



## PERFORMANCE OPTIMIZATION OF MODEL-FREE FAULT DIAGNOSIS SCHEMES

Mateusz KALISCH, Piotr PRZYSTAŁKA, Andrzej KATUNIN, Anna TIMOFIEJCZUK  
Institute of Fundamentals of Machinery Design, Silesian University of Technology  
Konarskiego 18A, 44-100 Gliwice, Poland, email: [mateusz.kalisch@polsl.pl](mailto:mateusz.kalisch@polsl.pl)

### Summary

This paper describes the method of model-free fault detection and isolation. The main purpose of the research is to present one possibility of the development of diagnostic schemes for which the component structure and behavioural parameters are tuned automatically in order to obtain the maximal efficiency of the fault detection and isolation system. The proposed approach can be viewed as the intersection of elementary methods (classic and soft computing) such as discrete wavelet analysis, machine learning (using decision trees or artificial neural networks), and evolutionary algorithms. The fundamental verification of the method was conducted for data made available within the benchmark problem involving a wind turbine. The achieved results confirm the effectiveness of the proposed approach while also showing its limitations.

Keywords: fault detection and isolation, evolutionary optimization, data mining techniques, wavelet analysis, wind turbine systems

### OPTIMALIZACJA SPRAWNOŚCI SCHEMATÓW DIAGNOSTYCZNYCH BEZ WSPARCIA MODELOWEGO

### Streszczenie

Artykuł opisuje metodę detekcji i izolacji uszkodzeń bez użycia modelu. Głównym celem badań jest pokazanie możliwości opracowania schematów diagnostycznych, których struktura oraz parametry są dostrajane automatycznie w celu osiągnięcia najwyższej możliwej sprawności detekcji i izolacji uszkodzeń. Zaproponowane podejście może być postrzegane jako połączenie elementarnych metod (klasyczne metody oraz obliczenia miękkie) jak np. analiza falkowa, metody uczenia maszynowego (drzewa decyzyjne i sztuczne sieci neuronowe) oraz algorytmy ewolucyjne. Weryfikacja metody została przeprowadzona na danych symulacyjnych wygenerowanych za pomocą modelu turbiny wiatrowej. Uzyskane wyniki potwierdziły wysoką skuteczność metody oraz pokazały jej ograniczenia.

Słowa kluczowe: detekcja i izolacja uszkodzeń, optymalizacja ewolucyjna, techniki eksploracji danych, analiza falkowa, turbiny wiatrowe

## 1. INTRODUCTION

Fault diagnosis is one of the most important directions of research in the field of automatic control because of the fact that industrial systems such as oil refineries, chemical plants, power stations, and many others, are required to be safely and reliably operated [1, 2]. In general, fault diagnosis can be carried out by applying various diagnostic schemes classified into two groups: model-based or model-free approaches. Fault diagnosis with the use of analytical or information redundancy is one of the most popular approaches in the area of process diagnostics [2]. Model-based fault diagnosis methods utilize quantitative and/or qualitative models of the supervised object in order to detect, isolate and identify faults affecting its

components. Despite the fact that much more attention is currently paid to model-based fault diagnosis methods, the authors proposed an approach that can be used to improve the development of model-free fault detection and isolation schemes for wind turbine systems. This method provides the opportunity to develop model-free diagnostic schemes for which the component structure and behavioural parameters are adjusted automatically in order to obtain the maximal efficiency of the fault detection and isolation system.

Wind energy is becoming more and more popular because of international and national government regulations governing the use of renewable power sources [3]. However, the energy obtained from a wind farm is expensive considering the costs of wind turbine manufacturing and

maintenance. Currently, industrial fault detection algorithms are based on simple approaches, and thus wind turbines must be turned off even in the cases when insignificant problems and faults occur. Moreover, it is important to reduce the costs of the operation and maintenance of wind turbines, especially for offshore wind farms, where weather conditions may prevent servicing for long periods of time [4]. The probability of wind turbine failure is high because of an aggressive and changing environment [3]. Considering this, faults may occur in different parts of a wind turbine like the generator [3,5], gearbox [6], controller, sensors, motors yawing, and others described e.g. by the authors of [7]. It follows that the diagnosis of a wind turbine in light of the proposed approach in this paper seems to be attractive while simultaneously challenging. Therefore, it has been selected as a diagnosed object in this study. Different studies in this matter (see e.g. [4,8-13]) emphasize two main directions in the development of fault detection and isolation approaches for wind turbines such as model-based methods (with the use of analytic models or data-driven models) and model-free methods (applying machine learning techniques together with signal processing approaches such as Kalman filtering, wavelet transformation, etc.). When one only takes into consideration the methods included in the group related to the subject of the paper, it can be easy to see that the most commonly used approaches adhere to classic methods (decision trees, naive Bayes algorithm, k-Nearest Neighbors algorithm, etc.) or soft computing methods (e.g. artificial neural networks, Bayesian networks, heuristic optimization algorithms, etc.). A survey on the application of such methods for fault detection and the diagnosis of wind turbines can be found in [14].

The rest of the paper is organized as follows. Section 2 contains the detailed description of the proposed method. In particular, it contains investigations on signal processing, classification methods, and performance optimization for the proposed fault detection and isolation schemes. Section 3 contains a brief description of the benchmark problem adapted and the most interesting results of the verification experiments, while the final section is devoted to concluding remarks.

## 2. MODEL-FREE FAULT DETECTION AND ISOLATION METHOD WITH PERFORMANCE OPTIMIZATION

In this paper, the general scheme presented in Fig. 1 is proposed for use in the fault detection and isolation of wind turbine systems. It may be seen that faults are detected and distinguished by use of primary and redundant process variables (filtered signals and residues). In this method, two separate classifiers should be introduced. The first classifier uses the subset of relevant diagnostic information as its input and is dedicated to generating diagnostic

signals, whereas the second one has the other subset of input variables and its task is to calculate a fault signature. This classifier is triggered in cases when the diagnostic signal indicates a fault scenario.

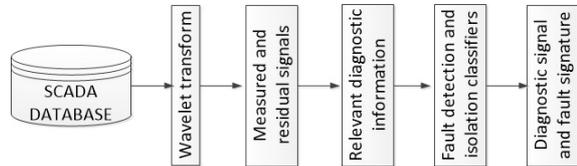


Fig. 1 General scheme of model-free fault detection and isolation schemes

The scheme described above can be included into well-known model-free fault diagnosis techniques. However, the novelty of the proposed approach is based on the fact that the structure and behavioural parameters of the scheme are tuned automatically, taking into account the general performance of the designed fault detection and isolation system. The idea of this concept is presented schematically in Fig. 2. As one may observe, the wavelet analysis and calculation of residues are performed in the classic way. On the other hand, differences can be seen in the estimation of scalar features and the selection of relevant inputs, as well as in the learning of the fault classifier. These parts of the scheme were subjected to optimization. A more detailed description of each step of the proposed approach is provided below.

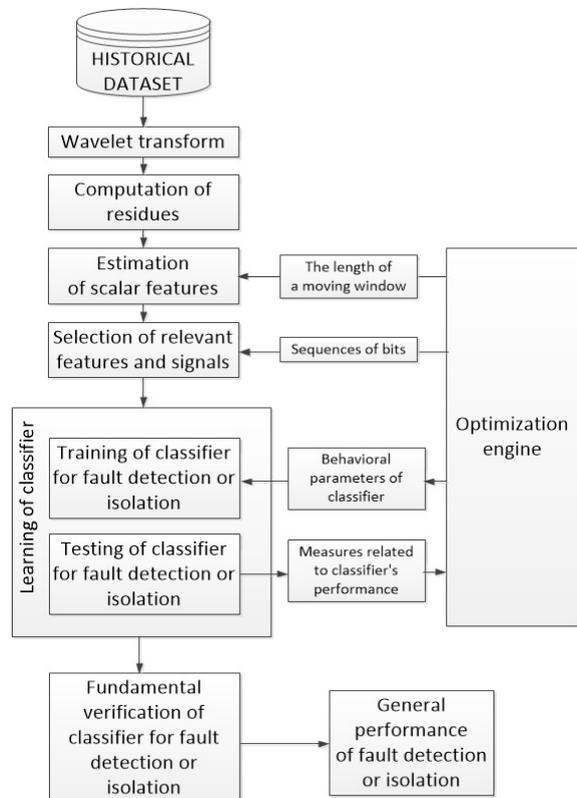


Fig. 2 Proposed approach for optimizing model-free fault detection and isolation schemes

## 2.1 Data preprocessing

Wavelet analysis was performed for the preliminary filtering and emphasizing of the features of raw signals generated through the use of the benchmark. The strong effectiveness of wavelet analysis in fault detection and isolation problems was shown by numerous researchers, with a variety of such applications being described in the review paper [15]. An important reason for the application of wavelet analysis in process signals acquired from the wind turbine is that these signals are highly non-stationary, and thus time-frequency analysis should be performed in order to properly extract the features. Discrete wavelet transform (DWT) was applied, which allows for the decomposition of the analysed signals into approximate and detailed parts. The decomposition was performed at a single-level with an application of a Daubechies wavelet of order 5. The reason for the selection of this wavelet is because of the stochastic nature of the analysed signals and higher number of vanishing moments of this wavelet, which results in the stronger emphasis of the detected features with simultaneously low-magnitude disturbances around the detected feature. As is known, decomposition using the DWT algorithm can be interpreted as a filtering procedure using a set of low- and high-pass filters. During this procedure we obtained the low-pass wavelet coefficients (approximations), which are the filtered versions of the original signals, and high-pass ones (details), which contain relevant diagnostic information including the locations of features responsible for the faulty states. It should also be mentioned that during decomposition, the resulting sets of coefficients reduce their length twice due to the downsampling procedure. However, this phenomenon is advantageous in this case, because the amount of diagnostic information increases while the amount of data remains the same as before the wavelet-based preprocessing. Both approximate and detailed wavelet coefficients were used for further analysis.

In the benchmark model, several signals are measured by redundant sensors. For this kind of signals, two physically redundant measurements are subtracted from each other to generate one residual signal. All obtained residues are added to the matrix of process variables. Next, the algorithm calculates the scalar features of all available signals (filtered by wavelet transform and residues). The authors choose several of the time domain features often used in fault diagnosis [16], such as: average, maximum and minimum values, standard deviation, root mean square, shape factor, kurtosis, time-domain energy, skewness, and entropy. All of the new values of the time domain features are added to the matrix of transformed signals and residues. In the next step, the optimization algorithm chooses the most relevant variables from the matrix of all available signals.

The selected variables are used in the process of training and testing of classifiers. In this step, each classifier is trained and tested ten times in a cross validation process. The learning data is prepared in a different way for fault detection and for fault isolation. The proportion of samples for different classes in each dataset is always equal. The process of the fundamental verification of a classifier is run again after the optimization process for optimal parameters. The final results are as follows: the confusion matrix, the average efficiency, and its standard deviation.

## 2.2 Classification methods

The problem of classification can be resolved in different ways, but in this paper three well-practised tools were utilized. The first two classification methods adhere to soft computing approaches, whereas the last one can be viewed as a classic approach.

Multilayer neural network (MLP) – this is a feedforward neural model in which multiple layers of neurons with nonlinear activation functions allow the network to learn nonlinear or linear relationships between input and output vectors [17]. In this paper, a multiple-layer network consists of three layers including  $n^1$  neurons in the input layer and  $n^2$  and  $n^3$  neurons in the first and the second hidden layers, respectively. In this case, the neural computation can be represented by the following equation:

$$y = LW^3 f^2(LW^2 f^1(LW^1 u + b^1) + b^2) + b^3 \quad (1)$$

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

Probabilistic neural network (PNN) – the first layer of this type of a neural model computes distances from the input signal to the training input patterns [18]. This layer returns a vector whose elements indicate how close the actual input is to a training input pattern. The next layer quantifies these contributions for each class of inputs so as to calculate a vector of probabilities as its net output. Finally, a competitive activation function on the output of the second layer is used in order to pick the maximum of these probabilities (it produces a set of values, where the value of 1 is used for the chosen class and 0 for the other classes). The architecture of this neural network can be expressed in a mathematical sense as:

$$y = f^2(LW^2 a^1) \quad (2)$$

$$a_i^1 = f_i^1(\|LW_i^1 u\|b_i^1) \quad (3)$$

where  $LW^{(1,2)}$  correspond to the weight matrices of the radial basis and competitive layer,  $f_i^1(\alpha)$  is the  $i$ th component of radial basis transform operator,  $\alpha$  is known as the spread or smoothing parameter determining the size of the receptive field of the Gaussian kernel,  $f^2$  is a competitive activation function,  $LW_i^1$  is a vector made of the  $i$ th row of  $LW^1$ ,  $i=1,2,\dots,n^1$ , and  $n^1$  is the number of neurons in the first layer.

Decision tree (DT) – the CART (Classification And Regression Tree) algorithm, which belongs to the group of supervised learning methods, was utilized in this paper. This requires labelled learning data to create a classifier. The decision tree is a collection of simple rules connected by branches in a hierarchical graph. One of the most important parts of the learning process is splitting. There are two widely used splitting algorithms: Gini splitting rule (4) and Twoing splitting rule (5) [19]:

$$i(t) = \sum_{k \neq l} p(k|t)p(l|t) \quad (4)$$

$$\Delta i(t) = \frac{P_l P_r}{4} \left[ \sum_{k=1}^K |p(k|t_l) - p(k|t_r)| \right]^2 \quad (5)$$

where  $k, l=1,\dots,K$  – the index of the class;  $p(k|t)$  – the conditional probability of the class  $k$  provided for the node  $t$ . There are several other methods based on entropy,  $\chi^2$ , and maximum deviation but it has been proven that the final tree is insensitive to the choice of splitting rule [19]. The comparison of two decision trees, learned on the same dataset but using different splitting rules, shows that structures of the trees are very similar. The difference can be seen only at the bottom of the tree, where the variables are less significant [20]. Another important part of the decision tree learning process is pruning. Pruning prevents overfitting and reduces the complexity of the tree. There are many available methods for pruning, like reduced error pruning, pessimistic error pruning, critical value pruning, cost-complexity pruning, and error-based pruning.

Each classifier described above was trained and tested  $n$  times during the cross-validation process. Each time the training and testing dataset was drawn again from the full data. The final result of the cross-validation process was the average efficiency of the classification and its standard deviation.

### 2.3 Performance optimization of the FDI scheme

The main purpose of the optimization process is to find the structure and values of the adjustable parameters of the fault detection and isolation

scheme in order to minimize the cost function, which can be formulated by taking into account the following two criteria: the first criterion is to obtain the maximal efficiency of the fault detection and isolation scheme, whereas the second one is to minimize the complexity of a fault classifier. The authors propose the adaptation of the global criterion method in which two objectives are combined. Therefore, one of the most general indirect utility functions is suggested in its simplest form as [21]:

$$U(q) = [(1 - eff) + \alpha_i \alpha_{eff}] c + \gamma \quad (6)$$

where  $eff$  is the average efficiency of a fault diagnosis scheme,  $\sigma_{eff}$  is the standard deviation of the efficiency,  $\alpha_i$  is the critical value corresponding to a given significance level,  $c$  denotes the complexity of an applied classifier,  $\gamma = 1 \cdot 10^6 \cdot H(eff_s - eff)$  is the penalty factor,  $H$  is the unit step function (Heaviside's function) and  $eff_s$  is the set-up value of the efficiency. The vector  $q$  is composed of adjustable parameters. These components represent the length of a moving window, sequences of bits denoting relevant input signals and their features, as well as the structure or behavioural parameters of a used classifier (e.g. the number of neurons in the hidden layer of a multilayer neural network or a spread parameter of the probabilistic neural network).

Due to the form of the cost function (6), and because there are continuous and discrete independent variables in the vector  $q$ , standard optimization methods such as gradient-based approaches cannot be adopted in this context. In spite of this, there are a large number of algorithms that can be used for solving the problem stated above. The authors decided to employ evolutionary optimization algorithms, which are based on the natural selection process that mimics biological evolution. The standard genetic operators for single-objective optimization are used to guarantee the convergence of a solution [22].

### 3. CASE STUDY

The authors of the paper applied the benchmark model of a wind turbine elaborated by the authors of [23] and implemented in MATLAB® Simulink® commercial software. The benchmark was developed in order to aid engineers and scientists working in evolving fault detection and isolation methods, and robust controllers of the wind turbines. The benchmark can be divided into several parts connected with the wind model, blade and pitch model, drive train model, generator/converter model, and controller. The benchmark allows for the use of several process variables such as measured wind

speed at hub position ( $V_{hub}$  [m/s]), rotor speed ( $\omega_r$  [rad/s]), generator speed ( $\omega_g$  [rad/s]), feedback pitch angle ( $\beta_f$ ), pitch angle ( $\beta$  [deg]), generator torque ( $\tau_g$  [Nm]), and electrical power ( $P_g$  [W]). In this benchmark model, several faults are considered. These faults cover sensor, actuator, and process faults in different parts of the wind turbine. Some of these faults are very serious and it is important to detect and isolate them very quickly. Other faults are less severe, and the controller can compensate for the influence of the fault while the turbine still works with reduced performance. The list of available faults implemented in the benchmark was grouped and presented in Tab. 1.

Table 1 Set of faults considered

Sensor faults	
F1	Fixed value on Pitch 1 position sensor 1
F2	Scaling error on Pitch 2 position sensor 2
F3	Fixed value on Pitch 3 position sensor 1
F4	Fixed value on Rotor speed sensor 1
F5	Scaling error on Rotor speed sensor 2 and Generator speed sensor 2
Actuator faults	
F6	Changed pitch system response pitch actuator 2 – high air content in oil
F7	Changed pitch system response pitch actuator – low pressure
System faults	
F8	Offset in Converter torque control
F9	Change Dynamic Drive train

The proposed method was examined in order to display a possibility for the design of wind turbine diagnostic schemes for which structure and behavioural parameters can be tuned automatically to obtain the maximal performance of the fault detection and isolation system. The case study was carried out on data collected during the simulation of the benchmark model of the wind turbine described above.

The authors decided to use only a subset of the available data in order to reduce the time of computations required for training and testing processes. To minimise the influence of the time on classification results, the chosen part of the data was acquired from the period where the power generated by a wind turbine was approaching the maximum value. It was assumed that only one fault might occur at the same time. The moment of fault occurrence was chosen randomly. The length of a signal was equal to 1000 seconds and the length of each sample with a fault was equal to 50 seconds. In this way, the prepared data contained approximately 50% of data without faults and 50% of data with all considered faults. This data was directly used in the preparation of fault detection and isolation schemes.

In this study, the performance optimization process was carried out using MATLAB<sup>®</sup> with the Genetic Algorithm and Direct Search toolbox. The most relevant parameters of the evolutionary algorithm were chosen according to the guidelines suggested in the literature [22]. The fitness function was based on the indirect utility function (6), wherein the critical value  $\alpha_t$  equalled 6 and the set-up value of the efficiency  $eff_c$  equalled 85%. The upper and lower limits of the independent variables ( $q_1, q_2, \dots$ ) were chosen, taking into account the number of the measured signals and the properties of a given classifier. It was determined that individuals in the population would be composed of genes representing real numeric and integer values. Genes in the chromosome correspond to the values of the elements of the vector  $q$ , where  $q_1$  represents the length of a moving window and  $q_2, q_3, \dots, q_{13}$  are used in order to indicate the relevant signals and scalar features (for each classifier). In the case of a multilayer neural network, variables  $q_{14}$  and  $q_{15}$  correspond to the number of neurons in the first and second hidden layer; when the classifier is based on a probabilistic neural network,  $q_{14}$  denotes the spread parameter. The feasible population method was adapted to create a randomly well-dispersed initial population that satisfies all bounds. The ranking method was used to scale the fitness function and the roulette method was employed to choose parents creating new individuals for the next generation. Reproductions were carried out by applying the elite count method (the number of individuals equalled 2) with crossover as well as mutation operators. The heuristic crossover was realized with a probability of 0.8. The remaining individuals were processed using an adaptive feasible mutation function. The maximum number of generations was chosen as the criterion for stopping the algorithm. On the one hand, the performance optimization of the fault detection scheme was run for 5 generations and with 10 individuals in the population. On the other hand, it was experimentally established when tuning the fault isolation scheme that the population size would equal to 30 and the maximum number of generations would equal to 50.

The overall results of the tuning procedure can be seen in Tab. 2. The smallest value of the fitness function was obtained for the fault detection and isolation scheme based on a decision tree algorithm. The highest value of this function was received for the case when the fault classifier was created using a probabilistic neural network. It can be observed that for this type of a classifier, if the spread parameter  $\alpha$  is near zero, the network acts as the nearest neighbour classifier. Alternatively, if it becomes larger, the designed network takes into account several nearby design vectors. In the case of a

multilayer neural network-based classifier, it was possible to have the final result of the performance optimization comparable to the result obtained for the scheme based on a decision tree algorithm.

Table 2 Overall results of the performance optimization of fault detection and isolation schemes

Class.	$\Delta k$	$\dim(u)$	Params.	$U(q)$
Fault detection				
MLN	211	202	$n^2 = 13$ $n^3 = 9$	2.16 E03
PNN	25	10	$\alpha = 0.05$	3.98E04
DT	248	201	-	0.86E03
Fault isolation				
MLN	236	195	$n^2 = 13$ $n^3 = 9$	0.66E03
PNN	140	190	$\alpha = 0.38$	9.59E05
DT	232	171	-	0.17E03

The optimization results can be better explained when taking into consideration the convergence plots of the normalized fitness function presented in Fig. 3. The most significant minimization of the fitness function was possible for the fault isolation scheme, with the classifier created based on a multilayer neural network. The other conclusion from this figure is that the optimization algorithm does not converge at the optimal solution for the case of a probabilistic neural classifier.

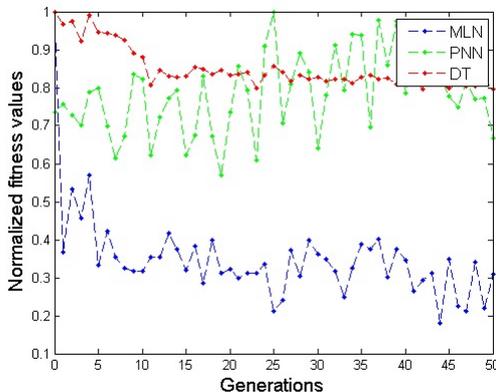


Fig. 3 Evolutionary optimization of the fault isolation scheme

In order to present some aspects more clearly, further discussion was focused on the efficiency of the fault detection and isolation schemes that were created taking into account the results of the performance optimization.

The authors prepared three test datasets for use during the verification process. Optimal parameters were used in the last step of the proposed approach in order to estimate the final result of the classifiers used in fault detection and isolation schemes. In the first dataset, the time of the occurrence of faults in training and test samples was the same. This

indicates that the training and testing samples were drawn from the same dataset. In the second dataset, faults in the testing area were moved 25 seconds forward. Therefore, the testing dataset contained 50% of samples consisting of the training data and 50% of new samples that were not included in the training interval. The last test dataset was completely different, due to the time of the fault occurrence. In this case, faulty samples were moved forward 50 seconds with respect to the training dataset, thus 100% of the data with faulty states in the testing data did not overlap with faulty states in the training data.

Table 3 General efficiency of fault detection and isolation for considered verification tests

		Test dataset 1	Test dataset 2	Test dataset 3
Fault detection	MLN	99 ± 2.55	75 ± 2.19	56 ± 2.55
	PNN	99 ± 0.10	61 ± 0.70	48 ± 0.70
	DT	84 ± 2.55	74 ± 3.56	66 ± 4.53
Fault isolation	MLN	99 ± 0.03	81 ± 4.12	56 ± 2.55
	PNN	99 ± 0.04	11 ± 0.33	11 ± 0.70
	DT	92 ± 2.28	81 ± 0.22	76 ± 0.43

As is shown in Tab. 3, the highest efficiency for each classifier was obtained within the first test dataset because the training and testing data were drawn from the same source. For Tests 2 and 3, the efficiency is worse because the faults in the testing dataset were moved in relation to the training dataset. The decision tree obtained the worst classification efficiency in the first test. In Test 2, it was able to reach a comparable result, and in Test 3 achieved a clearly better result for both considered diagnostic schemes (fault detection and isolation) was obtained. Confusion matrices for the two classifiers used in fault isolation by means of the decision tree and multilayer neural network were obtained. Tab. 4 and Tab. 5 contain results in percentages only for the second verification case. The general efficiency of these classifiers is almost the same, but the differences in the fault isolation of specific faults can be clearly seen. For example, the decision tree algorithm has a problem with the isolation of Fault no. 3, whereas the multilayer neural network for the same fault reaches 100% efficiency.

Table 4 Confusion matrix of decision tree for the second variant

	F1	F2	F3	F4	F5	F6	F7	F8	F9
F1	100	0	0	0	0	0	0	0	0
F2	0	100	49	0	0	0	0	0	0
F3	0	0	51	0	0	0	0	0	0
F4	0	0	0	100	0	0	0	0	0
F5	0	0	0	0	100	0	0	0	0
F6	0	0	0	0	0	82	8	1	0
F7	0	0	0	0	0	8	58	48	0
F8	0	0	0	0	0	10	34	51	0
F9	0	0	0	0	0	0	0	0	100

The similar situation occurred for Fault no. 7. On the other hand, a multilayer neural network has significant problems in isolating Fault no. 2 and no. 6. Furthermore, a multilayer neural network classified incorrect results into almost all available faults, whereas a decision tree algorithm classified incorrect results only for few classes. One may also see which faults were difficult to isolate in the confusion matrices of both classifiers, e.g. Faults no. 6, 7, and 8 or Faults no. 2 and 3.

Table 5 Confusion matrix of multilayer neural network for the second variant

	F1	F2	F3	F4	F5	F6	F7	F8	F9
F1	92	0	0	0	0	0	0	0	0
F2	0	59	0	0	2	0	0	0	0
F3	4	21	100	0	2	0	0	0	0
F4	2	8	0	100	2	0	0	0	0
F5	1	3	0	0	94	0	0	0	0
F6	1	2	0	0	0	59	0	2	2
F7	0	2	0	0	0	9	100	18	4
F8	0	1	0	0	0	32	0	75	5
F9	0	4	0	0	0	0	0	5	89

One of the tasks of the optimization algorithm was to choose the signals that were used to learn classifiers. After wavelet transformation, each measured signal was represented by two filtered signals – approximation and detail. The analysis of signals chosen by the optimization algorithm shows that for fault detection, the prepared matrix of signals had about 15% more detail signals than approximation signals; for the fault isolation, there were more signals based on approximations than based on details. This trend was the same for each classifier tested.

#### 4. CONCLUSIONS

The main purpose of this study is to present the method of fault detection and isolation. The basis of this method is a well-practiced model-free approach, where the diagnostic scheme is created by using elementary methods (classic and soft computing) such as discrete wavelet analysis, machine learning (e.g. decision trees and artificial neural networks), and algorithms. The extension in this matter is based on the fact that the proposed method enables the creation of diagnostic schemes for which the component structure and behavioural parameters are tuned automatically in order to obtain the maximal

efficiency of the fault detection and isolation system. The verification study was elaborated for the data obtained during the simulation of the numerical model of a wind turbine. The results achieved show the effectiveness of the proposed approach. This confirms that the method possesses great potential and should be further developed to improve the reliability of industrial systems.

#### ACKNOWLEDGMENTS

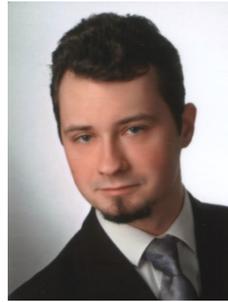
The research presented in this paper was partially financed by the National Centre of Research and Development (Poland) within the frame of the project titled “An integrated shell decision support system for systems of monitoring processes, equipment and hazards” carried out in the Path B of the Applied Research Programme – Grant No. PBS2/B9/20/2013. Part of the research was also financed by the statutory funds of the Institute of Fundamentals of Machinery Design.

#### REFERENCES

- [1] Korbicz J., Kościelny J. M., Kowalczyk Z., Cholewa, W.: (Eds.), *Fault diagnosis. Models, artificial intelligence, applications*, Springer, Berlin/Heidelberg, 2004.
- [2] Kościelny J. M.: *Diagnostyka zautomatyzowanych procesów przemysłowych*. Warszawa, Akademicka Oficyna Wydawnicza EXIT, 2001.
- [3] Kumar D., Kumar A., Ranjan A.: *Monitoring and fault diagnosis in wind turbine using wavelet transform*, Undergraduate Academic Research Journal, 1, 2012.
- [4] Tabatabaeipour S. M., Odgaard P. F., Bak T., Stoustrup J.: *Fault detection of wind turbines with uncertain parameters a set-membership approach*. *Energies*, 5, 2012, pp. 2424-2448.
- [5] Verma A., Kusiak A.: *Fault monitoring of wind turbine generator brushes: a data-mining approach*, *Journal of Solar Energy Engineering*, 2012, 134.
- [6] Yang S., Li W., Wang C.: *The intelligent fault diagnosis of wind turbine gearbox based on artificial neural networks*, In International Conference on Condition Monitoring and Diagnosis, Beijing, China, 2008.
- [7] Amirat Y., Benbouzid M. E. H., Al-Ahmar E., Bensaker B., Turri S.: *A brief status on condition monitoring and fault diagnosis in wind energy conversion systems*, *Renewable and Sustainable Energy Reviews*, 13, 2009, pp. 2629-2636.
- [8] Cross P., Ma X.: *Model-based condition monitoring for wind turbines*, *Automation and Computing (ICAC)*, 19th International Conference, 2013, pp. 1-7.
- [9] Hwas A., Katebi R.: *Model-based fault detection and isolation for wind turbine*,

Control (CONTROL), UKACC International Conference, 2012, pp. 876-881.

- [10] Kumar R., Manimozhi M., Enosh M.: *A survey of fault detection and isolation in wind turbine drives*, Power, Energy and Control (ICPEC), International Conference, 2013, pp. 648-652.
- [11] Kusiak A., Verma A.: *A data-mining approach to monitoring wind turbines*, Sustainable Energy, IEEE Transactions 3(1), 2012, pp. 150-157.
- [12] Laouti N., Sheibat-Othman N., Othman S.: *Support vector machines for fault detection in wind turbines*, 18th IFAC World Congress, Milano, Italy, 2011.
- [13] Wei X., Verhaegen M., van den Englen T.: *Sensor fault detection and isolation and Kalman filter techniques*, International Journal of Adaptive Control and Signal Processing, 24, 2010, pp. 687-707.
- [14] Manrique R., Giraldo F., Esmeral J.: *Fault detection and diagnosis for wind turbines using data-driven approach*, 7th Colombian Computing Congress (CCC), 2012, pp. 1-6.
- [15] Yan R., Gao R., Xuefeng C.: *Wavelets for fault diagnosis of rotary machines: A review with applications*, Signal Processing 96, 2014, pp. 1-15.
- [16] Jigar P., Vaishali P., Amit P.: *Fault diagnostics of rolling bearing based on improve time and frequency domain features using artificial neural networks*, International Journal for Scientific Research & Development, 1, 2013, pp. 781-788.
- [17] Haykin S.: *Neural Networks: A Comprehensive Foundation. 2nd edition*, Prentice Hall International, 1999.
- [18] Specht D.: *Probabilistic neural networks*, Neural Networks 3, 1990, pp. 109-118.
- [19] Breiman L., Richard Olshen J. F., Stone C.: *Classification and Regression Trees*, Chapman & Hall, 1984.
- [20] Timofeev R.: *Classification and Regression Trees (CART) Theory and Applications*, Master's thesis, Humboldt University, Berlin, 2004.
- [21] Marler R., Arora J.: *Survey of multi-objective optimization methods for engineering*, Structural and Multidisciplinary Optimization 26(6), 2004, pp. 369-395.
- [22] Deb K.: *Multi-objective optimization using evolutionary algorithms*, Wiley, 2009.
- [23] Odgaard P., Stoustrup J., Kinnaert M.: *Fault tolerant control of wind turbines - a benchmark model*, 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, 2011, pp. 155-160.



expert systems.

**Mateusz KALISCH** is a research assistant at the Institute of Fundamentals of Machinery Design at the Silesian University of Technology in Gliwice, Poland. His areas of research are process diagnostics, machine learning methods, context-based reasoning, and



fault tolerant-control, and the rapid prototyping of control systems and mobile.

**Piotr PRZYSTAŁKA** is an Assistant Professor at the Institute of Fundamentals of Machinery Design at the Silesian University of Technology. His research focus is on hard and soft computing optimization methods and techniques in the context of fault diagnosis and



diagnostics, wavelets, and fractals theory and applications.

**Andrzej KATUNIN** is an Assistant Professor at the Institute of Fundamentals of Machinery Design at the Silesian University of Technology in Gliwice, Poland. His areas of research cover the mechanics and non-destructive testing of composites, signal and image processing, technical



**Anna TIMOFIEJCZUK** is Professor at the Institute of Fundamentals of Machinery Design. The main areas of her research include time-frequency signal analysis, diagnostic inference, and image processing, analysis, and recognition.