00	1,00	0,40	0,94	0,93	0,74	1,00	0,33	1,00	0,66	1,00	0,98	1,00
MNB	0,90	1,00	1,00	0,36	0,98	1,00	1,00	0,91	0,38	1,00	0,44	1,00	0,97	1,00
MDT	0,99	1,00	1,00	0,00	1,00	0,86	1,00	1,00	0,37	1,00	0,46	0,99	0,97	0,99
MRI	1,00	1,00	1,00	0,39	0,68	0,99	1,00	1,00	0,28	1,00	0,79	0,99	0,98	1,00
MNN	1,00	0,99	1,00	0,38	0,99	0,90	1,00	1,00	0,00	1,00	0,99	1,00	0,97	1,00
MSVM	1,00	1,00	1,00	0,00	0,99	1,00	1,00	1,00	0,00	1,00	0,25	1,00	0,95	1,00

Table 7. The 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	0,92	1,00	1,00	0,72	0,80	0,84	0,80	0,98	0,72	1,00	0,74	1,00	0,98	0,98
NB	0,64	1,00	1,00	0,92	0,71	0,99	0,89	0,94	0,90	1,00	0,72	1,00	0,96	1,00
DT	1,00	1,00	1,00	0,77	0,59	1,00	0,95	0,92	0,79	1,00	0,63	0,99	0,91	0,81
RI	0,93	1,00	1,00	0,87	0,62	1,00	0,92	0,94	0,85	1,00	0,56	0,99	0,94	0,89
NN	0,94	0,99	1,00	0,81	0,85	0,99	0,89	0,97	0,81	1,00	0,80	1,00	0,98	0,71
SVM	0,97	0,99	0,98	0,79	0,88	0,92	0,86	0,92	0,99	0,80	0,80	0,99	0,96	0,97

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 8 presents the sensitivity of different types of classifiers which are presented in the same form as in the second method of fault isolation (Table 5). In the first column indicated as “All” there are included values of the accuracy of the meta-classifiers. In the next columns the sensitivity values of single fault isolation obtained by means of meta-classifiers are included.

Table 8. The accuracy and sensitivity 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	0,80	1,00	1,00	1,00	0,28	0,74	0,86	0,94	0,81	0,21	1,00	0,42	1,00	0,99	0,98
MNB	0,76	1,00	1,00	1,00	0,46	0,55	0,93	0,86	0,81	0,18	1,00	0,33	1,00	1,00	0,55
MDT	0,78	1,00	1,00	1,00	0,00	0,60	0,79	0,96	0,81	0,20	1,00	0,86	1,00	1,00	0,65
MRI	0,78	1,00	1,00	1,00	0,32	0,94	0,83	0,79	0,74	0,14	1,00	0,53	1,00	0,96	0,67
MNN	0,79	1,00	1,00	1,00	0,85	0,56	0,85	0,96	0,81	0,00	1,00	0,33	0,98	1,00	0,66
MSVM	0,77	0,99	1,00	1,00	0,04	0,88	0,85	0,96	0,81	0,21	1,00	0,48	0,98	1,00	0,62

Table 9 includes the precision of fault isolation based on the third scheme (Fig. 6). This 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 7 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 accuracy and sensitivity of meta-classifiers are presented in Table 8. The accuracy of this method is a little bit higher than in the first scheme of fault isolation (Fig. 4) but the sensitivity (Table 8) and precision (Table 9) for the third scheme are more stable and similar for each class.

Table 9. The precision of fault isolation for meta-classifiers with a bank of classifiers for isolating single faults

	F1	F2	F7	F8	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19
MkNN	1,00	1,00	1,00	0,35	0,47	0,95	0,72	0,96	0,36	1,00	0,57	1,00	0,84	0,91
MNB	1,00	1,00	1,00	0,32	0,44	0,69	0,70	0,99	0,30	1,00	0,84	1,00	0,84	0,75
MDT	0,99	1,00	1,00	0,00	0,82	0,70	0,72	0,95	0,40	1,00	0,34	1,00	0,84	0,90
MRI	1,00	1,00	1,00	0,38	0,55	0,70	0,94	0,96	0,36	1,00	0,41	1,00	0,81	0,89
MNN	1,00	1,00	0,75	0,31	0,84	0,96	0,72	1,00	0,00	1,00	0,99	1,00	0,84	0,86
MSVM	1,00	1,00	1,00	0,34	0,34	0,93	0,72	0,98	0,40	1,00	0,56	1,00	0,84	0,91

Five of the six single classifiers used in the first fault isolation process reached results between 0,67 and 0,78. One classifier is useless (decision tree) and its accuracy is equal to 0,07. Meta-classifiers were able to significantly increase the accuracy of fault isolation. In this case the accuracy values of all classifiers are between 0,78 and 0,87. The sensitivity and precision values are also more stable and similar in each class. The last approach based on the third scheme of fault isolation (Fig. 6) shows also small improvement compared to the first scheme (see Fig. 4). Moreover, in this scheme the general efficiency of fault isolation is close to the result achieved by means of the single classifier. Generally, meta-classifiers in fault-isolation schemes are more stable and their results are similar to each other. In case of the second and third schema it is possible to use classifier based on decision tree so data prepared by base classifiers are simpler and more varied.

In this study, the authors used a confusion matrix in order to evaluate fault diagnosis systems that were created applying different classification schemes. Nevertheless, the accuracy, sensitivity and precision obtained from a confusion matrix can be directly compared with false and true detection/isolation rates proposed by the authors of the DAMADICS simulator [2]. 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 [8,34]. Furthermore, in this study the whole set of potential faults were investigated, whereas in the related papers only selected states were taken into consideration.

5 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 knowledge representation in the diagnostic expert system that is realized within the frame of the DISESOR project. The research was carried out basing on the well-practised hard and soft computing classification methods. The current paper can be viewed as the extension of the research work presented in [20]. In this study, the authors proposed a new approach for searching proper values of

relevant parameters of classifiers used to fault diagnosis. The examined methods were tested in the context to be used for designing the engine of the DISESOR system. 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 procedures 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. The final results reached in this paper are much better than results shown in [20]. The features and parameters of classifiers can be automatically tuned to increase their accuracy and sensitivity.

Overall, the application of single or meta-classification strategies with optimizing of relevant parameters 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. Akthar, F., Hahne, C.: RapidMiner 5, Operator Reference (2012). <http://www.rapid-i.com>
2. Bartyś, M., Patton, R., Syfert, M., de las Heras, S., Quevedo, J.: Introduction to the DAMADICS actuator FDI benchmark study. *Control Eng. Pract.* **14**(6), 577–596 (2006). <http://dx.doi.org/10.1016/j.conengprac.2005.06.015>
3. Bartyś, M., Syfert, M.: Using damadics actuator benchmark library (dablib). <https://www.researchgate.net/publication/272478998-Using-damadics-actuator-benchmark-library-dablib>
4. Bhadra, T., Bandyopadhyay, S., Maulik, U.: Differential evolution based optimization of svm parameters for meta classifier design. *Procedia Tech.* **4**, 50–57 (2012). <http://dx.doi.org/10.1016/j.protcy.2012.05.006>
5. Blanke, M., Kinnaert, M., Lunze, J., Staroświecki, M.: *Diagnosis and Fault-Tolerant Control*. Springer, Heidelberg (2006)
6. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: *Classification and Regression Trees*. Wadsworth, Belmont, CA (1984)
7. Caccavale, F., Villani, L.: *Fault Diagnosis and Fault Tolerance for Mechatronic Systems: Recent Advances*. Springer Tracts in Advanced Robotics. Springer, Heidelberg (2003)

8. Calado, J., Sá da Costa, J., Bartyś, M., Korbicz, J.: FDI approach to the damadics benchmark problem based on qualitative reasoning coupled with fuzzy neural networks. *Control Eng. Practice* **14**(6), 685–698 (2006). <http://dx.doi.org/10.1016/j.conengprac.2005.03.025>
9. Cholewa, W.: Real-time diagnostic expert systems. *CAMES* **9**(1), 21–40 (2002)
10. Cichosz, P.: *Systemy uczące się*. Warszawa: WNT (2000)
11. Cohen, W.W.: Fast effective rule induction. In: *Twelfth International Conference on Machine Learning* (1995)
12. Cretulescu, R., Morariu, D., Breazu, M., Vintan, L.: Weights space exploration using genetic algorithms for meta-classifier in text document classification. *Stud. Inf. Control* **21**(2), 147–154 (2012)
13. Rutkowska, D., Piliński, M., Rutkowski, L.: *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*. Wydawnictwo Naukowe PWN (1997)
14. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *J. Mach. Learn. Res.* **3**, 1157–1182 (2003)
15. Haykin, S.: *Neural Networks: A Comprehensive Foundation*, 2nd edn. Prentice Hall International, Upper Saddle River (1999)
16. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University (2003). <http://www.csie.ntu.edu.tw/~cjlin/papers.html>
17. Isermann, R.: Model-based fault detection and diagnosis - status and applications. *Ann. Rev. Control* **29**(1), 71–85 (2005). <http://dx.doi.org/10.1016/j.arcontrol.2004.12.002>
18. Isermann, R.: *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer, Heidelberg (2006)
19. Jingyuan, T., Yibing, S., Longfu, Z., Wei, Z.: Analog circuit fault diagnosis using adaboost and svm. In: *Communications, Circuits and Systems*, pp. 1184–1187 (2008). <http://dx.doi.org/10.1109/ICCCAS.2008.4657978>
20. Kalisch, M., Przystałka, P., Timofiejczuk, A.: Application of selected classification schemes for fault diagnosis of actuator systems. In: *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, pp. 1381–1390 (2014). <http://dx.org/10.15439/2014F158>
21. Kerdprasop, K., Kerdprasop, N.: 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)
22. Korbicz, J., Kościelny, J.M., Kowalczyk, Z., Cholewa, W. (eds.): *Fault Diagnosis. Models, Artificial Intelligence, Applications*. Springer, Berlin (2004). <http://dx.doi.org/10.1017/S0263574704241133>
23. Korbicz, J., Kowal, M.: Neuro-fuzzy networks and their application to fault detection of dynamical systems. *Eng. Appl. Artif. Intell.* **20**(5), 609–617 (2007). <http://dx.doi.org/10.1016/j.engappai.2006.11.009>
24. Kościelny, J.M.: *Diagnostyka zautomatyzowanych procesów przemysłowych*. Akademicka Oficyna Wydawnicza EXIT, Warszawa (2001)
25. Kuncheva, L.: *Combining Pattern Classifier: Methods and Algorithms*. Wiley-Interscience, New Jersey (2004)
26. Lam, L.: Classifier combinations: implementations and theoretical issues. In: *Kitler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, p. 77*. Springer, Heidelberg (2000). http://dx.doi.org/10.1007/3-540-45014-9_7

27. Lile, A.: Analyzing e-learning systems using educational data mining techniques. *Mediterr. J. Soc. Sci.* **2**(3), 403–419 (2011). <http://dx.doi.org/10.5901/mjss.2011.v2n3p403>
28. Moczulski, W.: Inductive acquisition of diagnostic knowledge for states tree with complex structure. *Mech. Syst. Signal Process.* **15**(4), 813–825 (2001). <http://dx.doi.org/10.1006/mssp.2001.1389>
29. Mrugalski, M., Witczak, M., Korbicz, J.: Confidence estimation of the multi-layer perceptron and its application in fault detection systems. *Eng. Appl. Artif. Intell.* **21**, 895–906 (2008). <http://dx.doi.org/10.1016/j.engappai.2007.09.008>
30. Namdari, M., Jazayeri-Rad, H., Hashemi, S.J.: Process fault diagnosis using support vector machines with a genetic algorithm based parameter tuning. *J. Autom. Control* **2**(1), 1–7 (2014)
31. Patton, R., Uppal, F., Lopez-Toribio, C.: Soft computing approaches to fault diagnosis for dynamic systems: a survey. In: *IFAC Symposium SAFEPROCESS*, pp. 298–311, June 2000
32. Patton, R.J., Frank, P.M., Clark, R.N.: *Issues of Fault Diagnosis for Dynamic Systems*. Springer, Heidelberg (2000)
33. Pöyhönen, S.: Support vector machines in fault diagnostics of electrical motors. Helsinki University of Technology Control Engineering Laboratory, Technical report (2002)
34. Previdi, F., Parisini, T.: Model-free actuator fault detection using a spectral estimation approach: the case of the damadics benchmark problem. *Control Eng. Pract.* **14**, 635–644 (2006). <http://dx.doi.org/10.1016/j.conengprac.2005.04.001>
35. Puig, V., Witczak, M., Nejjari, F., Quevedo, J., Korbicz, J.: A GMDH neural network-based approach to passive robust fault detection using a constraint satisfaction backward test. *Eng. Appl. Artif. Intell.* **20**, 886–897 (2007). <http://dx.doi.org/10.1016/j.engappai.2006.12.005>
36. Moczulski, W.: *Diagnostyka techniczna. Metody pozyskiwania wiedzy*. Wydawnictwo Politechniki Śląskiej, Gliwice (2002)
37. Woźniak, M.: *Metody fuzji informacji dla komputerowych systemów rozpoznawania*. Oficyna Wydawnicza Politechniki Wrocławskiej (2006)
38. Wu, Q., Ni, Z.: Car assembly line fault diagnosis based on triangular fuzzy support vector classifier machine and particle swarm optimization. *Expert Sys. Appl.* **38**, 4727–4733 (2011). <http://dx.doi.org/10.1016/j.eswa.2010.08.099>
39. Yang, B.S., Di, X., Han, T.: Random forests classifier for machine fault diagnosis. *J. Mech. Sci. Tech.* **22**, 1716–1725 (2008). <http://dx.doi.org/10.1007/s12206-008-0603-6>
40. Yao, P., Liu, Z., Wang, Z., Bu, S.: Fault signal classification using adaptive boosting algorithm. *Elektronika ir Elektrotechnika* **18**(8), 97–100 (2012). <http://dx.doi.org/10.5755/j01.eee.18.8.2635>