Available online at http://scik.org
J. Math. Comput. Sci. 11 (2021), No. 3, 2914-2933
https://doi.org/10.28919/jmcs/5657
ISSN: 1927-5307

MATHEMATICAL AND NUMERICAL RESULTS FOR QUALITY CONTROL OF HOT METAL IN BLAST FURNACE

A. AZZEDINE[1], F.Z. NOURI[1,*], S. BOUHOUCHE[2]

[1]Mathematical Modeling and Numerical Simulation Research Laboratory, Badji Mokhtar University, Po. Box 12, Annaba 23000, Algeria

[2]Iron and Steel Applied Research Unit, URMA-Annaba, Algeria

Abstract. The blast furnace is a very complex industrial equipment producing hot metal from iron oxides. Its measuring, modeling and exploring functionalities is very crucial; this is due to the difficult measurement and control problems related to the unit.

To maintain high efficiency, the current work proposes new adaptive algorithms for data-driven methods. These methods are classified into supervised and unsupervised algorithms, well known in optimization problems as regression, classification and clustering. To extract their limitations, a comparative study between the proposed techniques is presented, where the obtained results are validated on real data from the steel processes ArcelorMittal-Annaba, proving the feasibility and effectiveness of the proposed models and numerical procedures.

Keywords: supervised and unsupervised learning; data analysis; modeling.

2010 AMS Subject Classification: Primary 62P30, 62M20, 68Q32; Secondary 65C20.

1. Introduction

Blast furnace is an extremely complicated nonlinear system consisting of several specific elements such as the loading equipment; the cooling circuit, the whole production of hot wind, and the big cylindrical shaft four.

*Corresponding author

E-mail address: tassili.nan09@gmail.com

Received March 08, 2021

Their principal operation is as follows: the solid raw material like iron ore and coke is charged in a mixture successive layers from the upper of the furnace. Under the action of their particular weight, they gradually descend to the bottom of the heating up until it melts; however, the hot combustion gases rise through the column of combustion materials. The molten metal consisting of cast iron and slag flows into the crucible. At the end of combustion, the materials split into two elements; molten slag on the one side and molten hot metal on the other, which accumulate according to their own specific masses. The slag is drained through the slag tap hole which is higher than the cast iron tap hole.

The heat and mass transfer process complexity combined with a wide variety of chemical reactions and high pressure, make blast furnace modeling an extremely difficult problem. Despite the best researchers's efforts to solve this type of problems, challenges do remain.

Prediction and control of product quality by virtual sensing technology is a key tool in real time, and has been extensively used in many manufacturing processes (see Zhang (2017) [13] and Zhou et al. (2017) [15]). Usually, it is split in two main groups, that are the data-driven techniques and the first-principle models. In this case, it is very hard to construct the first principle due to the large size of the reactor and the complex internal operating environments, while the data-driven model do not need previous knowledge of the full operation of the process and are based directly on the process data, Ge et al. (2017) [2]. Therefore it has been effectively applied to a various industrial processes, as typical examples the time series models by Saxen (2013), multivariate statistical approach (PLS, PCR).

For an overview on the Data-driven time discrete models for dynamic prediction of the hot metal silicon content, the reader is referrred to Gao et al (2011) [1] and Saxen et al. (2013) [8], where they proposed a model to predict the change of thermal state of blast furnace hearth with support vector machine; while Ling J. et al (2011) [5] developed an adaptive model using the sliding windows smooth support vector regression (SVR) for nonlinear blast furnace system.

Numerous dynamic systems are characterized by a non-linear dynamic behavior, such as the blast furnace, where nonlinear models are then necessary. Indeed, it has been shown that SVR and Neural Network (NN) can approach continuous nonlinearities, and have been applied to the modeling of complex nonlinear systems whose complexity is often due to the high number of weights in the network; in addition to the model identification principle using conventional Recursive Least Square (RLS) and its adaptive version. More details about these methods can be found in different documents, see for example [1], [2] and [9] and references therein.

In this study, we propose a new approach to measurement quality control of blast furnace using robust nonlinear modeling and identification techniques known as data driven methods; with an attempt to reduce modeling and identification errors concerning the structure and parameters of the model that is included in the uncertainty budget.

2. Big Data and Data Driven Methods

Big Data is the huge amount of pseudo infinite information being collected from different databases. These raw data are usually of a different special nature characterising the multi-scale actions of the system under consideration. Although the quantity of information stored is important, there are other many things that have the same importance, like the accuracy, variety (format, nature, and type) , and value (perspectives and impact) of these data, as well as the absence of noise, knowing that outliers, missing and incorrect values with the data sparsity could introduce noise.

Big Data analysis faces many challenges along with the previous important issues such as data quality and verification, high dimensions, spread and representation data resources, data visualization and testing and the capability to develop algorithms. In order to derive patterns from large-scale data and understand the value of big data analysis, we need to apply and adapt certain methods such as the Machine Learning paradigms and algorithms; and deep learning. Machine learning is an important field and a research frontier in artificial intelligence, which have significant part in big data analysis long with computational power. its task is to analyse the largest amount of data at all levels, whether simple or complex and ensure that more accurate results and decisions are obtained faster. This can be categorized into two model forms:

The first is called supervised learning, where for this type of input data, it is usually required to train reliable data representations that can be expanded to new data in the same big data applications area. It can be classified into two main groups, logistic regression and regression. Neural Network (NN) and support vector machines (SVM) are considered as classification methods to get the best prediction or explore different linear or nonlinear models, just enough to change the structure parameters, such as the kernel for the SVM, the activation function and number of layers in the NN. Different algorithms based on robust ones from gradient descent to Levenberg Marquardt, are used to minimize the modelling errors for these model types.

The second is named unsupervised learning, in which there is no modelling error to supervise and no direct learning algorithm to track a model output. The Partial least square (PLS), Principal components analysis (PCA), dimensionality reduction and clustering, belong to unsupervised learning. Generally, we use these methods to extract features from the noisy data, as a pre-processing step, then the pre-processed data is applied as inputs for the supervised learning step.

Note that this section is not to cover the Big Data in particular, but to provide a brief overview of its key concepts and challenges as detailed in next sections.

2.1. PLS technique.

The key step in a process modeling is to define the most important input variables and forecast the process response from the collected data. Nevertheless, the high size and co-linearity of these data make it complex to build a robust process model. the need to explain the process quality of these data has led to the development of a multivariate analysis for complex processes. In order to reach this goal we tried to use different strategies among them the PLS technique. As a supervised dimension reduction methodology, it was invented in 1983 by Herman Wold [10] and was taken to pick up an optimal subset of input variables called ”latent variables”, where its purpose is to construct new predictor variables as linear combinations of the original ones summarized in a matrix X and a vector y of response variables (class labels). It focuses on maximising the covariance between the extracted factors from both input and output process sets.

Let $X \in \mathbb{R}^m$ and $Y \in \mathbb{R}^n$ be m and n-dimensional spaces of m and n variables, respectively. From N observed samples for each $x \in X$ and $y \in Y$, we get two variable blocks $X \in \mathbb{R}^{N\times m}$ and $Y \in \mathbb{R}^{N\times n}$, and write the general formulation as :

$$X = \sum_{i=1}^{k} t_i.p'_i + \varepsilon_X = T.P' + \varepsilon_X, \tag{2.1}$$

$$Y = \sum_{i=1}^{k} u_i.q'_i + \varepsilon_Y = U.Q' + \varepsilon_Y, \tag{2.2}$$

where T *and* $U \in \mathbb{R}^{N\times S}$ are factor matrices for the score and latent variables, respectively; $P \in \mathbb{R}^{m\times S}$ and $Q \in \mathbb{R}^{n\times S}$ are loading matrices and $\varepsilon_X \in \mathbb{R}^{N\times m}$ and $\varepsilon_Y \in \mathbb{R}^{N\times n}$ are error terms. In the PLS regression, the optimization criterion is

$$argmax\{cov(X_{k-1}w_k, Y_{k-1}q_k)\}, \tag{2.3}$$

under the contraints $w_k^{'}.w_k = q_k^{'}.q_k = 1$ and $cov(X_{[k-1]}w_k, X_{[k'-1]}w_{k'}) = 0, k \neq k'$.
From (2.1), it is readly shown that the first PLS component $t^{(1)} = Xw^{(1)}$ is obtained by maximizing the Tucker criterion of the inter-battery factor analysis:

$$cov^2(Y, Xw^{(1)}) = r^2(Y; Xw^{(1)})Var(Xw^{(1)})Var(Y), \tag{2.4}$$

under the contraints $\|w\| = 1$. $(*)$
We thus try to simultaneously maximize the variance $t^{(1)}$ and the correlation between $t^{(1)}$ and Y. We are therefore looking for a normalized vector $w^{(1)}$ maximizing:

$$\langle t^{(1)}, Y\rangle = \langle Xw^{(1)}, Y\rangle = \|Xw^{(1)}\|.\|Y\|.cor(Xw^{(1)}, Y). \tag{2.5}$$

To obtain the expression of $w^{(1)}$, we solve the optimization problem under constraint $(*)$ by using the Lagrange multipliers method:

$$L(w^{(1)}, \lambda) = cov(Y, Xw^{(1)}) - \lambda(w'^{(1)}w^{(1)} - 1) = w^{'(1)}X^{'}Y - \lambda(w'^{(1)}w^{(1)} - 1), \tag{2.6}$$

with $\lambda \in \mathbb{R}^+$. If we set the first derivatives of L with respect to λ and $w^{(1)}$ to zero, we write

$$\frac{\partial L}{\partial \lambda} = -(w'^{(1)}w^{(1)} - 1) = 0 \; or \; w'^{(1)}w^{(1)} = 1, \tag{2.7}$$

$$\frac{\partial L}{\partial w^{(1)}} = X'Y - 2\lambda w^{(1)} = 0 \; or \; X'Y = 2\lambda w^{(1)}. \tag{2.8}$$

By multiplying (2.8) by $w'^{(1)}$ and using (2.7), we get: $w^{'(1)}X^{'}Y = 2\lambda$. Let $\theta \in \mathbb{R}$, by symetry we obtain:

$$\theta = 2\lambda = \langle t^{(1)}, Y\rangle = w^{'(1)}X^{'}Y = Y^{'}Xw^{(1)}. \tag{2.9}$$

From (2.8) and (2.9), we have:

$$(X'Y)Y'Xw^{(1)} = (\theta w^{(1)})\theta = \theta^2 w^{(1)}. \tag{2.10}$$

Consequently, $w^{(1)}$ is the associated eigenvector to the eigenvalue θ^2 of the matrix $X^{'}YY^{'}X$; and the maximization of $\langle Xw^{(1)}, Y\rangle$ amounts to considering θ^2 as being the maximum eigenvalue of $X'YY'X$.

We can therefore deduce an expression for $w^{(1)}$ and the associated eigenvalue.

We write: $X'YY'Xw^{(1)} = \lambda_1 w^{(1)}$ or $\lambda_1 = \langle Xw^{(1)}, Y\rangle^2 = (w'^{(1)}X'Y)'(w'^{(1)}X'Y) = Y'XX'Y$.

Because $Y'XX'Y \in \mathbb{R}$, we obtain $X'YY'Xw^{(1)} = (X'Y)Y'X(w^{(1)}) = \lambda_1 w^{(1)} = Y'XX'Yw^{(1)} = w^{(1)}Y'XX'Y = (w^{(1)})Y'X(X'Y) \Rightarrow w^{(1)} = X'Y$.

From the constraint $(*)$ we have $\left\| w^{(1)} \right\| = 1$ so that $w^{(1)} = \frac{X'Y}{\|X'Y\|}$.

In order to know if the first component $t^{(1)} = Xw^{(1)}$ can sufficiently explain the set of explanatory and the endogenous variables, we perform two regressions X on $t^{(1)}$ and Y on $t^{(1)}$ to get

$$X = X_{[0]} = t^{(1)}p'^{(1)} + X_1, \tag{2.11}$$

$$Y = Y_{[0]} = c^{(1)}t^{(1)} + Y_1, \tag{2.12}$$

with $p^{(1)} = \frac{X'_{[0]}t^{(1)}}{t'^{(1)}t^{(1)}}$ and $c^{(1)} = \frac{Y'_{[0]}t^{(1)}}{t'^{(1)}t^{(1)}}$.

Other weight vectors are iteratively computed by the PLS method, such that when $w^{(1)}$ and $c^{(1)}$ are available, the score vectors can be computed by $t^{(1)} = Xw^{(1)}$, $u^{(1)} = Yc^{(1)}$ and loadings, i.e. the first columns of P and Q can be computed by $p^{(1)} = \frac{X^{'}_{[0]}t^{(1)}}{t^{'(1)}t^{(1)}}$ and $q^{(1)} = \frac{Y'_{[0]}t^{(1)}}{u'^{(1)}u^{(1)}}$ respectively. The data matrices X and Y are then deflated by subtracting their rank-one approximations

$$X \leftarrow X - t^{(1)}p'^{(1)} \textit{ and } Y \leftarrow Y - u^{(1)}q'^{(1)} \tag{2.13}$$

The new X and Y are used to compute $w^{(2)}$ and $c^{(2)}$ based on

$$X'YY'Xw^{(1)} = \lambda_1 w^{(1)} \textit{ and } Y'XX'Yc^{(1)} = \lambda_1 c^{(1)} \tag{2.14}$$

This process is repeated until the residuals are small enough or a predefined number of weight vectors $\{w^{(1)}, ..., w^{(k)}\}$ and $\{c^{(1)}, ..., c^{(k)}\}$ are obtained (see [1, 11]).

2.2. SVR technique.

The support vector regression was introduced by Vapnik in the early 1994 (see [4]); since it became popular and widely applied in many fields. It is firmly grounded in the framework of statistical learning theory which has led to a large development, the SVR is strongly recommended for solving various classification and prediction problems; it is also the key to construct the Lagrange function from an objective function by converting the minimization problem into a dual one [9]. Suppose the training set sample is given by

$$E = \{(x_1, y_1), ...(x_i, y_i), ...(x_L, y_L)\}, \tag{2.15}$$

where $\forall i = 1, ..., L$, $x_i \in \mathbb{R}^n$ is the input of the training sample, and $y_i \in \mathbb{R}$ is the target value. Our goal is to determine a function that can approximate future values accurately. The model is then given by:

$$y = w.\varphi(x) + b, \tag{2.16}$$

where weights $w \in \mathbb{R}^n$ (i.e. the coefficients vector), $b \in \mathbb{R}$ is a constant, and φ a map function supposed to be a nonlinear transformation from $\mathbb{R}^n$ to a higher dimensional feature space.
The aim is to find the weight w and the bias b, such that x can be calculated by minimizing the regression risk defined as:

$$r_{reg}(f) = C \sum_{j=1}^{m} \Gamma(f(x_j) - y_j) + \frac{1}{2} \|w\|^2, \tag{2.17}$$

where $\Gamma(.)$ is the cost function and C is a constant.
The solution for minimizing this cost function is equal to a convex optimization problem with a soft margin loss function, which is represented as follows:

$$min \, \frac{1}{2} \|w\|^2 + C. \sum_{j}^{m} (\xi_j + \xi_j^*), \tag{2.18}$$

subject to

$$\left\{ \begin{array}{c} \langle w, \phi(x_i) \rangle + b - y_i \leq \zeta + \xi_j^*, \\ y_i - \langle w, \phi(x_i) \rangle - b \leq \zeta + \xi_j, \\ \xi_j, \xi_j^* \quad \geq 0, \end{array} \right. \tag{2.19}$$

where ζ is the insensitive parameter; ξ_j, ξ_j^* are slack variables which are introduced to relax the optimization constraints and the vector w can be written in the form:

$$w = \sum_{j=1}^{m} (\alpha_j - \alpha j^*).\phi(x_j). \tag{2.20}$$

By substituting equation (2.20) in (2.16), the generic equation can be rewritten as

$$y = f(x) = \sum_{j=1}^{m} (\alpha_j - \alpha_j^*).\langle \phi(x_j).\phi(x) \rangle + b = \sum_{j=1}^{m} (\alpha_j - \alpha_j^*).K(x_j, x) + b. \tag{2.21}$$

Here $k(x_j, x)$ is the kernel function. Kernel functions allow dot product to be computed in large dimensional feature space using small dimensional space data input without knowing the transformation ϕ. The most widely used cost function is the ζ insensitive loss function that has the form

$$\Gamma(f(x) - y) = \begin{cases} |f(x) - y| - \zeta & for \quad |f(x) - y| \geq \zeta, \\ 0 & otherwise. \end{cases} \tag{2.22}$$

By solving the quadratic optimization problem (2.18), the regression cost in equation (2.17) and the ζ insensitive function in (2.22) can be minimized. We write

$$\frac{1}{2} \sum_{j,l=1}^{m} (\alpha_j - \alpha_j^*)(\alpha_l - \alpha_l^*) K(x_j, x_k) - \sum_{j=1}^{m} \alpha_j^*(y_j - \zeta) - \alpha_j(y_j + \zeta), \tag{2.23}$$

Subject to

$$\sum_{j,l=1}^{m} (\alpha_j^* - \alpha_j) = 0 \quad , \quad \alpha_j, \alpha_j^* \in [0, C[, \tag{2.24}$$

where α_j^* and α_j are the Lagrange multipliers, representing solutions to the quadratic problem that act as forces pushing predictions towards target values y_j.

2.3. Neural Network technique.

The neural networks were invented by Mc Culloch and Pitts in 1950, and made popular by Hopfield [3]. As a nonlinear mapping between the input and the output sets, the NN emulates human systems (Brain), which is characterized by adaptation and self-organization. Moreover they have the ability to learn from the experience and generalization from the previous sample to solve new problems. They are composed of a number of very simple processing elements known as neurons (see Diagram 1). A neuron typically consists of four components: (1) input data (2) a group of weights, (3) a weighted summer (nodes) and (4) a nonlinear activation function ϕ such as sigmoid; note that each neuron has an activation function. The inputs x_i are connected to the nodes in the input layer and the outputs y_p are taken from the output one. Between the input and output layers, there exists one or more hidden layers. All the nodes in one layer are connected to all the nodes in the next one. The connection strength between the output of a node i with a node j is given by a weight W_{ij}. The weights are regression coefficients to be estimated from a sample data. The bias term is comparable with the intercept of conventional regression model, i.e. it allows us to add flexibility when learning. The model can be written as follows

$$u = \sum_{i}^{N} w_i x_i + b, \tag{2.25}$$

$$y = \phi(u), \tag{2.26}$$

where x_i represents the input value, y the model output, $W = \sum_i^N w_i x_i$ is the weight connecting input i with a hidden neuron j, and ϕ is nonlinear called activation function. Different activation functions can be used, among which we can cite, sigmoid, hyperbolic tangent and Gaussian.
In order to find the optimal architecture of a neural network, many different approaches exist. These methods are usually quite complex in nature and are difficult to implement. Moreover there are no hard or fast rules about the choice of the number of nodes and hidden layers to be used in an application. Usually some trials and errors are required to determine the best combination to minimize the error. Hence the connection weights are auto-adjusted using efficient nonlinear optimization algorithms, namely the basic back propagation (BBP) training algorithm [11]. The weights W are changed by an amount ∂W according to the following formula

$$\Delta W = -\eta \frac{\partial E}{\partial W}, \tag{2.27}$$

where the parameter η is the learning rate and E is a cost function quatifying the difference between the initial known values of the approximated function over the discrete set of data and their corresponding NN approximation, written in the form

$$E = \frac{1}{N} \sum_{n=1}^{N} (e_i)^2 = \frac{1}{N} \sum_{n=1}^{N} (y - \hat{y})^2, \tag{2.28}$$

where N is the number of error terms.
The most widely used Levenberg-Marquardt (LM) in an optimization algorithm, can be considered as a trust-region modification to Gauss Newton (GN) method or a more robust one as a bridge between the GN and the gradient descent algorithm. It has been readily shown in many cases, that it converges even if the error surface is much more complex than in the quadratic

situation. The LM is similar to the BBP in the sense that it requires only the gradient vector calculation, wether the LM computes in addition the Jacobian. The LM algorithm can be represented as:

$$W_{k+1} = W_k - (J_k^T J_k + \mu I)^{-1} J_k^T \varepsilon, \tag{2.29}$$

where I is the identity matrix, ε the total error for all patterns and μ a learning parameter, that has to be adjusted several times at each iteration so that the result with the greatest error reduction is selected. When the μ value is very large the LM algorithm becomes steepest decent or BBP, whereas when it is equal to zero it is the Newton Method.

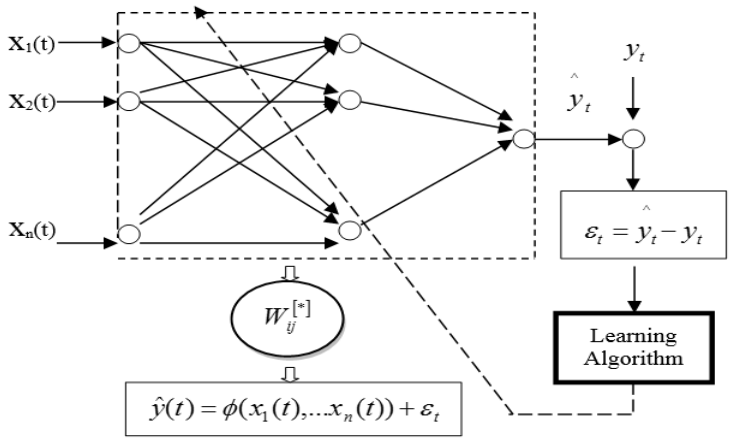
X₁(t)
X₂(t)
Xₙ(t)
$\hat{y}_t$
y_t
$\varepsilon_t = \hat{y}_t - y_t$
Learning Algorithm
$W_{ij}^{[*]}$
$\hat{y}(t) = \phi(x_1(t), \ldots x_n(t)) + \varepsilon_t$

Diagram 1: Principle of the model identification using the NN method

2.4. Adaptive techniques.

The PLS, SVR and NN methods presented previously, can be used in an adaptive form, that is needed particularly when we consider the system to be modelled in time. In this situation the adaptive form is realised by the use of sliding window algorithms (see Diagram 2).

If we consider a system defined by measured input and output data $X(t) \in \mathbb{R}^{\ltimes}$ and $y(t) \in \mathbb{R}$, to take into account the variability of $\langle X(t), y(t) \rangle$, a sliding window with width N is introduced and the input output data become $\langle X(t-N:t), y(t-N:t) \rangle$ for a large time $t = 1:M$, M is the maximum length of the window.

The actual modeling error is $e[t-N:t] = y_p[t-N:t] - y[t-N:t]$

The online recursive predicted model output that takes into account the past prediction errors is given by $y[t-N+1:t+1] = f(e[t-N:t], x[t-N+1:t+1])$, where f is a function obtained by a SVR learning input-output based algorithm.

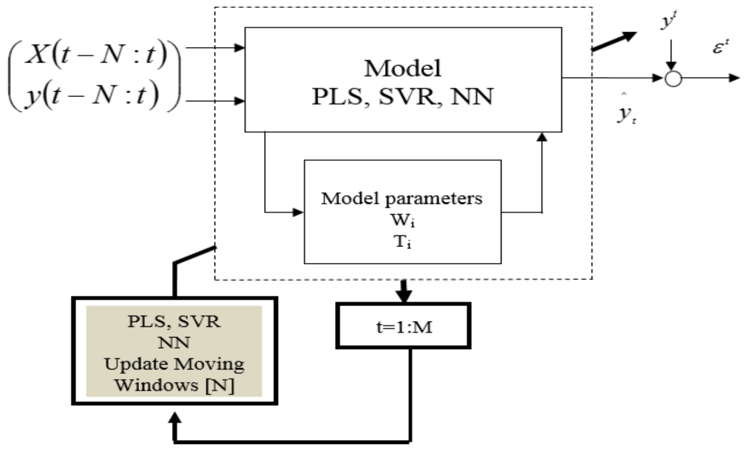

Diagram 2: Principle of the model identification using the LPS, SVR and NN adaptive forms

3. Application to quality control of hot metal in a blast furnace

3.1. Schematic of blast furnace with its zones and experimental results. In this section we present the schematic, the process and the structure of a blast furnace as shown in the Diagrams 3 and 4. The model of experimental inputs and outputs are shown in Table 1 and Figure 1. Table 1 defines the Process Parameters input and output data. The model structure given in Figure 1 shows the influence of input space on the output.

The model inputs are the natural gas flow rate, the heat wind flow rate and the oxygen purety as shown in Figure 1 (Fig.1a-Fig.1c), while the output model is the temperature of hot metal (see Figure 1, Fig1d.)

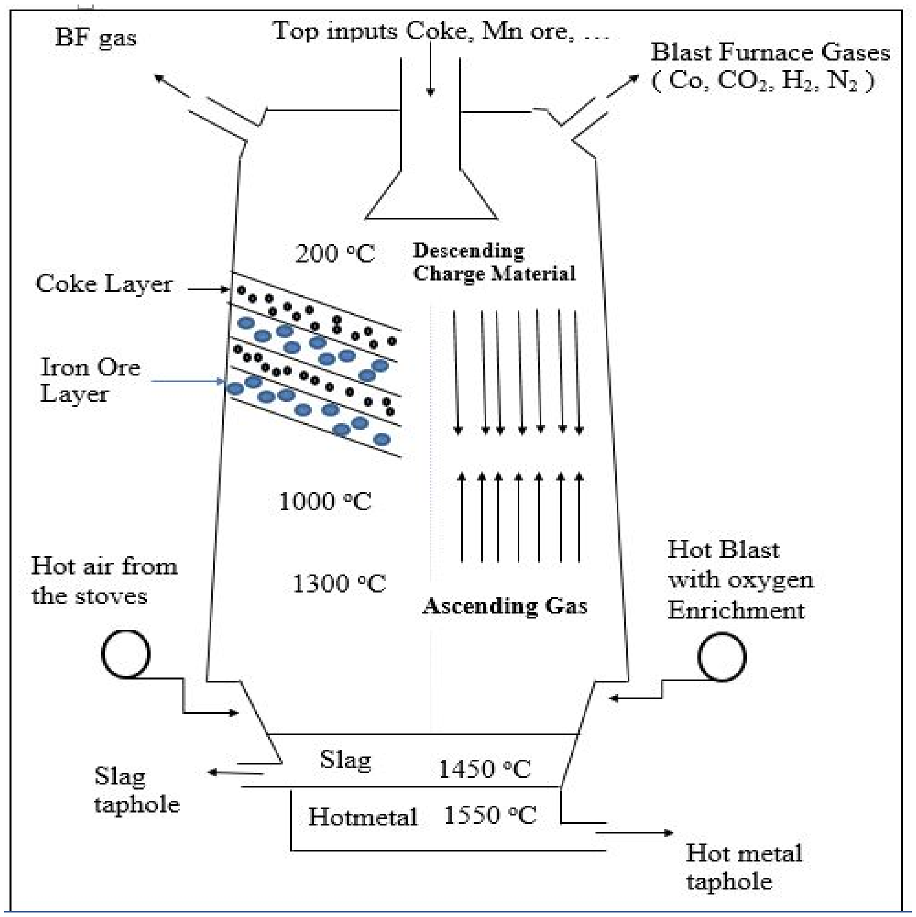

Diagram 3: A schematic of the blast furnace

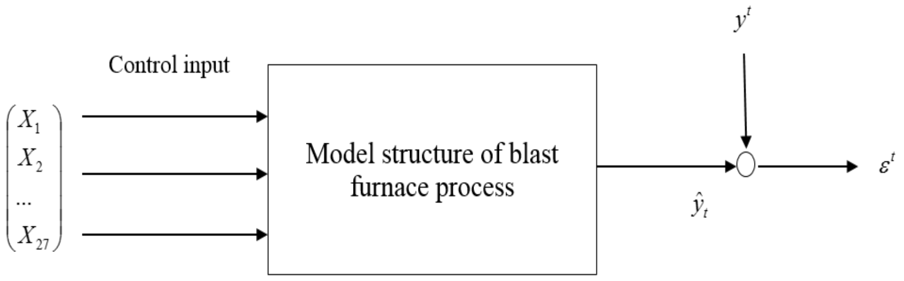

Diagram 4: Model structure

Table 1 :Nomenclature of inputs and outputs data

Variable	Designation	Unit	Variable	Designation
Run Parameters			Results Analysis	
X_1	Circular pressure	bar	X_{12}	Fe agglo
X_2	Pression centre	bar	X_{13}	Fe pellets
X_3	Hot wind temperature	°C	X_{14}	Fe min. cal.
X_4	Temperature of the pig iron	°C	X_{15}	Fe charge
X_5	G N		X_{16}	Fe pig iron
X_6	Steam	T/h	X_{17}	Fe pig iron+Losses
X_7	Flame temperature	°C	X_{18}	H2o coke
X_8	Wind velocity	m/s	X_{19}	H2o pellets
Injections inputs			X_{20}	CaO agglo
X_9	Naturel gaz		X_{21}	SiO2 agglo
X_{10}	Hot wind flow rate	Knm3/h	X_{22}	CaO pel
X_{11}	O2		X_{23}	Sio2 pel
			X_{24}	CaO min. cal
			X_{25}	SiO2 min. cal
			X_{26}	CaO Slag.
			X_{27}	SiO2 Slag.
Output				
y	Temperature of the pig iron			

Our goal is to find an input/output mathematical model $y = f(X_1,.,X_n)$.

3.2. Numerical Experimentation.

In this section we present the numerical algorithms used and the obtained results by the techniques that have been discussed in the previous sections.

3.2.1. *Numerical Algorithms.*

Algorithm 1. ***PLS Algorithm***

input : *Two matrices X and Y , arbitrary w with* $\|w\| = 1$

output: *Weight vectors w and residuals, loading vector S and score vectors* (t,u).

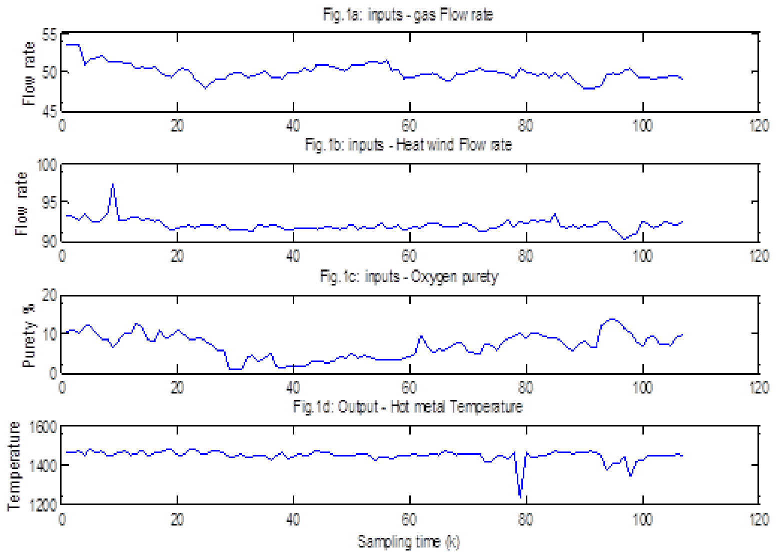

FIGURE 1. The model of experimental inputs and outputs

Step 1. Given a starting vector u_1, usually one of the columns in Y.

Step 2. Calculate the X-weights w, by $w_1 = \frac{X^T u_1}{\|X^T u_1\|}$

Step 3. Calculate X-scores T, by $t_1 = Xw_1$

Step 4. Calculate the Y-weights L, by $L_1 = \frac{u_1^T t_1}{\|u_1^T t_1\|}$

Step 5. Update the Y-scores U, by $u_1 = YL_1$

Step 6. Update S based on t by: $S_1 = \frac{X^T t_1}{t_1^T t_1}$, $S \leftarrow \frac{S}{\|S\|}$, $t \leftarrow \frac{t}{\|S\|}$, $w \leftarrow \frac{w}{\|S\|}$.

Step 7. Find the regression coefficient for the inner relation: $b_1 = \frac{u_1^T t1}{\|t_1^T t_1\|}$

Step 8. The residuals, ε_x and ε_y, are calculated by: $\varepsilon_x = X - t_1 S_1^T$, $\varepsilon_y = X - t_1 L_1^T$

Step 9 . Continue with next component until there is no more significant information.

Algorithm 2. ***SVR Algorithm***

Initialize the network weights W_{ij}^0 =[-0.5 to -0.5] and define the computing loop as:

For $K = 1 : L_k$

Step1: Compute ϕ

Step2: Compute the model output $\hat{y}$ from (2.21)

Step3: Compute the modeling error as: $e(k) = y(k) - \hat{y}(k)$

If $e(k) \approx 0$, $[w,b]^k = [w,b]^{k-1}$, *then stop :* $[w,b]^k = [w,b]^{[*]}$

Else, Adjust the SVR weights using the recursive Quadratic Programming Algorithm

Algorithm 3. ***NN Algorithm*** *The model structure can be defined as* $f \rightarrow NN$ *and its identification is simplified by the following steps:*

```
Initialize the network weights $W_{ij}^{0}$ =[-0.5 to -0.5] and define the computing loop as:
For $K = 1 : L_k$
Step 1: Acquisition of inputs/outputs.
Step 2: Compute u using equation (2.25).
Step 3: Compute the model output $\hat{y}(K)$ from (2.26).
Step 4: Compute the modeling error $\varepsilon(K) = y(K) - \hat{y}(K)$
a) If $\varepsilon(K) \approx 0$, $W_{ij}^{K} = W_{ij}^{K-1} \rightarrow$ stop: $W_{ij}^{K} = W_{ij}^{[*]}$
b) Else, Adjust the NN weights using the recursive Algorithm:
$W_{ij}^{K} = W_{ij}^{K-1} + G(K)\varepsilon(K)$ by Levenberg-Marquardt algorithm i.e equation (2.29)
End k.
```

3.2.2. *Results and Comments.* The pig iron temperature prediction results as a function of the oxygen flow rate, the oxygen purity and the hot wind flow rate for the blast furnace process are given in Figure 3, from top to bottom as follows:

The temperature prediction is carried out by the PLS model and its corresponding adaptive version. The latter approach gives better results of the predicted error which is lower than the conventional PLS. This is quite obvious because the adaptive model performs an immediate correction by obtaining a new projection at each iteration.

As the SVR approach is based on the Vapnik algorithm, its adaptive version is an execution of this conventional algorithm at every time using a sliding window mode. Similar to the PLS algorithm, the SVR adaptive version behaves better. However, compare to the PLS, the SVR approach gives better results in both cases (conventional and adaptive), and this is connected to the nature of the Vapnik algorithm which is based on the complex quadratic programming, while the PLS is just a projection technique.

For the NN technique, the activation function was LOGSIG and the size of the hidden layer used in artificial neural was 5, i.e. a single hidden layer of the neural network. The adaptive artificial neural network model does not work very well leading to very noisy values, the numerical implementation is still under consideration.

Based on the obtained results (see Figure 2 & 3 and Table 2), the SVR model gives the highest prediction accuracy compare to the proposed approaches PLS and NN and it can be qualified to be the best tool. The NN model may not be recommended, particularly in the case of noisy data as it induces a very high uncertainty.

The adaptive versions are implemented in the same way by using sliding windows as shown in Diagram 2.

Note that in table 2., PLSadap, SVRadap and NNadap are the adaptive versions of the PLS, SVR and NN, respectively.

Table 2:Indicators of the different methods

Methods	STD Learning	STD Prediction	Observations
PLS	27.5812	47.4724	double learning
PLSadap	19.6558	21.9312	almost equivalent to learning
SVR	1.8029	14.6761	more important than learning
SVRadap	0.4082	6.4565	Not big difference
NN	26.2949	46.3315	double learning
NNadap	25.0469	34.7740	Not big difference

4. CONCLUSION

In this paper, we explored three techniques and their adaptive versions, where we present the temperature predictions and their corresponding error estimates.The pig-iron temperature prediction as a function of the oxygen flow rate, the oxygen purity and the hot wind flow rate for the blast furnace process are solution of 2.16; where the results are shown in Table 2 and Figures 2 & 3. As reported in the previous section, the adaptive approaches give better results of the predicted error than the conventional ones in both the PLS and the SVR techniques; however, the SVR is more competetive than the PLS. The results of predicting the temperature of the cast iron by using the SVR approach are obtained by the use of a modified version based on the Vapnik algorithm [12]. An adaptive version of this latter is detailed in Algorithm 2 together with Diagram 2, and similar to the PLS case it also gives better results; however high uncertainty is noticed. Indicators for the proposed approaches are given in Table 2 and Figure 2. Therefore, we conclude that the adaptive based model approaches are more precise and can be recommended, more particularly they lead to a reduced uncertainties range in prediction compare to the partial least square model because all the input changes are considered; whereas the NN model may not be recommended, particularly in the case of noisy data. Furthermore the models can be combined with sensitivity analysis by the use of the Monte Carlo simulation technique.

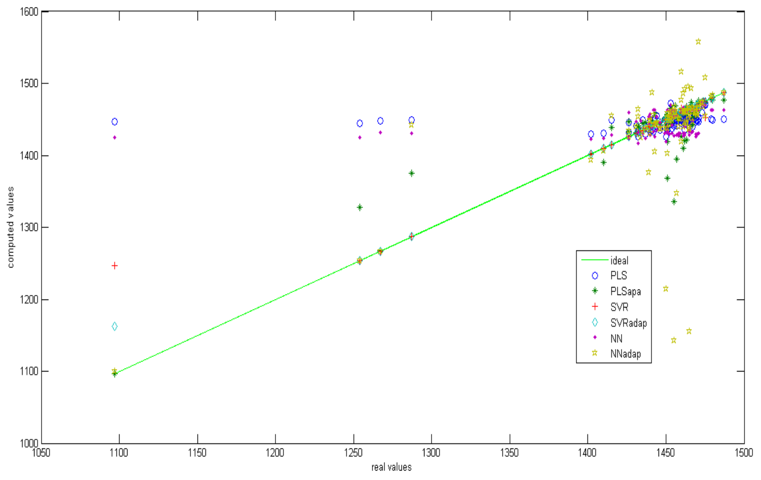

FIGURE 2. Computed and Predicted Results for stable data

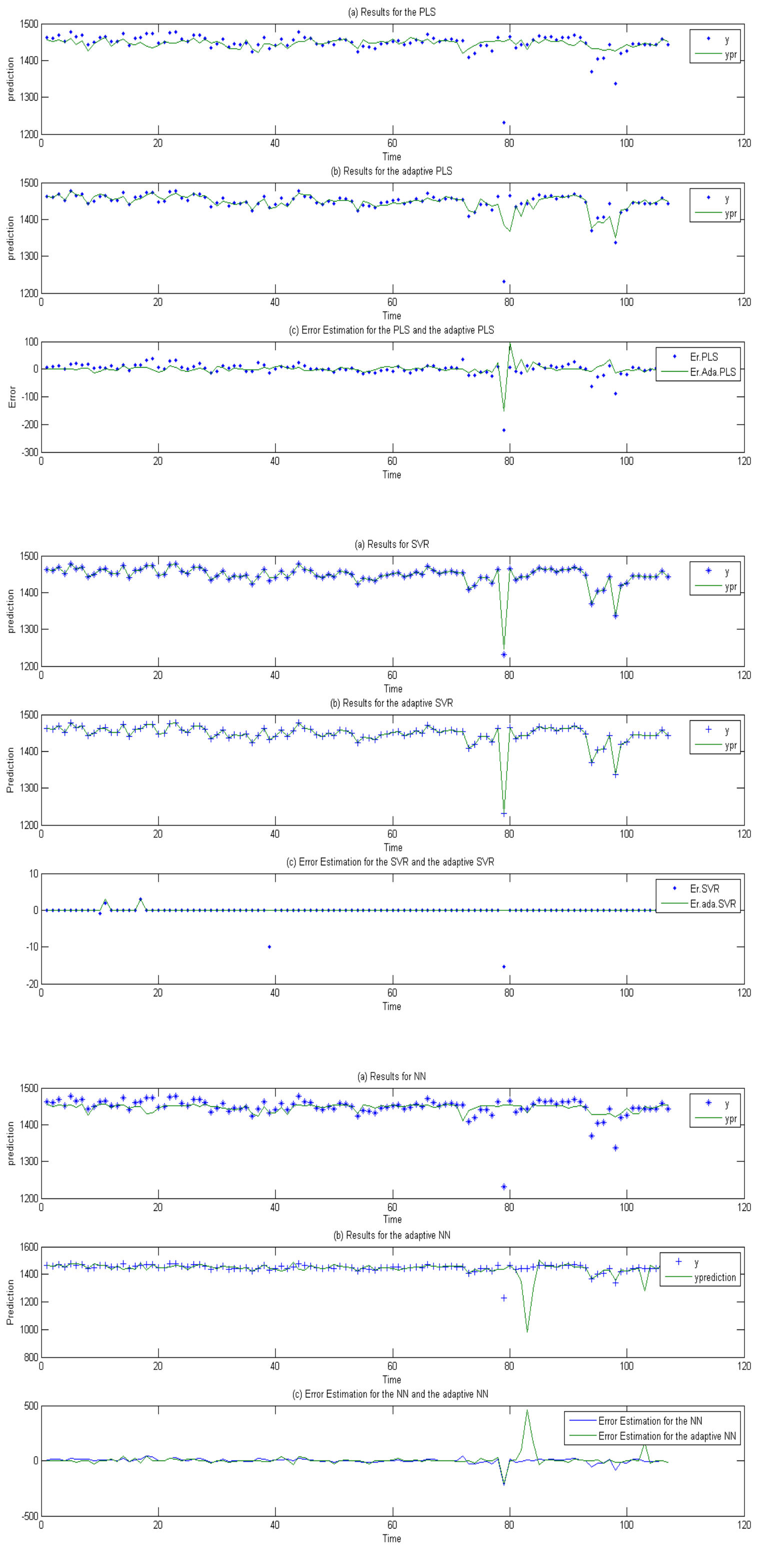

FIGURE 3. Prediction of the pig iron temperature and Errors

ACKNOWLEGEMENT

The authors are grateful and very thankful to the ArcelorMittal Annaba-Algeria for providing the real data to realise this modest work.

CONFLICT OF INTERESTS

The author(s) declare that there is no conflict of interests.

REFERENCES

[1] C. Gao, L. Jian, S. Luo, Modeling of the Thermal State Change of Blast Furnace Hearth With Support Vector Machines, IEEE Trans. Ind. Electron. 59 (2012), 1134–1145.

[2] Z. Ge, Z. Song, S.X. Ding, B. Huang, Data Mining and Analytics in the Process Industry: The Role of Machine Learning, IEEE Access. 5 (2017), 20590–20616.

[3] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Nat. Acad. Sci. 79 (1982), 2554–2558.

[4] J. Xiao, C. Wei, Y. Liu, Speed estimation of traffic flow using multiple kernel support vector regression, Physica A: Stat. Mech. Appl. 509 (2018), 989–997.

[5] L. Jian, C. Gao, Z. Xia, A Sliding-window Smooth Support Vector Regression Model for Nonlinear Blast Furnace System, Steel Res. Int. 82 (2011), 169–179.

[6] R. Östermark, H. Saxén, VARMAX-modelling of blast furnace process variables, Eur. J. Oper. Res. 90 (1996), 85–101.

[7] T.S. Rao, M.M. Gabr, Introduction to bispectral analysis and bilinear time series models, Lecture Notes in Statistics, vol. 24, Springer, Berlin, 1984.

[8] H. Saxen, C. Gao, Z. Gao, Data-Driven Time Discrete Models for Dynamic Prediction of the Hot Metal Silicon Content in the Blast Furnace—A Review, IEEE Trans. Ind. Inf. 9 (2013), 2213–2225.

[9] X. Tang, L. Zhuang, C. Jiang, Prediction of silicon content in hot metal using support vector regression based on chaos particle swarm optimization, Expert Syst. Appl. 36 (2009), 11853–11857.

[10] H. Wold, Estimation of principal components and related models by iterative least squares, in: P.R. Krishnaiaah (Ed.), Multivariate Analysis, Academic Press, 1966, pp. 391-420.

[11] G. Zhang, B. Eddy Patuwo, M. Y. Hu, Forecasting with artificial neural networks:, Int. J. Forecast. 14 (1998), 35–62.

[12] Q.X. Zhang, Y. Tao, Z.B. Cui, The Temperature Prediction in Blast Furnace Base on Fuzzy Least Squares Support Vector Machine, Appl. Mech. Mater. 336–338 (2013), 566–569.

[13] X. Zhang, M. Kano, Y. Li, Locally weighted kernel partial least squares regression based on sparse nonlinear features for virtual sensing of nonlinear time-varying processes, Computers Chem. Eng. 104 (2017), 164–171.

[14] B. Zhou, H. Ye, H. Zhang, M. Li, Process monitoring of iron-making process in a blast furnace with PCA-based methods, Control Eng. Practice. 47 (2016) 1–14.

[15] P. Zhou, H. Song, H. Wang, T. Chai, Data-Driven Nonlinear Subspace Modeling for Prediction and Control of Molten Iron Quality Indices in Blast Furnace Ironmaking, IEEE Trans. Control Syst. Technol. 25 (2017), 1761–1774.