Int. J. Appl. Math. Comput. Sci., 2012, Vol. 22, No. 2, 477–491
DOI: 10.2478/v10006-012-0036-3

amcs

IMPROVING PREDICTION MODELS APPLIED IN SYSTEMS MONITORING NATURAL HAZARDS AND MACHINERY

MAREK SIKORA [*,**], BEATA SIKORA [***]

[*] Institute of Informatics
Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland
e-mail: Marek.Sikora@polsl.pl

[**]Institute of Innovative Technologies EMAG
Leopolda 31, 40-189 Katowice, Poland

[***]Institute of Mathematics
Silesian University of Technology, Kaszubska 23, 44-100 Gliwice, Poland
e-mail: Beata.Sikora@polsl.pl

A method of combining three analytic techniques including regression rule induction, the k-nearest neighbors method and time series forecasting by means of the ARIMA methodology is presented. A decrease in the forecasting error while solving problems that concern natural hazards and machinery monitoring in coal mines was the main objective of the combined application of these techniques. The M5 algorithm was applied as a basic method of developing prediction models. In spite of an intensive development of regression rule induction algorithms and fuzzy-neural systems, the M5 algorithm is still characterized by the generalization ability and unbeatable time of data model creation competitive with other systems. In the paper, two solutions designed to decrease the mean square error of the obtained rules are presented. One consists in introducing into a set of conditional variables the so-called meta-variable (an analogy to constructive induction) whose values are determined by an autoregressive or the ARIMA model. The other shows that limitation of a data set on which the M5 algorithm operates by the k-nearest neighbor method can also lead to error decreasing. Moreover, three application examples of the presented solutions for data collected by systems of natural hazards and machinery monitoring in coal mines are described. In Appendix, results of several benchmark data sets analyses are given as a supplement of the presented results.

Keywords: natural hazards monitoring, regression rules, time series forecasting, k-nearest neighbors.

1. Introduction

Systems of natural hazards and machinery monitoring in coal mines visualize data and information acquired from sensors which are placed in mine undergrounds. The primary objective of monitoring is continuous supervision of a production process. Two fields of monitoring can be distinguished: natural hazards monitoring and machinery operation monitoring.

Natural hazards are one of the most frequent reasons of accidents and disasters in the mining industry. This concerns in particular underground mining, in which upsetting the stability of rock mass (the so-called microseismic hazards) and risks connected with concentration of dangerous gases in mine undergrounds (Grychowski, 2008; Kabiesz, 2005; Sikora and Wróbel, 2010; Sikora and Sikora, 2006) are the most serious and frequent hazards. Based on information delivered by the system, a dispatcher, if necessary, makes a decision concerning switching off the power in a given area of the mine, evacuation of the crew from endangered zones, temporary stoppage of mining and taking preventives that are meant to lower the degree of hazard (for example, executing relieving shooting or slowing down the mining process in order to decrease the concentration of dangerous gases). The dispatchers decisions are meant to minimize the risk of disaster dangerous for crew and mining machinery as well as to sustain the production process.

To date, the main objective of machinery operation monitoring has been supervision of its exploitation con-

ditions. Recently, information gathered from monitoring systems has been more and more often considered to be diagnostic information about the actual condition of the equipment (Jonak, 2002).

For a majority of natural hazards occurring in coalmines, no sufficiently accurate mathematical models for hazard forecasting have been developed so far. Therefore, new forecasting methods based on historical data collected in databases of monitoring systems are still being worked out. In the papers by Dixon (1992), Gale *et al.* (2001), Kabiesz (2005), Sikora and Wróbel (2010), Sikora and Sikora (2006), or Sikora *et al.* (2011), propositions of application of machine learning methods to improve the forecast of seismic and methane hazards are presented.

The objective of the present paper is to propose a combination of three techniques of data analysis and their application to gaseous hazard forecasting and analysis of a coal-cutting machine cutter operation. The basic analytic technique applied is the M5 algorithm enabling induction of rules with linear conclusions. To improve the accuracy of generated rules, two complementary analytic techniques are used. Firstly, during the time series analysis, the M5 algorithm was combined with a popular method of time series forecasting (ARIMA). Values of forecasts generated through the method define a new independent variable then used by M5. Secondly, regardless of the data type, the M5 algorithm was combined with the k-nearest neighbor method inducing rules solely in some neighborhood of a currently analyzed example.

The choice of data analysis methods was motivated by their simplicity, a small number of parameters and the possibility of full automation of the analysis process without user intervention. These properties will have great meaning for practical implementation of forecasting modules in monitoring systems.

The paper is organized as follows. In the next section, a concise overview of regression and forecasting methods is presented. All techniques and algorithms applied are presented in Section 3. A proposition of technique fusion into one stream of data processing is described in Section 4. Results of practical applications of the proposed methodology to tasks pertaining to hazard monitoring in coal mines (prediction of methane concentration, prediction of carbon dioxide concentration) and the efficiency of the production process (rock cutting energy analysis depends on the cutting blade alignment) are presented in Section 5. Section 6 includes a summary and proposition for further works. Additionally, applications of the proposed methodology on several benchmark data sets (*gas furnace, sunspot, housing, ozone, abalone, Mackey–Glass*) are presented in Appendix.

2. Methods of forecasting the values of a numerical variable

Among various methods applied to forecasting the values of a numerical variable, the following ones can be listed: soft computing methods (fuzzy logic, neural networks, fuzzy-neural networks (Czogała and Łęski, 2000; Yager and Filev, 1994)), kernel regression methods (Taylor and Cristianini, 2004; Vapnik, 1995), regression trees (Breiman *et al.*, 1994) or model trees (Friedman *et al.*, 1996; Quinlan, 1993; 1992a; Torgo, 1997, Wang, 1997), ensembles of rules (Dembczyński *et al.*, 2010) or ensembles of neural networks (Siwek *et al.*, 2009), and finally the classical approach using statistical methods (Box and Jenkins, 1994; Brockwell and Davis, 2002; Tong, 1990).

Methods of soft computing are characterized by very good generalization abilities. However, the methods have disadvantages. First, they usually apply all independent variables during forecasting. Secondly, they use optimization strategies which need repeated input data set processing (gradient methods, least squares methods, genetic algorithms (Czogała and Łęski, 2000; Goldberg, 1989; Yager and Filev, 1994)). In the case of soft computing, it is necessary to set appropriate values of parameters which can have great influence on the quality of these methods (the number of groups, the number of fuzzy sets into which the domain of an independent variable is divided, the defuzzification method, etc. (Czogała and Łęski, 2000; Duch *et al.*, 2000; Oh and Pedrycz, 2000; Yager and Filev, 1994)).

Kernel methods are a group of pattern analysis algorithms that are based on the assumption that finding patterns is performed in a modified feature space. The modification is described with the special mapping function called the kernel function (Taylor and Cristianini, 2004). The usage of the kernel function substitutes the process of increasing the number of feature space dimensions in such a way that the value of the kernel function for two objects is equal to their dot product in a higher dimensional feature space. One of the most popular kernel method is support vector machines, dedicated to classification tasks (Boser *et al.*, 1992). In this approach the separating margin width is maximized with regard to a specified loss function. If the solution is assumed to be nonlinear, an optimal separating hyperplane is found in the kernel space with the usage of the kernel function. It occurs that not all training points are required to describe the hyperplane—the required ones are called support vectors. This approach was also applied to regression problems (Vapnik, 1995). The modification is based on using different forms of the loss function, and the regression tube takes the separating hyperplane place.

Since the 1990s a lot of modifications of this algorithm have been proposed. In the work of Scholkopf *et al.*

(2000), a model called v-SVM is presented, where v means the fraction of total data points that become the support vectors. Increasing v gives a more complicated model but of better quality. As both models (standard and v-SVM) are based on the assumption that the level of noise is uniform in the whole data domain, the model called par-v-SVM (Hao, 2010) removes this limitation. The regression tube is defined by two functions: a regression function f and some boundary function g. The regression tube is defined as the space between $f - g$ and $f + g$. The symmetry of this solution is generalized with flexible SVR (Chen *et al*., 2011). In this case, the regression tube is defined with three functions: regression function f and two boundary functions h and l. The regression tube is the space between $f - l$ and $f + h$. Through all the years, support vector machines have been successfully applied for time series prediction (Cao and Tay, 2003; Michalak, 2011; Tay and Cao, 2002).

Methods of regression tree or model tree induction are characterized by a considerably smaller computational complexity; all these systems perform a top-down induction by recursively partitioning the training set. Model trees generalize the concept of regression trees in the sense that they approximate $g(x) = y$ by a piecewise linear function, that is, they associate leaves with multiple linear models (Quinlan, 1993; 1992a; Torgo, 1997; Wang and Witten, 1997). A further generalization is obtained in the SMOTI (Stepwise MOdel Trees Induction) algorithm (Malerba *et al*., 2005), which constructs model trees stepwise by adding, at each step, either a regression node or a splitting node. Regression nodes perform straight-line regression, while splitting nodes partition the feature space. Recently, attempts at adapting sequential covering rule induction algorithms to regression rule induction have been undertaken (Janssen and Fürnkranz, 2010b). Regression rules induction is carried out very similarly to the case of classification rules, but the usage of different measures evaluating the quality of the generated rule is the main difference. For regression rules, measures that evaluate both the rule generality and the accuracy of a regression model occurring in the conclusion of a rule are used. In the paper by Janssen and Fürnkranz (2010b), this is achieved by means of a properly adapted relative cost measure (Janssen and Fürnkranz, 2010a).

For solving regression problems, a lazy learning approach can be also applied. In particular, the lazy decision tree induction algorithm (Friedman *et al*., 1996) can be used there. In lazy decision tree induction, a tree is defined for each example which is to be classified. The process of building the tree (in principle, its one branch) is controlled so that a node covering a classified example and training examples from one decision class is obtained. The example put to classification is added to this class. This approach can also be applied for solving regression problems. In the case of regression trees, the criterion deciding about the node quality should be changed so that it minimizes the dependent variable variance (like in the case of the M5 algorithm) or maximizes the value of the quality measure used by separate-and-conquer regression. To recapitulate, as the M5 algorithm is a regressive version of the C4.5 algorithm, the lazy decision trees induction algorithm with the criteria of node quality evaluation changed is a regressive version of the lazy classification tree induction algorithm.

Due to unusual efficiency of regression trees and model trees (both computational and in the prediction error aspect), attempts to combine the methods with soft computing were made. Jang (1994) fuzzifies a regression tree obtained by the CART algorithm (Breiman *et al*., 1994); sharp division limits are replaced with fuzzy ones (sigmoidal or logistic membership functions). Another approach can be observed in the work of Nelles *et al*. (2000), where a feature space is divided into two parts iteratively (two Gaussian membership functions are used to divide the currently considered subset of the domain of each feature). Multidimensional rule premises, in conclusions of which multidimensional linear models are determined by the least squares method, are obtained in this way.

In machine learning, very popular are multistrategy methods joining two or more methodologies in order to improve the quality of the obtained classifiers or regression systems (Duch *et al*., 2000; Oh and Pedrycz, 2000). An additional improvement of classification and prediction abilities can be obtained by the so-called constructive induction (Bloedorn and Michalski, 2002; Wnek and Michalski, 1994). The method consists in introducing to the vector of independent variables a new variable whose values depend functionally (data driven constructive induction) or logically (hypothesis driven constructive induction) on values of the existing variables (Wnek and Michalski, 1994). In hypothesis driven constructive induction, the new variable introduced can be treated as a meta-variable whose values depend on the decision made by a simpler model (model which takes no feedback into consideration). The feedback frequently allows an improvement in the prediction accuracy in neuro-fuzzy networks used for time series forecasting (Chunshien and Kuo-Hsiang, 2007).

Statistical analysis of time series provides also good methods for developing forecasting models. Autoregressive and ARIMA models are designed for time series analysis. The Box and Jenkins guidelines (Box and Jenkins, 1994) pertaining to the possibility of model application, determination of their structure and a procedure of estimating values of their parameters turn out to be effective in many applications. The Box and Jenkins paper is so far the basic source of information about one- and two-dimensional time series forecasting methods. In newer papers (Brockwell and Davis, 2002), generalizations of the methods presented by Box and Jenkins that consider mul-

tidimensional time series analysis are also discussed. Moreover, new propositions concerning, among others, automation of the selection of the number of model parameters or application of nonlinear forecasting models are presented (Tong, 1990).

3. Basic notions and definitions

In the paper, the terminology and notations applied in the machine learning community are used. A derogation consists in naming conditional attributes independent variables, and a decision attribute—a dependent variable.

Let us assume that a finite set Tr of training examples is given. Each example is described by means of independent variables belonging to a set A. Each example is also characterized by a value of the dependent variable y. Independent features can be of symbolic (discrete-valued) or of numeric (real-valued) type. The dependent variable is of numeric type. In other words, each example $x \in Tr$ is characterized by a vector of values of independent variables $(x_1, x_2, \ldots, x_m)$, where $x_i = a_i(x)$, and by the dependent variable value $y(x)$.

3.1. Induction of regression rules. The idea of the M5 algorithm was taken from the so-called regression and classification trees (CART) (Breiman *et al*., 1994) and from the C4.5, algorithm (Quinlan, 1992b) that enables decision tree induction. M5 analyzes the training set Tr and makes it possible to generate rules of the form

$$\textbf{IF } w_1 \wedge w_2 \ \wedge \ldots \wedge w_k \textbf{ THEN } y = f(x), \quad (1)$$

where w_i is the so-called elementary condition which for discrete-valued variables has the form $a_i \in R_{a_i}$ for $R_{a_i} \subset V_{a_i}$ (e.g., *pressure* $\in$ *{small, average}*), and for real-valued attributes it takes the form $a_i \in \langle v_1, v_2 \rangle$ (e.g., *gas_concentration* $\in \langle 0.4, 1.3 \rangle$ or *gas_concentration* ≥ 2). The function f is a linear function of the form $s + s_{i1}a_{i1} + s_{i2}a_{i2} + \cdots + s_{it}a_{it}$, where $s, s_{i1}, s_{i2}, \ldots, s_{it}$ are real numbers (coefficients) and $\{a_{i1}, a_{i2}, \ldots, a_{it}\} \subset A$. Independent variables belonging to a rule conclusion should be real-valued variables.

The M5 algorithm builds a tree which is then transformed into a rule set (nodes that are not leaves create rule premises, and the function f which is the rule conclusion is found in a leaf). The tree is built based on the *divide-and-conquer principle*. At each stage of tree creation (in each node that is not a leaf), a procedure of checking which attribute $a \in A$ and cut-off point $q \in \mathbb{R}$ will divide an example set P connected with the given node into two subsets $P_{<q}$ and $P_{>q}$ in order to minimize the expected variance of dependent variable is invoked. Thus the objective is to maximize the value of

$$\triangle V = V(P) - \Big(\frac{|P_{<q}|}{|P|} V(P_{<q}) + \frac{|P_{>q}|}{|P|} V(P_{>q}) \Big), \quad (2)$$

where $V(P)$ is the variance of the dependent variable in the example set P. In the case of discrete attributes, an exhaustive procedure that consists in searching a power set of given attribute values is used. If the next partition no longer decreases the expected variance, the procedure of extending the tree stops (a node becomes a leaf).

In similar works focused on model trees or fuzzy tree building, a criterion minimizing the mean square error calculated on sets $P_{<q}$ and $P_{>q}$ (Chunshien and Kuo-Hsiang, 2007; Dembczyński *et al*., 2010; Nelles *et al*., 2000) is frequently used as the optimality criterion.

To limit the number of parameters in rule conclusions, M5 applies the exhaustive approach that consists in finding a linear model for all possible subsets of conditional attributes which are real attributes. An average absolute error calculated for a set of examples assigned to a given leaf is the optimality criterion. The average absolute error is exploited during the tree pruning procedure, too. The error is multiplied by $(n + v)/(n - v)$, where $n = |Tr|$, and v is the number of variables appearing in the linear model whose error we evaluate.

To improve prediction abilities of the obtained set of rules, M5 applies also the smoothing procedure. During the tree building, the order of creating successive nodes is remembered, and hence conditions appearing in rule premise generation. Before adding a next condition, the function f_i enabling us to calculate the value of the dependent variable is defined. Thus we have the sequence of rules $< r, r_{-1}, r_{-2}, \ldots, r_{root} >$, in which r is the output rule, r_{-1} is the rule r without one premise added as the last one, etc. The rule r_{root} includes no premise but the linear model determined for the whole training set. For rules r_{-i} and r_{-i-1}, the dependent variable value is transmitted from the rule r_{-i} to the rule r_{-i-1} and determined by the expression

$$PV(r_{-i-1}) = \frac{n_{-i} PV(r_{-i}) + SM(r_{-i-1})}{n_{-i} + s}, \quad (3)$$

where n_{-i} is the number of objects from Tr that satisfy the conditional part of the rule r_{-i}, s being a fixed constant (usually $s \cong 10$), $M(r_{-i-1})$ is the value of the dependent variable expected by the partial rule $r_{-i-1}, PV(r_{-i})$, and $PV(r_{-i-1})$ are the values of the dependent variable transferred to partial rules r_{-i}, r_{-i-1}. Finally, the value of the dependent variable predicted by the rule r is the value taken back by the partial rule r_{root}.

A more detailed description of the M5 algorithm can be found in the works of Quinlan (1993; 1992a) or Wang and Witten (1997). A commercial implementation of M5 is included in the Cubist program. A noncommercial one with certain modifications in relation to the original version can be found in the Weka environment (Witten and Frank, 2005). In experiments described in the farther part of the paper, the Cubist program and the C language libra-

ry enabling us to invoke the program from other applications are applied.

3.2. Univariate time series forecasting. During time series analysis we frequently encounter a situation in which the structure of the series built is unclear, and the variance of the random component is considerable. To facilitate generation of forecasts for such series, the ARIMA methodology has been developed (Box and Jenkins, 1994). Many time series consist of mutually dependent observations. In this case, consecutive elements of the series can be determined based on previous elements delayed in time

$$y_t = \xi + \phi_1 \cdot y_{(t-1)} + \phi_2 \cdot y_{(t-2)} + \phi_3 \cdot y_{(t-3)} + \cdots + \varepsilon, \quad (4)$$

where ξ is the free term, and ϕ_1, ϕ_2, ϕ_3 are parameters of the so-called autoregressive model.

Therefore the value of the time series is the sum of the random component and a linear combination of previous observations. Regardless of the autoregressive process, each element of the series may stay under the influence of past random component realizations. This impact cannot be explained by the autoregressive component, so we have

$$y_t = \mu + \varepsilon_t - \theta_1 \cdot \varepsilon_{(t-1)} - \theta_2 \cdot \varepsilon_{(t-2)} - \theta_3 \cdot \varepsilon_{(t-3)} - \ldots, \quad (5)$$

where μ is a constant, and $\theta_1, \theta_2, \theta_3$ are parameters of the so-called moving average model. In this case, each value of the time series consists of the random component (ε) and a linear combination of the random components from the past.

The ARIMA model introduced by Box and Jenkins contains both autoregressive and moving average parameters. Moreover, the model introduces a differentiation operator that is used in order to make the time series stable (the series should have the mean, variance and autocorrelation constant in time). Detailed information about determination of the number of autoregressive parameters (p) and moving average (q) based on autocorrelations and partial autocorrelations can be found in the work of Box and Jenkins (1994). In practical applications the number of parameters is usually limited to at most two. Estimation of coefficient values is made by mean square minimization algorithms (most frequently by the quasi-Newton method (Broyden, 1969)). Evaluation of the obtained model quality is based on residues (specifically, the residue correlogram should show no statistically relevant dependencies, and the residue distribution should be normal). The software package Statistica 8.0 by Statsoft© was used in conducted experiments.

3.3. Instance-based prediction. Instance-based learning algorithms apply a training set and a similarity concept for specific local data model generation. The value of the dependent variable in a test example is established based on the values of the dependent variable in training examples which is the most similar to the test one. In the simplest case, the decision is made based on the nearest example (metric distance minimization). The generalization of that approach is the method of k-nearest-neighbors (k-nn), in which k-nearest neighbors to the test example training examples are found (Wilson and Martinez, 2000). In the case of prediction tasks, the dependent variable is established as an average value of the value of the dependent variable in examples selected from the training set. Generalization of the k-nn method are distance-weighted (Macleod *et al.*, 1987) and feature-weighted (Wettschereck *et al.*, 1997) nearest neighbor methods. In a distance-weighted method the distance between already selected training examples and the test example is calculated. In the feature-weighted method, additional weights reflecting the significance of independent variables for classification or the regression process are assigned to the variables.

In the paper, to specify the similarity of examples x_i and x_j with respect to the independent variable a, the normalized Manhattan distance measure

$$\delta_a(x_i, x_j) = \frac{|a(x_i) - a(x_j)|}{\max^a - \min^a} \quad (6)$$

was used in the case of real-valued variables, and the Hamming measure

$$\delta_a(x_i, x_j) = \begin{cases} 0, & a(x_i) = a(x_j), \\ 1, & a(x_i) \neq a(x_j) \end{cases} \quad (7)$$

was applied for discrete-valued variables.

In the formula (6) $\max^a, \min^a$ denote maximal and minimal values of the variable a recorded in the training set, respectively. Finally, the similarity of vectors x_i and x_j is measured as $\rho(x_i, x_j) = \sum_{a \in A} \delta_a(x_i, x_j)$.

4. Combination of time series prediction techniques and the *k*-nearest neighbors method with the M5 algorithm

The idea of improving the quality of regression rules generated by the M5 algorithm, by using two additional analytic techniques, is presented in this section. The first consists in introducing into a set of variables based on which M5 makes rule induction a new meta-variable. The values of the meta-variable are established by the autoregressive model (in the case of data in the form of a time series) or the ARIMA model. Incentives of such procedures are twofold. One, from conducted research (Sikora and Krzykawski, 2005; Sikora *et al.*, 2011) it follows that for gaseous hazards the greater influence on future values of a dependent variable have their past values. Hence, it is reasonable to introduce the earlier values (so-called delayed

values) of the dependent variable into the vector of independent variables used by M5. On the other hand, research carried out by the authors (Sikora and Wróbel, 2010; Sikora and Krzykawski, 2005; Sikora *et al*., 2011) shows that using too many delays leads to obtaining models unduly matched to training data, which are burdened with a big error on new unknown data. This observation is the second reason for introducing the meta-variable represented by values returned by the autoregressive or ARIMA models. In practice the models use two parameters for both autoregression and the moving average, which enable us to get a simple and intelligible model of time series. Therefore, the model task is to pre-forecast the of values the dependent variable. This preliminary forecast can then be used by the M5 algorithm in order to improve it.

The second idea is a combination of the k-nearest neighbor method with the M5 algorithm. It assumes that during establishing the value of the dependent variable of a test example x, k-nearest neighbors of the example are selected from the training set. On the example set limited in such a manner, the M5 algorithm is initialized, and the obtained model is used for determining the value of the dependent variable of the example x. It is necessary to determine the most suitable value of k in order to use the method. In the present paper, the training set and leave-one-out testing are applied for establishing the optimal value of k. The presented proposition exploits experience with RISE and RIONA classification systems (Góra and Wojna, 2002), which join the idea of instance based learning with that of rule induction. The proposition presented in this paper is some kind of lazy learning approach, because it limits the space of examples on which rule induction is made by M5. In contrast to lazy regression trees, induction is made always on the same specific number of training examples being the nearest neighborhood of the analyzed test example. An optimal number of examples is denoted as k-opty.

Contrary to lazy regression trees, during rule induction information about the values of independent variables of the test example is not considered. The information is used solely for defining the dependent variable value after determining a tree.

It is obvious that the proposed combination of the above-mentioned methods will not always lead to an improvement in the forecast results. Therefore, the proposition for combining time series prediction techniques, the k-nn method and the M5 algorithm consists in sequential invoking and tuning of each of the methods. Obviously, time series prediction techniques can be used for data in the form of a time series only. A scheme of the analysis is presented in Fig. 1.

If data have the form of a time series, the ARIMA methodology is used. If the time series can be led to stationarity (by differentiation), parameters of the estimated model are statistically significant ($p_{val} < 0.05$), the residue distribution is normal and the residues are not correlated, then the forecasting model is recognized as satisfactory. In such a case a new independent variable (meta-variable) that represents the forecasted values is added to the training data set. This means that in each row of the time series which describes the time moment t a new independent variable y_{ARIMA} is added. Its value means a forecast of the ARIMA model calculated based on earlier values of the dependent variable y (i.e., $y_{t-k}, y_{t-(l-1)}, \ldots, y_{t-1}, y_t$, where l is implied from the form of the determined statistical model).

The next stage of the analysis is establishing the value of k-opty for the method combining the k-nn method with the M5 algorithm. Determining k-opty runs based on the training data set according to the algorithm presented below. In the algorithm description, nn$(e, Tr-\{e\}, k)$ denotes the set of examples from the set $Tr-\{e\}$, which are k-nearest to the example e, RRM5(S) stands for a set of regression rules determined by the M5 algorithm based on the set of examples S, e_y denotes the value of the dependent variable in the example e; e_{yM5} stands for the value of the dependent variable in the example e which is predicted by the model get by M5.

Algorithm *Find k-opty*

input: $Tr, k_{\max}$
output: k-opty
begin
k-opty=-1; RMS=$+\infty$;
  For $k = 1$ **to** $k_{\max}$
    error=0;
    For each $e \in Tr$
      Find nn$(e, Tr-\{e\}, k)$;
      Determine RR$_{M5}$(NN$(e, Tr-\{e\}, k)$);
      error=error+$(e_y - e_{yM5})^2$;
    RMS(k):=sqrt(error/$|Tr|$);
    If RMS(k)<RMS **then** k-opty:=k;
end.

As can be seen, for each training example e and each value $1 \leq k \leq k_{\max}$, k-nearest neighbors of the example are found in the training set (from which the currently considered example has been removed), and the set of examples obtained in such a manner is transferred to the M5 algorithm. Based on the set of examples, M5 generates a rule set which is then applied for determining the value of the dependent variable of e. In this way the whole set of examples is analyzed for each k. After the analysis, the RMS error is calculated. The value of k that led to the smallest error is recognized as k-opty.

Figure 1 shows that three analysis paths are realized simultaneously: ARIMA+k-nn+M5, k-nn+M5 and M5 only. Therefore we obtain three (if the analyzed data set has the form of a time series) or two (if the statistical model is wrong or data do not have the form of a time

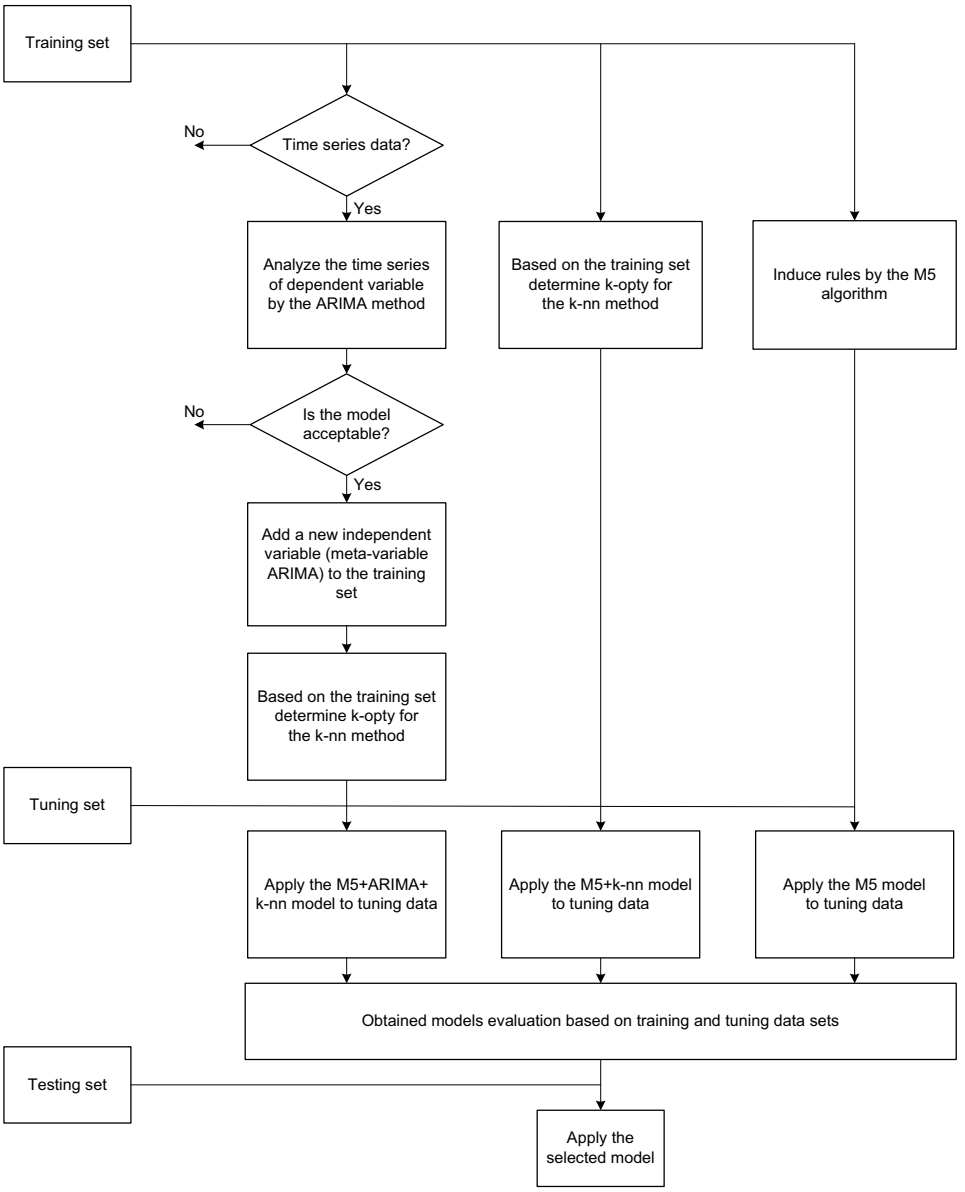

Fig. 1. Combination of k-nn and time series prediction with M5—data flow and analysis scheme.

series) forecasting models. A suitable model can be verified and selected on one of two data sets: the tuning one (which can be a training set in particular) and testing one. Obviously, to define a fully automatic method of model selection, verification cannot be the on the testing set. However, in the domain literature authors often present results of the same algorithm in various parameter configurations obtained on a training and a testing data set, while no unambiguous methodology exists for optimal values of parameters. Especially in the literature concerning neural-fuzzy systems such a situation is frequently met (due to a great number of fuzzy implications, values of learning parameters, fuzzification, defuzzification methods, etc.) (Czogała and Łęski, 2000; Oh and Pedrycz, 2000; Rutkowski, 2004).

In the present paper the model is selected automatically. In the case of data in the form of a time series a model which minimized the error obtained on the training set was selected as the best one. In the case of other data an independent tuning set was excluded from the training set and the quality of k-nn+M5 and M5 models was compared on this set.

5. Examples of practical applications of the methodology

5.1. Data analysis. The presented methodology was applied in three implementations of the M5 algorithm for

analysis of data coming from safety monitoring systems and technological processes in coal mines. Now we briefly present prediction problems and data sets pertaining to them.

The first problem concerns intermediate prediction (forecast horizon equal to ten minutes) of methane concentration in a mine excavation. The task is important from the perspective of foiling automatic, preventive current cut-offs which cause breaks in the mining process. A safety system turns off the current in mine tunnels if methane concentration exceeds a certain, fixed threshold value. The function of the forecasting system is to predict future methane concentration, and, if the forecast values approach threshold values, to inform a dispatcher about necessity of taking actions aimed at changing the manner of excavation ventilation or mining process. Both functions usually lead to reduction of methane concentration in the excavation.

The analyzed data set has the form of a time series. In the case considered here concentrations registered by the methanemeter $M32$ placed in the most troublesome area of the excavation (at the longwall face end) were the prediction subject. Aggregated data from ten-minute time periods were put to analysis. The forecast horizon equal to ten minutes is the next value of the dependent variable in a time series. Data from two methanemeters $M32, M31$ (the methanemeter at the longwall face end) and anemometer $AN31$ (the sensor of air flow speed) were used for the prediction. Information about output intensity on the wall (the Output variable) was also applied for the forecasting. Maximal values of the variables $M32, M31, AN21$ and Output registered at the actual and previous aggregation time t and $t-10, t-9, \ldots, t-1$ were used as a features vector. Moreover, the difference between the actual and previous aggregated values (e.g., $M32_t - M32_{t-1}$) was also calculated for each independent variable in order to convey the dynamism of changes of the measured quantities.

The dependent variable $M32Pred$ contained the value of methane concentration registered by the sensor $M32$ at the time $t+1$. By "the time t" we mean the ten-minute period. Training and testing data sets contained 679 and 286 examples, respectively. A detailed description of that application and the whole infrastructure of prediction system are presented by Sikora and Sikora (2006) as well as Sikora *et al*. (2011). However, in the papers no approach exploiting the k-nn algorithm is applied.

The second application concerns prediction of carbon dioxide concentration on the operating platform in a mine dewater station. Carbon dioxide is drawn out from the mine tunnels by the water column, in which dewater pumps are immersed, and emits into the atmosphere. Measurement of carbon dioxide concentration within the operating platform is notably significant, especially during maintaining or repairing works. The measurement system in one-minute gaps measures the following quantities: atmospheric pressure Ps, environmental humidity $RHOs$, humidity on the platform $RHPs$, environmental temperature TOs, temperature on the platform TPs. During the forecasting, $\Sigma CO2$, ΣPs, $\Sigma RHOs$, $\Sigma RHPs$, ΣTOs, ΣTPs were also applied as independent variables. The notation ΣV denoted the sum of the recent ten values of V (i.e., $\Sigma V = V_{t-9} + V_{t-8} + \ldots + V_t$). The dependent variable $CO2Pred$ included the value of carbon dioxide concentration at the time $t+6$. Training and testing example sets contained 1828 and 914 examples, respectively. A system of data acquisition and results of statistical analysis (manifold regression) are described in detail by Sikora and Krzykawski (2005). The analyzed data set had the form of a time series.

The third application concerns the process of rock cutting by conical rotary blades. The aim of the research was to determine such technological and geometrical parameters (settings) of blade that a unite cutting energy is minimal. The set of independent variables consisted of variables describing technological parameters of the blade work (t: cutting scale [mm], g: cutting depth [mm], m: mass of the cut material [g]) and geometrical parameters of the blade (β: blade's angle [°], δ: setting's angle [°], ρ: rotation angle [°]). A new independent variable that is the quotient of the scale (t) and the cutting depth (g) was also introduced. The dependent variable contains information about the value of the unit cutting energy Ec [MJ/m^3]. The analyzed data set does not have the form of a time series. The data set included 717 examples, and the 10-fold cross validation method was used as a testing methodology. Moreover, a tuning set which accounted for 10% of each training set was applied in the analysis, too. The set was found before the k-opty searching process.

Results of the data analysis are presented in Tables 1 and 2. The method ultimately recognized as the best one, based on which the error on a testing set was then searched, is in bold. In the case of time series it was the method minimizing the error on a training set, in the case of cross-validation—the method minimizing the error on a tuning set.

For the first data set (Methane), introducing a new variable including predicted values of methane concentration generated by the autoregressive model resulted in error decreasing and simplification of the forms of rules used for the forecasting. The statistical model of the forecasting consisted of one autoregressive component ($\xi = 0, \phi_1 = -0.2307$), and the series had to be put to single differentiation. An attempt at improving the forecast quality by adding the k-nn method to the analysis did not succeed, because an optimal value of k-opty was got during the tuning for the whole analyzed data set (k-opty$=|Tr|-1$). A difference of the error between models ARIMA+M5 and ARIMA+k-nn+M5 for k-opty$=|Tr|-1$ appeared only on the fourth decimal place. Results of sear-

Table 1. RMS error obtained on training data sets.

	ARIMA	M5	ARIMA+M5	ARIMA+k-nn+M5 ∨ k-nn+M5
Methane	0.093	0.087	**0.083**	0.083
CO_2	0.238	0.237	0.237	**0.059**
Ec	–	3.71 ± 0.26	–	**2.86**$\pm$0.18

Table 2. RMS error obtained on testing data sets.

	ARIMA	M5	ARIMA+M5	ARIMA+k-nn+M5 ∨ k-nn+M5
Methane	0.063	0.061	**0.056**	0.056
CO_2	0.368	0.220	0.220	**0.102**
Ec	–	3.84 ± 0.32	–	**3.66**$\pm$0.21 ($p = 0.049$)

Table 3. Comparison of the RMS error for constrained (k-opty $\leq$ 200) and complete (k-opty $\leq |Tr| - 1$) spaces of an optimal number of nearest neighbor searches: the training set.

	k-opty $\leq$ 200	k-opty$< |Tr|$
Methane	0.096 (200)	0.083 (677)
CO_2	0.051 (2)	0.051 (2)
Ec	2.86 (82)	2.86 (82)

Table 4. Comparison of the RMS error for constrained (k-opty $\leq$ 200) and complete (k-opty $\leq |Tr| - 1$) spaces of an optimal number of nearest neighbors searches: the testing set.

	k-opty $\leq$ 200	k-opty$< |Tr|$
Methane	0.103	0.056
CO_2	0.102	0.102
Ec	3.66	3.66

ching for an optimal value of k-opty for a limited (≤ 200) and whole ($|Tr|-1$) set of nearest neighbors are presented in Tables 3 and 4. It can be noticed that of the restriction k-opty searching space would lead to the worst results in the case of the Methane set.

The rules to determine the methane concentration forecast (without the ARIMA model usage) are as follows:

(i) **If** $M32_t \leq 0.9$, **then** $M32_{t+1} = 0.06 + 0.93M32_t$.

(ii) **If** $M32_t > 0.9$ **and** $\text{Output}_t = 0$, **then**
$$M32_{t+1} = 0.47 + 0.8M32_t + 0.05M32_{t-1} - 0.3AN31_t + 0.2AN31_{t-2} - 0.04AN32_t - 0.12AN32_{t-1} - 0.12(AN32_t - AN32_{t-1}).$$

(iii) **If** $M32_t > 0.9$ **and** $\text{Output}_t > 0$, **then**
$$M32_{t+1} = 0.51 + 0.33M32_t + 0.18M32_{t-1} + 0.21M32_{t-4} + 0.0013\text{Output}_t - 9.36AN31_{t-1} + 9.05AN31_t - 9.22(AN31_t - AN31_{t-1}) + 0.56AN32_t - 0.53(AN32_t - AN32_{t-1}) - 0.52AN32_{t-1}.$$

The rules to determine the methane concentration forecast (with the ARIMA model used as an additional independent variable) are as follows:

(iv) **If** $\text{ARIMA}_{t+1} \leq 0.9$, **then**
$$M32_{t+1} = 0.06 + 0.93M32_t.$$

(v) **If** $\text{ARIMA}_{t+1} > 0.9711$ **and** $\text{Output}_t = 0$, **then**
$$M32_{t+1} = 0.44 + 0.86M32_t - 0.27AN31_t - 0.17AN32_t + 0.2AN31_{t-2}.$$

(vi) **If** $\text{ARIMA}_{t+1} > 0.9711$ **and** $\text{Output}_t > 0$, **then**
$$M32_{t+1} = 0.74 + 0.39M32_t + 0.15M32_{t-4} + 0.12M32_{t-5} + 0.00156\text{Output}_t - 0.25AN31_{t-2} - 0.17AN31_t.$$

The usage of the values predicted by the ARIMA model (which boils down to the autoregressive model) as a new independent variable allowed us to simplify considerably input rules, and because of that the analysis of the rules (iv)–(vi) is simpler than that of (i)–(iii). Valuable for practical application of the methane forecasting system are the forecast maximal errors. In the analyzed time series the maximal rate of change of methane concentration during prediction period (for the testing data set) equaled 0.39, the maximal value of the error made by the predictor was equal to 0.22 for this set (and was registered in a different place than in the case of the maximal rate of change of CH_4 concentration). It is unusual that the RMS error on the testing set is smaller than the error on the training set. This results solely from selection of the training and testing sets in the case considered. The testing set describes the last two days of a week. In particular, the last part of the testing set describes the so-called maintenance shift when no mining works are conducted. Thereby a stabilization of methane concentration occurs, which can be seen in Fig.2. The figure also shows that the forecasting model makes utmost errors during sudden and dynamic changes of methane concentration.

The forecasting system has been implemented as an

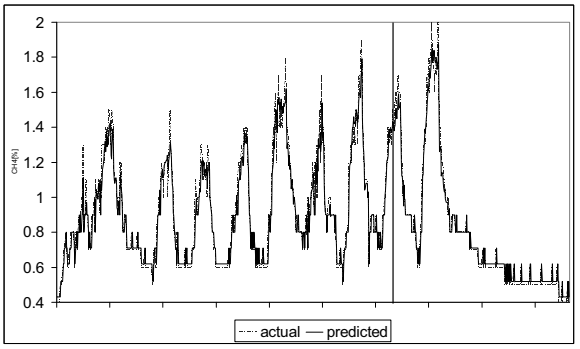

Fig. 2. Graphs of real and predicted methane concentrations. The vertical line separates the training set from the testing one.

additional module of the methane-fire disposal system SMP-NT developed at the Institute of Innovative Technologies EMAG (see Section 5.2). Detailed analysis of results for methane concentration forecasting in various mine excavations made by the M5 algorithms is presented by Sikora *et al*. (2011).

In the case of the second data set, application of the ARIMA methodology did not give better results. Though the obtained model parameters were statistically significant, the ARIMA variable occurred neither in the premise nor in the conclusion of any rule determined by M5. The noted decreasing of the error was obtained by combining k-nn with the M5 algorithm; k-opty=2 turned out to be the optimal value for the whole data set. The maximal error made during the prediction by the model applying M5 rules equaled only 2.86 for the testing set. The combination of k-nn and M5 allowed us to reduce the RMS error by half, but decreased the maximal error to 1.95 (Fig. 3) at the same time. It is worth noticing that the maximal change of CO_2 concentration in a six-minute forecast horizon was equal to 4.19. Establishing the value of k-opty as equal to 2 made M5 create one rule containing no premises with a multi-dimensional linear model in a conclusion (in this case the algorithm just realized the multiple regression algorithm). For examples describing a low concentration of carbon dioxide, in a predominant majority of examples, the regression model applied the variables CO_2, TOs (environmental temperature) and $\Sigma CO_2, \Sigma TOs$ only. For examples describing a higher concentration, the variables Ps (atmospheric pressure) and ΣPs were also applied, while the others were not used. Without the combination with k-nn, the M5 algorithm generated 21 rules which were created based on all independent variables.

The third data set does not have the form of a time series. Therefore the M5 algorithm and combined k-nn and M5 methods were possible to be applied for the analysis only. Average results with standard deviation are presented in Tables 1 and 2. The difference between M5

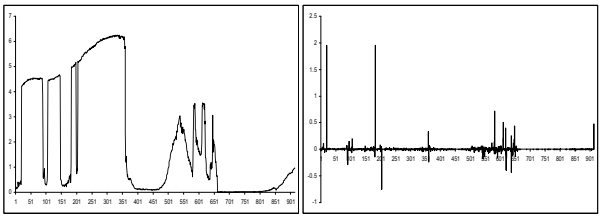

Fig. 3. Graphs of CO_2 concentration (testing set) and the error made by the model obtained by combined k-nn and M5 algorithms.

and the k-nn+M5 methods is equal to 0.18 on the average. In order to estimate the significance of differences made during each of the 10 experiments, the Wilcoxon signed-rank test was carried out. The statistically significant difference was obtained for the 95% level of significance ($p_{\text{value}} = 0.041$). The discovered rules show that low values of Ec, desired in terms of the analysis aim, were dependent on the cutting depth. If $g > 6$, then the cutting energy was low and belonged to the interval $\langle 2, 33\rangle$ MJ/m^3. The conclusion of the rule below decided about the precise value of the energy.

If $g > 6$, **then**

$$\text{EC}=-44.177 - 0.0037m - 0.64g + 0.18t - 2.1t/g - 0.23\rho + 0.68\beta + 0.4\delta.$$

It shows that the higher the values of blade parameters β, δ, the higher the cutting energy. In turn, the higher the cutting scale and depth, the lower the energy. For the blade's angle of rotation ρ, higher (but positive) angles of rotation contribute to the decreasing of the cutting energy, negative angles of rotation increase the energy. For the highest cutting energy values (rule's range: $\langle 33, 66\rangle$ MJ/m^3) the most typical was the following rule:

If $g \leq 6$ **and** $t > 10$, **and** $\rho \leq -10$, **then**

$$\text{EC}= 55.97 - 0.0155m - 0.66 - 0.23t.$$

The above rules are outcomes of the analysis of the whole available data set. During cross-validation, the M5 algorithm generated 3 to 4 rules. In the case of the combination of M5 and the k-nn method, the number of rules was equal to 1 to 4.

In order to compare obtained results, those achieved for the testing set by multiple regression, an artificial neural network, a neural-fuzzy network ANNBFIS (Czogała and Łęski, 2000) are also presented in Table 5. The values of all parameters of the above-mentioned methods were determined based on the training set. The regression and training of neural networks were carried out in the Statistica package. Especially for neural networks, those with a different architecture and various functions of neurons' activation were tested. This is enabled by the Statistica environment. The choice of the best of the tested

Table 5. Comparison of the obtained results with other forecasting methods.

	Test set: RMS error		
	Methane	CO_2	Ec
Our method	0.056	0.102	3.66
M5	0.061	0.220	3.84
Multiple regression	0.073	0.428	7.12
Neural network	0.072	0.223	3.72
ANNBFIS	0.068	0.197	3.82

networks was made in the same way as in the case of our method (see Section 4). A source code available in the paper by Czogała and Łęski (2000) was used for ANNBFIS network implementation.

For the data sets our method produced the best results each time. It is worth noticing that application of the sole M5 algorithm does not guarantee good results anymore.

The level of methane concentration predicted by the forecasting module together with information about changes in the concentration is used by a fuzzy reasoning system to determine the so-called potential methane risk.

5.2. Implementation of the proposed methodology in a methane concentration monitoring system. The proposed method was implemented in a forecasting module enabling medium-term prediction of methane concentration and methane risk estimation in hard-coal mines. The module aggregates and stores automatically data incoming from a monitoring system. These data are the basis for producing forecasting models that are then used for on-line forecasting methane hazards. During normal work of the system, its forecasting efficiency is monitored currently. If the efficiency diminishes, the repeated tuning of the system parameters takes place. The system efficiency is calculated as the RMS error. The values of absolute errors are also monitored. If the RMS error or the number of absolute errors greater than 0.09 or 0.19 or 0.29 exceeded within the last 24 hours (a moving time window) threshold values established in the system configuration, forecasting models are determined again.

The level of methane concentration predicted by the forecasting module together with information about changes in the concentration is used by a fuzzy reasoning system to determine the so-called potential methane risk.

A base of fuzzy rules has been developed by domain experts (Grychowski, 2008). Fuzzy rules consist of two premises: predicted methane concentration and the dynamics of concentration changes that follows from the forecast. Domains of both values were split into fuzzy sets according to domain knowledge. Methane concentration in atmosphere was split into four fuzzy sets (Fig. 4, middle chart). The dynamics of changes was reflected by means of three fuzzy sets (no changes, increasing, quickly increasing). The fuzzy set "no changes" takes also into account falls in the methane concentration (Fig. 4, left chart). Domain knowledge enables us to determine eight fuzzy rules that combine methane concentration and its changes dynamism with a risk degree in an excavation (Table 6).

Three risk states are distinguished (Fig. 4, right chart): normal state (point value 1), warning (point value 2), hazard (point value 3). These states were described by fuzzy sets with triangle membership functions that attain their maxima at points 1, 2, 3, respectively.

The system applies constructive inference of the Larsen type (Czogała and Łęski, 2000; Yager and Filev, 1994) in which the PROD operator (t-norm=PROD) is used for establishing the rule activation level. Rules aggregation consists in summing fuzzy sets derived by each rule (union of fuzzy sets—MAX operator). The standard center of gravity method (Yager and Filev, 1994) is applied as a defuzzification method. Input values are not put to fuzzification; they are treated as singletons.

The presented fuzzy reasoning system enables presenting to a dispatcher messages about actual (based on actual measured values) and predicted (based on predicted values) risk state understandable for him/her.

6. Conclusions

The idea of improving prediction abilities of rules generated by M5 by using the meta-variable that contains forecasts resulting from a one-dimensional statistical model and generating rules solely in a neighborhood of an analyzed testing example has been proposed.

The main motivation for our research was application of the developed method in solving tasks pertaining to the forecasting of natural hazards in coal mines and the monitoring of mine machinery. The presented method was applied for forecasting gaseous risks and analysis of a coal-cutting machine cutter operation. Results of experiments show that the presented proposition enables us to obtain the forecast quality better than in the case of each of the discussed method individually. Due to application of the M5 algorithm as the basic forecasting method, the presented technique is characterized by good generalization abilities and generates no models badly fitted to data.

It follows from experiments that the phase of partial model assessment is very important for the efficiency of the method, because the forecasting model combining all the three methods ARIMA+k-nn+M5 does lead to the best forecasts in each case. This claim is also supported by results obtained on benchmark data that are included in Appendix. In the present paper, models were selected based on the forecast error on validation and training sets.

The presented forecasting method has been applied in practice. It is used by the forecasting module that is a component of a methane risks monitoring system (Sikora *et al.*, 2011).

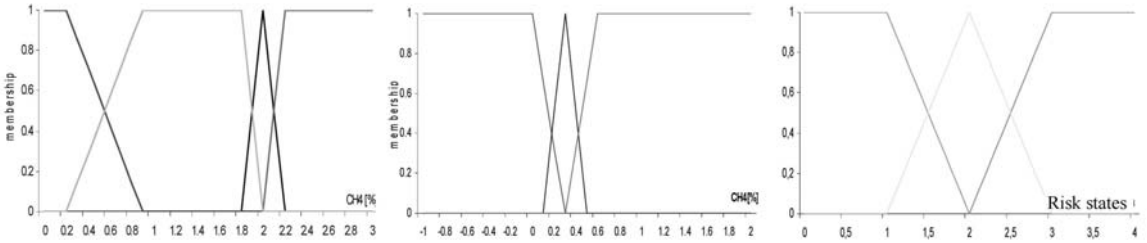

Fig. 4. Partition of CH_4 concentration, CH_4 evolution of changes and risk state domains for fuzzy sets.

Table 6. Rules connecting risk states with CH_4 concentration and evolution of changes.

Rule	CH_4 concentration	CH_4 changes dynamism	Risk state
1	normal	no changes	normal state
2	normal	increasing	normal state
3	normal	quickly increasing	warning
4	admissible	no changes	warning
5	admissible	increasing	warning
6	admissible	quickly increasing	hazard
7	boundary	–	hazard
8	exceeded	–	hazard

Our further research will focus on full automation of the process of the ARIMA model constructing and shortening the duration of searching values of the k-opty parameter.

Presently the process of tuning the parameters of the statistical model (p, q, r values) is not fully automatic but performed by an operator. However, one can attempt to define an algorithm for automatic selection of these values according to suggestions of Box and Jenkins (1994). The procedure of searching for an optimal value of the k-opty parameter is the most time-consuming operation of our methodology. Tables 9 and 10 (see Appendix) show that bounding the number of the nearest neighbors considered above does not allow us to achieve satisfactory results. Better outcomes are guaranteed for a method testing the whole possible range of the k parameters. Application of k-d trees (Wess *et al*., 1994) or SR-trees in the case of multi-dimensional data (Katayama and Satoh, 1997) may decrease the cost of determining nearest neighbors. The heuristic strategy that consists in searching for selected values of k only or the approach that constrains the training set (Wilson and Martinez, 2000) are also possible to be applied here. However, the time necessary for establishing the optimal k value is an unquestionable limitation of the presented method.

A benefit of the presented methodology is undoubtedly the relatively small number of parameters and a short time of learning for the fixed k-opty. It is also worth noticing that if the statistical model (in spite of satisfying conditions of parameters' statistical significance) does not contribute to the quality improvement of rules generated by M5, then it does not occur in these rules. This follows from the fact that the M5 algorithm performs feature selection during rule induction, which is rare in some neuro-fuzzy systems (Czogała and Łęski, 2000; Oh and Pedrycz, 2000; Rutkowski, 2004).

Acknowledgment

The authors wish to thank the anonymous reviewers for helpful feedback and comments on drafts of this paper.

References

Bloedorn, E. and Michalski, R. (2002). Data-driven constructive induction, *IEEE Intelligent Systems* **13**(2): 30–37.

Boser, B., Guyon, I. and Vapnik, V. (1992). A training algorithm for optimal margin classifiers, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, Pittsburgh, PA, USA,* pp. 144–152.

Box, G. and Jenkins, G. (1994). *Time Series Analysis: Forecasting and Control*, Prentice-Hall, Upper Saddle River, NJ.

Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1994). *Classification and Regression Trees*, Wadsworth, Belmont, CA.

Brockwell, P. and Davis, R. (2002). *Introduction to Time Series Forecasting*, Springer-Verlag, New York, NY.

Broyden, C. (1969). A new double-rank minimization algorithm, *Notices of the American Mathematical Society* **16**: 670.

Cao, L. and Tay, F. (2003). Support vector machine with adaptive parameters in financial time series forecasting, *IEEE Transactions on Neural Networks* **14**(6): 1506–1518.

Chen, X., Yang, J. and Liang, J. (2011). A flexible support vector machine for regression, *Neural Computing & Applications*, DOI 10.1007/s00521-011-0623-5.

Chunshien, L. and Kuo-Hsiang, C. (2007). Recurrent neuro-fuzzy hybrid-learning approach to accurate systems modeling, *Fuzzy Sets and Systems* **158**(2): 194–212.

Czogała, E. and Łęski, J. (2000). *Fuzzy and Neuro-Fuzzy Intelligent Systems. Studies in Fuzziness and Soft Computing*, Springer-Verlag, New York, NY.

Dembczyński, K., Kotowiski, W. and Słowiński, R. (2010). Ender: A statistical framework for boosting decision rules, *Data Mining and Knowledge Discovery* **21**(1): 52–90.

Dixon, W. (1992). *A Statistical Analysis of Monitored Data for Methane Prediction*, Ph.D. thesis, University of Nottingham, Nottingham.

Duch, W., Adamczak, R. and Grabczewski, K. (2000). A new methodology of extraction, optimization and application of crisp and fuzzy logical rules, *IEEE Transactions on Neural Networks* **11**(10): 1–31.

Friedman, J., Kohavi, R. and Yun, Y. (1996). Lazy decision trees, *Proceedings of AAAI/IAAI, Portland, OR, USA*, pp. 717–724.

Gale, W., Heasley, K., Iannacchione, A., Swanson, P., Hatherly, P. and King, A. (2001). Rock damage characterization from microseismic monitoring, *Proceedings of the 38th US Symposium of Rock Mechanics, Lisse, The Netherlands*, pp. 1313–1320.

Goldberg, D. (1989). *Genetics Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company, Boston, MA.

Góra, G. and Wojna, A. (2002). Riona: A new classification system combining rule induction and instance-based learning, *Fundamenta Informaticae* **51**(4): 369–390.

Grychowski, T. (2008). Hazard assessment based on fuzzy logic, *Archives of Mining Sciences* **53**(4): 595–602.

Hao, P. (2010). New support vector algorithms with parametric insensitive/margin model, *Neural Networks* **23**(1): 60–73.

Jang, J.-S. (1994). Structure determination in fuzzy modelling: A fuzzy cart approach, *Proceedings of the IEEE International Conference on Fuzzy Systems, Orlando, FL, USA*, pp. 480–485.

Janssen, F. and Fürnkranz, J. (2010a). On the quest for optimal rule learning heuristics, *Machine Learning* **78**(3): 343–379.

Janssen, F. and Fürnkranz, J. (2010b). Separate-and-conquer regression, *Proceedings of LWA 2010: Lernen, Wissen, Adaptivität, Kassel, Germany*, pp. 81–89.

Jonak, J. (2002). Hazard assessment based on fuzzy logic, *Journal of Mining Sciences* **38**(3): 270–277.

Kabiesz, J. (2005). Effect of the form of data on the quality of mine tremors hazard forecasting using neural networks, *Geotechnical and Geological Engineering* **24**(5): 1131–1147.

Katayama, N. and Satoh, S. (1997). The SR-tree: An index structure for high dimensional nearest neighbor queries, *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, New York, NY, USA*, pp. 369–380.

Macleod, J., Luk, A. and Titterington, D. (1987). A re-examination of the distance-weighted k-nearest-neighbor classification rule, *IEEE Transactions on Systems, Man and Cybernetics* **17**(4): 689–696.

Malerba, D., Esposito, F., Ceci, M. and Appice, A. (2005). Top-down induction of model trees with regression and splitting nodes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(5): 612–625.

Michalak, M. (2011). Adaptive kernel approach to the time series prediction, *Pattern Analysis and Applications* **14**(3): 283–293.

Nelles, O., Fink, A., Babuška, R. and Setnes, M. (2000). Comparison of two construction algorithms for Takagi–Sugeno fuzzy models, *International Journal of Applied Mathematics and Computer Science* **10**(4): 835–855.

Oh, S. and Pedrycz, W. (2000). Identification of fuzzy systems by means of an auto-tuning algorithm and its application to nonlinear systems, *Fuzzy Sets and Systems* **115**(2): 205–230.

Quinlan, J. (1992a). Learning with continuous classes, *Proceedings of the International Conference on Artificial Intelligence, Singapore*, pp. 343–348.

Quinlan, J.R. (1992b). *C4.5 Programs for Machine Learning*, Morgan Kaufman Publishers, San Mateo, CA.

Quinlan, J. (1993). Combining instance-based learning and model-based learning, *Proceedings of the 10th International Conference on Machine Learning, San Mateo, CA, USA*, pp. 236–243.

Rutkowski, L. (2004). Generalized regression neural networks in time-varying environment, *IEEE Transactions on Neural Networks* **15**(3): 576–596.

Scholkopf, B., Smola, A., Williamson, R. and Bartlett, P. (2000). New support vector algorithms, *Neural Computation* **12**(5): 1207–1245.

Schuster, H. (1998). *Deterministic Chaos*, VCH Verlagsgesellschaft, New York, NY.

Sikora, M. and Krzykawski, D. (2005). Application of data exploration methods in analysis of carbon dioxide emission in hard-coal mines dewater pump stations, *Mechanizacja i Automatyzacja Górnictwa* **413**(6): 57–67, (in Polish).

Sikora, M., Krzystanek, Z., Bojko, B. and Śpiechowicz, K. (2011). Application of a hybrid method of machine learning for description and on-line estimation of methane hazard in mine workings, *Journal of Mining Sciences* **47**(4): 493–505.

Sikora, M. and Sikora, B. (2006). Application of machine learning for prediction a methane concentration in a coal mine, *Archives of Mining Sciences* **51**(4): 475–492.

Sikora, M. and Wróbel, Ł. (2010). Application of rule induction algorithms for analysis of data collected by seismic hazard monitoring systems in coal mines, *Archives of Mining Sciences* **55**(1): 91–114.

Siwek, K., Osowski, S. and Szupiluk, R. (2009). Ensemble neural network approach for accurate load forecasting in

a power system, *International Journal of Applied Mathematics and Computer Science* **19**(2): 303–315, DOI: 10.2478/v10006-009-0026-2.

Tay, F. and Cao, L. (2002). Modified support vector machines in financial time series forecasting, *Neurocomputing* **48**(1): 847–861.

Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge.

Tong, H. (1990). *Non-linear Time Series: A Dynamical Systems Approach*, Oxford University Press, Oxford.

Torgo, L. (1997). Kernel regression trees, *Proceedings of Poster Papers, European Conference on Machine Learning, Prague, Czech Republic*, pp. 118–127.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*, Springer, New York, NY.

Wang, Y. and Witten, I. (1997). Inducing model trees for continuous classes, *Proceedings of Poster Papers, European Conference on Machine Learning, Prague, Czech Republic*, pp. 128–137.

Weigend, A., Huberman, B. and Rumelhart, D. (1990). Predicting the future: A connectionist approach, *International Journal of Neural Systems* **1**(3): 193–209.

Wess, S., Althoff, K. and Derwand, G. (1994). Using k-d trees to improve the retrieval step in case-based reasoning, *in* S. Wess, K.-D. Althoff and M. Richter (Eds.), *Topics in Case-Based Reasoning*, Springer-Verlag, Berlin, pp. 167–181.

Wettschereck, D., Aha, D. and Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms, *Artificial Intelligence Review* **11**(1–5): 273–314.

Wilson, D. and Martinez, T.R. (2000). An integrated instance-based learning algorithm, *Computational Intelligence* **16**(1): 1–28.

Witten, I. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, San Francisco, CA.

Wnek, J. and Michalski, R.S. (1994). Hypothesis-driven constructive induction in AQ17-HCI: A method and experiments, *Machine Learning* **14**(2): 139–168.

Yager, R. and Filev, D. (1994). *Essentials of Fuzzy Modeling and Control*, John Wiley and Sons, New York, NY.

Marek Sikora was born in Poland in 1969. He received the M.Sc. degree in applied mathematics from the University of Silesia in 1993 and the Ph.D. degree in informatics from the Silesian University of Technology in 2002. He is a member of the Scientific Council of the Institute of Innovative Technologies EMAG in Katowice and of the Polish Computer Society. His scientific interest is in rule induction and evaluation, machine learning, and application of intelligent systems in industry, biology and medicine. He is an author or coauthor of more than 60 scientific papers.

Beata Sikora was born in Poland in 1969. She received the M.Sc. degree in applied mathematics from the University of Silesia in 1995 and the Ph.D. degree in control engineering from the Silesian University of Technology in 2002. She is a member of the Polish Mathematical Society. Her scientific interest is controllability theory for linear dynamical systems with delays. Moreover, her current scientific interests are data analysis, especially the analysis of data coming from monitoring systems, and the application of machine learning methods for natural hazards assessment. She is an author or coauthor of about 20 scientific papers and 3 university textbooks.

Appendix

The presented method enabled us to achieve good results in the application domain we are interested in. In this appendix, several commonly known benchmark data sets are presented as analysis supplement.

The methodology presented in Section 4 was also applied to the analysis of commonly known benchmark data. As data in the form of a time series, the following data sets were selected: *gas furnace* (Box and Jenkins, 1994) (independent variables $u_{t-6}, \ldots, u_{t-1}, y_{t-4}, \ldots, y_{t-1}$, the dependent variable y), *sunspots* (Weigend *et al*., 1990) (independent variables $x_{t-12}, \ldots, x_{t-1}$, the dependent variable x_t) and a chaotic time series obtained on the basis of the solution to the Mackey–Glass differential delay equation (Schuster, 1998) (independent variables $x_{t-18}, x_{t-12}, x_{t-6}, x$, the dependent variable x_{t+6}). The sizes of training and testing data sets equal $|Tr| = 100, |Ts| = 189$ for *gas furnace*; $|Tr| = 100, |Ts| = 180$ for *sunspots*; $|Tr| = 500, |Ts| = 500$ for Mackey–Glass.

As data which do not have the form of a time series, the *Boston housing, ozone* and *abalone* sets from UCI Repository were selected. For the *Boston housing* and *ozone* sets the *10 fold cross validation* methodology was applied as the testing method. For the *abalone* set, which contains more than 1000 examples, *train and test* was employed.

The error values for the ANNBIFS fuzzy-neural network (Czogała and Łęski, 2000) were also given for comparison. The results of ANNBIFS presented in Tables 7 and 8 are the best ones obtained after the testing of several networks composed of two to ten fuzzy rules. RMS errors for the data sets obtained by other forecasting methods can be found, among others, in the papers by Czogała and Łęski (2000) as well as Rutkowski (2004).

In the case of time series, application of the ARIMA methodology combined next with the k-nn method allowed us to decrease the forecast error for *gas furnace* and *sunspots* data. For *Mackey–Glass*, the set on which rule induction is conducted did not succeed insomuch that the best results were obtained for the whole training set. This result is not surprising since the *Mackey–Glass* data set is generated in accordance with a mathematical equation. Therefore, the bigger the number of examples, the smaller

Table 7. RMS error obtained on training data sets.

	ARIMA	M5	ARIMA+M5	ARIMA+k-nn+M5 $\vee$ k-nn+M5	ANNBFIS
Gas furnace	0.376	0.134	0.134	0.121	**0.087**
Sunspots	0.093	0.075	0.063	0.060	**0.050**
Mackey–Glass	0.007	0.008	0.003	0.003	**0.002**
Boston	–	2.47 ± 0.14	–	2.10 ± 0.08	**1.96**± 0.16
Ozone	–	3.69 ± 0.27	–	3.14 ± 0.23	**2.80**± 0.27
Abalone	–	2.17	–	**2.17**	2.32

Table 8. RMS error obtained on testing data sets. The symbol '$+$' means that the result is statistically better on the level $p = 0.05$, '$-$' means that the result is statistically worse on the level $p = 0.05$. The Wilcoxon test was used for testing.

	ARIMA	M5	ARIMA+M5	ARIMA+k-nn+M5 $\vee$ k-nn+M5	ANNBFIS
Gas furnace	0.446	0.413	0.413	0.375	**0.366**
Sunspots	0.110	0.088	0.079	**0.074**	0.093
Mackey–Glass	0.008	0.012	0.004	0.004	**0.002**
Boston	–	3.19 ± 0.25	–	**3.01**$\pm 0.32^{+}$	3.35 ± 0.47
Ozone	–	**4.01**± 0.74	–	$4.43 \pm 1.22^{-}$	4.45 ± 0.85
Abalone	–	1.95	–	1.95	2.00

Table 9. Comparison of the RMS error for constrained k-opty$\leq$ 200 and complete k-opty$\leq |Tr| - 1$ space of an optimal number of nearest neighbors searching: the training set.

	k-opty$\leq$ 200	k-opty$\leq |Tr| - 1$
Gas furnace	0.121 (92)	0.121 (92)
Sunspots	0.060 (91)	0.060 (91)
Mackey–Glass	0.006 (200)	0.003 (499)
Boston	2.10 (104)	2.10 (104)
Ozone	3.14 (102)	3.14 (102)
Abalone	2.49 (200)	2.17 (2799)

Table 10. Comparison of the RMS error for constrained (k-opty$\leq$ 200) and complete (k-opty$\leq |Tr| - 1$) space of an optimal number of nearest neighbors searching: the testing set.

	k-opty$\leq$ 200	k-opty$\leq |Tr| - 1$
Gas furnace	0.375	0.375
Sunspots	0.074	0.074
Mackey–Glass	0.008	0.004
Boston	3.07	3.07
Ozone	4.43	4.43
Abalone	2.26	1.95

error obtained by the established analytic method. The decreasing trend of the error during establishing an optimal value of the parameter k proves this.

Results for data tested in cross validation mode are ambiguous. In one case, for combined M5 and k-nn methods, a statistically significant decrease in the error was obtained (the *Boston housing* data set) while in another case the combined methods led to statistically worse results (the *ozone* data set). While establishing the k-opty value for the *abalone* data set, along with the increasing parameter k, the error value decreased systematically (with small departures) and finally k-opty$= |Tr| - 1$. In this case, for the M5+k-nn method we obtained, like for the Ec set, the same error as for the sole M5 algorithm running on the whole training set (without one example removing). Tables 7 and 8 show that, in the case of the *ozone* data set, M5 was selected yet as the output method, thus giving better results than M5+k-nn. This case illustrates how the validation set can protect against selection of a model unduly matched to data (over-fitted model). An obvious observation is that the number of rules increases along with the growth in the parameter k. For $k < 10$, the M5 algorithm generated one rule in a majority of cases. Hence it generated a model of multiple regression.

For the data sets considered, the methodology we present proved better than the ANNBFIS network in four out of six cases. This concerns results obtained on testing sets; on training sets, ANNBFIS definitely wins (in five out of six cases). This means that the ANNBFIS network has no mechanisms protecting against unduly matching to training data. Such mechanisms are included in the M5 algorithm, which applies rules pruning. Because of that our method achieves better generalization results.

Received: 7 February 2011
Revised: 10 August 2011