XX IMEKO World Congress
Technical Diagnostics
September 9−14, 2012, Busan, Republic of Korea

DIAGNOSTICS OF WIND TURBINES BASED ON INCOMPLETE SENSOR DATA

Dr. Zs. J. Viharos, K. B. Kis

The Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, H-1111 Hungary, e-mail: zsolt.viharos@sztaki.hu

Abstract: A typical wind turbine monitors tens of parameters such as temperatures at different locations, rotation speed of the components, power produced, availability, etc. In many cases sensor data are not collected and stored continuously, because of different reasons like sensor or communication failure, storage size restrictions, condition and situation based information collection. The amount of the resulted incomplete information is typically a significant part of the whole collected dataset; consequently, there is a requirement for such diagnosis solutions that are able to handle incomplete data.

The paper introduces an artificial intelligence based solution for exploring dependencies among monitoring parameters using up the whole incomplete dataset in order to serve with reliable models for supervision of wind turbines.

Keywords: Artificial Neural Network, Incomplete Data, Supervision and Diagnostics, Wind Energy

1. INTRODUCTION

Wind turbines monitor tens of parameters to provide sufficient information for diagnostics and productivity measurements [1]. These parameters receive values from sensors, which detect temperatures at different locations, rotation speed of the components, power produced, availability, etc. [2][3]. Usually sensor data are not collected and stored continuously, because of different reasons like sensor or communication failure, storage size restrictions, condition and situation based information collection. The amount of missing data is typically a significant part of the whole collected dataset; consequently, there is a requirement for such diagnosis solutions that are able to handle incomplete datasets.

Reliable process models are extremely important in the different field of operating technical systems [4]. On the base of the applied knowledge, fundamental, heuristic and empirical models can be distinguished [5]. The paper introduces an artificial intelligence based solution for exploring dependencies among monitoring parameters using up the whole incomplete dataset in order to serve with reliable models for supervision of wind turbines [6][7].

The paper contains eight sections. After the introduction the second section describes the importance of analysing the collected data of wind turbines. After the third section discussing the topic of incomplete sensor data and the related definitions the forth section introduces an artificial Neural Network (ANN) based solution, which is able to deal with incomplete data. The fifth section describes the results of testing the algorithm on incomplete data collected from wind farms. The last three sections are conclusions, acknowledgments and references.

2. THE IMPORTANCE OF WIND TURBINE DIAGNOSTICS

Wind energy industry has experienced an extensive and worldwide growth during the past years. Certain forecasts indicate that the share of wind in Europe's energy production will reach up to 20% in the close future [8]. The efficient operation of installed turbines has an increasing significance. Among operational decisions, the supervision of the operation is decisive regarding both turbine availability and operational costs. Considering the spread of off-shore installations and the fact that their operational costs can be estimated to be 50% higher than that of the onshore farms [9], their supervision will have more emphasis. This strategy requires a full understanding of the wind turbine system and a detailed understanding of its failure mechanisms. Wind turbine supervisory control and data acquisition (SCADA) system data provides a rich resource to achieve this, because it archives comprehensive historical signal, alarm and fault log information, as well as the environmental and operational conditions. Research studies on SCADA alarms for detecting wind turbine failures and improving WT reliability through alarm optimization are rare [10].

Wind turbine generators are data intensive information sources because they incorporate various sensors similar to other branches, e.g. like manufacturing. This allows real condition monitoring and supervision of wind turbines and wind farms also from different locations and supports the preparation of reliability models with statistical information, too. There are also some differences, e.g., wind turbines are operating in continuously changing environmental conditions with sometimes extreme circumstances that is not typical e.g. in production system because they try to ensure stable and unchanging operation. This variety in environmental effects gives a great difficulty for handling changing conditions but it has also positive side: for statistical and further Artificial Intelligence (AI) analysis and modelling it can ensure a data set collected under various conditions. From the other side the data intensity requires sophisticated data processing techniques and knowledge related to them.

Learning process models, cause-effect relations, automatically recognising different working changes and degradation and intervening in the wind turbine operation in order to ensure economic and safe energy production are sophisticated approaches with high potential. They are the subjects of intensive research and development work world-wide. The complexity of the problem and the associated uncertainties necessitate the application of learning techniques to get closer to the realisation of intelligent manufacturing systems. Further integration of different techniques, such as AI, machine learning and agent-based approaches can be predicted. E.g. the ability for mapping non-linear and multidimensional dependencies among wind turbine parameters is the key in the realization of a non-conform situation [11].

3. INCOMPLETENESS OF SENSOR DATA

In order to treat different disturbances and their effects, monitoring systems collect several measured data coming from several sensors. Through analysing an applied monitoring system, one can recognise that the amount of data information collected in this way is very large and usually partly incomplete. Monitoring systems typically provide huge amount of information which is hard to collect and store. Once the data is collected, it will be usually organized into large databeses which is a characteristic of monitoring systems [12]. Because of their size, monitoring databases often become slow and too big to efficiently handle with SQL or other traditional methods. So the need arises to find such solutions that provide scalability and quick response time while they are able to deal with the increased order of volume of sensor data. A promising solution is to use NoSQL based methods which are capable to solve the mentioned problems above [13].

Beyond the large size of monitoring databases in many cases sensor data are not collected and stored continuously. The dataset, which was used in the test procedure described later, is a typical example of what is called "incomplete data". Because of e.g. storage size restrictions, a sensor's data value is only stored if there is a certain change in the measured value. This significant change and storage threshold is predefined as a percentage value of the last stored value, however the experiences with the incompleteness of the analyzed dataset shows that there are much more values missing. Many reasons, like sensor or communication failure, storage size restrictions, condition and situation based information collection can cause the resulted level of information lack.

The problem of missing data arises in several fields of real-life applications, for example in production lines, cutting process supervision, stock market datasets, questioners etc. [14]. Examining these fields, it can be seen that the rate of incompleteness can reach sometimes more than 50% of the whole dataset. In both of the analyzed datasets introduced in the paper the values were collected by SCADA systems from operating wind turbines. The first data collection was performed earlier in the past and resulted in a dataset where half of the values are missing. The other dataset was collected later and 30% of the data are missing in it.

	A	B	C	D	E	F	G	H
1	Parameters/ TimeStamp	parameter 1	parameter 2	parameter 3	parameter 4	parameter 5	parameter 6	parameter 7
2	1. time point						59	
3	2. time point					29	60	
4	3. time point				40		60	
5	4. time point						61	
6	5. time point							
7	6. time point			34	41		62	57
8	7. time point				43		63.75	58
9	8. time point			35			66	59
10	9. time point				44		66	
11	10. time point							
12	11. time point							
13	12. time point						65	
14	13. time point							
15	14. time point						64	
16	15. time point					30	63	
17	16. time point							58
18	17. time point	100	100	34	43	30	63	58
19	18. time point						62	
20	19. time point				42			
21	20. time point						62	

Fig. 1. An example of a typical incomplete dataset collected from wind turbines

As example the Fig. 1. shows a small part of the dataset collected from wind turbines, where 30% of data is missing. The horizontal axis represents the different parameters and vertical axis represents the units of time in which the parameter values were recorded. It is seen e.g. that the amount of missing data is varying by parameter.

4. ARTIFICIAL NEURAL NETWORK BASED SOLUTION FOR HANDLING INCOMPLETE DATA

The problem of missing data arises in several fields of real-life applications. There are some examples from several fields, together with the applied methods for handling missing data [14]. One solution is to use function-based interpolation to determine values for missing data and ANNs or other approximators are able to replace the functions and estimate the missing values.

Different approaches were tested to replace missing data by Pesonen *et al.* while building up a neural network to study medical data. They compared four methods to replace the missing data: substituting means, random values, data based on the nearest neighbour and a neural network based substitution [15].

An interesting solution can be found in the paper of Keeler et al. for handling missing data. A data preprocessing module extends the missing part of the input vector before conveying it to the ANN model and serves various parameters about this extension, too. The so-called decision processor receives these together with the model output and uses both of them to decide about the necessary changes in the system [16].

4.1. The method for handling incomplete data

Some methods were listed in the previous paragraphs to solve the problem of missing data. These methods try to complete the missing part of the data vectors in several different ways. Instead of completing data vectors another approach could be to generate neural networks, which can handle incomplete data directly as described in the paper. The algorithm is based on the main idea of turning the neurons corresponding to the missing part of certain data vectors into protected state and leaves the other neurons in unprotected state. Basically the neuron's protection state provides a simple way to exclude any neuron from the given structure and reattach it later. When a neuron is protected, it becomes temporarily "dead" with all of it's links. In other words the neural network behaves like if the protected neurons were never part of the net – they not involved in the learning process or any other calculation of the net. However the protected neurons and links preserve all the information they learned in their unprotected state and can be easily "reanimate" to work as the part of the net again. This process is beyond the simple extension of a neural network with new components because the "reanimated" neurons and its links take part in the calculations having the recent numerical parameters, consequently, these components of the structure "remember" the "knowledge" collected before.

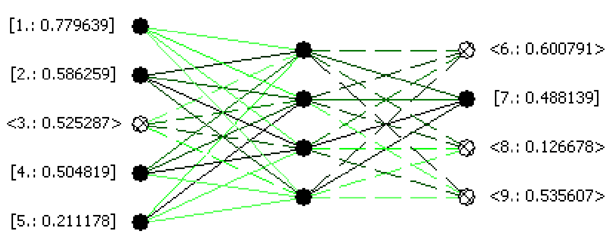

Fig. 2. Neuron protection according to missing data

As example, the Fig. 2. shows how the neural network structure is adapting to missing data values of one pattern (data vector) by the protection state of neurons. The full circles in the picture represent the active "unprotected" neurons while the empty ones are in protected state. The values in brackets are valid and the ones in <,>-s are missing or incorrect, i.e. they are not containing reliable information. In databases, incompletness does not always mean missing values, sometimes they appear to be abnormal, e.g. they are negative when they are not supposed to be or overstep a certain prescribed limit. For example a sensor that monitors a component's angular offset, can not measure larger value than which is physically possible.

The described approach needs further information i.e., to describe which part of the input- and output vector is missing or incorrect. A flag called validity is used for indicating whether a date in the data vector is valid or not.

As example the Fig. 3. shows a view of data vectors. The values in brackets [,] are valid and the ones in <,>-s are missing or incorrect. In the application a so named (binary) validity vector is attached to all of the input and output vectors (values) to describe the validity state of the data incorporated in these vectors. The protection of the input and output layer of the ANN structure changes according to the validity vector of the data vector, in question, namely if a data is invalid in the input or output vector, the corresponding neuron is set to protected, otherwise the neuron will be unprotected. It ensures that the protections of the input and output neurons change by all the learning data vector pairs.

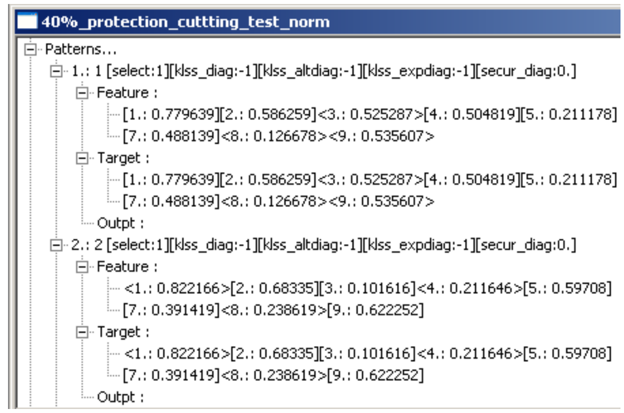

Fig. 3. Validity of data vectors

4.2. The method for dependency exploration/submodel decomposition

The previous paragraph describes how the method handles incomplete data directly. Another characteristics of the wind turbine diagnostics is being a highly complex system which implies models which - due to the high number of parameters and the dense network of interrelations - can be handled as a whole only at staggering computational costs. It is thus advantageous to decompose these complex system to several smaller interconnected subsystems (submodels) which can be easily handled one by one (moreover, always the set of submodels relevant to the given problem can be selected, so that submodel decomposition results in subtask decomposition as well) [17].

The novel model building method in [18] allows the flexible exploration of submodels, as well as free assignment of a given variable for input or output. The highest number of outputs is found in an input/output search based on ANN-learning [19]. However, attempting to learn a potential output by an ANN can only signalize that there is a dependency ''somewhere within the set of selected variables'' but cannot weed out parameters totally independent of the given subsystem. This would result in a single ANN struggling to learn the entire structure in question; therefore, the reduction to smaller, easy-to-handle submodels must be cared for by other means. While the vast majority of such approaches determines the submodel structures before any ANN training takes place, this new method identifies the submodel structures dynamically, leaning on the results of earlier ANN training periods. This is accomplished by an extended feature selection algorithm running on the complete set of variables and setting up an optional tree for submodel selection. The extended feature selection algorithm applied here assumes a pure classification task, onto which even continuous parameters can be mapped with an appropriate heuristics. In the first step, a given parameter is selected and its values encountered in the training data set are grouped into clusters (i.e., intervals of equal length), so that at least one element is contained in each interval (this, in itself, being the first heuristic decision). Next, the algorithm checks how ‘‘distinct’’ these clusters are, i.e., how far apart the weight centres of the clusters are and how large the distances of the cluster weight centre and the cluster’s discrete points are. This test is performed for all parameters that can come in question, then, the one exhibiting the most ‘‘distinct’’ clustering of values is chosen. Having selected the first parameter of interest, all remaining variables are tested again, each of them together with the already highlighted parameter, for the same measure of class distinctness, using Euclidean distances. Again, the parameter chosen to form a potential submodel together with the first preferred variable will be the one exhibiting the best class separability together with the parameter already selected in the first run. In every subsequent step, yet another unselected variable is tested the same way, and in every case, the one corresponding to best class separation is chosen (note that this incremental selection, as opposed to a combinatorically exhaustive test, is the second heuristic decision in the algorithm’s layout).

Adding new parameters to the ones already selected, a deterioration of class separability can be observed which is guaranteed to be worst when all variables are taken for classification. However, since our goal is the creation of submodels, each containing only a relevant part of the model’s entire parameter set, a suitable heuristics (the third such case in the algorithm) should be used to decide when adding new parameters should be stopped. By selecting only a part of the model’s entire parameter set as the best performing variable group for one given clustering, a candidate for a submodel is created.

output	feature sequence				
0	5	2	4	1	3
1	0	2	4	5	3
2	5	3	0	1	4
3	4	5	2	1	0
4	5	1	0	2	3
5	3	1	4	2	0
	distance sequence				
0	0.573	0.548	0.469	0.43	0.37
1	0.388	0.339	0.320	0.284	0.262
2	0.641	0.545	0.502	0.462	0.413
3	0.502	0.438	0.382	0.337	0.311
4	0.580	0.534	0.497	0.452	0.403
5	0.543	0.533	0.503	0.492	0.478

Fig. 4. Feature selection mechanism for determining possible submodels

As example the Fig. 4. shows how the feature selection determines possible submodel candidates in a six-dimensional parameter space. The column called “output” indicates which parameter is clustered i.e. chosen to be the base of generated classes. The distance sequence matrix which is in the bottom half of the picture, contains the distance values for ordering the parameters while the feature sequence matrix which is in the top half of the picture indicates the order of parameters. In the first step the algorithm calculates the average distance of the clusters based on the remaining five parameters one by one than it chooses the maximum of the values and the belonging parameter will be the first selected feature (the chosen maximum distance is the first value of a row of the distance sequence matrix and the chosen parameter is the first value of the corresponding row of the feature sequence matrix). In the second step the algorithm calculates the distances based on two parameters – the first one is the parameter selected in the previous step and the second one is one of the four remaining parameters – than choses the maximum as in the first step. These values are the second column of the matrices. The steps continue increasing the numbers of parameters on which measuring the distance of clusters are based until there is no more parameter to add. Once a sequence of parameters and corresponding distances is determined, the algorithm finds the greatest difference between two adjacent distance values and ignores the parameters which are on the right of this position. These cuts are indicated with blue rectangles in the picture. The parameter group on the left of the cut is a possible submodell which is indicated with a red ring in the picture. These groups will be examined further for discovering the dependencies between the parameters.

Since three heuristic decision steps were taken to obtain the candidate submodel, this can be considered only an assumption which is to be either verified or rejected by the ANN algorithm. The latter begins validating a given part of the submodel structure - at a given point in the decision tree - and delivers training results. Examining these and removing the successfully learned submodel from the ‘‘pool’’ of unclassified variables, feature selection is run again on the remaining data set and the decision tree is reconfigured if needed. Hereafter, ANN training takes place again. Thus, the method does not separate pre-selection and ANN training into disjoint tasks - in fact, feature selection and training complement each other with their alternate execution until all submodels are identified and learned.

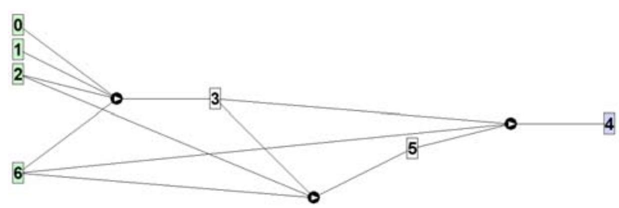

Fig. 5. Visual representation of a submodel structure

As example the Fig. 5. shows a tipical submodel structure. The numbers represents the parameters, a circle with a triangle in it indicates a submodel where the inputs are the numbers connected to the circle from left and the outputs are the ones connected from right. For example there are three submodels on this picture: the first one’s input parameters are 0, 1, 2, 6 and the output is 3, the second one’s inputs are 2, 3, 6 and the output is 5 and the third one’s inputs are 3, 5, 6 and the output is 4.

5. REALIZATION OF WIND TURBINE DIAGNOSTICS BASED ON INCOMPLETE SENSOR DATA

Wind turbines monitor tens of parameters such as temperatures at different locations, rotation speed of the components, power produced, availability, etc. The used dataset was collected from the sensors of wind turbines of one wind farm for several months. This is a typical example of incomplete data where 30% of the values are missing. The parameters of the dataset were partitioned into the following groups:

- *Availability*: operational availability related parameters
- *Power*: parameters measuring produced energy, frequency, voltage related values
- *Temperature*: parameters which measure temperatures at different wind turbine locations
- *Wind turbine speed*: parameters measuring speed, velocity and pitch angles

Each of the parameter groups contains one or two dozen parameters. The algorithm was searching for submodels in these groups and resulted in a submodel structure for each of them. The submodel structure is a connected stream of parameters which means that one submodel's output can be another one's input. It means that there are real inputs in a submodel structure which are only appear in the identified submodels as inputs, but not as outputs or parts of any of the submodels. From the real inputs of the structure, every other parameters of the given group can be estimated. A fifth group called "common" was generated from the real inputs of each group. With running the algorithm on this group more dependencies can be discovered among the other groups, reducing the input number of the whole, connected model structure. This fifth submodel structure describes how the above groups of parameters (availability, power, temperature and wind turbine speed) depend on each other.

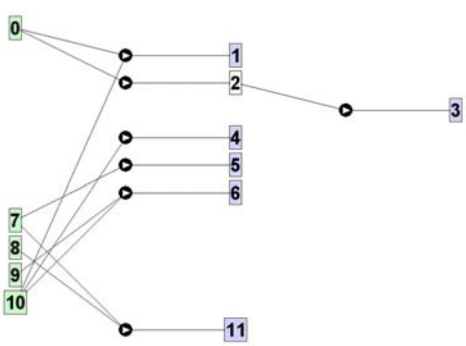

Fig. 6. Found submodels in Power parameter group

As example the Fig. 6. shows the submodel structure of the power parameter group. The algorithm has found seven submodels where the outputs can be estimated from the inputs with the given error limit. In other words, the algorithm explored seven dependencies among the parameters.

Two main aspects were considered during the test procedure for evaluating the methods. The first aspect is the accuracy of the identified submodels. The algorithm accepts a submodel only if the outputs can be estimated with an error which does not overstep a certain limit. The error calculation uses average squared error (RMS). This limit may be different for every parameter and is predetermined by human decisions. This decision is based on the calculated "next error limit" of the rejected submodels. Submodels are rejected when no one output can be estimated within the actual error limit. A submodel's next error limit gives information about to what value of the error limit should be to accept a possible submodel. With increasing the error limit of the algorithm based on the next limit of the rejected submodels more parameters can be estimated but with less accuracy. Finding the optimal balance between the error limit and the number of outputs is a compromise of the maximal dependency exploration among parameters and the satisfaction of field specific model accuracy requirements.

The average percentage error limits of the analysed groups and the sum of output parameters are the base of comparison for the tested datasets.

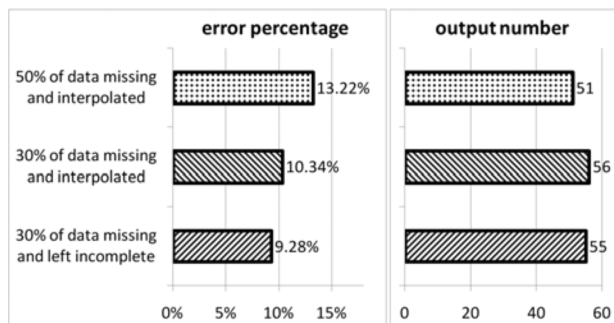

Fig. 7. Average error and summed output number

As example the Fig. 7. shows the results of submodel decomposition algorithm. The vertical axis indicates the datasets, the first one from the top is the "older" dataset with the rate of 50% incompleteness, where the missing values were determined by interpolation, the second one was also extended with interpolated data, but this dataset had 30% of values missing and the third dataset – which is the same as the second one without interpolation – was left incomplete and handled directly by the introduced algorithm. The left column shows the average modelling error on the whole dataset. The right column shows how many parameters were found as outputs which mean that for every one of these parameters there is a submodel where it can be estimated. The figure shows that the introduced solution for handling incomplete data directly results the most accurate model structure and almost the same number of estimated parameters as the methods that are not able to handle missing data. *This fact proves that it is worth to handle incomplete data sets directly by the modelling algorithms instead of eliminate the missing data as a preliminary step before the modelling takes place.* With high probability this effect is given because the step for eliminating missing data before the model building usually does not consider the features of the applied modelling methods, however the direct handling of incomplete data by the modelling solution solves this requirement in an integrated way.

6. CONCLUSIONS

Monitoring systems and complex diagnostics are extremely important in many fields like wind energy. Because of the high rate of incompleteness of collected data there is a requirement for such diagnosis solutions that are able to handle datasets with missing values.

The paper introduced an artificial intelligence based solution for exploring dependencies among monitoring parameters using up the whole incomplete dataset in order to serve with reliable models for supervision of wind turbines. The main advantage of the solution is that it uses all the accessible information of a given incomplete dataset directly without the preprocessing of data for determining the missing values before the model building phase.

Tests of the introduced method on incomplete wind turbine data showed that the resulted submodels are more accurate when the incomplete dataset is handled directly by the algorithm than they are when the missing values are determined by interpolation while the number of estimated outputs is basically the same in both cases.

Only few methods are able to handle incomplete datasets without manipulating the missing values before. However, the results showed that it is worth developing such solutions because they use all the accessible information providing more accurate and realistic model that is a key feature for wind turbine diagnostics, too.

7. ACKNOWLEDGMENTS

The authors acknowledge the support of grants ‘Digital, real-time enterprises and networks’, OMFB-01638/2009, ‘Quantifiable Closed Quality Control’, CORNET_7_09-1-2009-007, TAMOP-4.2.1/B-09/1/KMR-2010-0002, the Fraunhofer Project Center for Production Management and Informatics at SZTAKI, Budapest, Hungary and the hopefully approved Bolyai János Research Fellowship of Dr. Zsolt János Viharos having the title “Supervision and diagnostics of wins turbines”.

8. REFERENCES

[1] A. Jabłonski, T. Barszcz, M. Bielecka, “Automatic validation of vibration signals in wind farm distributed monitoring systems”, Journal of the International Measurement Confederation, vol. 44, pp. 1954-1967, 2011.

[2] Zs. J. Viharos, L. Monostori, K. Novák, G. Tóth, Z. Csongrádi, T. Kenderesy, T. Sólymosi, Á. Lőrincz, T. Koródi, “Monitoring of complex production systems, in view of digital factories”, XVII IMEKO World Congress - Metrology in the 3rd Millennium, Dubrovnik, Croatia, pp. 1463-1468, 2003.

[3] L. Monostori, Zs. J. Viharos, “Technical monitoring and diagnostics: Indispensable element of intelligent manufacturing systems”, XVIII IMEKO World Congress - Metrology for a Sustainable Development, Rio de Janeiro, Brazil, paper nr.: 32., 2006.

[4] M. E. Merchant, “An interpretive look at 20th century research on modeling of machining”, Inaugural Address, Proc. of the CIRP International Workshop on Modeling of Machining Operations, Atlanta, Georgia, USA, pp. 27-31, 1998.

[5] E. Pinheiro, O. Postolache, P. Girão, “Empirical Mode Decomposition and Principal Component Analysis implementation in processing non-invasive cardiovascular signals”, Journal of the International Measurement Confederation, vol. 45, pp. 175-181, 2012.

[6] M. Catelani, A. Fort , C. Alippi, “A fuzzy approach for soft fault detection in analog circuits”, Journal of the International Measurement Confederation, vol. 32, pp. 73-83, 2002.

[7] F. Rahimi-Ajdadi, Y. Abbaspour-Gilandeh, “Artificial Neural Network and stepwise multiple range regression methods for prediction of tractor fuel consumption”, Journal of the International Measurement Confederation, vol. 44, pp. 2104-2111, 2011.

[8] MI. Blanco, “The Economics of Wind Energy”, Renewable and Sustainable Energy Reviews vol. 13, pp. 1372–1382, 2009.

[9] J. Markard, R. Petersen, “The Offshore Trend: Structural Changes in the Wind Power Sector”, Energy Policy vol. 37., pp. 3545–3556, 2009.

[10] A. Kusiak, A. Verma, “The future of wind turbine diagnostics”. Wind Systems Magazine, vol. April, pp. 66-71, 2010.

[11] Zs. J. Viharos, G. Erdős, A. Kovács, L. Monostori, “AI supported maintenance and reliability system in wind energy production”, European Wind Energy Conference and Exhibition (EWEC), Tuesday 20 - Friday 23 April 2010, Warsaw, Poland, paper nr.: 286, 2010.

[12] L. Monostori, G. Haidegger, J. Váncza, Zs. J. Viharos, “Digital Enterprises: A national project in Hungary”, 1st CIRP(UK) International Seminar on Digital Enterprise Technology, Durham, UK, 16-17th September, pp. 269-272, 2002.

[13] M. Spaniol, A. Benczúr, Zs. J. Viharos, G. Weikum, “Big Web Analytics: Toward a Virtual Web Observatory”, ERCIM News (Journal of the European Consortium for Informatics and Mathematics), in print, 2012.

[14] Zs. J. Viharos, L. Monostori, T. Vincze, “Training and application of artificial neural networks with incomplete data”, The Fifteenth International Conference on Industrial & Engineering Application of Artificial Intelligence & Expert Systems, Lecture Notes of Artificial Intelligence, Springer Computer Science Book, Springer-Verlag Heidelberg, Cairns, Australia, pp. 649-659, 2002.

[15] E. Pesonen, M. Eskelinen, M. Juhola, “Treatment of missing data values in a neural network based decision support system for acute abdominal pain”, Artificial Intelligence in Medicine, Vol. 13, pp. 139-146, 1998.

[16] E. Keeler, J. David, H. Hartman, E. Jon, J. Ferguson, R. Bruce, “Method for operating a neural network with missing and/or incomplete data”, United States Patent, No. 5 819 006

[17] Zs. J. Viharos, Zs. Kemény, “AI techniques in modelling, assignment, problem solving and optimisation”, Journal of Engineering Applications of Artificial Intelligence, vol. 20, pp. 691-698, 2007.

[18] Zs. J. Viharos, “Automatic generation a net of models for high and low levels of production control”, 16th IFAC Word Congress, Prague, Czech Republic, 2005.

[19] L. Monostori, Zs. J. Viharos, S. Markos, “Satisfying various requirements in different levels and stages of machining using one general ANN-based process model”, Journal of Materials Processing Technology, vol. 107: (1-3), pp. 228-235, 2000.