Publishing House
AKAPIT

COMPUTER METHODS IN MATERIALS SCIENCE
Informatyka w Technologii Materiałów

Vol. 14, 2014, No. 3

COMPARISON OF THE CBR METHODOLOGY AND THE CART ALGORITHM APPLIED TO PROBLEM OF CASTING DEFECTS CLASSIFICATION

KRZYSZTOF REGULSKI, GABRIEL ROJEK*

AGH University of Science and Technology
Faculty of Metals Engineering and Industrial Computer Science
Department of Applied Computer Science and Modelling
al. A. Mickiewicza 30, 30-059 Krakow
**Corresponding author: rojek@agh.edu.pl*

Abstract

The main scope of presented in this article research is the analysis of application of artificial intelligence methodologies at building of a computer system that should aid at problem of casting defects classification. The computer system is designed as a decision support tool in the diagnosis of casting defects for small and medium-sized plants, which implies restrictions according to the usage of data that, in this case, are not measured in real time of production. Without access to control data, the diagnosis of casting defects can be based on observations made by a technologist responsible for the inspection of ready castings. Those observations concern usually the type of damage, its distribution, location, occurrence or even color of surface. The problem of such observation based diagnosis can be resolved by building of a computer tool, which uses classification methodologies in order to give aid at casting defects classification. Presented research focus on two methodologies within artificial intelligence: Case-Based Reasoning (CBR) and Classification And Regression Trees (CART). The CBR methodology enables to use knowledge according to previously made classifications in order to help predict the present classification problem. The decision support system with the applied CBR methodology is able to learn basing on the knowledge which is acquired in the result of classifications performed by this system. The CART algorithm enables to generate classification tree, which can be easily used by a technologist or by an expert system, giving support at defect diagnosis. Presented in this article research concerns comparison of those two methodologies in terms of their usefulness at designing the system operating in conditions of small and medium-sized casting factory.

Key words: casting defects, classification, Case-Based Reasoning, Classification And Regression Trees

1. INTRODUCTION

Building of computer system which is a tool for diagnosis and classification of casting defects is a problem known and studied for a long time. Previous works of the authors in this scope (Górny et al., 2010) show that small and medium-sized foundries often lack proper equipment allowing continuous measurement of parameters, which are responsible for formation of casting defects. This is the reason for late detection of defects, which is usually done at the stage of technical inspection of ready made products, that is, at the end of the entire production process. In such situations, it is difficult to evaluate the causes of defects. The evaluation is usually based on the experience of an expert being a technologist, who by proper classification of the type of the defect can predict the production parameters, which may be responsible for its formation. Such role of the expert detecting casting defect shows that the most important issue is proper classification of defect, which is the main basis for later improvement of products quality. The main goal of the research presented here is to support the diagnosis of defects under the technical conditions

 ISSN 1641-8581

of an individual foundry, where specific types of defects occur. Such support may significantly accelerate the discovery of defects and improve the effectiveness of their classification in order to reduce production costs.

2. CASTING DEFECTS

The quality of castings is determined by a number of parameters that are important at different stages of the casting process. Defects that may result from improper choice of parameters of the moulding and core sand include: pinholes resulting from the presence of hydrogen, sand inclusions, deformations, gas inclusions, fractures, and shape imperfections. Defects that may result from improper construction or improper assembly of the pattern and mould (Baler & Köppen, 1994) are fractures, shape imperfections, sand inclusions, misruns, cracks, gas inclusions, surface defects, injry (mechanical damage), knob, flash, mismatch, pushing up and warping. Gases tend to dissolve in liquid steel at all stages of castings production, i.e. during melting in the furnace, during tapping, during pouring of moulds, and even after pouring of the mould before complete solidification of the casting takes place. Therefore, reducing or eliminating casting defects such as blowholes, voids in the cast structure, pinholes, non-metallic inclusions or porosity, and scaling on the surface of casting requires strict control of the whole process of melting and casting. Control of the charge and its compliance with technological regime during melting of alloys in the furnace for casting are particularly important in the absence of vacuum treatment of liquid metal (in an induction furnace or ladle). Possible defects caused by faulty melting process include misruns, slag inclusions, tears (caused by excessively high temperature), gas inclusions, incorrect chemical composition, and pinholes.

2.1. Casting defects diagnosis

Classification of the type of defect depends on its visible and hidden features such as: the type of damage, its visibility, size of damage, amount of material, distribution, location, mould material, inclusions, rate of occurrence, configuration, penetration, surface color, orientation, shine, surface oxidation, surface of defect. Identification of visible attributes is performed by the technologist who observes the surface of made products. However, identification of hidden features involves deeper research and is often costly and time consuming so it is usually not performed by small and medium-sized foundries. Therefore, the technologist evaluating the type of defect does not always have complete and accurate knowledge of the defect. With incomplete and improper information, he frequently makes use of his experience in order to take decision.

Without complete knowledge of the defect, which would usually make it possible to avoid costly and time-consuming research, the technologist has to state his diagnosis. Such diagnosis which indicates several possible types of defects featuring the described attributes, helps considerably in the process of quality improvement. From a point of view of small or medium-sized foundries, it can be also stated, that certain types of defects are more common than others, and thus it is easier to choose the right one.

2.2. Formalization of knowledge of casting defects

Former research on casting defects (Fałęcki, 1997), standards related to defects in castings (e.g. PN-85/H-83105) and Atlas of Defects in Castings

Table 1. *Fragment of knowledge base concerning attributes of casting defects*

Damage name	Damage type	Distribution	Location	Occurrence	Damage shape	Technological operation
Cold lap	wrinkles, scratch, erosion scab	local	insert wall, chaplet surface	numerous	narrow, rounded edges	casting design, pouring, cooling
Cold lap	fissure, scratch	local	Surface	single	narrow, rounded edges	gating system design, pouring
Cold shots	metal beads		Interior		spherical	gating system design, pouring
Cold lap, cold shots	discontinuity, fissure	widespread	surface, sub-surface area	numerous	rounded edges, narrow	feeding system design, pouring
Cold lap near core or other metallic part	discontinuity	local	near inserts		curved walls	pouring, solidification

(2004) indicate possible features of different types of casting defects. Based on these sources, the descriptions of defects were collected in the form of sets of attribute values and later specified in an array of attributes (Górny et al., 2010). This array, a fragment of which is illustrated in table 1, is the origin of knowledge base which must be made use of by computer methods in building a computer system helping at classification problem of casting defects.

Considering the computer interference techniques presented in subsequent chapters, the knowledge base in this form has to be brought to a form in which only one record of data corresponds to each individual value of attributes. It should be noted that many records reproducing all possible combinations of permissible attribute values may correspond to a single type of defect.

3. THE CBR METHODOLOGY APPLIED TO THE PROBLEM OF CASTING DEFECTS CLASSIFICATION

The main paradigm of the Case-Based Reasoning methodology is the inference regarding the current problem by reuse of knowledge relating to previously solved problems, what makes this approach different from other methodologies within artificial intelligence e.g. expert systems or systems based on fuzzy logic paradigm. The CBR methodology also provides a possibility of learning by considering the current results of system functioning in later made inference (Bergmann et al., 2009). This characteristic should provide customization of the system to the specific nature of the problem solved, which is important for all problems related to the diagnosis of defects in small and medium-sized enterprises active in the foundry industry. As presented e.g. by McKenna and Smyth (2000) the Case-Base Reasoning methodology can be used in order to implement classification systems, which constitutes another reason for using this methodology in solving the casting diagnosis problem.

3.1. Construction of CBR system

Preliminary task at designing the CBR system is the construction of a case base that contains all information regarding the domain of the problem being solved. That knowledge is stored in the form of cases, corresponding to the previously occurred and solved cases of classification problem. The case is a formal description of an individual problem and its solution. In the domain of casting defects diagnosis the problem is described by attributes of a defect and the solution is its classification to one of several types of casting defects. Taking into consideration the remarks presented in previous sections on knowledge concerning casting defects and its formalization, a single case consists of sixteen attributes specifying the problem (e.g. size of the damage, location and inclusions) and one attribute that specifies the solution being the type of casting defect. Fourteen of the attributes specifying the problem have discrete values, while two of them are in the form of real numbers from a fixed range.

Before the first run of the CBR application the case base should be filled with cases related to previous experiences in the domain of casting defects diagnosis, however cases formed during functioning of the implemented system will be added to this case base. Fragment of case base being filled with data resulting from sources of knowledge concerning casting defects:

Case 1, snagging, 6.0, clear, insufficiency, local, surface, metal mould, not observed, 1.0, irregular, superficial, metal color, protruding, mat, oxidized, no data, injry

Case 2, snagging, 6.0, clear, insufficiency, local, surface, metal mould, not observed, 1.0, irregular, superficial, metal color, shifted, mat, oxidized, no data, injry

Case 3, snagging, 6.0, clear, insufficiency, local, surface, metal mould, not observed, 1.0, irregular, superficial, metal color, adjacent, mat, oxidized, no data, injry

As has been mentioned earlier, every case contains 16 attributes concerning observations made according to a casting defect and one attribute concerning the name of type of this casting defect. The first attribute of three presented cases has a value 'snagging' which is a description of a type of damage. The second attribute has value 6.0, which indicates that the defect is well visible. The third one has value 'clear', which means that the size of damage is clearly visible without any additional optical instruments. The type of casting defect is specified at the end of every case – it is 'injry' (that means mechanical damage).

The main algorithm of every CBR application consists of the following four sequential phases, called a CBR cycle (Aamodt & Plaza, 1994):

1. *Retrieve* the most similar case (or cases) from the *case base*

2. *Reuse* the information and knowledge in the retrieved case (cases) in order to find a solution for a current problem
3. *Revise* the proposed solution
4. *Retain* the parts of this experience in the *case base* to be used in future

CBR cycle begins when the current case is introduced to the inference system, which defines a new problem to be solved, meaning that a new defect has to be classified. After input of all attributes describing presently solved problem, the algorithm moves to the retrieve phase. In this phase the inference system searches the case base in order to find a past case, which is most similar to the current problem. The most important task of this phase is to define the function, which is a measure of the similarity between the description of the current problem and the description of the problem, which is one of the cases included in the case base. This function returns information about the full similarity if the attributes, that are strings, have consistent values. For attributes that are real numbers, the similarity function uses the measure of Euclidean distance. The item found in the database of past cases, which has the highest similarity to the current problem, becomes the basis of next phase i.e. the phase of reuse.

In the phase of reuse, the solution of the found out case is adapted or, as usually in classification systems, directly copied to the current case and returned as the solution for the current problem. The phase of revision makes an assessment of the solution that has been returned by the system in the previous phase. This assessment usually cannot be performed automatically and requires expert intervention. In this phase, some verifications of the proposed solution are possible if such is the suggestion of an expert. After the revision phase, the retention phase follows. Its aim is to add the current case with ready description and solution of the problem to the case base. This adding allows the CBR system to learn, i.e. to use knowledge of the problems solved by the system in later inference. After completing the database of cases, the system is ready to solve another new problem, which involves the performance of the next full CBR cycle.

3.2. Implementation of CBR system

As part of the presented here research, a CBR system was implemented according to presented in previous subsection remarks on its design. The created system has been implemented using Java and jCOLIBRI programming library, which contains a set of tools helpful at implementation of the CBR system. The main programming activities are related to determining the development of the formal description of a single case, creating a database of cases and determining the course of the four phases of the CBR cycle. Figure 1 presents user interface of the developed application.

After the developed computer system starts operating all fields which contain values of attributes or name of the type of defect are empty. The use of the system can be presented in five steps:

1. The user specifies the problem – feeds in the values of individual parameters of the defect according to the observations performed (type of damage, visibility, size of damage etc.) and presses “Find the type of the defect” button.
2. The system performs the retrieve phase – it retrieves the most similar case in the case base.
3. The system performs the reuse phase – as a solution to the problem, the system indicates the name of the defect by copying it from the case being found in the retrieve phase.
4. The system performs the revise phase – the user has the ability to put in a different name of the defect than the name that was returned by the system in the reuse step. This is important when the system is supervised by an expert (particularly at the initial period of the system usage).
5. The system performs the retain phase – adds to the case base the current case described as the problem (associated with the input of attributes in the first step) and its solution in the form of the name of defect type.

After the retain phase the case base is supplemented with knowledge related to the currently solved case. The way of description of this case does not differ from those initially introduced, which allows its use in further operation of the system to support the diagnosis of casting defects. It is important to make the system operate within one company since it enables a spontaneous adjustment of the system to the specific characteristics of the production process (it occurs through adding problem cases and their solutions which presently occur in the enterprise).

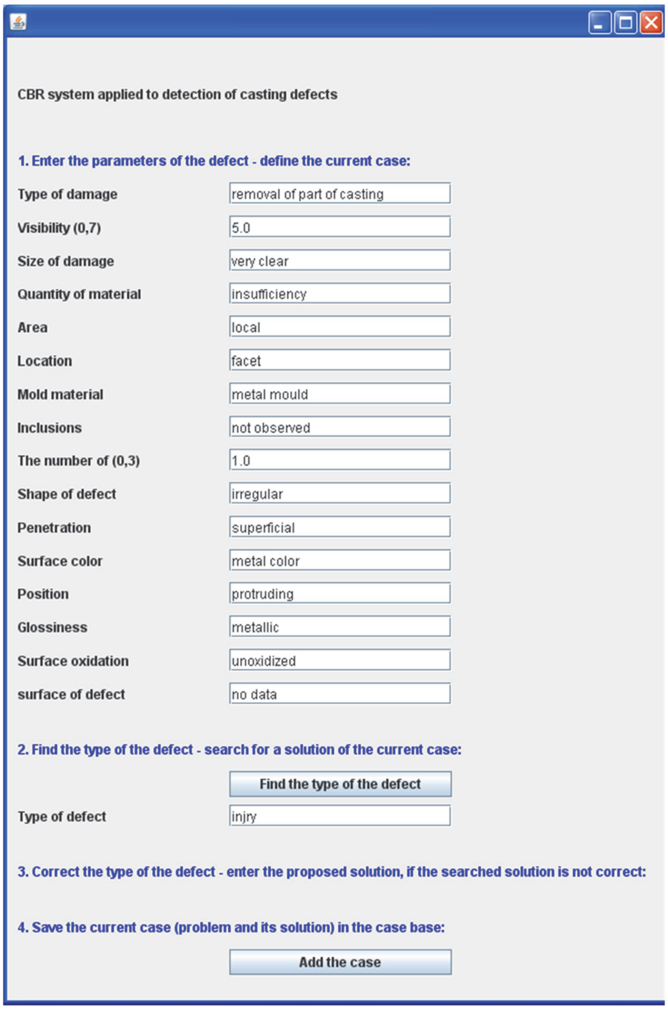

Fig. 1. *User interface of the developed system with the applied CBR methodology.*

4. THE CART ALGORITHM APPLIED TO THE PROBLEM OF CASTING DEFECTS CLASSIFICATION

Decision trees are an advanced form of knowledge representation, which creates rich interpretive possibilities, both at the stage of knowledge acquisition (data mining) and in the phase of its subsequent use in a decision-making process. Decision tree is a graphical method of supporting the decision-making process. It is a tree structure in which internal nodes contain tests on attribute values, and leaves determine decisions about the classification of objects. The decision tree is a graphical encoding of a series of conditional statements (Regulski & Kluska-Nawarecka, 2012).

Building of a decision (induction) tree requires a corresponding resource of data, which are acquired by the methods of exploration of large data sets, or may result from physical experiments carried out. Classification can occur when dependent variable is categorical and its possible values (or membership in the class or group) should be known on the basis of the knowledge of the value of one or more continuous predictor variables and, possibly, categorical variables. The aim of this model is to predict the value of dependent variable on the basis of the variation of the independent variables. Tree-building algorithm partitions a set of learning objects (samples) iteratively until each partition (node) contains the data belonging to one class. This allows construction of rules of inference that is based on explanatory (independent) variables.

The methodology using the CART (Classification And Regression Trees) algorithm solves the classification problem by building a binary decision tree according to some splitting rules based on the predictor variables. In this method, the space of predictor variables (whole dataset is enclosed in the root node) is partitioned recursively in a binary way. Predictor variables can be both: continuous or qualitative. The partition is intended to increase node purity. The partitioning is repeated until new splits do not improve homogeneity. When the splitting is stopped, the terminal node is called a leaf. Prediction is determined by leaves, and takes a form of a class value (Breinman et al., 1984).

Induction of the tree with the CART algorithm is conducted as follows (Williams, 2011):

Step 1: CART starts with the first variable, splits a variable at all of its possible split points (at every value that the variable takes in the sample). Then algorithm examines every allowable split on each predictor. The split is a test on a variable. If variable X_i is categorical taking a finite number of values $\{c_1, c_2, .. c_k\}$, split takes form: $X_i \in C$ as C ranges over all subsets of $\{c_1, c_2, .. c_k\}$. In case of continuous variable, split takes form: $X_i > c$ for all c in the range of X_i. As there are many variables to consider, there is a large number of possible choices for split points. Step 1 is repeated for each of the remaining variables at the root node.

Step 2: CART ranks all of the best splits on each variable according to the reduction in impurity achieved by each split. Then algorithm evaluates goodness criteria of a split to each split point and evaluates the reduction in impurity, or heterogeneity due to the split. CART selects the best of the splits (the best in criterion of lowest impurity of descendant nodes). In CART algorithm the partitioning is based on the Gini measure:

$$G_I(t) = \sum_k p(j|t)p(i|t)C(i|j) \tag{1}$$

where the sum extends over all k categories, $p(j|t)$ is the probability of category j at the node t and $C(i|j)$

is the probability of misclassifying a category j case as category i. Impurity measure becomes zero only when in the node there are cases which belong to only one class. If the apriori probabilities are determined on the basis of class frequencies, misclassification costs are equal, the Gini measure of heterogeneity in the node is calculated as the sum over all pairs of classes in the node. The maximum is reached when class frequencies in the node are equal. However, when all cases in the node belong to one class, the Gini measure is 0 (Hill & Lewicki, 2007).

Step 3: Node splitting stop when the stop criterion is satisfied.

The most popular strengths of the CART algorithm are (Yohannes & Hoddinott, 1999):

- the explanatory variables can be qualitative and continuous;
- CART does not require any apriori knowledge about the researched data/phenomena, no distributional assumptions;
- building tree process helps in evaluation of the significance of explanatory variables;
- CART is not at all affected by outliers, collinearities, heteroscedasticity. Outliers are isolated into a node, and do not have any effect on splitting;
- CART can detect relationships among variables in the data set;
- algorithm deals with a large number of variables submitted for analysis, it can produce useful results using only a few important variables.

Application of the CART algorithm should lead to the creation of such a model tree, for which the predictive ability is the greatest, i.e. the variance for each class is the smallest. The algorithm of the CART allows not only creating the decision rules, but also determining the validity of individual predictive variables. A given variable is considered as important in the classification process (i.e. carrying information about the class), if this variable often participates in the process of classifying objects from the training set. The attribute 'readiness' to participate in the classification process is measured during the construction of classification trees. On this basis, the model makes the ranking of variables in respect of their weights. Validity means a high degree of covariance (expressed as covariance or correlation) of a given factor with the dependent variable. The validity of predictors is calculated as the sum over all nodes of the tree of the node purity and is expressing this value as a fraction of the maximum amount (for all predictors). The predicative variables may assume values <0,1> where 0 means that the variable has no effect on the dependent variable and 1 means the greatest impact.

The effectiveness of the model generated with the CART algorithm can be measured with the use of the number of correctly (or incorrectly) classified cases.

4.1. Results of the CART algorithm application

The CART algorithm was used in order to generate classification tree according to the data described in section 2.2. CART algorithm is implemented in several commercial statistical computing packages such as SPSS, SAS and STATISTICA, often as add-on modules and libraries. Authors use STATISTICA to generate classification tree.

The result of the algorithm is presented in Figure 2. The classification tree obtained from CART algorithm can be also transformed to the set of rules. For example: take under consideration leaf ID=6, it indicates that the node of the defect called 'pinholes' is the most frequent. Global frequency of that leaf is N=180 records. All of them are pinholes, that means the rule is 100 percent sure. We can transform the whole branch of the generated tree into a rule:

IF 'type of damange' is not 'lack of part of casting' **AND** 'penetration' is not 'perpendicular to the surface' **AND** 'inclusions' are 'grafit' **THEN** 'type of defect' is 'pinholes'

The whole tree can be formulated as a set of rules (with probability of correct classification):

1. IF type of damage = 'lack of part of casting' THEN type of defect = 'short run' (100%)
2. IF type of damage is not 'lack of part of casting' AND penetration is 'perpendicular to the surface' THEN type of defect = 'flach' (87,5%)
3. IF type of damage is not 'lack of part of casting' AND penetration is not 'perpendicular to the surface' AND inclusions is 'graphite' THEN type of defect = 'pinholes' (100%)
4. IF type of damage is 'shape distortion' AND penetration is not 'perpendicular to the surface' AND inclusions is not 'graphite' AND mould material = 'wet mass' THEN type of defect = 'swell' (52%)
5. IF type of damage is 'shape distortion' AND penetration is not 'perpendicular to the surface' AND inclusions is not 'graphite' AND mould material = 'metal mould' THEN type of defect = 'distortion' (100%)

6. IF type of damage is 'fracture' AND penetration is not 'perpendicular to the surface' AND inclusions is not 'graphite' THEN type of defect = 'injry' (100%).

able means less uncertainty of rules which in turn means fewer misclassifications.

There are six rules in the presented tree. Four of them are 100 percent sure (certain) – that means, in

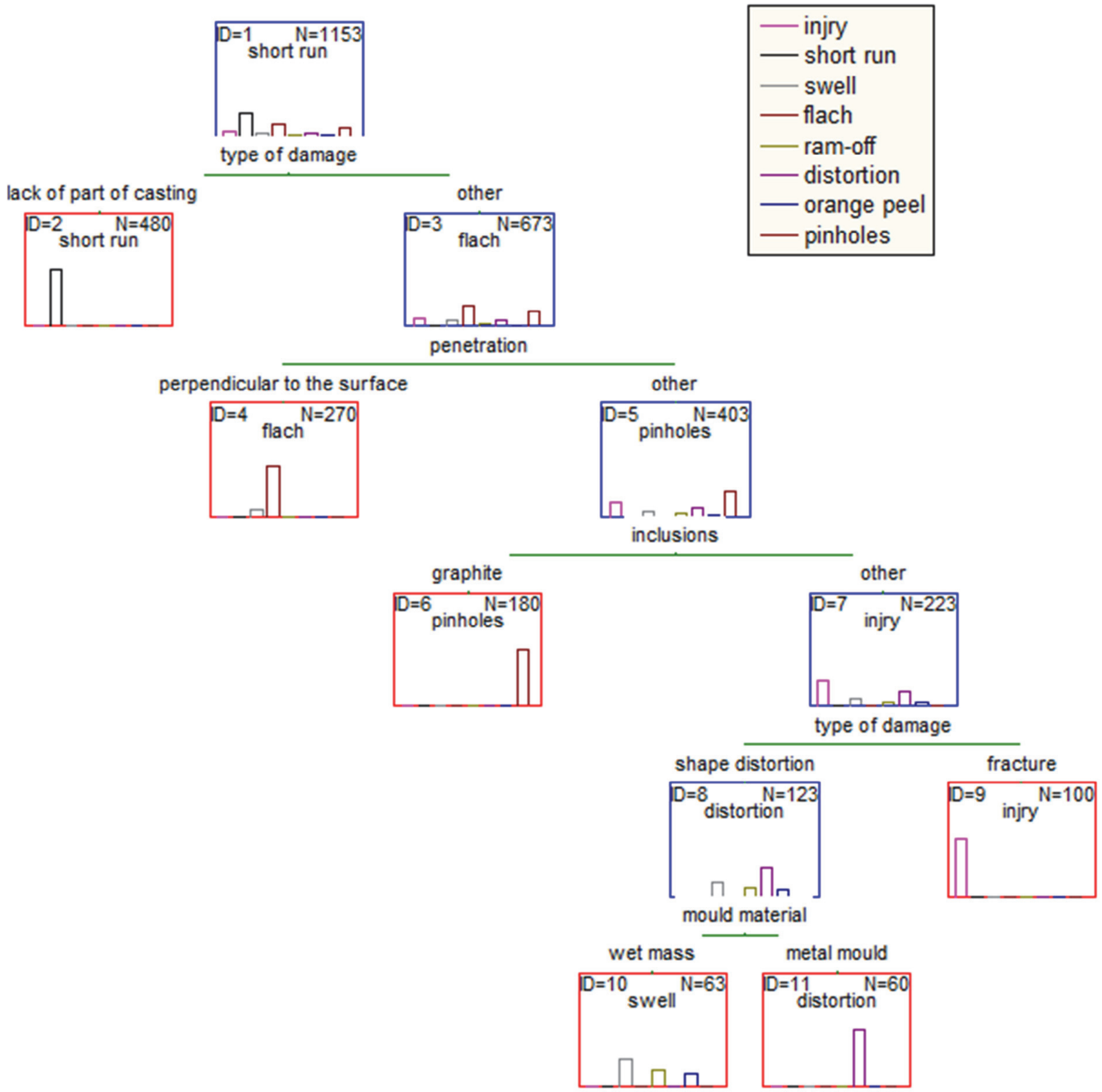

Fig. 2. Decision tree generated with CART algorithm.

The generated tree gives six rules that establish knowledge base for decision support system for casting defects diagnosis. It is very important to determine how uncertain are those rules. CART algorithm is not able to detect probability of its classification error in other way than by history of misclassification on the training and test dataset. Classification error is associated with impurity of leaves, and impurity is related to 'noise' in dataset – that means misclassification is always concerned as a consequence of data used to machine learning. More data representing each class of dependent variable means less uncertainty of rules which in turn

each leaf all cases were from the same class. Leaves were 100 percent homogenous. Two of the rules were uncertain. We can assume that probability of correct classification with the application of these rules in case of rule no. 3 is 87,5% and for rule no. 4 it is only 52%. To decrease that uncertainty we should obtain more cases for 'flach' and 'swell'.

Classification error is only present for class 'swell' and 'flach' (as presented in figure 3). It follows the selection of cases for defects learners: ram-off, orange peel and swell. Algorithm was not able to learn a rule for ram-off and orange peel due to a small number of cases indicates those defects. All cases concerning one of these defects are put in one node with swell which was the nearest defect in relation to attribute values. Once again: the more cases the better trained tree.

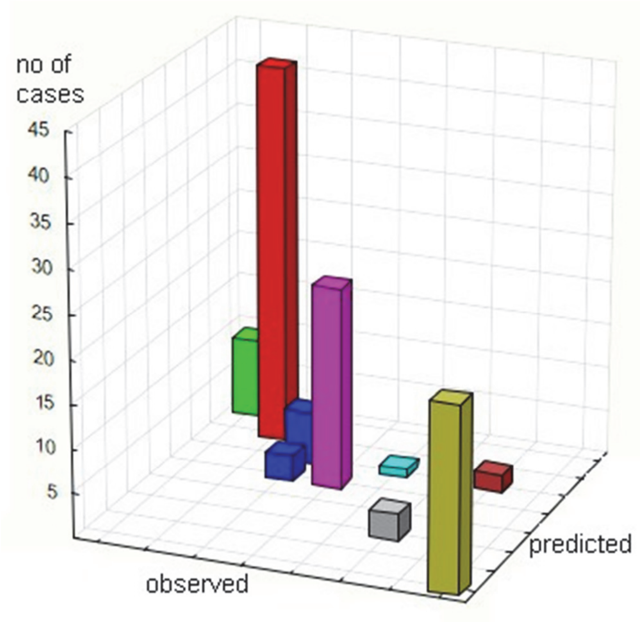

Fig. 3. Classification matrix for the decision tree in graphic form. Each row represents one of the type of defects. Observed from the left: injry, short run, swell, flach, ram-off, distortion, orange peel, pinholes.

variables taken into account are the most important for classification. We are not obligated to choose predictors as those are chosen at the process of tree induction. However it is possible to check ex-post how important they were. Figure 4 shows the graph of predictors significance calculated during tree-building process.

5. CONCLUSIONS REGARDING THE CBR METHODOLOGY AND THE CART ALGORITHM

The Case-Based Reasoning methodology allows solving the problem of classification. Additionally, the system with applied CBR methodology is able to learn through adding the data related to classifications made by the system. The data added to the case base is the source of knowledge of the system. Due to this learning the user is not obligated to rebuild the model or to launch another additional module for the learning mode. The system functioning in one individual foundry should adapt to specific problems of casting in this enterprise, which is possible also thanks to the learning performed together with problem solving as the main paradigm of the CBR methodology.

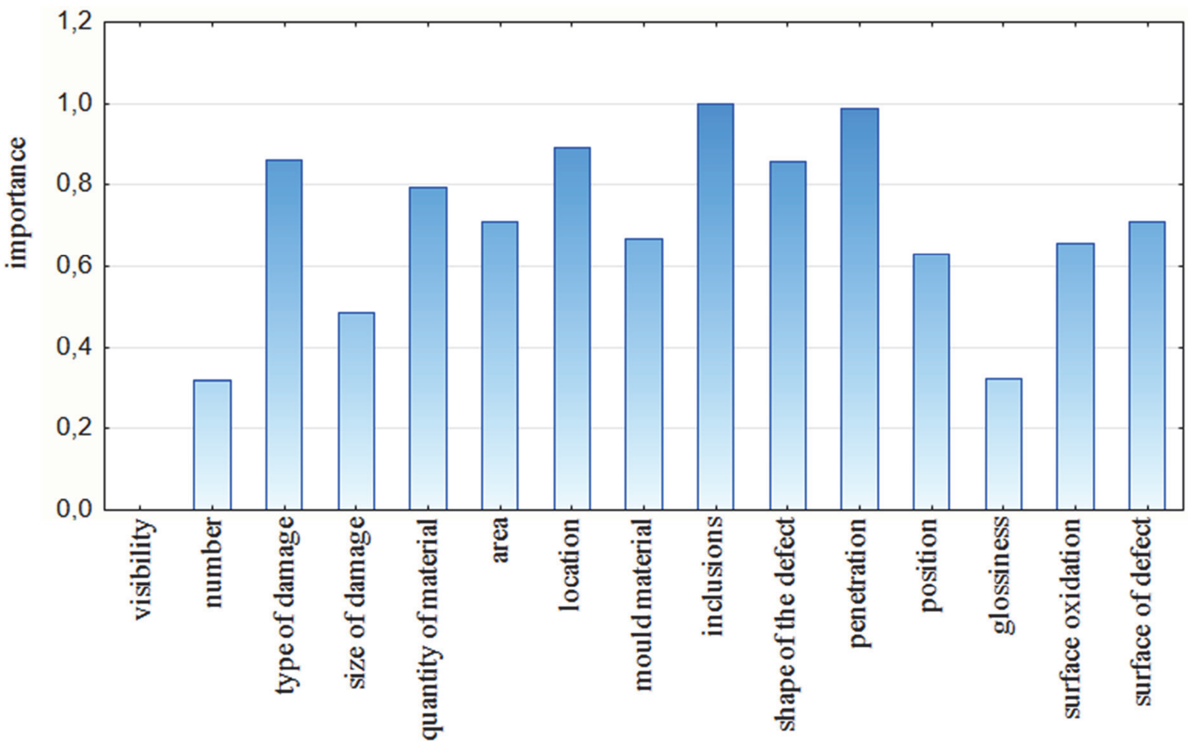

Fig. 4. Significance of predictors calculated during tree-building process.

As it has been said, the CART algorithm allows determining the validity of individual predictive variables. We can indicate which of explanatory

Classification of each case in the CBR approach involves searching for one, the most similar case from the database. The method can effectively assign a scenario even if a new case of defects is 'quaint' (if only one similar case was found in the knowledge base, this case is a basis for solution proposed by the developed system). In contrast to the CBR methodology, the CART algorithm is insensitive to outliers cases – a singular objects can be wrongly classified. Classification and regression trees are used for making generalizations from the set of cases. Learning a tree (induction rules) goes along with the development of the knowledge base; significant changes occur when changing the structure of the population, so a single record usually cannot affect the form of the model – outliers are put in the one, heterogeneous node. It should be noted, that if new cases are added (in extreme situation even only one case) other variables become more significant and will be chosen as the first split. This will change the tree completely. It proves that the CART is not a stable method and the addition of new cases requires building a new model. Moreover, the resulting model can be quite different from the previous one. It is simultaneously a feature of the model, and the drawback. On one hand the algorithm is not able to learn about less common cases, but on the other, the model is insensitive to noise in data and erroneous cases (e.g. wrongly classified by the user in past made classification).

The CART algorithm generates rules that represent generalized knowledge. These rules are understandable to the technologist who can use them to verify the established diagnosis. These rules can also be applied to reasoning systems (e.g. expert systems). The reasoning system constructed in such a way will operate basing on the rules generated earlier. Therefore the learning is not possible. In opposition to the methodology using the CART algorithm, the CBR methodology focuses on solving individual problems instead of generalizing the knowledge being the basis for the obtained solutions.

Concluding the remarks presented in the paper on application of the CBR methodology and the CART algorithm to problem of casting defects classification, it has to be stated that each individual approach has its own strengths and weaknesses. Comparing these two methodologies, it can be concluded that both algorithms complement each other perfectly. When the CBR methodology fails (in extracting of generalized knowledge from dataset), the CART algorithm can do the job. If, on the other hand, the CART algorithm is helpless (at situation of outliers and small number of cases or in learning), the CBR approach can provide assistance.

In further works the authors would like to undertake the research oriented on designing a coherent decision support system that implements both presented methodologies within a single computer application, using each of the models in the context of needs.

ACKNOWLEDGEMENTS

Financial support of the Ministry of Science and Higher Education (AGH UST, project no. 11.11.110.300) is acknowledged.

REFERENCES

Aamodt, A., Plaza, E., 1994, Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches, *AI Communications*, 7, 1, 39-59.

Atlas of defects in castings, 2004, collective work, Foundry Research Institute in Cracow, Cracow.

Baler, J., Köppen, M., 1994, *Casting Defects Handbook. Disadvantages associated with the masses and prevention flasks*, IKO-Erbslöh.

Bergmann, R., Althoff, K. D., Minor, M., Reichle, M., Bach, K., 2009, Case-Based Reasoning, Introduction and Recent Developments, *Kunstliche Intelligenz: Special Issue on Case-Based Reasoning*, 23, 1, 5-11.

Breinman, L., Friedman, J. H., Olshen, R. A., Stone, C. J., 1984, *Classification and regression trees*, Wadsworth and Brooks, California.

Fałęcki, Z., 1997, *Analysis of defects in castings*, AGH, Kraków.

Górny, Z., Kluska-Nawarecka, S., Wilk-Kolodziejczyk, D., Regulski, K., 2010, Diagnosis of casting defects using uncertain and incomplete knowledge, *Archives Of Metallurgy And Materials*, 55, 3, 827-836.

Hill, T., Lewicki, P., 2007, *STATISTICS: Methods and Applications*, StatSoft, Tulsa OK.

McKenna, E., Smyth, B., 2000, Competence-Guided Case-Base Editing Techniques, *Lecture Notes in Computer Science*, 1898, 186-197.

Regulski, K., Kluska-Nawarecka, S., 2012, *Knowledge integration computer tools and algorithms in the improvement of the production processes of cast-steel castings*, *Artificial Intelligence in the Knowledge and Information Systems*, Instytut Odlewnictwa, Kraków.

Williams, G., 2011, *Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery*, Springer, New York.

Yohannes, Y., Hoddinott, J., 1999, *Classification and Regression Tress: An Introduction*, International Food Policy research Institute, Washington, U.S.A.

PORÓWNANIE METODOLOGII CBR ORAZ ALGORYTMU CART W ZASTOSOWANIU DO ROZWIĄZANIA PROBLEM KLASYFIKACJI WAD ODLEWNICZYCH

Streszczenie

Przedstawionym w artykule głównym obszarem badań jest analiza zastosowania metod sztucznej inteligencji w budowaniu systemu komputerowego, mogących wspierać problem klasyfikacji wad odlewniczych. System komputerowy jest projektowany jako narzędzie wspomagania decyzji w diagnostyce wad odlewniczych dla małych i średnich zakładów, co powoduje ograniczenia w wykorzystaniu danych, w tym przypadku bowiem parametry procesu nie są mierzone w rzeczywistym czasie produkcji. Bez dostępu do danych odnoszących się do parametrów sterowania, klasyfikacja wad może opierać się na obserwacjach dokonywanych przez technologów odpowiedzialnych za badanie gotowych odlewów. Obserwacje dotyczą zazwyczaj rodzaj uszkodzenia, jego rozmieszczenia, położenia oraz występowania lub nawet koloru powierzchni badanego materiału. Problem rozpoznawania typu wady może zostać rozwiązany poprzez budowę narzędzia komputerowego, które używa metod klasyfikacji w celu wsparcia użytkownika w zakresie poprawnej detekcji wad. Badania koncentrują się na dwóch metodologiach z zakresu sztucznej inteligencji: wnioskowaniu epizodycznym (CBR) oraz drzewach klasyfikacyjnych i regresyjnych (CART). Metodologia CBR pozwala na wykorzystaniu wiedzy o poprzednio dokonanych klasyfikacjach w celu przewidywania rezultatu bieżącego problemu klasyfikacji. Metodologia CBR umożliwia również naukę systemu wspomagania decyzji na podstawie wiedzy o dokonywanych przez ten system klasyfikacjach. CART pozwala generować drzewo klasyfikacyjne, które może być łatwo użyte przez technologów albo może być wykorzystywane przez system ekspertowy. Prezentowane porównanie dotyczy użyteczności tych metod w projektowaniu systemu działającego w warunkach małego i średniego zakładu odlewniczego.

Received: October 28, 2012
Received in a revised form: November 21, 2012
Accepted: December 2, 2012