World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:4, No:1, 2010

Comparative Study of Decision Trees and Rough Sets Theory as Knowledge Extraction Tools for Design and Control of Industrial Processes

Marcin Perzyk and Artur Soroczynski

***Abstract*—**General requirements for knowledge representation in the form of logic rules, applicable to design and control of industrial processes, are formulated. Characteristic behavior of decision trees (DTs) and rough sets theory (RST) in rules extraction from recorded data is discussed and illustrated with simple examples. The significance of the models' drawbacks was evaluated, using simulated and industrial data sets. It is concluded that performance of DTs may be considerably poorer in several important aspects, compared to RST, particularly when not only a characterization of a problem is required, but also detailed and precise rules are needed, according to actual, specific problems to be solved.

***Keywords*—**Knowledge extraction, decision trees, rough sets theory, industrial processes.

I. Introduction

In recent years an increasing interest of data mining (DM) applications in industrial enterprises can be observed. Large amounts of collected data, related to designs, manufacturing processes, materials and equipment, can be potentially used for improvement of the quality and economics of production. A comprehensive and insightful characterization of the problems in manufacturing enterprises as well as the potential benefits from application of DM in this area is presented in [1]. Examples and general characteristics of problems related to the usage of data mining techniques and systems in manufacturing environment can be found in several review papers [2]–[4]. A substantial progress in development of complex DM systems for manufacturing organizations can be also observed [5]–[10].

DM techniques can provide various types of information. Most frequently, methods of automated knowledge extraction from the recorded past data in the form of logic rules of the type: 'IF (conditions) THEN (decision class)' are utilized. Also another types of information may be important for industrial applications, such as relative significance of input variables (usually process parameters) [11], prediction of continuous-type output (usually process results) as well as grouping (clustering) of variables.

In principle, for extraction of logic rules from data, any classification system or model can be used. Typical learning algorithms include direct rule induction, decision trees (DT), naïve Bayesian classifier and algorithms based on the rough sets theory (RST). Detailed information on these methods can be found in [12] and the literature quoted there. Artificial neural networks have also been successfully utilized for logic rules extraction [13]–[16], often involving fuzzy numbers. This approach facilitates processing continuous-valued variables, handling uncertainties appearing in data and usage of linguistic variables.

For manufacturing problems DTs are probably the most frequently used tools for rules extraction from data (e.g. [4], [9], [10], [17]–[19]), whereas the RST-based methods seem to be their newer alternative (e.g. [12], [20]–[22]). Both algorithms are relative simple, especially compared to neural or fuzzy-neural systems, and easy to interpret by users. Both of them treat the data in a natural way however, they are based on completely different principles and algorithms.

The practical aspects of application of those tools are also different. The computation times of DT are generally short and the interpretation of rules obtained from DT can be facilitated by the graphical representation of the trees. The RST theory may require long computational times and may lead to much larger number of rules, compared to DT, if one seeks a detailed information from the knowledge system. It should be noticed, that whereas DT are widely spread both in handbooks and in commercially available DM software, the RST can be rather seldom found, except for scientific literature.

Making a right choice of the rules extraction algorithm is important, particularly in construction of DM systems. However, there are very little comparative studies available, which could show the advantages and weakness of individual tools [12], [20]. The purpose of the present paper is to show important differences in performances of the two algorithms mentioned above, i.e. DT-based and RST-based, chiefly from the standpoint of industrial manufacturing processes.

M. Perzyk is with the Institute of Manufacturing Technologies, Warsaw University of Technology, Warsaw, Poland (corresponding author, phone and fax: +48228499797; e-mail: M.Perzyk@ wip.pw.edu.pl).

A. Soroczynski – affiliation as above (e-mail: asoroczy@wip.pw.edu.pl).

This work was supported by grant N R07 0015 04 from Ministry of Science and Higher Education, Poland.

The work presented in this paper consists of three main parts. First, the expectations and requirements for the knowledge rules systems which could be successively used in design and control of industrial manufacturing processes are formulated and characterized. Then the characteristic behavior of DT-type models is presented and compared to RST theory, using simple demonstration data sets. Finally, the significance of drawbacks of the DT models as rules extraction system is evaluated, using several simulated and real data sets.

II. Requirements for Knowledge Rules Applicable for Industrial Processes

General requirements for knowledge rules which could be useful in manufacturing industry are rather obvious and similar to those for other areas of applications.

First, the rules should be reliable, which means that there is a real chance that application of a rule will bring the predicted result. This can be expressed by rules quality parameters, of which the most important are confidence (also known as consistency or precision) and support (also known as strength). Confidence is defined as a ratio of number of records with given combination of input values and the given output value to the number of records which have that combination of input values only. It estimates the probability, that application of input values appearing in the rule will give the result expressed by the rule's output (decision class). Support, defined as the ratio of number of number of records with combination of input and output values appearing in a rule to the total number of records, reflects the breadth of observation basis of the rule.

The second general requirement is that the rules should not be unnecessarily demanding, i.e. they do not comprise conditions which are not important, particularly redundant.

The tools used for knowledge extraction are first of all oriented at generation of a set of rules which best characterize the problem, i.e. most reliable ones. However, in many industrial processes, particularly in manufacturing, some more specific requirements are relevant, related to design and development of new processes or control of currently running ones. Hence, typical questions to be answered by using the rules can be formulated as follows:

- What are the most effective and reliable ways (i.e. process parameters - input values) to achieve an assumed result (class variable)? This is not the same as finding the most reliable rule set as the requested result may be included in the rules of minor quality only.
- What will happen if we are not able to apply certain input values, i.e. what will we get if we use different ones? Do we still have a chance (and how big) to get the required result?
- What will be predictions (and how reliable) in case some input variables cannot be specified, e.g. they may be out of control?
- What are all alternative ways to achieve our goal? How reliable they are?

It should be noticed that answering some of the above questions may result in necessity of predictions for combination of parameters (input variables values) which have never appeared in the past (i.e. are not present in the data). Users may be interested not only in obtaining a one-time prediction for such input values but also in having a rule or rules with estimated quality parameters.

The requirements for rules system and the knowledge extraction tools, suitable for manufacturing industry applications, are not only a consequence of the issues described above, but also the specificity of available data. Typically, the number of independent variables (i.e. problem dimensionality) is not large, it seldom exceeds 10. Number of available records can vary within broad ranges, from only a few to many thousands, especially when the automatic data acquisition system is utilized. The variables can be continuous or categorical. In the former case the discretization, necessary for some classification systems also for input variables, can be often done by the user, based on his/her experience and feeling. Typical industrial data are noisy which results in their inconsistency, i.e. an occurrence of different output variable values (decision classes) for an identical combination of input values, i.e. conditions in a rule. The other types of inconsistencies, which can be also found in real data, will be not considered here. Finally, strongly unbalanced representativeness of classes can be often observed, both for input and output variables.

From the characteristics of industrial processes problems presented above the following requirements for rules systems seem to be essential or at least important:

- The rules should make use of *all* information in data. This means, for example, that all output values (classes) must be represented. Even single cases can be valuable and therefore should be reflected in the rules system.
- The rules should not contain redundant conditions as they can be misleading for the user.
- It should be possible to find a rule 'tailored' to the user specifications, including combinations of input variables values which are not represented in the data.
- Reliability of all rules should be evaluated, using the confidence and support as the primary parameters.

III. Characteristic Behavior of DTs and RST in Rules Extraction

A structure of a DT model is uniquely defined by a set of the logic expressions, corresponding to the knowledge rules. The nature of DT models, based on recursive partitioning of the data records, results in a set of conditions, which may be different from the combinations of input variables in the training data records. Some of the combinations appearing in the data set may be absent in the tree and vice versa, also some sequences of conditions in the data may be abbreviated in the tree.

The lack of some combinations of input values in DTs which are present in training data, may result in the rule system in which some important rules are missing. Another consequence is that DTs can give wrong predictions for training data. In case of consistent data, this may be a result of improper tree structure, i.e. in which the given combination of input values (attributes) is connected with a class of the output variable which is different from that which appears in the data. Partly incorrect predictions may be a consequence of the fact, that DTs are able to give only *one* prediction for a given combination of input variables values. For noisy, inconsistent data it must always lead to a fraction of false predictions. Considering a DT as a knowledge rules system it means that for that type of data DTs must omit some rules, potentially also important for a user. In particular, those omitted rules can be the only ones which give a certain output. An illustrative example is given in Table I. Due to ambiguous output classes D appearing in the last four records, the DT has ignored the output class D = u, although its occurrence in the data was more frequent that D = r.

TABLE I
EXAMPLE OF DT FOR INCONSISTENT DATA

A_1	A_2	D	Comments
Data set:			
b	f	r	Present in rules, 1 record
b	g	s	
c	f	s	
c	g	t	Present in rules
c	g	t	Present in rules
c	g	u	Missing in rules, 1 of 2 records
c	g	u	Missing in rules, 1 of 2 records
Rules:			
b	f	r	
b	g	s	
c	f	s	
c	g	t	The only rule for these A1 and A2

In contrast with DTs, RST theory is able to offer all possible rules resulting from the data, with specification of their confidence and other quality parameters.

Rules obtained from DTs may include redundant conditions as the splitting variable used in the core must appear in all rules (generally, the splitting variable in a node must appear in all rules resulting from subsequent splits). In contrast, RST provides ‘fitted’ rules, i.e. without unnecessary conditions. That type of behavior of the both algorithms was commented in detail in [12]. The rules extracted by RST were described as ‘more individualized’ and made the authors to chose RST for their application.

In principle, both DTs and RST can offer predictions for combinations of input values absent in the data, however, they treat such cases in entirely different ways. Interrogation of DTs for such values may lead to one of the following results: (a) prediction which is a good reflection of the general dependencies in the training data, (b) prediction which is far from the expectations and (c) impossibility of the prediction, when the requested path does not exist in DT. Some DT induction algorithms provide mechanisms which may help in situations (c).

The cases (a) and (b) mean that DT includes paths (leading to leaves) which, in principle, correspond to rules not supported by data. However, those logic expressions cannot be treated as valuable rules not only because they may not meet the actual user’s demands but also because the quality parameters of such rules cannot be specified. The confidence would be indeterminate (0/0) whereas the support would be equal to zero.

RST-based algorithm will find a rule with reduced number of conditions, so that they include only those combinations of input values which appear in the data. This ‘substitute’, shorter rule has its confidence and support defined. However, it may result in obtaining predictions (rules) which are not expected. An illustrative example is given in Table II.

TABLE II
EXAMPLE OF RST RULE FOR NEW DATA

A_1	A_2	D
Training data set:		
1	1	1
2	1	2
1	2	2
1	4	3
3	2	3
1	5	4
4	2	4
5	3	5
2	4	4
New data:		
2	3	***3***
Matched (shorter) rule found from RST:		
	3	***5***

The training data in Table II contains ordinal-type variables and was created by computing output values from the simple formula: $D=A_1+A_2$, for random values of inputs, followed by the normalization and categorization of the output. Thus, the expected class for the new input combination can be easily calculated. It can be seen that the RST-based algorithm has led to the result which is far from the expected one.

IV. SIGNIFICANCE OF DRAWBACKS OF DTS AS RULES EXTRACTION SYSTEMS

A. *Methodology*

The significance of problems which may appear in application of DT models for rules extraction, was evaluated with a use of simulated and industrial data sets. The first type sets were obtained by assuming an analytical formula of the type $Y=f(X_1, X_2, \ldots)$, from which, for random values of continuous-type input variables X_1, X_2, …, the dependent continuous-type variable Y was calculated. Then a Gaussian-type noise with maximum deviations ±20% was imposed on the input variables, and finally all the continuous values variables were converted to categorical ones assuming equal intervals method. Each of the simulated data sets had 1000 records. Three numbers of the intervals for discretization were assumed: 3, 5 and 7, and two basic formulas were used: $Y=X_1+2\cdot X_2+3\cdot X_3+4\cdot X_4+5\cdot X_5$ (Sim1 3cl, Sim1 5cl and Sim1 7cl data sets) and $Y=X_1\cdot X_2+X_3+X_4+X_5$ (Sim2 3cl, Sim2 5cl and Sim2 7cl data sets). The first group of data sets reflects the situation, where the input variables have highly differentiated effect on output whereas the second data sets is an example of interaction between two variables with overall significance equal to significances of the remaining variables. Similar situations often appear in practice.

All the real (industrial) data sets concern foundry production. The Ind1 data set correlates chemical composition of ductile cast iron, defined by 5 main elements (Mn, Si, Cr, Ni and Cu) with its four grades, obtained as a result of the melting process, as the output class variable. The number of classes (categories) for all 5 input variables was assumed equal 5. The second type of industrial data (Ind2 3cl, Ind2 5cl and Ind2 7cl data sets) correlates chemical composition of the ductile cast iron, defined by 9 elements with its tensile strength (details can be found in [23]).

Another type of industrial data were obtained as readouts from a semi-empirical nomograph which permits to calculate solidification shrinkage of grey cast iron as a function of four variables: carbon contents (5 different values – categories), sum of silicon and phosphorus content (4 values), casting modulus (4 values) and pouring temperature (4 values). In Ind3 data set the output was the iron shrinkage expressed by 7 different levels (classes) and in the last two data sets the outputs were the decision concerning necessity and size of application of feeders to avoid shrinkage defects: in Ind4 data set the output Feeder had 2 classes (No and Yes) and in Ind5 data set the output had 3 classes (No, Small and Large). Each of the last 3 data sets contained 190 records. Further details can be found in [24].

The requirements for rules and knowledge extraction tools formulated in Section II have brought about utilization of the procedures which ensured possibly the largest choice of rules available from the data, passing over possible overfitting of the models. Binary DTs were obtained using CART algorithm and MineSet commercial software package. Various splitting conditions, stopping criteria and pruning parameters were tried out. The smallest trees which ensured the smallest fraction of false predictions for training sets were chosen. RST procedure, oriented at generation of full set of rules, was written by the present authors with a somewhat similar approach as used in the Explore algorithm [25]. First, all the combinations of single input variables appearing in the data are placed in the rules (i.e. rules with only one conditions were generated) and their confidences are calculated. Then the further conditions are added, providing the confidence of a rule thus obtained is increased, compared to the rule with shorter conditional part.

B. *Results*

In Fig 1 the fractions of wrong predictions obtained from DTs for all consistent data subsets (i.e. all the discernible input values combinations pointing at one output value only) are shown, for all the training data sets. It can be observed that the fraction of wrong predictions for simulated data increases with the number of classes (categories) which can be attributed to a limited accuracy of DT models. The general level of false predictions for real data is much lower, compared to simulated data. An interpretation of this observation would require a deeper analysis of the data sets structures, e.g. representativeness of the classes of input and output variables.

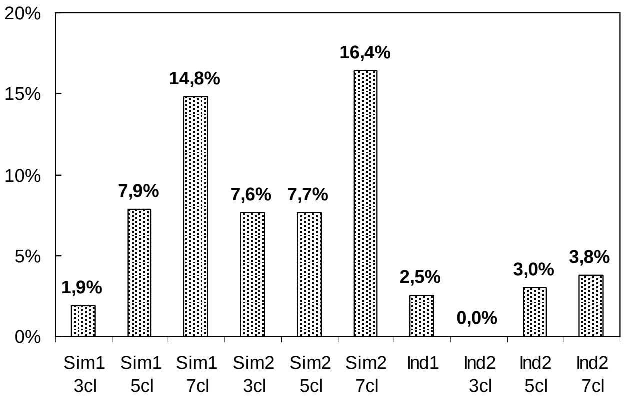

Fig. 1 Average fractions of false predictions obtained from DTs for consistent data subsets (including single records)

In Figs. 2 and 3 some statistical information obtained for inconsistent data subsets is shown. The ratios of the number of inconsistent data subsets to all subsets of the same input values, were fairly similar in all presented data sets (20% – 30%). The fractions of false predictions also result from the distributions of the classes in the inconsistent data subsets. It is interesting to note that in several cases DTs have pointed at the decision classes which are not predominant for the given combination of input values.

The results presented in Figs. 1, 2 and 3 indicate that the rules systems represented by DTs may be significantly incorrect for inconsistent data as well as for consistent data with variables large number of classes (categories) of variables.

In Fig. 4 the fractions of rules included in DTs which are not supported by data are shown, exhibiting quite large values in several cases.

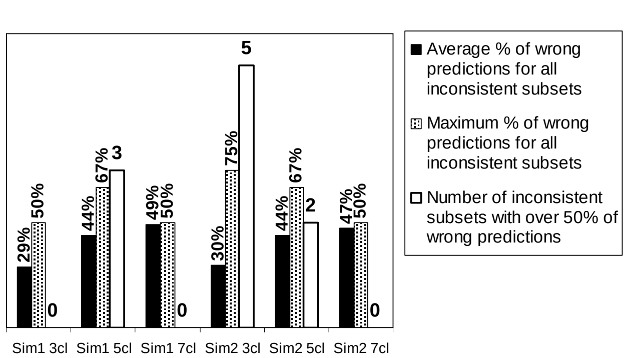

Fig. 2 Statistics of false predictions obtained from DTs for inconsistent data subsets for simulated data sets

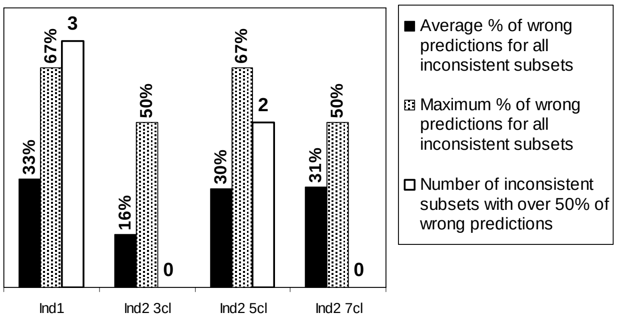

Fig. 3 Statistics of false predictions obtained from DTs for inconsistent data subsets for industrial data sets (for Ind3, Ind4 and Ind5 no inconsistent subsets were found)

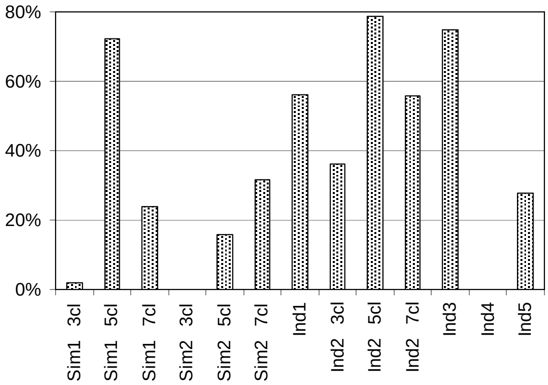

Fig. 4 Fractions of rules in DTs not supported by data

In principle, this can be a positive feature of DTs as such rules may be desired by a user (see comments in Section II). However, the usefulness of such rules may be questionable. First, because they do not necessarily meet the user's specific needs and second because their reliability, defined by confidence and support, is not determined, as pointed in Section III.

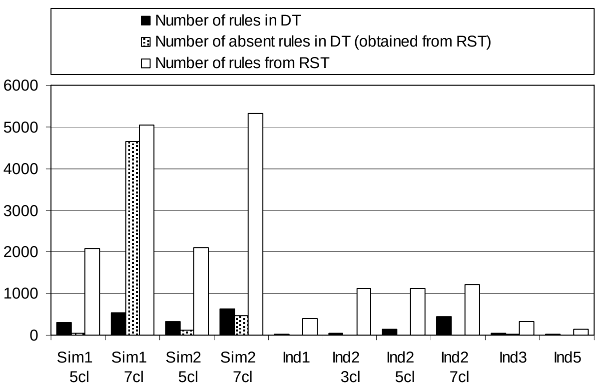

Fig. 5 Quantities of rules in DTs and obtained from RST – total and missing in DTs

In Fig. 5 the numbers of rules absent in DTs, but extracted by RST, are presented, together with total numbers of rules in DTs and from RST. Note, that if a conditional part corresponding to a RST rule was found in a longer DT rule, then such rule was not qualified as 'absent in DT'.

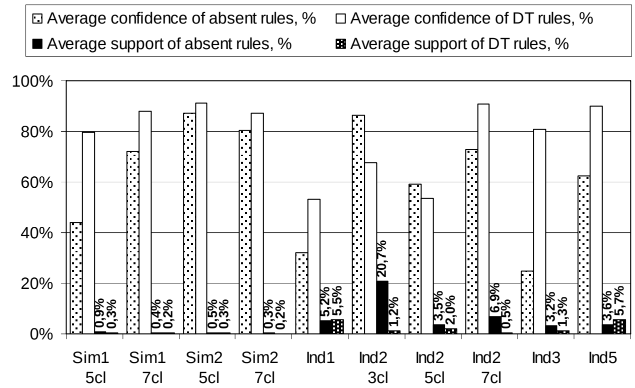

Fig. 6 Quality parameters of rules obtained from RST and omitted by DTs

The average confidence and support values of the missing rules in DTs are shown in Fig. 6, together with those for the rules which are present in DTs. It can be seen that the missing rules may be valuable for a user as their confidences are relatively high and comparable with those for the rules included in DTs. It is worth noticing that for some of the simulated data sets, some of the missing rules had 100% confidence. The support values are generally low for both groups of rules, which is obviously a result of the nature of the data sets. However, the support values for rules absent in DTs are often higher then for the rules present in DTs which is probably a consequence of the fact that the rules from RST do not have redundant conditions.

In Fig. 7 fractions of DT rules with redundant conditions are shown. Obviously, the RST rules taken as reference had this same confidence values.

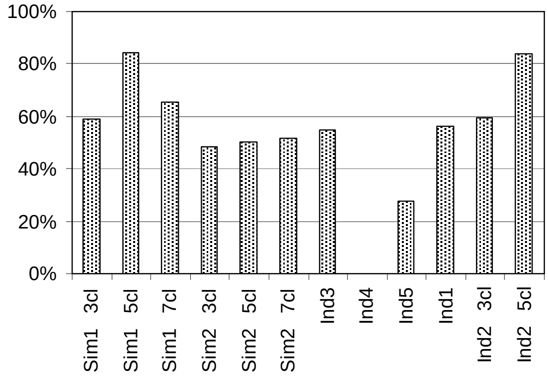

Fig. 7 Fractions of rules in DTs with redundant conditions

In Fig. 8 some characteristics of DT rules containing redundant conditions (denoted as 'oversized') are presented with reference to the corresponding rules obtained from RST (denoted as 'fitted'), for selected data sets. It can be seen that percent of redundant input variables in DT rules is high. The conclusion is that the presence of redundant conditions in rules obtained from DTs, being a result of the nature of that type models, may be their significant disadvantage. However, it is worth noticing that some DT induction algorithms, such as C4.5, contain a mechanism of dropping conditions that are irrelevant to the class, which may reduce the redundancies appearing in the rules.

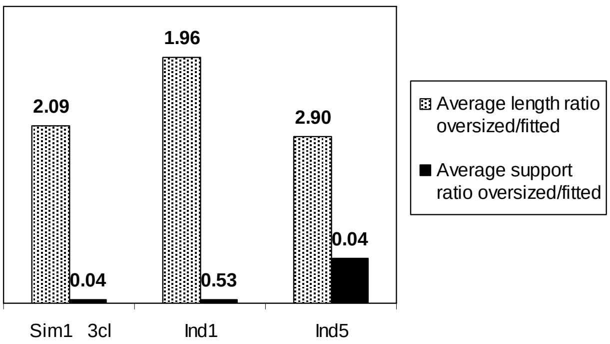

Fig. 8 Characteristics of DT rules with redundant conditions (oversized) compared to the corresponding rules obtained from RST (fitted)

Only very limited tests were made for predictions for new data, using Sim1 3cl data set and its 20 records representative subset. The results were substantially dependent on the selection of the new data. The results obtained for DTs for the cases where the predictions were available, appeared to be similar or significantly better for some selections of the new data, compared to RST, in spite of the fact that independent testing data were not used for pruning the trees. On the other hand, for several cases DTs were unable to give predictions for the desired new input values combinations, as mentioned in Section III. The relatively large fractions of false predictions for RST theory inclined present authors to treat this problem in a more detail in a separate work.

V. Conclusion

Decision trees have revealed several disadvantages as knowledge extraction tools for the applications where not only a characterization of a problem is required, but also detailed and precise rules are needed, according to actual, specific problems to be solved. For such applications rules obtainable from RST turned out to be generally better. However, an improvement of predictive capabilities of RST-based rules for new combinations of input values is needed.

Although the present paper is focused on industrial processes, it can be expected that the obtained results can be useful also for other application areas.

Acknowledgment

The authors are grateful to Prof. Jerzy Stefanowski from Institute of Computing Science of Poznan University of Technology, Poland, for his highly valuable advice in preparation of this paper.

References

[1] M. Shahbaz, M. Srinivas, J. A. Harding and M. Turner, „Product design and manufacturing process improvement using association rules", *Proc Inst Mech Eng Part B J Eng Manuf*, vol. 220, no. 2, pp. 243–254, 2006.

[2] A. Kusiak, "Data mining: manufacturing and service applications", *International Journal of Production Research*, 2006, vol. 44, no. 18–19, pp. 4175–4191, September 2006.

[3] J.A. Harding, M. Shahbaz, M. Srinivas and A. Kusiak, A. Data mining in manufacturing: A review. J Manuf Sci Eng Trans ASME, 2006, 128(4), 969-976, November 2006.

[4] K. Wang, "Applying data mining to manufacturing: The nature and implications", *J Intell Manuf*, vol. 18. no.4, pp. 487–495, August 2007.

[5] A. Mahl and R. Krikler, "Approach for a rule based system for capturing and usage of knowledge in the manufacturing industry", *J Intell Manuf*, vol. 18, pp. 519–526, July 2007.

[6] K. F. Tsang, H. C. W.Lau, and S. K. Kwok, "Development of a data mining system for continual process quality improvement", *Proc Inst Mech Eng Part B J Eng Manuf*, vol. 221, no. 2, pp. 179–193, 2007.

[7] C. H. Dagli and H. C. Lee, "Engineering smart data mining systems for internet aided design and manufacturing", *Int J Smart Eng Syst Design*, vol. , no. 4, pp. 217–225, 2001.

[8] W. C. Chen, S. S. Tseng, K. R. Hsiao and C. C. Liu, "A data mining project for solving low-yield situations of semiconductor manufacturing", in *IEEE Int Symp Semicond Manuf Conf Proc*, Boston, 2004, pp. 129–134.

[9] R. S. Chen, R. C. Wu, and C. C. Chang, "Using data mining technology to design an intelligent CIM system for IC manufacturing", in *Proc. Sixth Int. Conf. Softw. Eng. Atif. Intell. Netw. Parallel/Distr. Comput. Self-Assemb. Wireless Netw., SNPD/SAWN 2005*, Towson, MD, USA, 2005, pp. 70–75.

[10] J. Hur, H. Lee, F. G. Baek, "An intelligent manufacturing process diagnosis system using hybrid data mining", *Lect. Notes Comput. Sci.*, vol. 4065 LNAI, pp. 561–575, July 2006.

[11] M. Perzyk, R. Biernacki and J. Kozlowski, "Data mining in manufacturing: significance analysis of process parameters", *Proc Inst Mech Eng Part B J Eng Manuf*, vol. 222, no. 11, pp. 1503–1511, 2008.

[12] A. Kusiak and C. Kurasek, "Data mining of printed-circuit board defects", *IEEE Trans Rob Autom*, vol. 17, no. 2, pp. 191–196, April 2001.

[13] T. A. Etchells and P. J. G. Lisboa, "Orthogonal Search-Based Rule Extraction (OSRE) for Trained Neural Networks: A Practical and Efficient Approach", *IEEE Trans Neural Networks*, vol. 17, no. 2, pp. 374–384, Match 2006.

[14] R. K. Brouwer, "Fuzzy rule extraction from a feed forward neural network by training a representative fuzzy neural network using gradient descent", in *Proc IEEE Int Conf Ind Technol*, Hammamet, Tunisia, 2004, pp. 1168–1172.

[15] W. Duch, R., Adamczak and K. Grabczewski, "A new methodology of extraction, optimization and application of crisp and fuzzy logical rules", *IEEE Trans Neural Networks*, vol. 12, no. 2, pp. 277–306, March 2001.

[16] S. H. Huang and H. Xing, "Extract intelligible and concise fuzzy rules from neural networks", *Fuzzy Sets Syst*, vol. 132, no. 2, pp. 233–243, December 2002.

[17] H. Huang and D. Wu, "Product quality improvement analysis using data mining: A case study in ultra-precision manufacturing industry", *Lect. Notes Comput. Sci.*, vol. 3614 LNAI, pp. 566–580, 2006.

[18] L. Rokach, O. Maimon, "Data mining for improving the quality of manufacturing: a feature set decomposition approach", *J Intell Manuf*, vol. 17, no. 3, pp. 285–299, June 2006.

[19] D. Koonce, C. H. Fang and S. C. Tsai, "Data mining tool for learning from manufacturing systems", *Comput Ind Eng*, vol. 33, no. 1–2, pp. 27–30, Oct 1997.

[20] T. L. Tseng, M. C. Jothishankar, T. Wu, G. Xing and F. Jiang, "Applying data mining approaches for defect diagnosis in manufacturing

industry", *IIE Annual Conference and Exhibition*, Houston, 2004, pp. 1441–1447.

[21] L. Shen, F. E. H. Tay, L. Qu and Y. Shen, "Fault diagnosis using Rough Sets Theory", *Computers in Industry*, vol. 43, no. 1, pp. 61–72, August 2000.

[22] H. Sadoyan, A. Zakarian and P. Mohanty, "Data mining algorithm for manufacturing process control", *Int J Adv Manuf Technol*, vol. 28, no. 3–4, pp. 342–350, March 2006.

[23] M. Perzyk and A. Kochanski, "Prediction of ductile cast iron quality by artificial neural networks", *J Mater Proc Technol*, vol. 109, no. 3, pp. 305–307, February 2001.

[24] M. Perzyk and A. Soroczynski, "Comparison of selected tools for generation of knowledge for foundry production", *Archives of Foundry Engineering*, vol. 8, no. 4, p. 263–266, December 2008.

[25] J. Stefanowski and D. Vanderpooten, "Induction of decision rules in classification and discovery-oriented perspectives", *Int J Intell Syst*, vol. 16, no. 1, pp. 13–27, January 2001.