

# A COMPARATIVE STUDY OF DATA-DRIVEN MODELING METHODS FOR SOFT-SENSING IN UNDERGROUND COAL GASIFICATION

JÁN KAČUR\*, MILAN DURDÁN, MAREK LACIAK, PATRIK FLEGNER

*Technical University of Košice, Faculty BERG, Institute of Control and Informatization of Production Processes, Němcovej 3, 040 01 Košice, Slovak Republic*

\* corresponding author: [jan.kacur@tuke.sk](mailto:jan.kacur@tuke.sk)

**ABSTRACT.** Underground coal gasification (UCG) is a technological process, which converts solid coal into a gas in the underground, using injected gasification agents. In the UCG process, a lot of process variables can be measurable with common measuring devices, but there are variables that cannot be measured so easily, e.g., the temperature deep underground. It is also necessary to know the future impact of different control variables on the syngas calorific value in order to support a predictive control. This paper examines the possibility of utilizing Neural Networks, Multivariate Adaptive Regression Splines and Support Vector Regression in order to estimate the UCG process data, i.e., syngas calorific value and underground temperature. It was found that, during the training with the UCG data, the SVR and Gaussian kernel achieved the best results, but, during the prediction, the best result was obtained by the piecewise-cubic type of the MARS model. The analysis was performed on data obtained during an experimental UCG with an ex-situ reactor.

**KEYWORDS:** Underground coal gasification, syngas calorific value, underground temperature, time series prediction, machine learning, soft-sensing.

## 1. INTRODUCTION

### 1.1. UNDERSTANDING UCG TECHNOLOGY

Underground coal gasification (UCG) represents an in-situ controlled combustion of coal where valuable gases (i.e., syngas) are produced. The UCG represents an alternative to traditional coal mining methods. The UCG allows to mine coal from deep coal seams, seams affected by tectonic disturbances, seams with a low grade, or seams that have a thin stratum profile. Various coal types can be gasified, e.g., lignite or bituminous. The UCG offers a low surface damage, low solid waste discharge and lower emissions of SO<sub>2</sub>, NO<sub>x</sub> to the air than the traditional coal mining.

For an industrial gasification, at least two boreholes should be drilled (i.e., inlet and outlet). Inlet borehole serves as a supply well for gasification agents (i.e., air, oxygen, and steam), and outlet borehole as the exhaust of the produced syngas. Inlet and outlet boreholes are usually linked by various methods in order to create a gasification channel [1].

The main chemical reactions that occur during the UCG are drying, pyrolysis, combustion, and gasification of solid hydrocarbons. For the improvement of the UCG, it must be ensured that combustion reactions produce sufficient energy for the heating of reactants. It is also necessary to overcome the heat losses from the georeactor and to support the rate of endothermic gasification reactions [2]. The UCG is performed as an autothermic process where the heat in the coalbed is generated by an injection of oxygen from the injection well and by means of combustion re-

actions with carbon. The UCG essentially represents the acquisition of a spatially and thermally decomposed reaction zone in the coalbed, which overlaps regions of coal oxidation, coal reduction, and coal pyrolysis. The incoming air causes that the coal burns, the exothermic process releases heat and consumes oxygen. When coal is heated and CO is produced, the Boudouard chemical reactions (i.e.,  $\text{CO}_2 + \text{C} \rightleftharpoons 2\text{CO}$ ) is one of the most important chemical reaction. Raw, pure gas from the UCG consists predominantly of H<sub>2</sub>, CO, CO<sub>2</sub>, CH<sub>4</sub>, higher hydrocarbons, tar, impurities and small quantities of SO<sub>x</sub>, NO<sub>x</sub>, and H<sub>2</sub>S [3]. In terms of the calorific value, gases, such as CO, H<sub>2</sub> and CH<sub>4</sub>, are valuable, but higher hydrocarbons also contribute to the calorific value. The syngas can be used for generating electricity, to produce synthetic natural gas or various chemical products.

### 1.2. MEASUREMENT AND MONITORING IN UCG

The efficiency of the coal-to-gas transformation depends on the UCG monitoring and control and the various coal seam parameters. The main reason for the UCG monitoring is operating the technology more efficiently and increasing the quality of the produced gas, cost reduction and to meet regulatory requirements. Monitoring also informs about the control decision effects, injection rates, syngas composition, temperatures, pressures, cavity size, fractures, and when to stop the gasification.

In the UCG, various process variables can be monitored. These variables can be used for the data-driven

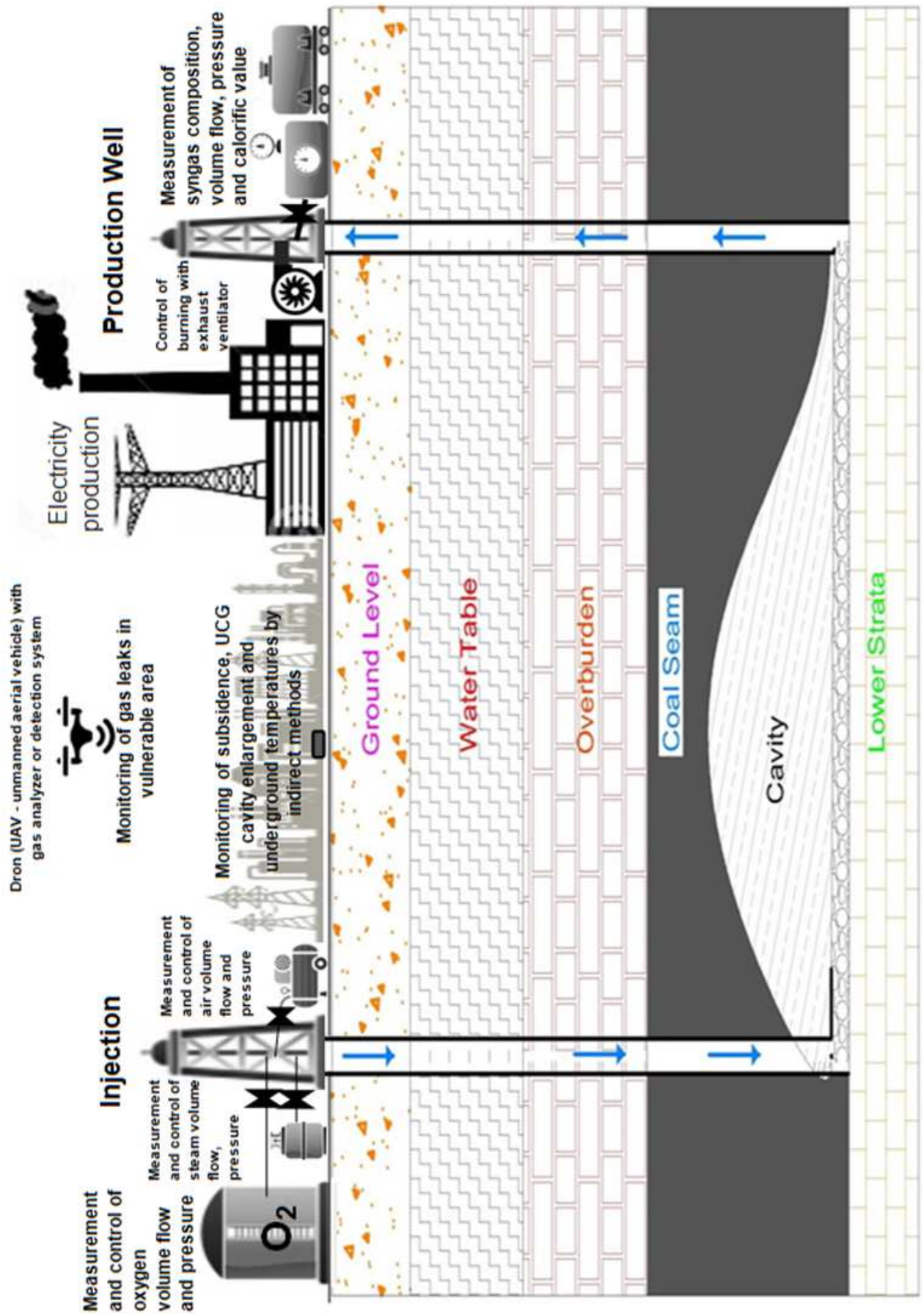


FIGURE 1. Scheme of measurement and control in UCG.

modeling of the process behaviour. In the terms of the process control, it is needed to monitor volume flows and pressures (i.e., overpressure) of injected oxidizers (i.e., air, oxygen, water vapor). On the outlet, the volume flow of produced syngas and regulated underpressure can be monitored. Of course, it is needed to monitor concentrations of syngas components that affect the calorific value (e.g., CO, CO<sub>2</sub>, CH<sub>4</sub> and H<sub>2</sub>). The volume flow, pressure, and composition of injected gasification agents can substantially affect the composition of the produced syngas [2].

The measurement of the temperature inside oxidizing and reducing zones is most problematic. There are two methods of an indirect measurement of the underground temperature - from the current syngas composition [4]) or method based on rules of heat transfer [5, 6].

Recently, methods of monitoring the underground temperature by measuring carbon isotopes and by measuring emissions of radon to the surface appeared [7]. Figure 1 presents the basic scheme of the process variables measurement.

### 1.3. MODELING AND PREDICTION IN UCG

In the last years, an increased demand for an online and accurate measurement of some process variables that cannot be measured by conventional methods has occurred. This is the measurement of process variables in an aggressive (e.g., high temperature) or physically inaccessible (e.g., underground) environment. Similarly, in the UCG, modelling and prediction methods need to be applied in order to determine the process parameters when going deep underground. These variables are decisive in increasing the efficiency and quality of the production. For this reason, different predictors and models, which can calculate desired process variables based on other observations, are developed and applied. These models often serve as support systems for the control of the technological process.

Predictive modeling usually uses statistics to predict the future behaviour of the process and is often associated with machine learning. The most popular are methods of a regression analysis where the output is a regression model. Almost every regression model can serve for the prediction. Some primordial regression analyses of the UCG were previously performed in [8].

The time series prediction is a challenging research area with broad application prospects. The soft-sensing methods for the data estimation and prediction are widely used in the industry. Various approaches to modelling and data prediction have been explored in the world. Unfortunately, there is only scarce evidence of the UCG models oriented for process control and soft-sensing.

Soft sensors based on data-driven predictive modelling are very useful in industry, especially in operations where important process variables cannot be

measured directly by a conventional hardware. Soft sensors use various models that enable real-time estimating process variables without a hardware sensor. They can provide less expensive and quicker process data than slow and costly hardware devices. However, the soft sensors can be run in parallel with the hardware devices for the measurement [9].

Well-known software algorithms that can be seen as soft sensors include Kalman filters. More recent implementations of soft sensors in the UCG use Neural Network (NN) or Fuzzy Computing. Unfortunately, there is only scarce evidence of using the machine learning methods for a prediction of the underground temperature, syngas calorific value or syngas composition in the UCG.

For example, Ji and Shi [10] have used a hybrid radial basis function (RBF) NN as a learning scheme for the temperature prediction of Texaco gasifier. In order to increase the performance of the NN, the number of hidden neurons was determined by a fuzzy C-means algorithm and particle swarm optimization algorithm. Recently, Uppal et al. [11–13] have proposed a control oriented one dimensional packed bed model of the UCG for an estimation of the syngas composition. This model works with a connection to the sliding mode controller to maintain the desired syngas calorific value.

Learning schemes for the coal gasification to support the process control can also be found in [14]. Multiple Neural Network (MNN) for the syngas composition prediction and dynamic principal component analysis was proposed in [15]. Other researchers, e.g., Guo et al. [16] have modeled coal gasification with a hybrid NN. A model of a coal gasification was developed, incorporating a first-principles model with an NN parameter estimator. The hybrid NN was trained with experimental data for the two coals and gave a good performance in the process modeling.

Other effective methods have also been applied to the gasification. Liu et al. [17] have proposed a data-driven modeling for fixed-bed intermittent gasification processes inside UGI gasifiers by an enhanced lazy learning combined with a relevance vector machine. Authors have used the Bayesian learning framework for the modeling gasifier's temperature. The effectiveness of the enhanced lazy learning approach combined with the relevance vector machine for the modelling of the UGI gasification processes has been verified by a series of experiments based on the data collected from practical fields. Similarly, for the same problem of the data-driven modeling for the UGI gasification process, a variable structure of a genetic BP NN was used in [17]. The UGI represents a gasification named by UGI Company. The UGI gasifier is an atmospheric fixed bed, solid-state slag coal gasification equipment. The prediction of the syngas composition based on the thermodynamic model can be found in [18, 19]. In the past, the application of the one-dimensional time-dependent numerical computational model of the

UCG in a packed bed has also been investigated with a verification on laboratory measurements [20]. The model based on nonlinear partial differential equations was capable of estimating the syngas composition and temperature distribution. A novel dynamic soft-sensing method based on an impulses response template for the Shell coal gasification process was developed in [21]. The proposed model can predict the syngas composition during the coal gasification.

The application and comparison of the efficiency of various learning methods, i.e., NN, Adaptive Regression Splines (MARS) or Support Vector Machines (SVMs) in the UCG data prediction has not yet been the subject of an extensive study, but similar applications in steel making processes and biomass gasification are registered (e.g., [22, 23]).

The purpose of the UCG monitoring is to provide a better understanding of how the syngas is produced. For this reason, it is needed to know what temperature is reached in the underground oxidizing zone.

This work examines a potential learning method that can be implemented in the proposal of the soft sensor for the data prediction in the UCG. An underground geo-reactor is different from another industrial plant because the coal seam was created by nature and it is not possible to see what is in the underground when the UCG is in progress. For the UCG, new research and technologies are aimed to make measurements of process variables faster and non-destructive, which would allow having a smart non-intrusive quality sensor at hand. In this paper, in order to training data-driven modeling, a data set from an experimental UCG was used.

In the data-driven (i.e., black box) modelling, input and output data are used in order to create a statistical model. In order to find a prediction apparatus for the UCG data prediction, the machine learning approach has been examined. One of the interesting advantages of the machine learning is that a system, randomly initialized and trained on some data sets, will eventually learn good feature representations for a given task.

In the following sections, three learning methods, Back-Propagation NN (BPNN), MARS and Support Vector Regression (SVR), are examined in order to support soft-sensing in the UCG. Predictive methods are evaluated using statistical approaches and calculating the performance index. The methods were applied to the experimental data obtained from the experimental trial of the UCG. The results from the three methods are compared to each other for determining which method is the most suitable for the UCG.

## 2. ANALYSIS OF SELECTED MODELING METHODS

### 2.1. MULTILAYER FEED-FORWARD NEURAL NETWORKS

The inherent non-linear structure of the NN is well suited for solving many real-world problems. In recent years, several models of NNs have been designed and optimized to solve a specific problem. NNs models have an excellent ability to learn from experience and are also suitable non-parametric methods that do not require many limiting factors. A multilayer feed-forward neural networks are most commonly used as a universal means for classification and prediction. They consist of sensoric units, so-called input nodes, that form an input layer, one or more hidden layers with counting nodes and an output layer also with counting nodes. The signal passes through the network forwards across the individual layers. In a multi-layer feed-forward NN, all neurons of the previous layer are linked to each neuron of the following layer. However, there are no interconnections between the neurons at the level of the same layer as well as the direct interconnection of the input layer neurons with the neurons that are two layers further.

In this paper, the back-propagation algorithm was used for the NN modeling of the UCG. This simple gradient algorithm was proposed by [24, 25]. There are more approaches to explaining the principle of NNs and the back-propagation method, e.g., using the projection pursuit regression (PPR) [26]. In this paper, a graph-oriented approach with an extensive description that can be found in [27] has been used. The input and output scheme considered for the NN for the UCG data prediction is shown in Section 4.1 (see Figure 7).

Formally, the NN is defined as the oriented graph  $G = (V, E)$  where  $V = \{v_1, v_2, \dots, v_N\}$  is the set of vertices and  $E = \{e_1, e_2, \dots, e_M\}$  the set edges. Denote a non-empty vertex or edge set of the graph  $G$ , containing  $N$  nodes (neurons) and  $M$  (connections). The set of  $V$  neurons is distributed to disjunctive subsets of  $V = V_I \cup V_H \cup V_O$  where  $V_I$  contains  $N_I$  input neurons, which are adjacent to only the outgoing edges.  $V_H$  contains  $N_H$  hidden neurons, which are adjacent to the outgoing edges as well as to the incoming ones. Finally,  $V_O$  contains  $N_O$  output neurons, which are adjacent only to incoming edges.

For an acyclic NN, the neurons can be arranged into layers where  $L_1 = V_I$  is an input layer (i.e., contains only input neurons),  $L_2, L_3, \dots, L_{t-1}$  are hidden layers and  $L_t$  is an output layer. The NN determined by the acyclic graph is usually chosen so that the neurons from the two adjacent layers are joined together by all possible connections. Neurons and connections are rated by real numbers. Each neuron  $v_i$  is rated by a threshold  $\vartheta_i$  and an activity  $x_i$ . Similarly, each connection  $(v_j, v_i)$  is rated by a weighting coefficient (or simply, by weight)  $w_{ij}$ . The activities of hidden

and output neurons are determined by the following equation [27]:

$$x_i = t(\xi_i) \tag{1}$$

$$\xi_i = \sum_{j \in \Gamma_i^{-1}} w_{ij} x_j + \vartheta_i \tag{2}$$

where a summation runs through neurons which are predecessors of the neuron  $v_i$  and variable  $\xi$  is called the potential of the neuron  $v_i$ .

For defining the oriented graph  $G$ , we used the view of  $\Gamma$  that assigns to each vertex  $v \in V$  a subset  $\Gamma(v) \subset V$  that contains those neurons that are endpoints on the connections that are going out from the vertex  $v$  [28].

The neurons of the  $\Gamma(v)$  subset are called the descendants of the vertex  $v$  in the graph  $G$ . The “inverse”  $\Gamma^{-1}$  view will assigns to each vertex  $v \in V$  a subset  $\Gamma^{-1}(v) \subset V$  of the gamma composed of the “descendants” of the vertex  $v$  in the graph  $G$ .

Neuronal activities form a vector  $\mathbf{x} = (x_1, x_2, \dots, x_N)$ . This vector can be formally decomposed into three subsets containing input, hidden, and output activities

$$\mathbf{x} = \mathbf{x}_I \oplus \mathbf{x}_H \oplus \mathbf{x}_O \tag{3}$$

Hidden activities are not explicitly mentioned; they only play the role of intermediate results. In general, to calculate activities from the layer  $L_i$  (where  $i > 1$ ), it is only needed to know the activities from the lower layers  $L_1, L_2, \dots, L_{i-1}$ . In this recursive manner, the activities of all neurons can be gradually calculated. Activities of output neurons are calculated as the last. For this reason, for NNs represented by the acyclic graph, the name *εfeed-forward NNsε* is used. The adaptation of the NN is based on searching for such threshold and weighting coefficients, which, for a given pair of input vector of activities  $\mathbf{x}_I$  and desired output vector of activities  $\hat{\mathbf{x}}_O$  i.e.,  $\mathbf{x}_I/\hat{\mathbf{x}}_O$  and the calculated output vector  $\mathbf{x}_O$ , minimize the difference between the output activities  $\mathbf{x}_O$  and  $\hat{\mathbf{x}}_O$ . Vector  $\hat{\mathbf{x}}_O$  represents the desired measured (i.e., experimental) data.

The aim of the adaptation process is to find such thresholds and weighting coefficients that minimize the objective function  $E$ . For more pairs of input and output vectors

$$\mathbf{x}_I^{(1)}/\hat{\mathbf{x}}_O^{(1)}, \mathbf{x}_I^{(2)}/\hat{\mathbf{x}}_O^{(2)}, \dots, \mathbf{x}_I^{(r)}/\hat{\mathbf{x}}_O^{(r)}, \tag{4}$$

which represent the training set, the objective function has the following form:

$$E = \sum_{i=1}^r E^i = \sum_{i=1}^r \frac{1}{2} (\mathbf{x}_O^{(i)} - \hat{\mathbf{x}}_O^{(i)})^2 \tag{5}$$

where  $\mathbf{x}_O^{(i)}$  is the output vector of the NN as a response to the input vector  $\mathbf{x}_I^{(i)}$  and  $\hat{\mathbf{x}}_O^{(i)}$  the desired output vector is assigned to the input  $\mathbf{x}_I^{(i)}$ .

This minimization of the non-linear objective function can be performed by many optimization methods known in numerical mathematics.

The most effective are so-called gradient methods, based on the use of a gradient of the objective function for the iterative construction of an optimal solution.

When calculating the gradient of the objective function that contains more than one pair of input-output vectors  $\mathbf{x}_I/\hat{\mathbf{x}}_O$  (see equation (5)) then, the overall gradient of the objective function is simply determined as the sum of the gradients for all pairs  $\mathbf{x}_I/\hat{\mathbf{x}}_O$  of the training set (4).

$$\text{grad } E = \sum_{i=1}^r \text{grad } E^{(i)} \tag{6}$$

where the objective function  $E^{(i)}$  is defined for  $i$ -th pair  $\mathbf{x}_I/\hat{\mathbf{x}}_O$  of the training set.

Formally, the adapted NN is described by the coefficients determined as

$$(\bar{\mathbf{w}}, \bar{\boldsymbol{\vartheta}}) = \underset{(\mathbf{w}, \boldsymbol{\vartheta})}{\text{argmin}} E(\mathbf{w}, \boldsymbol{\vartheta}) \tag{7}$$

The settings of an applied BPNN and evaluation of its performance on the UCG data prediction is discussed in Section 4.1.

## 2.2. MULTIVARIATE ADAPTIVE REGRESSION SPLINES

Multivariate Adaptive Regression Splines (MARS) is the method of regression that was developed by [29]. Many works have been published that discussed the MARS method [26, 30–34]. It is a non-parametric regression technique that looks like an extension of linear models. This technique automatically models non-linearities and interactions between variables. This technique is also more flexible than linear models and is suitable for processing large data series. This technique can serve for a quick prediction of time series. MARS is similar to recursive partitioning where input data are divided into discontinuous regions of varying size. Then, the local model is created for each region. The size of each area is set by MARS as required. In MARS, these regions are smaller when the relationship between input and output is more complex. MARS, like the recursive partitioning technique, performs the automatic selection of variables, so the model includes important (useful) variables and excludes non-essential (as opposed to NN). The MARS model is adapted based on the input training data, and cross-validation is used to validate the resulting model. The resulting model may not only be stored in a PC but is also portable, and there is easy to see the impact of each predictor (the model is easier to understand by humans). In order to create the MARS model, the training data vectors, i.e., the inputs (observations) and outputs (targets) are needed. Training data are split into several splines on an equivalent interval basis [29].

The data are, in each spline, split into many subgroups, and several knots are created that can be placed between different input variables or different intervals in the same input variable to separating subgroups [31].

In MARS, the regression function called a basis function (BF) is approximated by smoothing splines for a general representation of data in each subgroup. Between any two knots, the model can characterize the data either globally or by using the linear regression. The BF is unique between any two knots and is shifted to another BF at each knot [29, 35].

The two BFs in two adjacent domains of data intersect at the knot to make the model outputs continuous. MARS creates a curved regression line to fit the data from subgroup to subgroup and from one spline to another spline. For evading over-fitting and over-regressing, the shortest distance between two neighboring knots is predetermined to prevent too few data in a subgroup.

In the MARS method, the goal is to find the dependency of variables  $y_i$  on one or more independent variables  $\mathbf{x}_i$ . The following regression sample is considered:

$$D = \{\mathbf{x}_i, y_i\}_{i=1}^N = \{x_{1i}, \dots, x_{ni}, y_i\}_{i=1}^N \quad (8)$$

where  $\mathbf{x}_i \in \mathbb{R}^p$  is the  $i$ -th vector of the independent variable,  $y_i$ , ( $i = 1, \dots, N$ ) is the dependent variable,  $N$  is the number of independent variables,  $n$  is the number of data in  $\mathbf{x}_i$ . The relationship between  $y_i$  and  $\mathbf{x}_i$  ( $i = 1, \dots, N$ ) can be represented as:

$$y_i = f(x_i^1, x_i^2, \dots, x_i^p) + \varepsilon = f(\mathbf{x}_i) + \varepsilon, \quad (9)$$

where  $f$  is an unknown function, and  $\varepsilon$  is an error ( $\varepsilon \sim N(0, \sigma^2)$ ). The single valued deterministic function  $f$ , captures the joint predictive relationship of  $y_i$  on  $(x_i^1, x_i^2, \dots, x_i^p)$ . The additive stochastic component  $\varepsilon$ , whose expected value is determined to be zero, usually reflects the dependence of  $y_i$  on values other than  $(x_i^1, x_i^2, \dots, x_i^p)$ , that are neither controlled nor observed.

In the one-dimensional case, splines are expressed in terms of piecewise linear basis functions,  $(x - t)_+$  and  $(t - x)_+$  with the node in  $t$ . The “+” means a positive part. These functions are truncated linear functions, for  $x \in R$ .

$$(x - t)_+ = \begin{cases} x - t, & \text{If } x > t, \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

and  $(t - x)_+ = \begin{cases} t - x, & \text{If } x < t, \\ 0, & \text{otherwise} \end{cases}$

Each function (i.e.,  $(x - t)_+$  and  $(t - x)_+$ ) is piecewise linear, with a knot at the value  $t$ . They are marked as *linear splines*. These two functions are named as a reflected pair.

In the multidimensional case the idea is to form *reflected pairs* for each input component  $x^j$  of the vector  $\mathbf{x} = (x^1, \dots, x^j, \dots, x^p)^T$  with knots at each observed

value  $x_i^j$  of that input ( $i = 1, 2, \dots, N; j = 1, 2, \dots, p$ ). Thus, a set of constructed basis functions can be represented in the form:

$$\mathcal{C} = \left\{ (x^j - t)_+, (t - x^j)_+ \mid t \in \{x_1^j, x_2^j, \dots, x_N^j\}, \right. \\ \left. j \in \{1, 2, \dots, p\} \right\} \quad (11)$$

If all input data are different, then, in the set of  $2Np$  basis functions, each of them depends on only one variable  $x^j$ . For example,  $B(\mathbf{x}) = (x^j - t)_+$  is regarded as a function over the entire input space  $\mathbb{R}^p$ . The basic functions used for approximation are as follows:

$$B_m(\mathbf{x}) = \prod_{k=1}^{K_m} \left[ s_{k,m} \cdot (x^{v(k,m)} - t^{km}) \right]_+ \quad (12)$$

where  $K_m$  is the total number of truncated linear functions in the  $m$ -th basis function (i.e., it is the number of “splits” that gave rise to  $B_m$ ),  $x^{v(k,m)}$  is the component of the vector  $\mathbf{x}$ , related to the  $k$ -th truncated linear function in the  $m$ -th basic function,  $t^{km}$  is the corresponding node, and  $s_{km} \in \{\pm 1\}$ .  $K_m$  is the user defined degree order of the interaction term and  $s_{k,m}$  represents the direction of the univariate term, which could be positive or negative.

The model-building strategy is like a forward stepwise linear regression, but instead of using the original inputs, it is allowed to use functions from the set  $\mathcal{C}$  of their products. Therefore, the MARS model can be expressed by the following equation [26]:

$$y = \hat{f}(\mathbf{x}) + \varepsilon = c_0 + \sum_{m=1}^M c_m B_m(\mathbf{x}) + \varepsilon \quad (13)$$

where  $y$  is the output variable,  $\mathbf{x}$  is the vector of input variables,  $M$  is the number of basis functions in the model (i.e., number of spline functions,  $c_0$  is the coefficient of the constant basis function  $B_0$ , and sum is over the basis functions  $B_m$  produced by algorithm that implements the stepwise forward part of the MARS strategy by incorporating the modification to recursive partitioning. The coefficients  $c_m$  are estimated by minimizing the residual sum-of-squares (i.e., by standard linear regression).  $B_m(\mathbf{x})$  is the  $m$ -th function in  $\mathcal{C}$ , or a product of two or more such functions.

The most important thing in this model is the choice of basis functions. In the beginning, the model contains a single function  $B_0(\mathbf{x}) = 1$  and all functions from the set  $\mathcal{C}$  are possible candidates for an inclusion in the model. As in the linear regression, setting  $B_m$ , the coefficients  $c_m$  can be found by the method of least squares.

Another subroutine of MARS performs the backward deletion strategy wherein each iteration causes one unnecessary (i.e., redundant) basis function to be deleted. The inner loop of the algorithm will select



one function to be deleted. A function whose removal either mostly improves the fit or at least degrades it will be deleted. However, the constant basis function  $B_1(\mathbf{x}) = 1$  is never removed.

The settings of MARS, its result forms, and evaluation of its performance are discussed in Section 4.2.

### 2.3. SUPPORT VECTOR REGRESSION

Back-propagation NNs are capable of representing general non-linear functions, but its disadvantage is often very difficult teaching, because, practically, there is always a risk of a deadlock in the local minimum of the error function, and in addition, the learning is highly complicated by looking for a high number of weights in the multidimensional space. An alternative and relatively new approach are the so-called Support Vector Machines (SVMs). The SVMs are used for time series prediction and classification tasks. These methods represent the field of the so-called kernel machines and exploit the benefits provided by effective algorithms for finding a linear boundary while being able to represent highly complex non-linear functions. Kernel functions methods try to find an optimal linear separator. The optimal linear separator in an SVM algorithm is searched using the quadratic programming method.

In Support Vector Regression (SVR) the data  $\mathbf{x} \in \mathcal{X}$  are mapped into a high-dimensional feature space  $\mathcal{F}$  via a nonlinear mapping  $\Phi$ , and to do linear regression in this space [36, 37].

Input data (i.e., observations) are represented by vector  $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_l)$  where  $l$  denotes the size of the sample.

Taking into account regression with one target variable  $y$ , so observations on the examined process can be written as a sequence of pairs  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_i, y_i), \dots, (\mathbf{x}_l, y_l)$ ,  $\mathbf{x}_i \in \mathbb{R}^n$ ,  $y_i \in \mathbb{R}$ . Vector  $\mathbf{x}_i$  represents one pattern of input observations  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{in})$ . In the case of observing process variables during the UCG, this vector may contain measured data from the database.

Thus, a linear regression in a high dimensional (feature) space corresponds to a non-linear regression in the low dimensional input space  $\mathbb{R}^n$ .

The whole problem of the SVR can be rewritten in terms of dot products in the low dimensional input space [38].

$$f(\mathbf{x}) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})) + b = \sum_{i=1}^l (\alpha_i - \alpha_i^*) k(\mathbf{x}_i, \mathbf{x}) + b \quad (14)$$

Given two points  $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ , the function that returns the inner product between their images in the space  $\mathcal{F}$  is known as the kernel function. In equation (14) a kernel function  $k(\mathbf{x}_i, \mathbf{x}_j) = (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j))$  is introduced.

Kernel type	Kernel function
Gaussian (RBF) kernel	$k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \ \mathbf{x}_i - \mathbf{x}_j\ ^2}$
Linear kernel	$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
Polynomial kernel	$k(\mathbf{x}_i, \mathbf{x}_j) = (\gamma (\mathbf{x}_i^T \mathbf{x}_j + 1))^d$
Sigmoid kernel	$k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i^T \mathbf{x}_j + d)$

TABLE 1. Overview of common kernels used by SVR ( $\gamma$  is a kernel parameter controlling the sensitivity of the kernel function, and  $d$  is an integer).

Parameters  $\alpha_i, \alpha_i^*$  are the solutions of the quadratic programming problem [37]. These parameters have an intuitive interpretation as forces pushing and pulling the estimate  $f(\mathbf{x}_i)$  towards the measurements  $y_i$ . Parameter  $b$  is a threshold.

Common kernels are summarized in Table 1.

In this paper, the linear epsilon-insensitive SVM ( $\varepsilon$ -SVM) regression has been used.

For this special cost function, the Lagrange multipliers  $\alpha_i, \alpha_i^*$  are often sparse, i.e., they result in non-zero values after the optimization only if they are on or outside the boundary, which means that they fulfill the Karush-Kuhn-Tucker (KKT) conditions.

The  $\varepsilon$ -insensitive cost function is given by

$$C(f(\mathbf{x}) - y) = \begin{cases} |f(\mathbf{x}) - y| - \varepsilon & \text{for } |f(\mathbf{x}) - y| \geq \varepsilon \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

In  $\varepsilon$ -SVM regression, the set of training data includes predictor variables and observed response values. The goal is to find a function  $f(\mathbf{x})$  that deviates from  $y$  by a value no higher than  $\varepsilon$  for each training point  $\mathbf{x}$ , and at the same time is as flat as possible.

In SVR the kernel matrix  $\mathbf{K} = (k(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^l$  ( $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ ) is introduced. It is a symmetric positive definite matrix of inner products between all pairs of points  $\{\mathbf{x}_i\}_{i=1}^l$ . Each element represents the inner product of the predictors transformed by  $\Phi$ . However, it is not needed to know  $\Phi$ , because the kernel function can generate the kernel matrix directly. Using this approach, the non-linear SVR finds the optimal function  $f(\mathbf{x})$  in the transformed predictor space. The prediction of new values is based on a function that depends only on the support vectors:

$$f(\mathbf{x}) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \mathbf{K}(\mathbf{x}_i, \mathbf{x}) + b \quad (16)$$

where  $\alpha$  and  $\alpha^*$  are non-negative Lagrange multipliers for each observation  $\mathbf{x}$ . A threshold  $b$  can be determined from the Lagrange multipliers.

Lagrange coefficients can be found by minimization of the following function [39]:

$$L(\alpha) = \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) + \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) - \sum_{i=1}^l y_i (\alpha_i^* - \alpha_i) \quad (17)$$

subject to the constraint

$$\sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0; \quad \forall i : 0 \leq \alpha_i \leq C; \quad (18)$$

$$\forall i : 0 \leq \alpha_i^* \leq C$$

The KKT complementarity conditions are

$$\begin{aligned} \forall i : \alpha_i (\varepsilon + \xi_i - y_i + f(\mathbf{x}_i)) &= 0 \\ \forall i : \alpha_i^* (\varepsilon + \xi_i^* + y_i - f(\mathbf{x}_i)) &= 0 \\ \forall i : \xi_i (C - \alpha_i) &= 0 \\ \forall i : \xi_i^* (C - \alpha_i^*) &= 0 \end{aligned} \quad (19)$$

where slack variables  $\xi$  and  $\xi^*$  for each point ensure that regression errors are up to the value of  $\xi$  and  $\xi^*$  and meet the desired conditions. KKT complementarity conditions are optimization constraints required to obtain the optimum. These conditions indicate that all observations strictly inside the epsilon tube have Lagrange multipliers  $\alpha_i = 0$  and  $\alpha_i^* = 0$ . Observations with nonzero Lagrange multipliers are called *support vectors*. The constant  $C$  is the box constraint, a positive numeric value that controls the penalty imposed on observations that lie outside the epsilon margin ( $\varepsilon$ ) and helps to prevent overfitting, i.e., regularization [40].

The minimization problem can be solved by common quadratic programming techniques, e.g., the Chunking and working set method, Sequential minimal optimization (SMO), or Iterative single data algorithm (ISDA).

For the modeling of the UCG process, the epsilon-insensitive SVM ( $\varepsilon$ -SVM) regression has been used. To solve the optimization problem, the algorithm of SMO has been used.

### 3. EXPERIMENTAL UCG IN EX-SITU REACTOR

For the purpose of verifying the UCG, laboratory equipment has been created. The base is an experimental coal gasifier, i.e., an ex-situ reactor or so-called syngas generator (see Figure 2) and a set of devices for measurement and control. The ex-situ reactor was constructed so that the bedding of coal with overburden and under-burden layers can simulate the real coal seam. Several experiments as trials of a real UCG were performed with the ex-situ reactor. This laboratory gasification equipment was well described in [41, 42].

Similar trials of the UCG on a laboratory ex-situ reactor can be found in [43, 44]. Various gasification agents (i.e., oxidizers), ways of bedding coal and monitoring of the UCG process were tested there.

The gasification in the ex-situ reactor is based on the control of the flow of the inlet gasification agents (i.e., air and oxygen) and pressure on an outlet. Lignite coal from the Slovak mine that is suitable for the UCG was gasified. The composition of the coal that was



FIGURE 2. Experimental coal gasifier (ex-situ reactor).

gasified and factors that affect the UCG can be found in [41]. The influence of various gasification agents (i.e., its flows and pressures) on the syngas quality was discussed in [2].

The bedding of coal in the ex-situ reactor was made on the basis by the rules of the similarity theory. The goal was to obtain the similarity with the real coal seam. Blocks of coal merged into one coal unit were used when preparing a physical model of the coalbed. In order to make the physical model airtight, layers of over-burden and under-burden contained sand mixed with water glass. In addition, the reactor was tilted at  $10^\circ$ , in order to get as close as possible to the real coal seam. The coal used in the experiment was extracted from the underground coalbed with the same inclination. This coalbed (i.e., in mine Cigel - Slovakia, overburden bed) has the potential to be mined in the future by the UCG.

Air, as the primary oxidant, was blown into the pressure vessel by two compressors. The pressure of air injected into the ex-situ reactor was adjusted by a reducing valve. The air flow was controlled by the servo valve and measured by a differential pressure sensor with a centric orifice that was installed in a pipeline. Similarly, the flow of the produced syngas was measured, but the segment orifice was used. The oxygen flow and pressure were controlled by two reducing valves. As a source of the technical oxygen, pressure cylinders were used. Technical oxygen was injected as an auxiliary oxidant into the mixing chamber where it was mixed with air, and the mixture was then injected into the ex-situ reactor. The pressures of the oxidants were measured by a set of pressure transducers. The K-type thermocouples were used to measure the temperatures in the coal model. These



thermocouples allow to measure the temperature up to 1300 °C. Thermocouples were placed in ceramic tubes (for protection in an oxidation-reducing atmosphere) and then inserted into holes drilled in the physical model. The temperature of the coal along the gasification channel, in the overburden and the under-burden was measured. On the outlet from the reactor, the sample of syngas was captured to be analysed by two stationary analysers. The concentration of CO, CO<sub>2</sub>, O<sub>2</sub>, CH<sub>4</sub>, H<sub>2</sub> in the syngas was constantly measured. The syngas calorific value was continually calculated from the composition of the syngas. The pressure at the outlet was controlled by a vacuum fan. The power of the fan was controlled by a frequency inverter. The outlet pressure was measured by one pressure transducer. The produced syngas was finally burned in the combustion chamber. The UCG was tried in two different reactors.

Figure 3 shows a complete technological scheme of the experimental gasification with one generator. All devices for measurement and control (i.e., pressure transducers, differential pressure transducers, servo-valve, thermocouples, switching relays, frequency converter with fan, compressors and gas analysers) were connected to a PLC that provided the basic control loop gasification (i.e., on-off control of compressors, air flow stabilization, temperature stabilization, oxygen concentration stabilization). The PLC was connected to a PC that performed data storage, optimal control [45], and process monitoring [46].

The SCADA/HMI system Promotic was used for the monitoring of the process and setting the controllers [46]. A detailed description of all devices that were used for the measurement and control was presented in [41]. The measured process variables (i.e., flows of gasification agents, pressures, temperatures, syngas composition and calorific value) were recorded in a database which may later be processed as a set of time series. The PC with CPU Intel®Core™ i5-4300U (2.9 GHz) and 8 GB RAM was used for all calculations.

## 4. RESULTS AND DISCUSSION

The flowchart for the proposed soft-sensing in the UCG is shown in Figure 4. This paper focuses on the evaluation of the potential predictive methods that could be used in soft-sensing. Machine learning models are generalized to data similar to those on which they were trained. Although static models, which are time-independent, i.e., they work on a single data set, have been used, their application to the dynamic process should be improved by the continual updating of the training set with the online data. The development of practical on-line prediction soft sensors consists of two stages: training and on-line prediction. A data set from the experimental UCG was used in order to train a data-driven modeling algorithm.

Three prediction methods analysed in the previous section have been applied in order to predict the un-

derground temperature in the oxidizing zone of ex-situ reactor and syngas calorific value. The data obtained during the experimental UCG with the laboratory equipment have been used in the analyses. As an underground temperature in the oxidizing zone, the highest temperature along the gasification channel during the experiment is considered. Observations and target data measured during a one well-running experiment were used in this paper for the analysis.

When the learning method is applied, it is convenient to divide observed data into a training set  $A_{train}$  and test set  $A_{test}$  (i.e., validation set). When choosing a training and test sets, it is recommended to ensure that the data used for the model testing covers a significant range of variations that are supposed to be encountered during the use of the soft sensor. Given that there is no exact rule in the literature how to divide the data into a training and test set for specific learning methods (there are some different recommendations and instruction for experimentation), the models were tested on data from the 10 % and 20 % of the experiment. In general, a higher performance of selected methods was obtained with more data for training (i.e., the ratio between the training set and the test set was 90:10). To compare the performance of three different methods, this paper presents only results when 10 % of the experiment was used for the test. The used test set consisted of 10 % of the data from the end of the experiment. The whole experiment lasted for 70 hours. In simulations, there were 4201 patterns in total and 3781 patterns were used for training. Overview of all regarded observations and targets shows Table 2. The pressure on the outlet is the relative pressure measured on the output pipe from the gasifier. This pressure can also be negative when the power of the exhaust fan is increased. The behaviour of the measured data from the UCG experiment is shown in Figure 5 and Figure 6.

Due to the fact that the highest temperature along the gasification channel is considered to be the temperature in the oxidizing zone weakly correlates with the operating variables (i.e., flows of gasification agents and pressure) and it has some inertia, the decision to ensure its estimation from the composition of syngas measured on the outlet was made. Since the composition of the syngas depends on the temperature in the oxidizing zone, there is an inverse way to determine the temperature that corresponds to the measured concentrations. This decision was also supported by the existence of a large number of uncertainties that occur in the UCG. The propagation of the temperature in the underground is not uniform, i.e., there are different temperatures in the coal, along the gasification channel, in the underburden and overburden. In addition, there is a continual shift of the combustion front. Due to the changing conditions in the underground gasifier (e.g., groundwater, cracks, fractures, gas leaks and shift of combustion front and surface subsidence) it is the process control in conditions of

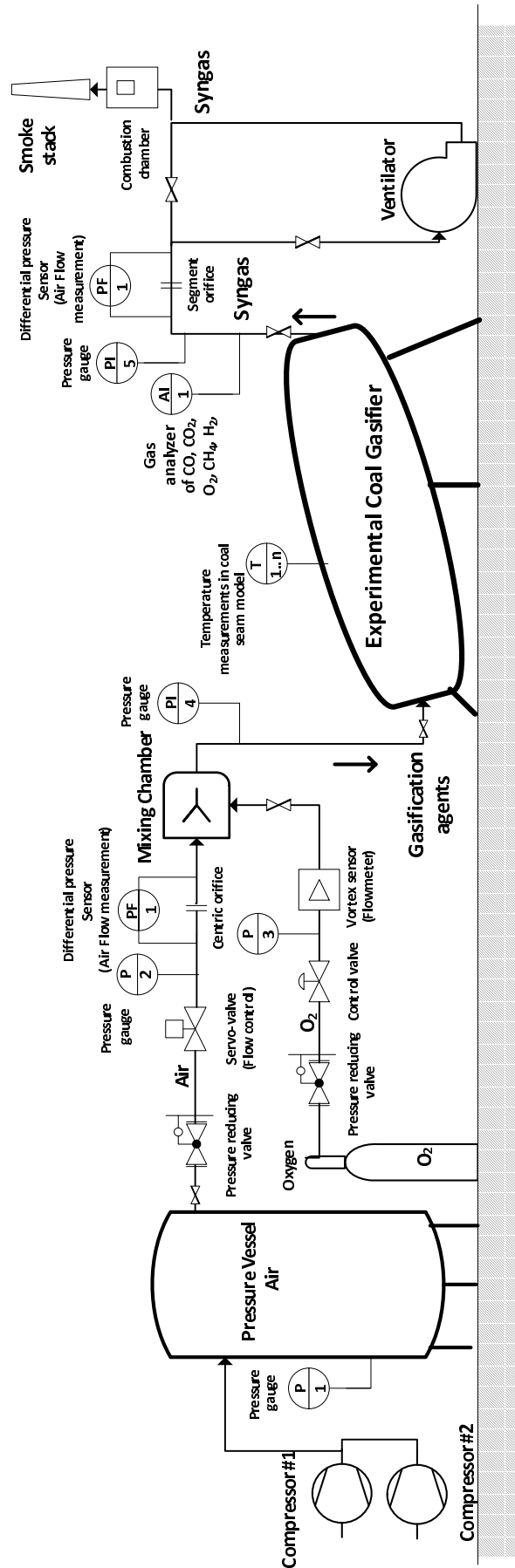


FIGURE 3. Scheme of gasification equipment with one ex-situ reactor (modified after [41]).

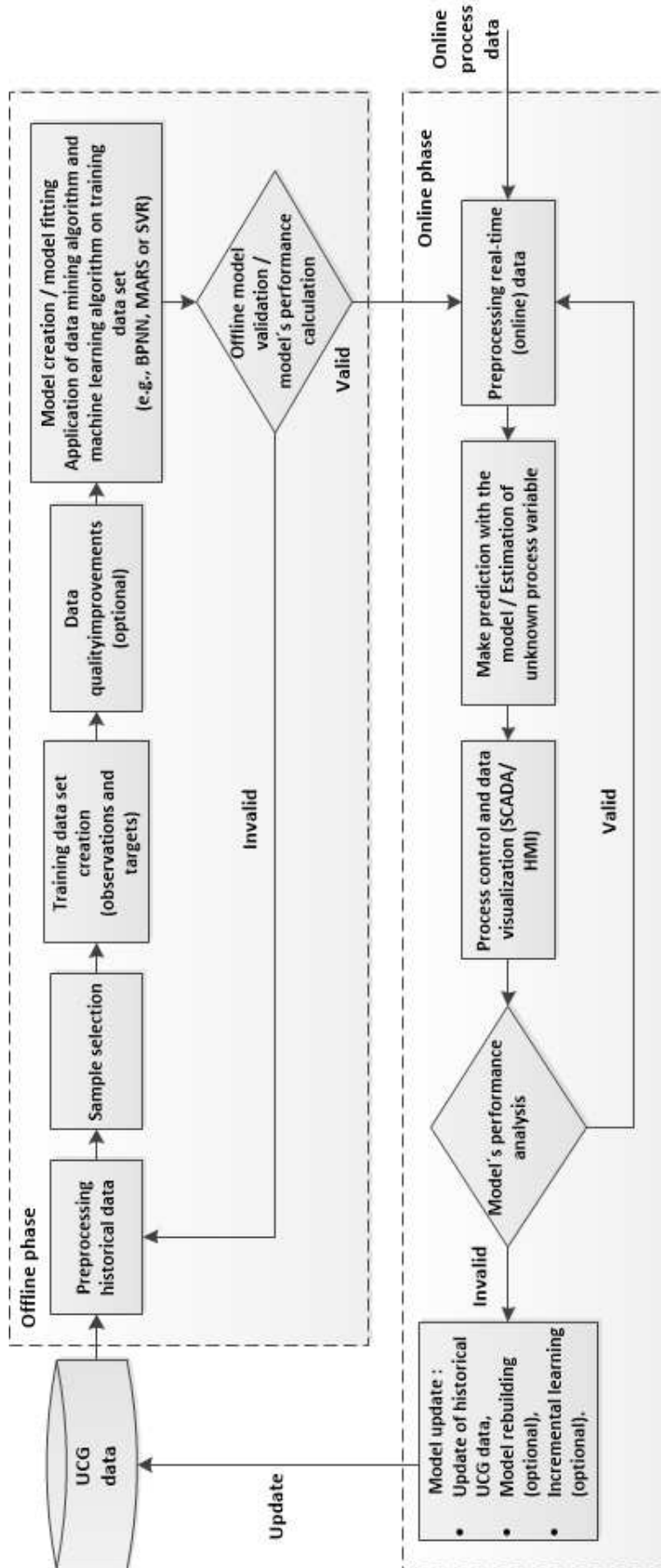


FIGURE 4. The principle of soft-sensing in UCG.

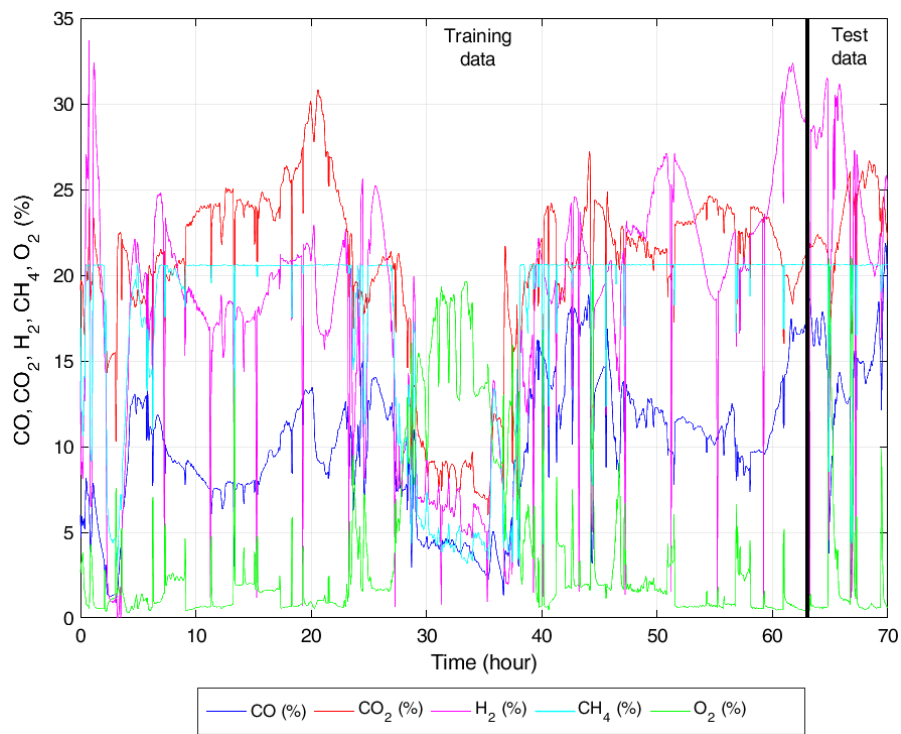


FIGURE 5. Time series of measured syngas composition divided into training and measured data set.

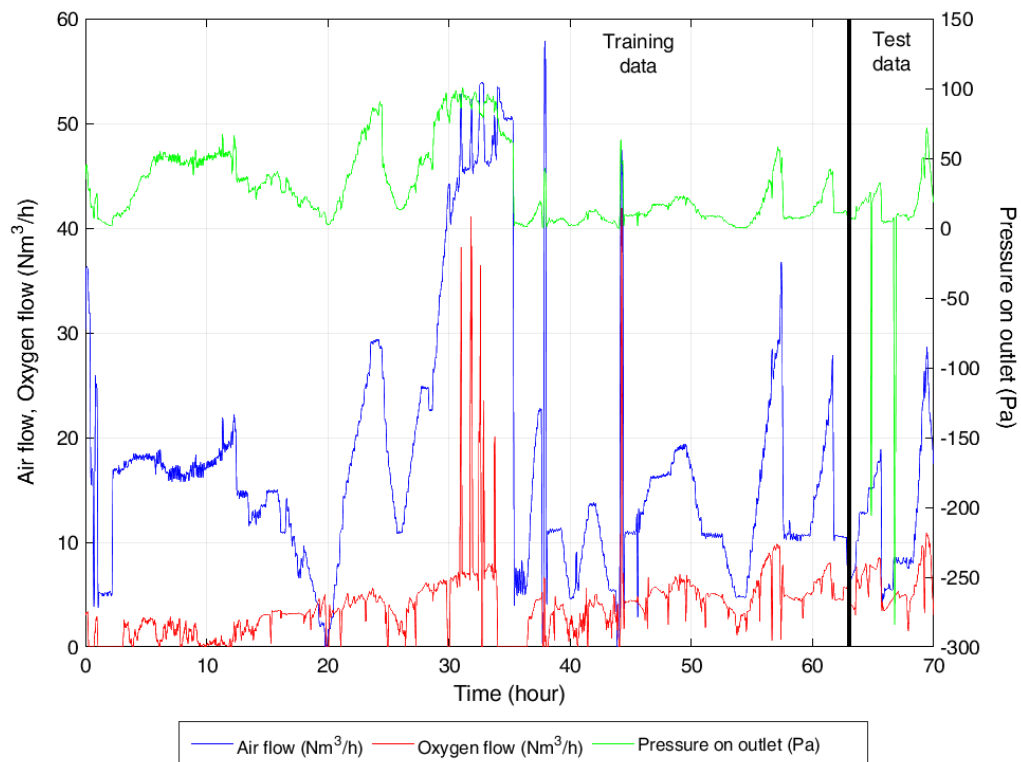


FIGURE 6. Time series of measured control variables divided into training and measured data set.

Target (output process variable $y$ )	Observation (input process variable $x$ )
Calorific value of syngas (MJ/Nm <sup>3</sup> )	$x_1$ - Air flow (Nm <sup>3</sup> /h)
	$x_2$ - Oxygen flow (Nm <sup>3</sup> /h)
	$x_3$ - Pressure on outlet (i.e., underpressure/overpressure) (Pa)
Underground temperature (°C)	$x_1$ - Concentration of CO in syngas (%)
	$x_2$ - Concentration of CO <sub>2</sub> in syngas (%)
	$x_3$ - Concentration of H <sub>2</sub> in syngas (%)
	$x_4$ - Concentration of CH <sub>4</sub> in syngas (%)
	$x_5$ - Concentration of O <sub>2</sub> in syngas (%)

TABLE 2. Observations and targets used in modeling.

uncertainty. In such conditions, the process of measuring the process variables, identification, and finally automated control is more complicated. Partly, these uncertainties can be reduced by the more detailed geological survey. But even this does not guarantee the elimination of such uncertainties, as evidenced by long-term experience in the traditional coal mining technology.

Predictive methods were evaluated using statistical approaches and calculating the performance index. The following indicators were used to compare the performance of the individual prediction methods. Variable  $y$  represents the measured target, and  $Y$  represents its prediction.  $\bar{y}$  is the average of target values  $y_i$  and  $\bar{Y}$  is the average of predicted values  $Y_i$  ( $i = 1, \dots, N$ ).  $N$  represents the number of patterns in the training or testing set.

- *Coefficient of correlation ( $r_{yY}$ )* - The coefficient expresses the force of the linear relationship between two variables. Determines the degree of dependence of two variables and acquires the value from the interval (-1; 1). Its definition is based on the consideration of the sum of deviations of individual values of two correlated characters and their averages. Several equations are used to calculate the correlation coefficient, but the following are used in this work:

$$r_{yY} = \frac{\sum_{i=1}^N (Y_i - \bar{Y})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (Y_i - \bar{Y})^2} \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}} \quad (20)$$

If  $r_{yY} = 1$ , the dependence is completely direct;  $r_{yY} = -1$  i.e., the correlation is completely indirect; if  $r_{yY} = 0$ , between the variables is independence. More precisely:  $r_{yY} < 0.3$  - low tightness;  $0.3 \leq r_{yY} < 0.5$  - slight tightness;  $0.5 \leq r_{yY} < 0.7$  - significant tightness;  $0.7 \leq r_{yY} < 0.9$  - high tightness;  $0.9 \leq r_{yY}$  - very high tightness.

- *Coefficient of determination ( $r_{yY}^2$ )* - It expresses the degree of the causal dependence of two variables. It

is a statistic that will give some information about the goodness of the fit of a model. The correlation coefficient is the square root of the determination coefficient. Degrees of tightness depending on the coefficient of determination are as follows:  $r_{yY}^2 < 0.1$  - low tightness;  $0.1 \leq r_{yY}^2 < 0.25$  - slight tightness;  $0.25 \leq r_{yY}^2 < 0.50$  - significant tightness;  $0.5 \leq r_{yY}^2 < 0.80$  - high tightness;  $0.8 \leq r_{yY}^2$  - very high tightness.  $r_{yY}^2 = 1$  indicates that the model perfectly fits the measured target data.

$$r_{yY}^2 = 1 - \frac{\frac{1}{N} \sum_{i=1}^N (Y_i - \bar{Y})^2}{\frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2} \quad (21)$$

- *Relative root mean squared error (RRMSE)* - This error can be calculated by dividing root mean square error RMSE by the average of actual values  $y_i$ . RMSE represents the square root of mean square error (MSE) calculated as follows:

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{N} \sum_{i=1}^n (Y_i - y_i)^2} \quad (22)$$

The MSE is a useful statistic measure for assessing the accuracy of the prediction. The RRMSE can be calculated by the following equation [34, 47]:

$$\text{RRMSE} = \frac{\text{RMSE}}{\frac{1}{N} \sum_{i=1}^N y_i} \times 100 = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (Y_i - y_i)^2}}{\frac{1}{N} \sum_{i=1}^N y_i} \times 100 \quad (\%) \quad (23)$$

- *Mean absolute percentage error (MAPE)* - This statistic indicator expresses a percentage prediction error. It can be calculated as follows:

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - Y_i|}{|y_i|} \times 100 \quad (\%) \quad (24)$$

This error has certain disadvantages. At zero values of  $y_i$ , a division by zero can occur, and MAPE



produces undefined values. In very low values of  $y_i$ , MAPE can exceed 100 % extremely. When actual values  $y_i$  are very high (i.e., above  $Y_i$ ), MAPE will not exceed 100 %.

- *Performance index* (PI) - indicates the overall performance of the given prediction method. The PI value ranges is from 0 to  $+\infty$ . The smaller values of PI indicate a better performance. The PI is calculated as follows [47, 48]:

$$\text{PI} = \frac{\text{RRMSE}}{r_{yY} + 1} \quad (25)$$

The above statistical indicators were calculated individually for training and testing data. The time needed to train the predictive model was also measured.

#### 4.1. PREDICTION BY THE BACK-PROPAGATION NN

The back-propagation algorithm (or gradient algorithm) is based on an “error-correction” learning rule, i.e., learning by the network error. This algorithm performs the steepest descent procedure. The learning consists of two pass-overs across different layers of the network, i.e., the forward pass (i.e., forward activation flow of outputs) and backward pass (i.e., the backward error propagation of weight adjustments).

Before the training, it is appropriate to standardize all inputs of the NN for the effective scaling of the weights.

The learning takes place in cycles (i.e., epochs), always with new input patterns. It is based on the gradual minimizing weights so that the error  $E$  (5) is reduced. This error is minimized iteratively in different epochs so that the required accuracy  $\varepsilon$  was achieved. After learning, it is desirable that the output from the NN network is equal to the required output or to get it as close as possible, considering all input patterns from the set of  $A_{train}$ .

The ability of the NN to determine the output for inputs outside the  $A_{train}$  set is called the generalization. This is also the main role of the NN in the prediction. Weights are modified using the set  $A_{train}$  and, using  $A_{test}$ , the generalization error is detected.

Five input variables i.e., concentration of  $O_2$  ( $x_1$ ),  $CO_2$  ( $x_2$ ),  $CO$  ( $x_3$ ),  $H_2$  ( $x_4$ ) and  $CH_4$  ( $x_5$ ) (measured in vol. %) have been regarded for the initialization of input neurons and the prediction of the underground temperature in the UCG. Two stationary analysers that measured concentrations of only these five gases have been used during the UCG experiment. These concentrations have been considered to be the most significant.

In the prediction of the syngas calorific value, three input process-relevant variables were used i.e., injected air flow ( $x_1$ ), flow of supplementary oxygen ( $Nm^3/h$ ) ( $x_2$ ) and pressure on outlet (Pa) ( $x_3$ ). These variables are adjustable by the automatized control system.

The general scheme of the NN considered from the UCG data prediction is shown in Figure 7. The methods of Batch Gradient Descent with Momentum and Gradient Descent with Variable Learning Rate have been applied. In this approach, weights and biases are updated according to the gradient descent momentum and an adaptive learning rate.

It is the most widely used way to realize this minimization of error (5)) within gradient optimization methods, in which weighting and threshold factors are recurrently updated according to following equations [27]:

$$\begin{aligned} w_{ij}^{(k+1)} &= w_{ij}^{(k)} - \lambda \frac{\partial E}{\partial w_{ij}} + \mu \Delta w_{ij}^{(k)} \\ \vartheta_j^{(k+1)} &= \vartheta_j^{(k)} - \lambda \frac{\partial E}{\partial \vartheta_j} + \mu \Delta \vartheta_j^{(k)} \end{aligned} \quad (26)$$

where parameter  $\lambda > 0$  represents the learning rate and must be small enough to ensure the monotone convergence of the optimization algorithm and, at the same time, large enough to provide a sufficiently high convergence rate.

Calculating the partial derivations  $\frac{\partial E}{\partial w_{ij}}$  and  $\frac{\partial E}{\partial \vartheta_j}$  for the entire NN, running recurrently from the highest to the lowest layer, i.e., against the direction of dissemination of information in the NN, runs from the lowest to the highest layer.

Initial values of the threshold and weighting coefficients  $\vartheta_j^{(0)}$  and  $w_{ij}^{(0)}$  are randomly generated from a small center-to-zero interval e.g., from an open interval  $(-1, 1)$ . The last member  $\mu$  in (26) represents the so-called moment member that is determined by the difference of the coefficients from the last two iterations,  $\Delta w_{ij}^{(k)} = w_{ij}^{(k)} - w_{ij}^{(k-1)}$  and  $\Delta \vartheta_j^{(k)} = \vartheta_j^{(k)} - \vartheta_j^{(k-1)}$ . The momentum is important for the “skip” of the local minima in the initial optimization phase. The value of the parameter  $\mu$  is usually chosen from the interval  $0.5 \leq \mu \leq 0.7$ .

The adaptive learning rate tries to maintain a stable learning and a largest size of the learning step. The mean squared error (MSE) was used as the function for the error calculating during the training.

The number of hidden layers and neurons is usually determined on the basis of experimentation, where an NN model is selected for which  $E_{test}$  is minimal. However, a small number of hidden layers in the NN may not well model non-linearities in the data. It is, therefore, necessary to look for an optimal number of hidden neurons.

Two variants of the NN, i.e., with one and two hidden layers, were used, but the number of neurons was estimated in previous experimentation. It has also been tried to set the number of neurons in the hidden layer to  $2m + 1$ , where  $m$  is the number of input neurons. In all variants that were tried only one neuron in the output layer was used. The momentum constant was set-up to 0.9. Within the results, the goal was also to show what impact has the number of hidden neurons on the quality of prediction. The results of training and testing are shown in Table 3.

Predicted variable	Hidden layers	Number of neurons (L1:L2)	Observations (inputs)	Training										Testing									
				$r_p \gamma$	$r_p^2 \gamma$	RRMSE (%)	PI	MAPE (%)	MSE	RMSE	Time (s)	$r_p \gamma$	$r_p^2 \gamma$	RRMSE (%)	PI	MAPE (%)	MSE	RMSE					
Temperature	2	5000:15	CO, CO <sub>2</sub>	0.2707	0.0733	8.0062	6.3008	6.4108	6484.2537	80.5249	44.8312	0.1985	0.0394	8.6818	7.2440	7.2707	6938.2153	83.2960					
	2	5000:15	CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.0567	0.0032	11.7869	11.1542	9.1223	14054.2843	118.5508	50.8077	0.0611	0.0037	18.2237	17.1749	10.8640	30570.4682	174.8441					
	2	800:8	CO, CO <sub>2</sub>	0.2043	0.0417	8.0649	6.6967	6.6452	6579.6878	81.1153	6.8560	0.6755	0.4563	8.5075	5.0775	7.4526	6662.3759	81.6234					
	2	800:8	CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.1820	0.0331	8.8428	7.4810	6.9526	7910.2629	88.9397	7.1921	0.0810	0.0066	9.9893	9.2407	6.8042	9185.3462	95.8402					
	1	5000	CO, CO <sub>2</sub>	0.0557	0.0034	69.5635	65.7078	53.1500	489523.6382	699.6597	38.4061	0.1021	0.0104	95.0738	105.8881	73.4974	832048.0252	912.1667					
	1	5000	CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.1865	0.0348	42.8462	36.1105	31.0099	185709.9328	430.9408	43.1599	0.1611	0.0260	70.0982	83.5614	48.0393	452313.8233	672.5428					
	1	800	CO, CO <sub>2</sub>	0.0428	0.0018	23.6400	22.6700	17.8908	56533.4172	237.7676	1.7451	0.1623	0.0263	48.7514	41.9446	35.9242	218776.5470	467.7356					
	1	800	CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.0778	0.0061	16.0174	14.8608	11.9061	25953.3424	161.1004	1.8475	0.6381	0.4072	28.3447	17.3032	22.0818	73955.5871	271.9478					
	1	5	CO, CO <sub>2</sub>	0.1495	0.0223	8.1083	7.0538	6.7466	6650.7120	81.3519	0.6702	0.3373	0.1138	8.4245	6.2996	7.1668	6533.0962	80.8276					
	1	11	CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.4210	0.1772	7.4552	5.2466	6.1409	5622.4662	74.9831	0.7461	0.6787	0.4606	7.4301	4.4261	5.8129	5081.7732	71.2866					
	Calorific value	2	5000:15	Air, O <sub>2</sub>	0.5588	0.3122	33.3750	21.4108	37.6195	9.6441	3.1055	55.2753	0.0366	0.0013	30.4626	29.3861	30.6660	12.4353	3.5264				
2		5000:15	Air, O <sub>2</sub> , Outlet pressure	0.1311	0.0172	119.9483	106.0431	126.1949	124.5076	11.1610	73.1432	0.0270	0.0007	86.6737	84.3988	70.8208	100.6693	10.0334					
2		800:8	Air, O <sub>2</sub>	0.7383	0.5451	22.6638	13.0378	28.3859	4.4472	2.1088	7.1023	0.0999	0.0100	24.8010	22.5486	22.6788	8.2425	2.8710					
2		800:8	Air, O <sub>2</sub> , Outlet pressure	0.7392	0.5464	22.6569	13.0271	28.0108	4.4444	2.1082	7.5802	0.6906	0.4769	21.5027	12.7190	19.5456	4.5877	2.1419					
1		5000	Air, O <sub>2</sub>	0.0398	0.0016	159.1270	153.0306	183.3015	219.2321	14.8065	44.2146	0.1474	0.0217	173.3974	151.1207	148.0262	402.9098	20.0726					
1		5000	Air, O <sub>2</sub> , Outlet pressure	0.2339	0.0547	133.3327	108.0600	135.6404	153.9182	12.4064	44.6760	0.4976	0.2476	139.3664	93.0571	113.2722	260.2789	16.1332					
1		800	Air, O <sub>2</sub>	0.4106	0.1686	55.1188	39.0761	61.2308	26.3036	5.1287	7.7348	0.1819	0.0331	87.2729	73.8429	75.1442	102.0661	10.1028					
1		800	Air, O <sub>2</sub> , Outlet pressure	0.4590	0.2107	51.0458	34.9874	51.7013	22.5599	4.7497	7.9496	0.6693	0.4479	57.1718	34.2496	47.0393	43.8013	6.6183					
1		5	Air, O <sub>2</sub>	0.6714	0.4507	24.9076	14.9025	32.9435	5.3713	2.3176	0.9649	0.4011	0.0016	19.8082	19.0445	22.9766	5.2579	2.2930					
1		7	Air, O <sub>2</sub> , Outlet pressure	0.7281	0.5301	23.0707	13.3503	28.6974	4.6083	2.1467	1.0306	0.7187	0.5166	15.8219	9.2055	16.8804	3.3546	1.8316					

TABLE 3. Results of simulations with NNs where 10 % of the experiment was used to test.

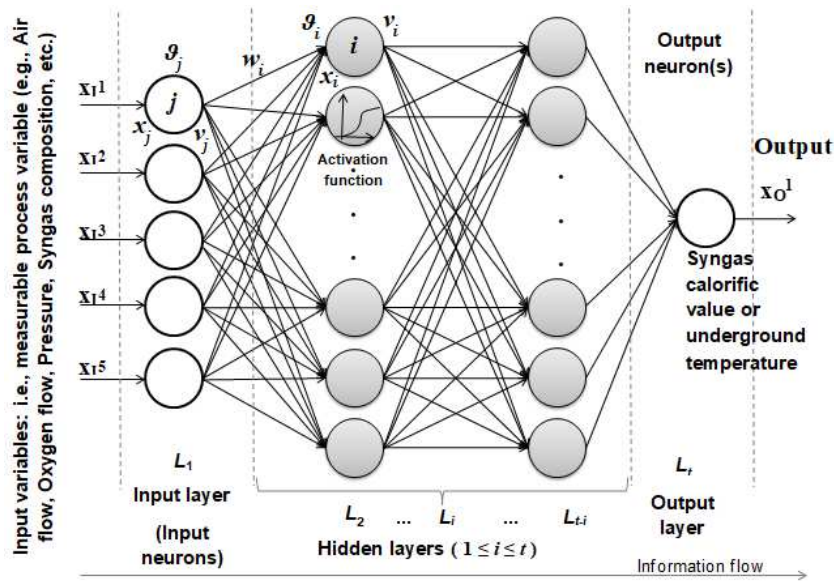


FIGURE 7. Proposal of Neural network considered for UCG data prediction.

When training and testing the prediction of underground temperature, it can be seen that the lowest values of statistic indicators (i.e., RRMSE, MSE, MAPE, and RMSE) were obtained in the case of one hidden layer with 11 neurons (i.e.,  $2m + 1$ , according to a general recommendation, where  $m$  is the number of input neurons). In this case, the RRMSE and performance index had the lowest values (PI = 4.42 for testing and PI = 5.24 for training). The value of the coefficient of determination was the highest in this case ( $r_{yY}^2 = 0.46$  for testing and  $r_{yY}^2 = 0.17$  for training). This result was obtained when five input observations were used for training the NN model and for the testing of prediction. This means that this variant predicts the target the best. The second interesting result was obtained in the case of using an NN with two hidden layers and the structure of 800:8 neurons. It was the case when the two observation inputs were used. The worst results in the prediction of the underground temperature were achieved in the case of one hidden layer with 5000 neurons. When training and testing the NN for the calorific value, it can be seen that the lowest values of statistic indicators were also obtained in the case of one hidden layer with 7 neurons (i.e.,  $2m + 1$ ). In this case, the RRMSE and performance index was the lowest (e.g., PI = 9.20 for testing). This result was obtained when three input variables were used for training the NN model and for the testing of prediction. The value of the coefficient of determination was the highest in this case (i.e.,  $r_{yY}^2 = 0.51$  for testing). Similarly, as in the case of the calorific value prediction, the second interesting result was obtained in the case of using an NN with two hidden layers and the structure of 800:8 neurons. This is the case where three observation inputs were used. The worst results in the prediction of the calorific value were achieved

in the case of one hidden layer with 5000 neurons (see Table 3).

It can be stated that the use of the NN model for the temperature prediction has achieved better results in terms of the performance index than in the case of the calorific value. The best prediction of the calorific value and underground temperature by NN, where 10 % of the experiment was used for the test is shown in Figure 8 and Figure 9. The black vertical line in figures divides the prediction into training and testing.

#### 4.2. PREDICTION BY THE MARS

The algorithm of a regression model creation with MARS runs in two phases as it was analysed in detail in Section 2.2. In the forward phase, the algorithm begins with a model that has only an intercept term. Then, in the cycle, reflected pairs of BFs are added so that the training error is reduced as much as possible. This is done until, for example, the maximum number of BFs is not reached. In the backward phase, the model is simplified by deleting one of the least important BFs, which also reduces the training error. Then, more “best” models of different sizes are obtained. At the end of this phase, only one model with the lowest GCV is selected from these best models (excluding models larger than maximal final BFs).

Several different variants of simulations with MARS have been performed. The maximum number of BFs included in the model in the forward phase has been experimentally changed.

The initial number of BFs in the forward phase was determined according to the formula:  $\min(200, \max(20, 2d)) + 1$ , where  $d$  represents the number of input variables.

The initial number of BFs was set to 21 in all simulations. In modelling, we have considered a maximal interactivity between input variables without

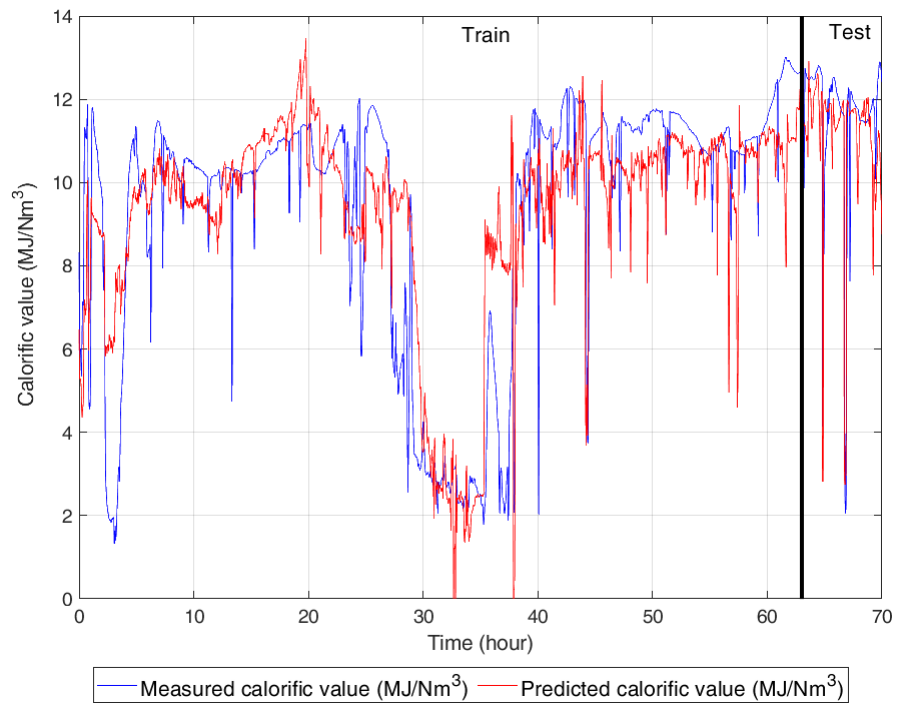


FIGURE 8. Measured and predicted calorific value of syngas by NN, where three inputs were used in the test.

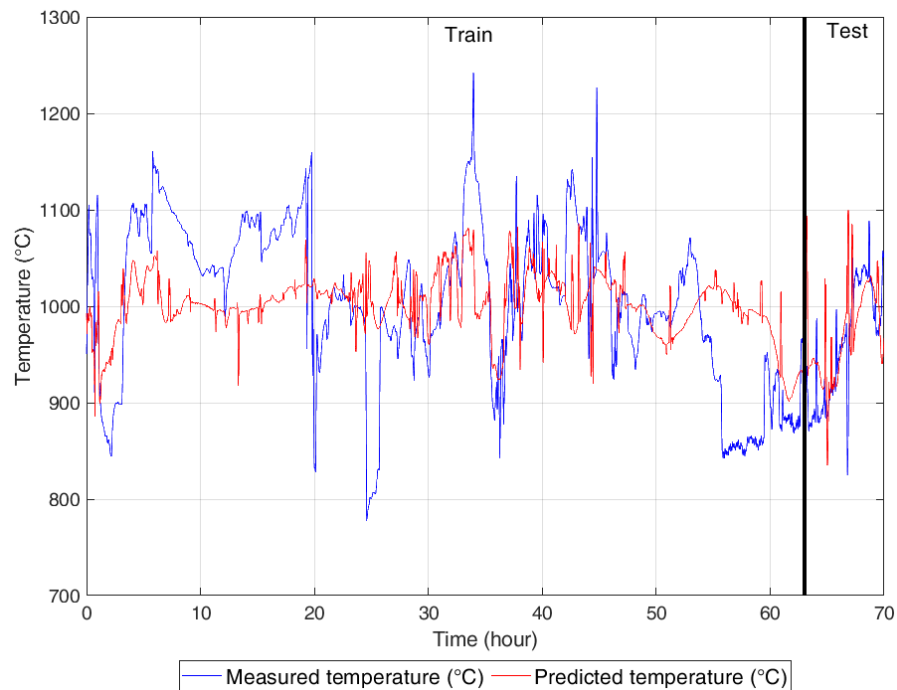


FIGURE 9. Measured and predicted underground temperature by NN, where five inputs were used in the test.

self-interactions. Since there were not smoother data, the piecewise-cubic and piecewise-linear type of modeling have been analysed in order to know the prediction performance. In default, all MARS models are created as piecewise-linear and are transformed into piecewise-cubic after the backward phase.

The best or optimal number of maximal BFs in the final MARS model was estimated by the GCV criterion and by the 10-fold Cross-Validation. In an all Cross-Validation iteration, a new MARS model is created and reduced using the GCV in the in-fold (training) data. In addition, there also is a calculated MSE criterion on out-of-fold (test) data ( $MSE_{\text{oof}}$ ) in the reducing phase for each model.

Figure 10 shows a comparison of the behavior of the GCV and  $MSE_{\text{oof}}$  criterion calculated for each new model after the 10-fold iteration. In the simulation, the model for predicting syngas calorific value was considered. The figure shows two vertical dashed lines at the minimum of the two solid lines. The figure also shows the number of optimum BFs estimated by the GCV (cyan) and Cross-Validation (magenta). Ideally, these two lines would coincide. Similarly, Figure 11 shows the determination of the optimal number of basis functions for the MARS model for the underground temperature prediction. Figures show simulations of the piecewise-linear type of MARS models with all inputs considered in a given target variable. The results from the training of the model and testing the prediction are shown in Table 4.

The original MARS approximation method uses a cubic function to smooth the truncated piecewise-linear functions. The cubic function has the following general form:

$$C(x|s = +1, t_-, t, t_+) = \begin{cases} 0, & x \leq t_- \\ \alpha_+(x - t_-)^2 + \beta_+(x - t_-)^3, & t_- < x < t \\ x - t, & x \geq t_+ \end{cases} \quad (27)$$

where

$$\alpha_+ = \frac{2t_+ - 3t + t_-}{(t_+ - t_-)^2}, \quad \beta_+ = \frac{-t_+ + 2t - t_-}{(t_+ - t_-)^3}, \quad (28)$$

and

$$C(x|s = -1, t_-, t, t_+) = \begin{cases} t - x, & x \leq t_- \\ \alpha_-(x - t_+)^2 + \beta_-(x - t_+)^3, & t_- < x < t_+ \\ 0, & x \geq t_+ \end{cases} \quad (29)$$

where

$$\alpha_- = \frac{-t_+ + 3t - 2t_-}{(t_- - t_+)^2}, \quad \beta_- = \frac{t_+ - 2t + t_-}{(t_- - t_+)^3}. \quad (30)$$

where  $t$  represents a univariate knot, which is selected for each of the factor variables  $x$ .

The piecewise-linear type of MARS models better fits the training data but, in the prediction on untrained data, better results with the piecewise-cubic type of the model (see Table 4) are obtained.

Equation (31) represents a resulting piecewise-cubic type of MARS model for the prediction of the underground temperature. Basis functions in the equation are calculated according to Table 5. This MARS model takes into account five inputs, i.e., concentration of measured gasses:  $x_1$  - CO,  $x_2$  - CO<sub>2</sub>,  $x_3$  - H<sub>2</sub>,  $x_4$  - CH<sub>4</sub>,  $x_5$  - O<sub>2</sub>. Similarly, equation (32) represents the piecewise-cubic type of MARS model for the prediction of the calorific value of the syngas. Basis functions in the equation are calculated according to Table 6. The MARS model, in this case, has three inputs  $x_1$  - air flow and  $x_2$  - oxygen flow and  $x_3$  - controlled pressure on outlet. In these models, the best results for the prediction were obtained in terms of all statistical indicators. The lowest performance index  $PI = 5.66$  was obtained in the testing of the prediction of the underground temperature on untrained data with the piecewise-cubic type of the MARS model. When testing the prediction of the calorific value on untrained data with piecewise-cubic type of MARS model, the performance index  $PI = 12.1382$  was obtained.

$$\begin{aligned} \text{Temperature } (^{\circ}\text{C}) &= 855.36 \\ &- 2108.8 \times \text{BF1} \\ &+ 8.7366 \times \text{BF2} + 14.266 \times \text{BF3} \\ &+ 20.014 \times \text{BF4} - 13.465 \times \text{BF5} \\ &- 18.233 \times \text{BF6} + 1.2743 \times \text{BF7} \\ &- 3.764 \times \text{BF8} - 1.3594 \times \text{BF9} \\ &- 6.7334 \times \text{BF10} + 1.1376 \times \text{BF11} \\ &+ 0.61503 \times \text{BF12} + 0.79404 \times \text{BF13} \\ &- 1.766 \times \text{BF14} + 0.20451 \times \text{BF15} \\ &- 1.8976 \times \text{BF16} - 5.8421 \times \text{BF17} \\ &+ 0.72146 \times \text{BF18} + 0.86513 \times \text{BF19} \end{aligned} \quad (31)$$

$$\begin{aligned} \text{Calorificvalue (MJ/Nm}^3) &= 13.528 \\ &- 0.0044217 \times \text{BF1} - 0.13132 \times \text{BF2} \\ &- 0.14469 \times \text{BF3} - 1.1482 \times \text{BF4} \\ &+ 0.015322 \times \text{BF5} - 0.0034767 \times \text{BF6} \\ &- 0.019788 \times \text{BF7} + 0.11264 \times \text{BF8} \\ &+ 0.55397 \times \text{BF9} - 0.21132 \times \text{BF10} \\ &- 0.84369 \times \text{BF11} + 0.032498 \times \text{BF12} \\ &- 0.16806 \times \text{BF13} + 0.12228 \times \text{BF14} \\ &+ 0.084482 \times \text{BF15} + 0.00096036 \times \text{BF16} \\ &+ 0.00075854 \times \text{BF17} \end{aligned} \quad (32)$$

Due to the fact that the piecewise-linear type of the model gives, during the training, better results in terms of all indicators, these variants of the model are also presented. These are models with all considered inputs for the prediction of the given target.

Piecewise-linear type of MARS model usage  $\max(0, x-t)$  function, where  $t$  is the knot. The  $\max()$  function represents positive part of  $(0, x-t)$  which can be formally expressed as the following:

$$\max(0, x - t) = \begin{cases} x - t, & \text{if } x \geq t \\ 0, & \text{otherwise} \end{cases} \quad (33)$$



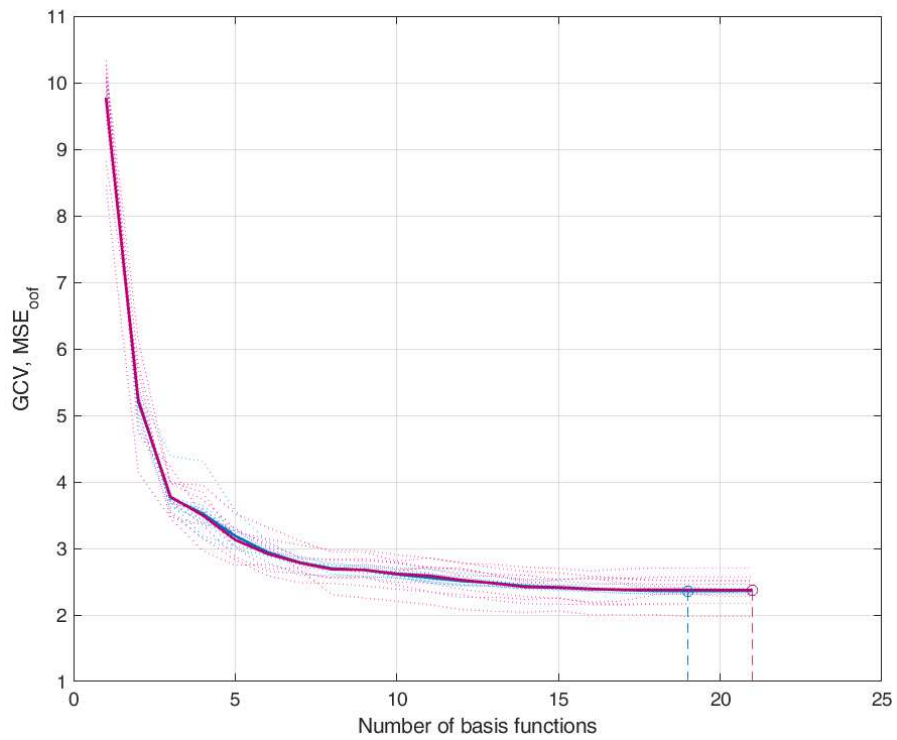


FIGURE 10. Example of the estimation of the “best” number of BFs of the calorific value MARS model with three inputs by GCV and 10-fold Cross-Validation (i.e.,  $MSE_{oof}$ ).

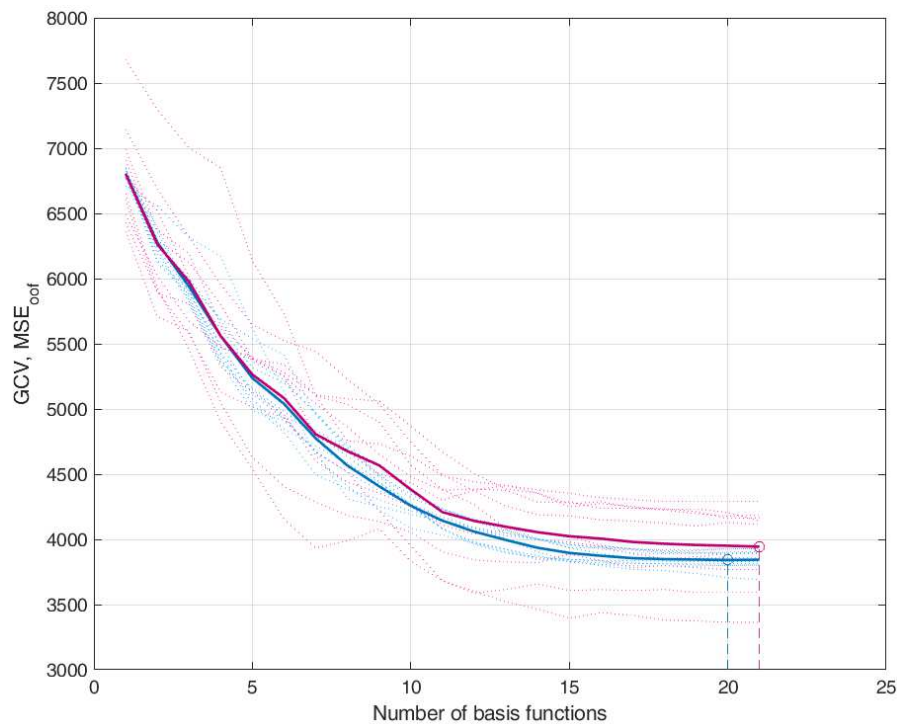


FIGURE 11. Example of estimation of the “best” number of BFs in MARS model of underground temperature with five input by GCV and 10-fold Cross-Validation (i.e.,  $MSE_{oof}$ ).

Predicted variable	Type of MARS model	Number of BFs in final model including BF0	Observations (inputs)	Training						Testing								
				$r_{py}$	$r_{py}^2$	RRMSE (%)	PI	MAPE (%)	MSE	RMSE	Time (s)	$r_{ty}$	$r_{ty}^2$	RRMSE (%)	PI	MAPE (%)	MSE	RMSE
Temperature	piecewise-cubic	16	CO, CO <sub>2</sub>	0.4707	0.2216	7.2342	4.9189	5.7432	5294.1481	72.7609	10.5619	0.2312	0.0534	10.5865	8.5986	8.8779	10316.4417	101.5699
	piecewise-linear	20	CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.5859	0.3433	6.6445	4.1897	5.2014	4466.1544	66.8293	17.8538	0.6031	0.3637	6.4275	4.0094	5.1623	3802.8747	61.6675
	piecewise-linear	16	CO, CO <sub>2</sub>	0.4985	0.2485	7.1080	4.7434	5.6262	5110.9554	71.4909	12.1525	0.3282	0.1077	10.6824	8.0427	9.1629	10504.3010	102.4905
Calorific value	piecewise-cubic	20	CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.6666	0.4444	6.1119	3.6673	4.7285	3778.9200	61.4729	17.8482	0.5788	0.3350	8.9462	5.6665	7.5283	7367.2269	85.8326
	piecewise-linear	15	Air, O <sub>2</sub>	0.7863	0.6183	20.7582	11.6208	24.0946	3.7307	1.9315	8.0520	0.0419	0.0018	83.9179	80.5431	32.5398	94.3695	9.7144
	piecewise-linear	18	Air, O <sub>2</sub> , Outlet pressure	0.8661	0.7501	16.7964	9.0010	17.3057	2.4426	1.5629	13.5414	0.4386	0.1924	17.4620	12.1382	17.8493	4.0861	2.0214
		16	Air, O <sub>2</sub>	0.7991	0.6386	20.1965	11.2256	23.0084	3.5316	1.8792	9.1635	0.0419	0.0018	80.7365	77.4897	30.3193	87.3500	9.3461
		17	Air, O <sub>2</sub> , Outlet pressure	0.8730	0.7621	16.3866	8.7489	16.5883	2.3249	1.5247	13.4410	0.4238	0.1796	17.8945	12.5681	18.3341	4.2910	2.0715

TABLE 4. Results of simulations with MARS models where 10 % of the experiment was used to test.

BF	Equation	BF	Equation
BF1	$C(x_4   +1, 20.571, 20.579, 20.607)$	BF11	$C(x_1   +1, 3.6356, 6.3881, 7.6506) \times C(x_2   -1, 19.073, 26.51, 28.665)$
BF2	$C(x_4   -1, 20.571, 20.579, 20.607)$	BF12	$BF8 \times C(x_1   +1, 9.2894, 9.6658, 14.265)$
BF3	$C(x_2   +1, 7.967, 11.636, 19.073)$	BF13	$BF8 \times C(x_1   -1, 9.2894, 9.6658, 14.265)$
BF4	$C(x_2   -1, 7.967, 11.636, 19.073)$	BF14	$BF9 \times C(x_1   +1, 7.6506, 8.9131, 9.2894)$
BF5	$C(x_1   -1, 3.6356, 6.3881, 7.6506)$	BF15	$BF9 \times C(x_1   -1, 7.6506, 8.9131, 9.2894)$
BF6	$C(x_3   +1, 15.747, 26.344, 30.018)$	BF16	$BF5 \times C(x_3   +1, 2.5749, 5.1498, 15.747)$
BF7	$C(x_3   -1, 15.747, 26.344, 30.018)$	BF17	$BF5 \times C(x_3   -1, 2.5749, 5.1498, 15.747)$
BF8	$BF3 \times C(x_4   +1, 11.785, 20.563, 20.571)$	BF18	$BF14 \times C(x_5   +1, 2.1322, 3.9609, 12.268)$
BF9	$BF3 \times C(x_4   -1, 11.785, 20.563, 20.571)$	BF19	$BF14 \times C(x_5   -1, 2.1322, 3.9609, 12.268)$
BF10	$C(x_1   +1, 3.6356, 6.3881, 7.6506) \times C(x_2   +1, 19.073, 26.51, 28.665)$		

TABLE 5. Basis functions of piecewise-cubic type of MARS model of underground temperature (five inputs).

BF	Equation	BF	Equation
BF1	$C(x_1   +1, 5.866, 11.732, 17.503) \times C(x_3   +1, 8.357, 16.442, 32.428)$	BF10	$C(x_1   -1, 5.866, 11.732, 17.503) \times C(x_2   +1, 0.239, 0.478, 2.505)$
BF2	$C(x_1   +1, 5.866, 11.732, 17.503) \times C(x_3   -1, 8.357, 16.442, 32.428)$	BF11	$C(x_1   -1, 5.866, 11.732, 17.503) \times C(x_2   -1, 0.239, 0.478, 2.505)$
BF3	$C(x_2   +1, 7.763, 9.83, 25.869)$	BF12	$C(x_1   +1, 17.503, 23.274, 31.139) \times C(x_2   +1, 2.505, 4.532, 5.114)$
BF4	$C(x_2   -1, 7.763, 9.83, 25.869)$	BF13	$C(x_1   +1, 17.503, 23.274, 31.139) \times C(x_2   -1, 2.505, 4.532, 5.114)$
BF5	$BF4 \times C(x_3   +1, 32.428, 48.414, 74.478)$	BF14	$BF11 \times C(x_3   +1, -0.155, 0.272, 8.357)$
BF6	$BF4 \times C(x_3   -1, 32.428, 48.414, 74.478)$	BF15	$BF11 \times C(x_3   -1, -0.155, 0.272, 8.357)$
BF7	$C(x_1   +1, 5.866, 11.732, 17.503) \times C(x_2   +1, 5.114, 5.696, 7.763)$	BF16	$BF5 \times C(x_1   +1, 31.139, 39.004, 48.437)$
BF8	$C(x_1   +1, 5.866, 11.732, 17.503) \times C(x_2   -1, 5.114, 5.696, 7.763)$	BF17	$BF5 \times C(x_1   -1, 31.139, 39.004, 48.437)$
BF9	$C(x_1   -1, 17.503, 23.274, 31.139)$		

TABLE 6. Basis functions of piecewise-cubic type of MARS model of syngas calorific value (three inputs).

$$\begin{aligned}
 &\text{Temperature } (^{\circ}\text{C}) = 851.31 \\
 &+3036.4 \times \text{BF1} + 11.023 \times \text{BF2} \\
 &+21.043 \times \text{BF3} + 7.8072 \times \text{BF4} \\
 &-7.1445 \times \text{BF5} - 28.58 \times \text{BF6} \\
 &+1.264 \times \text{BF7} - 483.77 \times \text{BF8} \\
 &-2.2149 \times \text{BF9} - 6.5355 \times \text{BF10} \\
 &+0.25511 \times \text{BF11} + 33.098 \times \text{BF12} \\
 &+57.057 \times \text{BF13} - 1.0868 \times \text{BF14} \\
 &+0.2452 \times \text{BF15} - 4.2728 \times \text{BF16} \\
 &-8.3842 \times \text{BF17} + 0.56517 \times \text{BF18} \\
 &+0.74386 \times \text{BF19}
 \end{aligned} \tag{34}$$

$$\begin{aligned}
 &\text{Calorificvalue (MJ/Nm}^3) = 10.969 \\
 &-0.003224 \times \text{BF1} - 0.13121 \times \text{BF2} \\
 &-0.63223 \times \text{BF3} - 0.51773 \times \text{BF4} \\
 &+0.008499 \times \text{BF5} - 0.0075648 \times \text{BF6} \\
 &+0.04656 \times \text{BF7} + 0.050259 \times \text{BF8} \\
 &+0.43584 \times \text{BF9} - 0.13449 \times \text{BF10} \\
 &-1.2256 \times \text{BF11} - 0.043978 \times \text{BF12} \\
 &-0.08991 \times \text{BF13} + 0.22099 \times \text{BF14} \\
 &+1.4463 \times \text{BF15} + 0.00078717 \times \text{BF16} \\
 &+0.0011077 \times \text{BF17}
 \end{aligned} \tag{35}$$

The piecewise-linear type of MARS model for the underground temperature prediction represents equation (34). Corresponding basis functions are shown in Table 7. The performance index obtained during the training of the MARS model was  $PI = 3.66$ . In this case, the coefficient of determination was the highest ( $r_{yY}^2 = 0.44$ ). The piecewise-linear type of the MARS model for the syngas calorific value prediction is represented by the equation (35). Corresponding basis functions are shown in Table 8. The performance index obtained during the training of the MARS model

was  $PI = 8.74$ . In this case, the coefficient of determination was the highest ( $r_{yY}^2 = 0.76$ ).

Due to the comparison with other methods, this paper presents the behaviour of the prediction with only the piecewise-cubic type of MARS models, because the best results on untrained data have been achieved for them.

The best prediction of the calorific value and underground temperature by the piecewise-cubic type of the MARS model where 10 % of the experiment was used for the test of the prediction is shown in Figure 12 and Figure 13. The black vertical line divides the prediction into training and testing part. It can be said that better predictions with the MARS model were achieved in the case of the underground temperature.

The experimental results demonstrate that the piecewise-cubic type of MARS model is better than the piecewise-linear both in the temperature and the calorific value prediction.

### 4.3. PREDICTION BY THE SUPPORT VECTOR REGRESSION

The SVR model has been trained on the predictor data similarly as in the previous method. The predictor data were mapped using three kernel functions, and the SMO method for the objective-function minimization was used. The training data table was used where one row of the table represented one observation and individual columns were predictors  $x$ . The table contains one additional column for the response variable  $y$ . The standardized predictor matrix has been used for the training. The standardization was performed using corresponding weighted means of predictors and weighted standard deviations. Predictors

BF	Equation	BF	Equation
BF1	$\max(0, x_4 - 20.579)$	BF11	$\max(0, x_1 - 6.3881) \times \max(0, 26.51 - x_2)$
BF2	$\max(0, 20.579 - x_4)$	BF12	$\text{BF8} \times \max(0, x_1 - 9.6658)$
BF3	$\max(0, x_2 - 11.636)$	BF13	$\text{BF8} \times \max(0, 9.6658 - x_1)$
BF4	$\max(0, 11.636 - x_2)$	BF14	$\text{BF9} \times \max(0, x_1 - 8.9131)$
BF5	$\max(0, 6.3881 - x_1)$	BF15	$\text{BF9} \times \max(0, 8.9131 - x_1)$
BF6	$\max(0, x_3 - 26.344)$	BF16	$\text{BF5} \times \max(0, x_3 - 5.1498)$
BF7	$\max(0, 26.344 - x_3)$	BF17	$\text{BF5} \times \max(0, 5.1498 - x_3)$
BF8	$\text{BF3} \times \max(0, x_4 - 20.563)$	BF18	$\text{BF14} \times \max(0, x_5 - 3.9609)$
BF9	$\text{BF3} \times \max(0, 20.563 - x_4)$	BF19	$\text{BF14} \times \max(0, 3.9609 - x_5)$
BF10	$\max(0, x_1 - 6.3881) \times \max(0, x_2 - 26.51)$		

TABLE 7. Basis functions of the piecewise-linear type of MARS model of underground temperature (five inputs).

BF	Equation	BF	Equation
BF1	$\max(0, x_1 - 11.732) \times \max(0, x_3 - 16.442)$	BF10	$\max(0, 11.732 - x_1) \times \max(0, x_2 - 0.478)$
BF2	$\max(0, x_1 - 11.732) \times \max(0, 16.442 - x_3)$	BF11	$\max(0, 11.732 - x_1) \times \max(0, 0.478 - x_2)$
BF3	$\max(0, x_2 - 9.83)$	BF12	$\max(0, x_1 - 23.274) \times \max(0, x_2 - 4.532)$
BF4	$\max(0, 9.83 - x_2)$	BF13	$\max(0, x_1 - 23.274) \times \max(0, 4.532 - x_2)$
BF5	$\text{BF4} \times \max(0, x_3 - 48.414)$	BF14	$\text{BF11} \times \max(0, x_3 - 0.272)$
BF6	$\text{BF4} \times \max(0, 48.414 - x_3)$	BF15	$\text{BF11} \times \max(0, 0.272 - x_3)$
BF7	$\max(0, x_1 - 11.732) \times \max(0, x_2 - 5.696)$	BF16	$\text{BF5} \times \max(0, x_1 - 39.004)$
BF8	$\max(0, x_1 - 11.732) \times \max(0, 5.696 - x_2)$	BF17	$\text{BF5} \times \max(0, 39.004 - x_1)$
BF9	$\max(0, 23.274 - x_1)$		

TABLE 8. Basis functions of the piecewise-linear type of MARS model of syngas calorific value (three inputs).

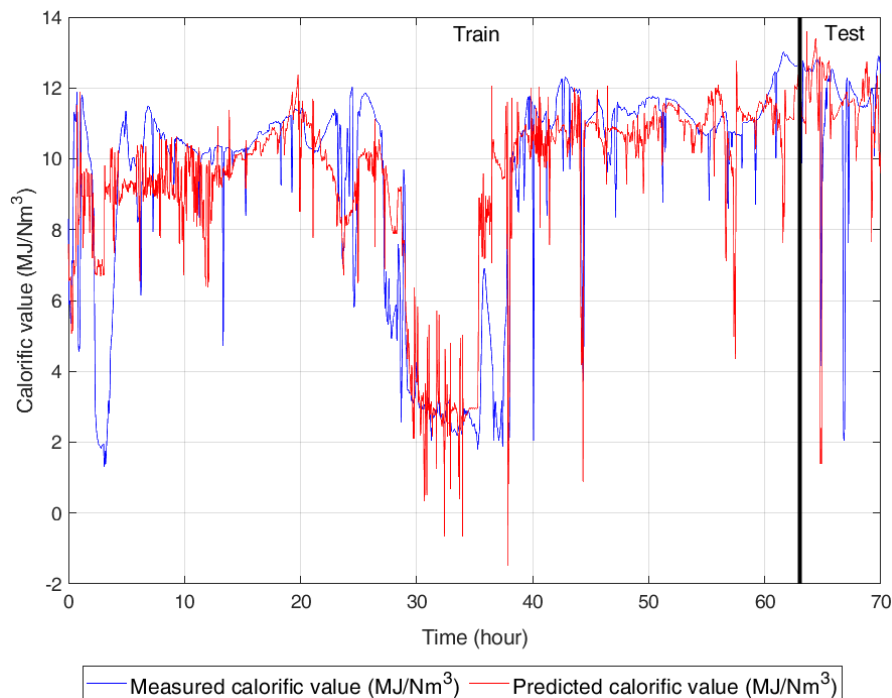


FIGURE 12. Measured and predicted calorific value of syngas by piecewise-cubic type of MARS model where 10 % of the experiment and three inputs were used for the test.

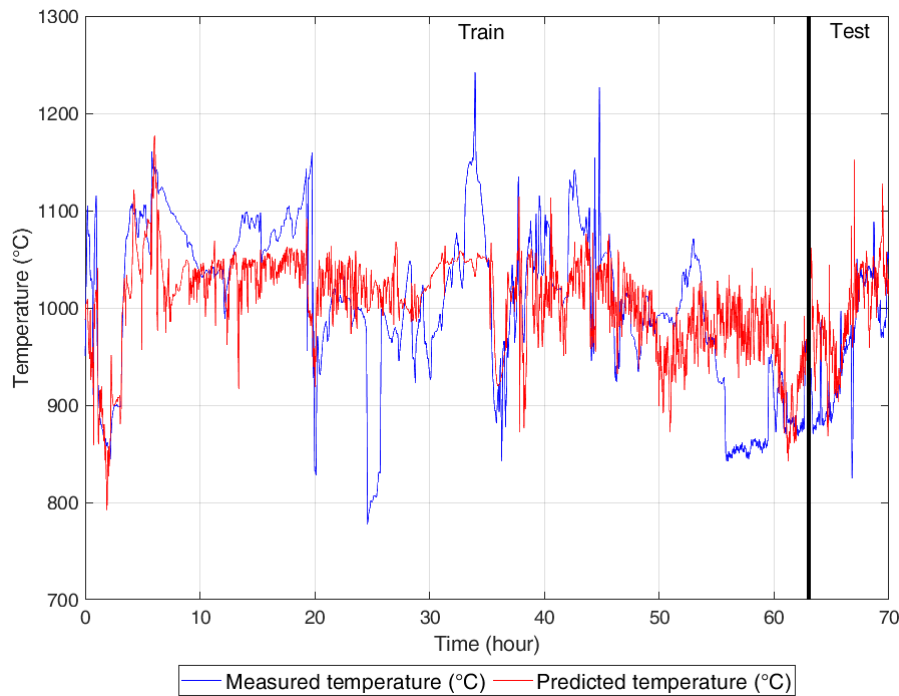


FIGURE 13. Measured and predicted underground temperature by piecewise-cubic type of MARS model where 10 % of the experiment and five inputs were used for the test.

are less sensitive to the scale on which they are measured on when standardization is used. Similarly, a table for the test of the model on untrained data was prepared. As kernel functions, the Linear, Gaussian and Polynomial kernels were used (see Table 1). The final values of  $\alpha$  were stored in the memory of the computer.

Table 9 shows the results of applying a various types of the kernel function in the SVR where 10 % of the UCG experiment was used for the test. The table shows that the best results were obtained in the utilization of Gaussian kernel function, regarding all input variables. This result was obtained for the temperature prediction and also for the syngas calorific value prediction. It is possible to see that the temperature model with five input variables gives the best tightness with the real measured data (see parameter  $r_{yY}^2 = 0.82$  for training and  $r_{yY}^2 = 0.47$  for test). Also, the performance index calculated for the training and testing has reached the lowest value in this case (see parameter  $PI = 1.80$  for training and  $PI = 4.67$  for test). The best performance of prediction is also indicated by other statistical parameters. Similarly, the best result for the SVR model of the syngas calorific value was also obtained when three inputs and Gaussian kernel were used. It is possible to see that the calorific value model with three inputs gives the best tightness with the real measured data (see parameter  $r_{yY}^2 = 0.83$  for training and  $r_{yY}^2 = 0.35$  for test). Also, the performance index calculated during the training and testing has reached the lowest value in this case (see parameter  $PI = 7.20$  for training and  $PI = 8.21$  for test). The best quality of prediction is

also indicated by other statistical parameters. The worst results for prediction on untrained data were obtained using the polynomial kernel, both in the case of the temperature model and also the calorific value model. Figure 14 and Figure 15 show the best prediction of the calorific value and underground temperature by the SVR on untrained data where 10 % of the experiment was used for the test. The black vertical line in figures divides the prediction into the training and testing phase. Even with this method, the results were better for the temperature prediction.

#### 4.4. OVERALL RESULTS

A ranking of the results from three evaluated methods is presented in Table 10 and Table 11. These tables show the comparison of the best results when two variants of observations (i.e., input variables) of each predicted target were used. In the training phase, the model was verified on training data in order to predict the target variable. The results from the training phase of each method are shown in Table 10. It is possible to see that the SVR model with Gaussian kernel better fits the measured target data in the case of modelling the temperature with five observations. The SVR model also achieved a better performance when using two input variables. The other interesting results were obtained with the piecewise-linear type of the MARS model, both in the case of two and five observations. However, MARS models have consumed more time than BPNN and SVR.

When fitting the calorific value, the SVR and Gaussian kernel also reached the best performance. It was the case with five inputs variables. In the case of



Predicted variable	Kernel function	Observations (inputs)	Training						Testing								
			$r_{pY}$	$r_{pY}^2$	RRMSE (%)	PI	MAPE (%)	MSE	RMSE	Time (s)	$r_{pY}$	$r_{pY}^2$	RRMSE (%)	PI	MAPE (%)	MSE	RMSE
Temperature	Linear	CO, CO <sub>2</sub>	0.2794	0.0781	8.2694	6.4635	6.8399	6917.6854	83.1726	0.4813	0.3650	0.1332	8.4538	6.1933	7.2103	6578.6165	81.1087
		CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.2945	0.0867	7.8480	6.0625	6.4595	6230.6605	78.9345	0.4821	0.4754	0.2260	8.4061	5.6975	6.8881	6504.5925	80.6511
	Gaussian	CO, CO <sub>2</sub>	0.5888	0.3466	6.6362	4.1770	4.8036	4455.0438	66.7461	0.6605	0.2053	0.0421	12.4967	10.3683	10.4411	14375.3449	119.8972
		CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.9096	0.8274	3.4414	1.8022	1.9718	1198.0793	34.6133	0.6688	0.6913	0.4779	7.9051	4.6740	6.9661	5752.3317	75.8441
	Polynomial	CO, CO <sub>2</sub>	0.2518	0.0634	8.0087	6.3977	6.2560	6488.3474	80.5503	2.1646	0.1704	0.0290	12.7474	10.8914	10.2197	14957.7685	122.3020
		CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.5531	0.3059	6.9284	4.4609	4.8213	4855.9229	69.6845	3.4230	0.0772	0.0060	16.7335	15.5342	14.1215	25774.9745	160.5459
Calorific value	Linear	Air, O <sub>2</sub>	0.6927	0.4798	24.5832	14.5231	31.2777	5.2323	2.2874	1.5294	0.3222	0.1038	19.2421	14.5531	20.0508	4.9617	2.2275
		Air, O <sub>2</sub> , Outlet pressure	0.7061	0.4986	24.2518	14.2145	31.6478	5.0922	2.2566	1.5539	0.3850	0.1482	16.1073	11.6297	16.1572	3.4767	1.8646
	Gaussian	Air, O <sub>2</sub>	0.7868	0.6190	21.0848	11.8006	22.3990	3.8491	1.9619	0.5178	0.4171	0.1740	18.9023	13.3387	19.4816	4.7880	2.1881
		Air, O <sub>2</sub> , Outlet pressure	0.9130	0.8335	13.7896	7.2085	9.9955	1.6463	1.2831	0.5446	0.5997	0.3596	13.1471	8.2184	11.3808	2.3162	1.5219
	Polynomial	Air, O <sub>2</sub>	0.7042	0.4959	25.9054	15.2012	32.7633	5.8103	2.4105	109.8509	0.3785	0.1433	20.8256	15.1075	23.5170	5.8119	2.4108
		Air, O <sub>2</sub> , Outlet pressure	0.6500	0.4225	35.7762	21.6826	44.4540	11.0817	3.3289	121.1258	0.2066	0.0427	28.6155	23.7158	32.3065	7.9859	3.3126

TABLE 9. Results of simulations with SVR where 10 % of the experiment was used to test.

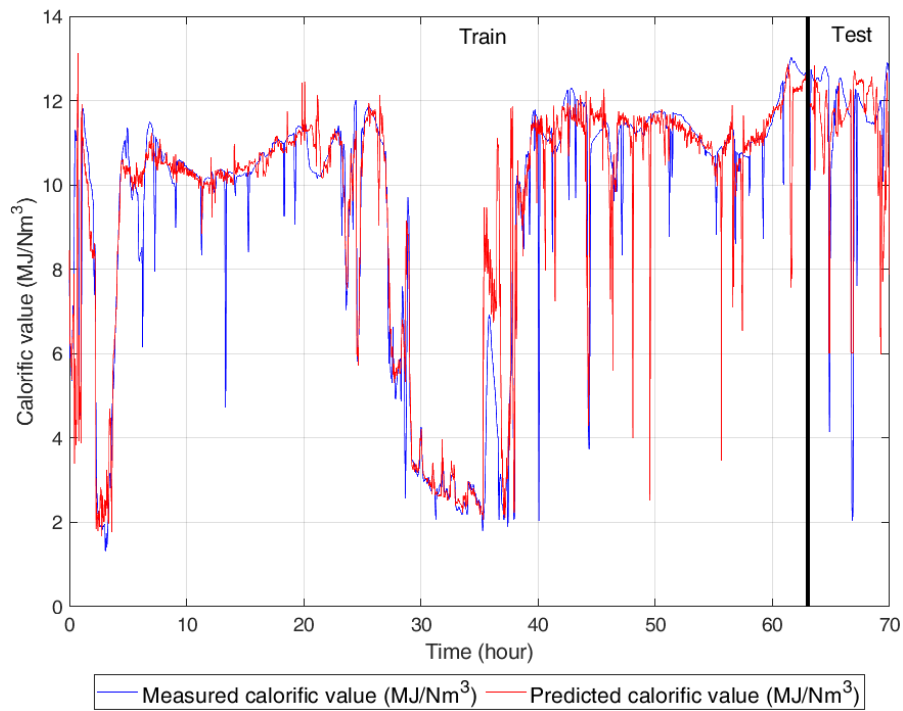


FIGURE 14. Measured and predicted calorific value of syngas by SVR with Gaussian kernel function where 10 % of the experiment and three inputs were used for test.

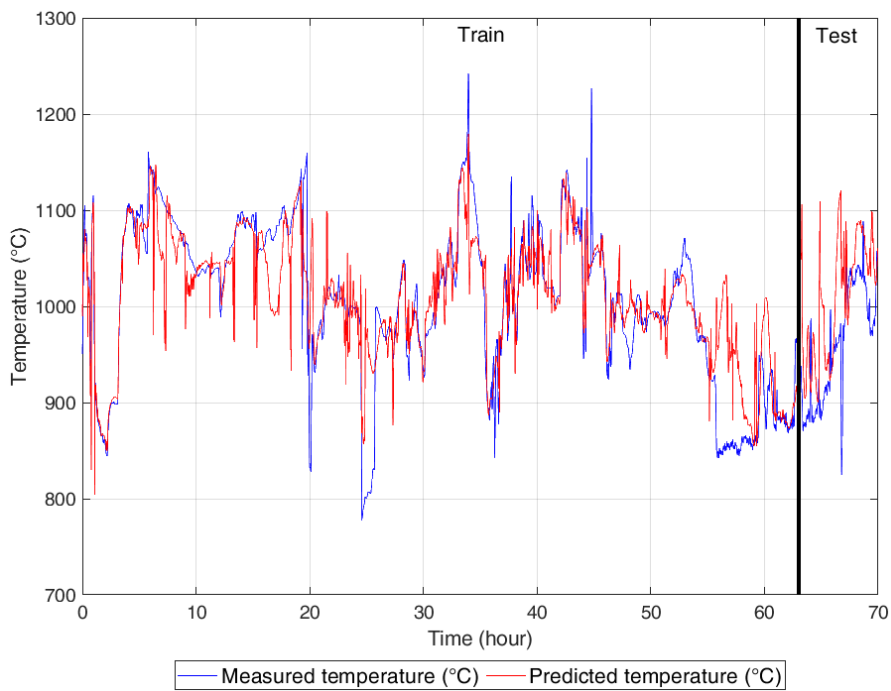


FIGURE 15. Measured and predicted underground temperature by SVR with Gaussian kernel function where 10 % of the experiment and five inputs were used for test.

Predicted variable	Method	Observations	Training									
			$r_{yY}$	$r_{yY}^2$	RRMSE (%)	PI	MAPE (%)	MSE	RMSE	Time (s)		
Temperature	BPNN, Layers: 2, Neurons: (L1:L2) 800:8	CO, CO <sub>2</sub>	0.2043	0.0417	8.0649	6.6967	6.6452	6579.6878	81.1153	6.8560		
	BPNN, Layer: 1, Neurons (L1) 11	CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.4210	0.1772	7.4552	5.2466	6.1409	5622.4662	74.9831	0.7461		
	MARS, piecewise-linear, 16 BFs	CO, CO <sub>2</sub>	0.4985	0.2485	7.1080	4.7434	5.6262	5110.9554	71.4909	12.1525		
	MARS, piecewise-linear, 20 BFs	CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.6666	0.4444	6.1119	3.6673	4.7285	3778.9200	61.4729	17.8482		
	SVR, Gaussian kernel	CO, CO <sub>2</sub>	0.5888	0.3466	6.6362	4.1770	4.8036	4455.0438	66.7461	0.6605		
	SVR, Gaussian kernel	CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.9096	0.8274	3.4414	1.8022	1.9718	1198.0793	34.6133	0.6688		
Calorific value	BPNN, Layers: 2, Neurons: (L1:L2) 800:8	Air, O <sub>2</sub>	0.7383	0.5451	22.6638	13.0378	28.3859	4.4472	2.1088	7.1023		
	BPNN, Layers: 2, Neurons: (L1:L2) 800:8	Air, O <sub>2</sub> , Outlet pressure	0.7392	0.5464	22.6569	13.0271	28.0108	4.4444	2.1082	7.5802		
	MARS, piecewise-linear, 15 BFs	Air, O <sub>2</sub>	0.7991	0.6386	20.1965	11.2256	23.0084	3.5316	1.8792	9.1635		
	MARS, piecewise-linear, 17 BFs	Air, O <sub>2</sub> , Outlet pressure	0.8730	0.7621	16.3866	8.7489	16.5883	2.3249	1.5247	13.4410		
	SVR, Gaussian kernel	Air, O <sub>2</sub>	0.7868	0.6190	21.0848	11.8006	22.3990	3.8491	1.9619	0.5178		
	SVR, Gaussian kernel	Air, O <sub>2</sub> , Outlet pressure	0.9130	0.8335	13.7896	7.2085	9.9955	1.6463	1.2831	0.5446		

TABLE 10. Overall results from the training phase.

Predicted variable	Method	Observations	Testing									
			$r_{yY}$	$r_{yY}^2$	RRMSE (%)	PI	MAPE (%)	MSE	RMSE			
Temperature	BPNN, Layers: 2, Neurons: (L1:L2) 800:8	CO, CO <sub>2</sub>	0.6755	0.4563	8.5075	5.0775	7.4526	6662.3759	81.6234			
	BPNN, Layer: 1, Neurons (L1) 11	CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.6787	0.4606	7.4301	4.4261	5.8129	5081.7732	71.2866			
	MARS, piecewise-linear, 16 BFs	CO, CO <sub>2</sub>	0.3282	0.1077	10.6824	8.0427	9.1629	10504.3010	102.4905			
	MARS, piecewise-cubic, 20 BFs	CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.6031	0.3637	6.4275	4.0094	5.1623	3802.8747	61.6675			
	SVR, Linear kernel	CO, CO <sub>2</sub>	0.3650	0.1332	8.4538	6.1933	7.2103	6578.6165	81.1087			
	SVR, Gaussian kernel	CO, CO <sub>2</sub> , H <sub>2</sub> , CH <sub>4</sub> , O <sub>2</sub>	0.6913	0.4779	7.9051	4.6740	6.9661	5752.3317	75.8441			
Calorific value	BPNN, Layer: 1, Neurons (L1) 5	Air, O <sub>2</sub>	0.0401	0.0016	19.8082	19.0445	22.9766	5.2579	2.2930			
	BPNN, Layer: 1, Neurons (L1) 7	Air, O <sub>2</sub> , Outlet pressure	0.7187	0.5166	15.8219	9.2055	16.8804	3.3546	1.8316			
	MARS, piecewise-linear, 15 BFs	Air, O <sub>2</sub>	0.0419	0.0018	80.7365	77.4897	30.3193	87.3500	9.3461			
	MARS, piecewise-cubic, 18 BFs	Air, O <sub>2</sub> , Outlet pressure	0.4386	0.1924	17.4620	12.1382	17.8493	4.0861	2.0214			
	SVR, Gaussian kernel	Air, O <sub>2</sub>	0.4171	0.1740	18.9023	13.3387	19.4816	4.7880	2.1881			
	SVR, Gaussian kernel	Air, O <sub>2</sub> , Outlet pressure	0.5997	0.3596	13.1471	8.2184	11.3808	2.3162	1.5219			

TABLE 11. Overall results from the testing phase.

model fitting with two observations, slightly better results for training were achieved when the piecewise-linear type of the MARS model was used. The MARS model also had the highest time consumption during the training in this case. Fitting the model of the calorific value has reached a worse performance on average than in the case of the temperature. The higher performance index in the training phase is due to the higher variability of inputs and low correlation between the inputs and target. The results of fitting with the BPNN take the third-place, because of the worst performance index (PI), both in the case of the calorific value and the underground temperature. In general, to improve the performance index in the training phase, it is suggested to use more input variables.

The results from the test phase are shown in Table 11. These results are not so consistent as in the training phase, especially for the temperature prediction where the best results are scattered over individual methods. In the testing phase, the model was verified on untrained data in order to predict the target variable. The time consumption was not evaluated in the testing phase.

For the temperature prediction with five input variables, the best result in terms of performance index was obtained in the utilization of piecewise-cubic type of the MARS model with 20 BFs. The worst result in terms of the performance index was obtained by the SVR with the Gaussian kernel and five input variables. When the SVR with five inputs was used, the predicted temperature value correlated the best with the measured one. With two input variables, the best results were obtained by the BPNN and the worst by the piecewise-linear type of the MARS model.

For the calorific value prediction with three input variables, the results with the lowest performance index were obtained for the utilization of the SVR and Gaussian kernel. Results obtained with the utilization of the BPNN take the second-place. When the BPNN with three inputs was used, the predicted calorific value correlated the best with the measured one. The worst results in terms of the performance index were obtained by the piecewise-cubic type of the MARS model with 18 BFs and three inputs. With two input variables, the best results were obtained for the SVR and Gaussian Kernel and the worst in the case of the piecewise-linear type of the MARS model. The prediction of the underground temperature has reached a higher performance index on average than in the case of temperature.

## 5. SUMMARY AND CONCLUSIONS

In this paper, three approaches were examined in order to find the best prediction method for the UCG data soft-sensing. A comparison of methods suitable for predicting the UCG data has not been published yet. In the UCG process, it is a complicated to predict some process variables because it is not possible to

see the state of the process that runs in an inaccessible environment. The goal was to find the valid data-driven learning method that allows to estimate the underground temperature or the syngas calorific value from other measurable process variables. Predicting these variables will make it possible to control the UCG process more efficiently. In this paper, only a small amount of measurable input variables from one UCG experiment were used to get a comparison of learning methods. All methods that have been applied considered only one output variable. The resulting MARS model can be stored in a PC and is even portable as an analytic equation and the impact of each predictor can clearly be seen (i.e., the model is easier to understand by humans). In MARS, the prediction is based on a simple and quick calculation of the MARS model formula. In SVR, each variable is multiplied by the corresponding element of each support vector, which can be a slow process if there are many variables and a large number of support vectors. Individual SVR models have been obtained by using the  $\varepsilon$ -SVR. Applying the kernel trick in the SVR, it allows modelling expert knowledge of the UCG process. The SVR is defined as a convex optimization problem where there are no local minima, and therefore, effective optimization methods such as SMO can be used. In the case of NNs, the training is often complicated because there is always a risk of a deadlock at the local minimum of the error function. Also, the learning of NNs is highly complicated by looking for a high number of weights in a multidimensional space. In MARS and SVR, it is needed to package program code that provides the prediction with the optimized weights or support vectors. It can be said that all three methods have achieved satisfactory results in terms of the underground temperature and syngas calorific value prediction. Regarding the training, the SVR with the Gaussian kernel was the winner. This model best matched the measured data, both in the case of the temperature and the calorific value. Regarding the prediction, the best result was obtained by piecewise-cubic type of MARS model. In these cases, the better results were achieved at all considered input variables of the target variable. The results show that a higher number of input variables increases the predictive performance. Obtained results can be applied to the model predictive control of the UCG process.

## ACKNOWLEDGEMENTS

This work was supported by the EC Research Programme of the Research Fund for Coal and Steel (Grant No. RFCR-CT-2013-00002), by the Slovak Grant Agency for Science under grant VEGA 1/0273/17, and by the Slovak Research and Development Agency under the contract No. APVV-14-0892.

## REFERENCES

- [1] G. Ökten, V. Didari. Underground gasification of coal. in: Kural, o. (ed.) coal. Technical report, Istanbul

- Technical University, University, Istanbul, Turkey, p. 371-378, 1994.
- [2] J. Kačur, M. Durdán, M. Laciak, P. Flegner. Impact analysis of the oxidant in the process of underground coal gasification. *Measurement* **51**:147–155, 2014. DOI:10.1016/j.measurement.2014.01.036.
- [3] M. Sury, M. White, J. Kirton, et al. Review of Environmental Issues of Underground Coal Gasification, Technical Report COAL R272 DTI/Pub URN 04/1880. Technical report, WS Atkins Consultants Ltd. Department of Trade and Industry, 2010.
- [4] J. Kačur, M. Durdán, G. Bogdanovská. Monitoring and measurement of the process variable in UCG. In *SGEM 2016: 16th International Multidisciplinary Scientific GeoConference, Bulgaria - Sofia : STEF92 Technology*, pp. 295–302. 2016. DOI:10.5593/SGEM2016/B21/S07.038.
- [5] M. Durdán, K. Kostúr. Modeling of temperatures by using the algorithm of queue burning movement in the UCG process. *Acta Montanistica Slovaca* **20**(3):181–191, 2015.
- [6] K. Kostúr. Mathematical modeling temperature's fields in overburden during underground coal gasification. In *ICCC 2014 : Proceedings of the 2014 15th International Carpathian Control Conference (ICCC) Velke Karlovice, May 28-30*, pp. 248–253. Danvers : IEEE, 2014. DOI:10.1109/CarpathianCC.2014.6843606.
- [7] M. Koenen, F. Bergen, P. David. Isotope measurements as a proxy for optimising future hydrogen production in underground coal gasification, news in depth, 2015.
- [8] M. Benková, M. Durdán. Statistical analyzes of the underground coal gasification process realized in the laboratory conditions. In *SGEM 2016: 16th International Multidisciplinary Scientific GeoConference, Bulgaria - Sofia : STEF92 Technology*, pp. 405–412. 2016. DOI:10.5593/SGEM2016/B21/S07.052.
- [9] L. Fortuna, S. Graziani, A. Rizzo, G. M. Xibilia. *Soft Sensors for Monitoring and Control of Industrial Processes*. Springer London, 2007. DOI:10.1007/978-1-84628-480-9.
- [10] T. Ji, H. Shi. Soft sensor modeling for temperature measurement of texaco gasifier based on an improved RBF neural network. In *2006 IEEE International Conference on Information Acquisition*, pp. 1147–1151. IEEE, 2006. DOI:10.1109/icia.2006.305907.
- [11] A. A. Uppal, A. I. Bhatti, E. Aamir, et al. Control oriented modeling and optimization of one dimensional packed bed model of underground coal gasification. *Journal of Process Control* **24**:269–277, 2014. DOI:10.1016/j.jprocont.2013.12.001.
- [12] A. A. Uppal, A. I. Bhatti, E. Aamir, et al. Optimization and control of one dimensional packed bed model of underground coal gasification. *Journal of Process Control* **35**:11–20, 2015. DOI:10.1016/j.jprocont.2015.08.002.
- [13] A. A. Uppal, Y. M. Alsmadi, V. I. Utkin, et al. Sliding mode control of underground coal gasification energy conversion process. *IEEE Transactions on Control Systems Technology* **26**(2):587–598, 2018. DOI:10.1109/tcst.2017.2692718.
- [14] Q. Wei, D. Liu. Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification. *Transactions on Automation Science and Engineering, IEEE* **11**(4):1020–1036, 2014. DOI:10.1109/TASE.2013.2284545.
- [15] R. Guo, G. X. Cheng, Y. Wang. Texaco coal gasification quality prediction by neural estimator based on dynamic PCA. In *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, pp. 1298–1302. 2006. DOI:10.1109/ICMA.2006.257660.
- [16] B. Guo, Y. Shen, F. Zhao. Modelling coal gasification with a hybrid neural network. *Fuel* **76**(12):1159–1164, 1997. DOI:10.1016/s0016-2361(97)00122-1.
- [17] S. Liu, Z. Hou, C. Yin. Data-driven modeling for fixed-bed intermittent gasification processes by enhanced lazy learning incorporated with relevance vector machine. In *11th IEEE International Conference on Control & Automation (ICCA)*, pp. 1019–1024. IEEE, 2014. DOI:10.1109/icca.2014.6871060.
- [18] M. Laciak, J. Kačur, K. Kostúr. The verification of thermodynamic model for UCG process. In *ICCC 2016: 17th International Carpathian Control Conference*, pp. 424–428. 2016. DOI:10.1109/CarpathianCC.2016.7501135.
- [19] M. Laciak, D. Ráškayová. The using of thermodynamic model for the optimal setting of input parameters in the UCG process. In *ICCC 2016: 17th International Carpathian Control Conference*, pp. 418–423. 2016. DOI:10.1109/CarpathianCC.2016.7501134.
- [20] A. M. Winslow. Numerical model of coal gasification in a packed bed. *Symposium (International) on Combustion* **16**(1):503–513, 1977. DOI:10.1016/s0082-0784(77)80347-0.
- [21] P. Ji, X. Gao, D. Huang, Y. Yang. Prediction of syngas compositions in shell coal gasification process via dynamic soft-sensing method. In *Proceeding of 10th IEEE International Conference on Control and Automation (ICCA)*, pp. 244–249. 2013. DOI:10.1109/ICCA.2013.6565140.
- [22] I. H. AL-Qinani. Multivariate adaptive regression splines (MARS) heuristic model: Application of heavy metal prediction. *International Journal of Modern Trends in Engineering & Research* **3**(8):223–229, 2016. DOI:10.21884/ijmter.2016.3027.7nuqv.
- [23] A. Aryafar, R. Gholami, R. Rooki, F. D. Ardejani. Heavy metal pollution assessment using support vector machine in the shur river, sarcheshmeh copper mine, iran. *Environmental Earth Sciences* **67**(4):1191–1199, 2012. DOI:10.1007/s12665-012-1565-7.
- [24] D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning internal representation by error propagation. in: D. E. Rumelhart, J. L. McClelland, and PDP Research Group. *Parallel Distributed Processing. Explorations in the Microstructure of Cognition*. Vol 1: Foundation, 1987.
- [25] G. Sampson, D. E. Rumelhart, J. L. McClelland, T. P. R. Group. *Parallel distributed processing: Explorations in the microstructures of cognition. Language* **63**(4):871, 1987. DOI:10.2307/415721.

- [26] T. Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning - Data Mining, Inference, and Prediction, Second Edition*. Springer New York, 2009. DOI:10.1007/b94608.
- [27] V. Kvasnička, . Beňušková, I. F. J. Pospíchal, et al. *Úvod do teórie neuronových sietí*. IRIS, Bratislava, 1997.
- [28] J. Sedláček. *Úvod do teorie grafů*. Academia, Praha, 1981.
- [29] J. H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics* **19**(1):1–67, 1991. DOI:10.1214/aos/1176347963.
- [30] P. Sephton. Forecasting recessions: can we do better on MARS?, 2001.
- [31] M. Chugh, S. S. Thumsi, V. Keshri. A comparative study between least square support vector machine(lssvm) and multivariate adaptive regression spline(mars) methods for the measurement of load storing capacity of driven piles in cohesion less soil. *International Journal of Structural and Civil Engineering Research* 2015. DOI:10.18178/ijscer.4.2.189-194.
- [32] V. R. Tselykh. Multivariate adaptive regression splines. *Machine learning and data analysis* **1**(3):272–278, 2012. DOI:10.21469/22233792.
- [33] P. Samui, D. P. Kothari. A multivariate adaptive regression spline approach for prediction of maximum shear modulus and minimum damping ratio. *Engineering Journal* **16**(5):69–78, 2012. DOI:10.4186/ej.2012.16.5.69.
- [34] W. Zhang, A. T. C. Goh. Multivariate adaptive regression splines and neural network models for prediction of pile drivability. *Geoscience Frontiers* **7**(1):45–52, 2016. DOI:10.1016/j.gsf.2014.10.003.
- [35] A. Abraham, D. Steinberg. MARS: Still an alien planet in soft computing? In *International Conference on Computational Science - ICCS (Proceedings), Part II, vol. 2*, pp. 235–244. Springer Berlin Heidelberg, 2001. DOI:10.1007/3-540-45718-6\\_27.
- [36] B. E. Boser, I. M. Guyon, V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory - COLT'92, Pittsburgh, PA*, pp. 144–152. ACM Press, 1992. DOI:10.1145/130385.130401.
- [37] V. N. Vapnik. Constructing learning algorithms. In *The Nature of Statistical Learning Theory*, pp. 119–166. Springer Verlag, New York, 1995. DOI:10.1007/978-1-4757-2440-0\\_6.
- [38] K. R. Müller, A. J. Smola, G. Rätsch, et al. Predicting time series with support vector machines. In *Lecture Notes in Computer Science*, pp. 999–1004. Springer Berlin Heidelberg, 1997. DOI:10.1007/bfb0020283.
- [39] J. Kačur, M. Laciak, M. Durdán, P. Flegner. Utilization of Machine Learning Method in Prediction of UCG Data. In *ICCC 2017: 18th International Carpathian Control Conference*, pp. 1–6. IEEE, 2017. DOI:10.1109/carpathiancc.2017.7970411.
- [40] MathWorks. Understanding Support Vector Machine Regression, In: *Statistics and Machine Learning Toolbox User's Guide (R2018ab)*. regression.html, 2018.
- [41] J. Kačur, K. Kostúr. Approaches to the Gas Control in UCG. *Acta Polytechnica* **57**(3), 2017. DOI:10.14311/ap.2017.57.0182.
- [42] M. Laciak, K. Kostúr, M. Durdán, et al. The analysis of the underground coal gasification in experimental equipment. *Energy* **114**:332–343, 2016. DOI:10.1016/j.energy.2016.08.004.
- [43] R. L. I. Dobbs, W. B. Krantz. Combustion front propagation in underground coal gasification, Final Report, Work Performed under Grant No. DE-FG22-86PC90512. Technical report, University of Colorado, Boulder Department of Chemical Engineering, 1990. DOI:10.2172/6035494.
- [44] K. Stańczyk, A. Smoliński, K. Kapusta, et al. Dynamic experimental simulation of hydrogen oriented underground gasification of lignite. *Fuel* **89**(11):3307–3314, 2010. DOI:10.1016/j.fuel.2010.03.004.
- [45] K. Kostúr, J. Kačur. Developing of optimal control system for UCG. In *Proceedings of the 13th International Carpathian Control Conference (ICCC)*, pp. 347–352. IEEE, 2012. DOI:10.1109/carpathiancc.2012.6228666.
- [46] K. Kostúr, J. Kačur. Development of control and monitoring system of UCG by promotiv. In *2011 12th International Carpathian Control Conference (ICCC)*, pp. 215–219. IEEE, 2011. DOI:10.1109/carpathiancc.2011.5945850.
- [47] A. H. Gandomi, D. A. Roke. Intelligent formulation of structural engineering systems. In *Seventh MIT Conference on Computational Fluid and Solid Mechanics-Focus: Multiphysics and Multiscale*, pp. n/a–n/a. Cambridge, USA, 2013.
- [48] A. H. Gandomi, D. A. Roke. Assessment of artificial neural network and genetic programming as predictive tools. *Advances in Engineering Software* **88**:63–72, 2015. DOI:10.1016/j.advengsoft.2015.05.007.