



Predictive modeling in a steelmaking process using optimized relevance vector regression and support vector regression

Simone Massulini Acosta¹ · Anderson Levati Amoroso¹ ·
Ângelo Márcio Oliveira Sant'Anna² · Osiris Canciglieri Junior³

Accepted: 20 March 2021 / Published online: 20 April 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

The existence of contaminants in metal alloys products is the main problem affecting the product quality, which is an important requirement for competitiveness in industries. This paper proposes the application of a relevance vector machine for regression (RVR) and a support vector machine for regression (SVR) optimized by a self-adaptive differential evolution algorithm for regression to model the phosphorus concentration levels in a steelmaking process based on actual data. In general, the appropriate choice of learning hyperparameters is a crucial step in obtaining a well-tuned RVM and SVM. To address this issue, we apply a self-adaptive differential evolution algorithm, which is an evolutionary algorithm for global optimization. We compare the performance of the RVR and SVR models with the ridge regression, multiple linear regression, model trees, artificial neural network, and random vector functional link neural network models. RVR and SVR models have smaller RMSE values and better performance than the other models. Our study indicates that the RVR and SVR models are adequate tools for predicting the phosphorus concentration levels in the steelmaking process.

Keywords Relevance vector regression · Support vector regression · Differential evolution · Process modeling · Steelmaking process

✉ Simone Massulini Acosta
smacosta@utfpr.edu.br

Anderson Levati Amoroso
andersonl@utfpr.edu.br

Ângelo Márcio Oliveira Sant'Anna
angelo.santanna@ufba.br

Osiris Canciglieri Junior
osiris.canciglieri@pucpr.br

¹ Academic Department of Electronics, Federal University of Technology of Paraná, Curitiba, Paraná, Brazil

² Department of Mechanical Engineering, Federal University of Bahia, Salvador, Bahia, Brazil

³ Industrial and Systems Engineering Graduate Program, Pontifical Catholic University of Paraná, Curitiba, Paraná, Brazil

1 Introduction

Ensuring product and process quality is a constant challenge in industrial organizations. One of the main impurities found in the steel making process is phosphorus. A high phosphorus content can considerably reduce the quality of steel alloys. Therefore, a method for process analysis and control to ensure high process reliability and quality is necessary (Barella et al. 2017).

Regression models can be used to predict output data based on various input data and explain the underlying phenomenon behind the collected data. Regression models are used for monitoring response variables as functions of one or more input variables.

Most statistical methods are parametric in that they make assumptions about the distributional properties and autocorrelation structure of the process parameters. Several distribution-free or nonparametric methods based on machine learning techniques have been proposed in the literature. These methods are nonparametric in that they do not need to assume specific probability distributions for implementation (Camci et al. 2008). Artificial neural networks (ANNs), support vector machines (SVMs), and relevance vector machines (RVMs) are the most commonly used machine learning techniques.

ANNs can be defined as information processing systems based on the behavior of the human nervous system (Vapnik 1998). Haykin (2009) suggested a decision rule to minimize the error in the training data based on the general induction principle. Mazumdar and Evans (2009) described modern steelmaking processes along with physical modeling, mathematical modelling, and applications of ANN and genetic algorithm. Ghaedi and Vafaei (2017) reviewed the applications of ANN, SVM, and adaptive neuro fuzzy inference system (ANFIS) for adsorption removal of dyes from aqueous solution.

In recent years, SVMs have been introduced as one of several kernel-based techniques available in the field of machine learning for classification, prediction, and other learning tasks (Vapnik 1998). Kernel-based methods are based on mapping data from the original input feature space to a kernel feature space of higher dimensionality and then solving a linear problem in the feature space (Schölkopf and Smola 2002). SVMs were first introduced by Vapnik for solving classification problems. Support vector techniques have since been extended to the domain of regression. These techniques are called support vector regression (SVR) (Vapnik 1998).

Applications of SVR to model different chemical and industrial processes have been presented in recent years. Ghaedi et al. (2014) proposed a multiple linear regression (MLR) and least square support vector regression (LS-SVM) method for modeling of methylene blue dye adsorption using copper oxide loaded on activated carbon. Zaidi (2015) proposed a unified data-driven model for prediction of the boiling heat transfer coefficient in a thermosiphon reboiler using SVR as the modeling method. Ghaedi et al. (2016b) studied the predictive ability of a hybrid SVR and genetic algorithm optimization model for the adsorption of malachite green onto multiwalled carbon nanotubes. Cheng et al. (2016) performed a study which suggested that the SVR model can provide an important theoretical and practical guide for experimental design and for controlling the tensile strength of graphene nanocomposites via rational process parameters. Ghaedi et al. (2016a) presented the application of least squares support vector regression (LS-SVR) and MLR for modeling removal of methyl orange onto tin oxide nanoparticles loaded on activated carbon and activated carbon prepared from *Pistacia atlantica* wood. Jia et al. (2017) proposed a mathematical model for optimizing the dividing wall column process with a combination of SVM and particle swarm optimization algorithm. Ghugare et al. (2017) performed a study that

utilized genetic programming, ANN, and SVR for developing nonlinear models to predict the carbon, hydrogen, and oxygen fractions of solid biomass fuels.

Tipping (2000) introduced the relevance vector machine (RVM), a Bayesian sparse kernel technique for regression and classification of functional forms identical to the SVMs. The RVM for regression, called relevance vector regression (RVR), constitutes an approximation that can be used to solve nonlinear regression models.

In recent years, some applications of RVR to model and predict industrial processes applied to different areas of engineering have been reported. Zhang et al. (2015) utilized an RVM to estimate the remaining useful life of a lithium-ion battery based on denoised data. He et al. (2017) presented a new fault diagnosis method based on RVM to handle small-sample data. The results showed the validity of the proposed method. Liu (2017) utilized just-in-time (JIT) and RVM for soft-sensor modeling. The proposed methodologies were successfully applied for predicting hard-to-measure variables in wastewater treatment plants. Verma et al. (2017) utilized three different kernel-based models (SVR, RVR, and Gaussian process regression) to predict the compressive strength of cement. The performance of SVR and RVR was found to be comparable to that of ANN. Imani et al. (2018) examined the applicability and capability of extreme learning machine (ELM) and RVR models for predicting sea level variations and compared their performances with the SVR and ANN models. The results showed that the ELM and RVR models outperformed the other methods.

The performance of RVR and SVR models depends heavily on the choice of the hyperparameters. In actual applications, many practitioners select the hyperparameters in RVR and SVR empirically by trial and error or use a grid search technique together with a cross-validation method. Apart from consuming enormous amounts of time, such procedures for selecting the hyperparameters may not result in the best performance. Here, we use a differential evolution algorithm to optimize the RVR and SVR hyperparameters with different kernel functions. Differential evolution (DE) is a variant of evolutionary algorithms proposed by Storn and Price (1997).

There is no consistent methodology for determining the control parameters in DE (scale factor F , crossover rate Cr , and population size Np). These parameters are frequently arbitrarily set within predefined ranges. The control parameters are, in general, key factors affecting the convergence of DE (Price et al. 2006). Das et al. (2016) summarized and organized the current developments in DE, and presented recent proposals for parameter adaptation in DE algorithms.

This work is an applied study to predict the phosphorus concentration levels in the manufacture of FeMnMC in a Brazilian steelmaking company. For the same process, Pedrini and Caten (2010) have developed seven models with the MLR, and Acosta et al. (2016) have developed a multilayer perceptron (MLP) neural network to predict the phosphorus concentration levels.

The main objectives of this study are to apply RVR and SVR techniques to predict the phosphorus concentration level in the steelmaking process. In addition, we applied a DE algorithm to optimize the RVR and SVR hyperparameters with different kernel functions. To the best of our knowledge, no previous studies have analyzed the use of RVR and SVR combined with a self-adaptive DE approach to predict phosphorus concentration levels in the steelmaking process.

The main contributions of this paper can be summarized as follow: (i) RVR, and SVR models are proposed for the predictive modeling of phosphorus concentration levels in a steelmaking process with actual data, (ii) a self-adaptive DE algorithm is utilized to optimize the RVR and SVR hyperparameters with different kernel functions, and (iii)

the performance of the RVR and SVR models are compared with ridge regression, MLR, model trees, ANN, and random vector functional link (RVFL) neural network.

The rest of this paper is organized as follows: Section 2 presents the fundamental theory of SVR, RVR, and DE; Section 3 presents the proposed monitoring strategy; and Section 4 presents an applied study with a brief description of the steelmaking process, the implementation of the models, and a comparison with the ridge regression, MLR, model trees, ANN, and RVFL. Finally, Section 5 presents the conclusions of the study.

2 Background

2.1 Support vector regression

This section describe the fundamental theory of SVR. For more details on SVM, please refer to Vapnik (1998), Kecman (2001), Schölkopf and Smola (2002), Smola and Schölkopf (2004) and Cherkassky and Mulier (2007). Basak et al. (2007) have reviewed the existing theory, methods, developments, and scope of SVR. The principle idea in SVR is to compute a linear regression function in a high-dimensional feature space which the input data are mapped into via a nonlinear function.

The construction of SVR uses the ϵ -insensitive loss function proposed by Vapnik (1998),

$$|y - f(x)|_\epsilon = \begin{cases} 0 & \text{if } |y - f(x)| \leq \epsilon \\ |y - f(x)| - \epsilon & \text{otherwise} \end{cases} \tag{1}$$

where y is the measure (target) value and $f(x)$ is the predicted value.

Vapnik’s ϵ -insensitive loss function in Eq. (1) defines a tube with radius ϵ fitted to the data, called the ϵ -tube. Consider training data $\{(x_1, y_1), \dots, (x_N, y_N)\}$, where \mathfrak{X} denotes the space of the input. A linear function $f(x)$ can be written in the form of (Smola and Schölkopf 2004)

$$f(x) = \langle \omega, x \rangle + b \text{ with } \omega \in \mathfrak{X}, b \in \mathfrak{R} \tag{2}$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product in \mathfrak{X} and b is the bias.

The problem can be written as a convex optimization problem formulated using slack variables (ξ_i and ξ_i^*) to measure the deviation of the training samples outside the ϵ -insensitive zone (Vapnik 1998),

$$\text{minimize } \frac{1}{2} \|\omega\|^2 + C \left(\sum_{n=1}^N (\xi_i + \xi_i^*) \right) \text{ subject to } \begin{cases} y_i - \langle \omega, x_i \rangle - b \leq \epsilon + \xi_i \\ \langle \omega, x_i \rangle + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{3}$$

The regularization parameter C influences the tradeoff between the approximation error and the weight vector norm $\|\omega\|$. Fig. 1 illustrates the SVR model. According to Smola and Schölkopf (2004), only the points outside the ϵ -tube (shaded region) contribute to the cost, insofar as the deviations are penalized linearly.

The optimization problem in Eq. (3) can be transformed into a dual problem utilizing Lagrange multipliers (α_n, α_n^*) (Vapnik 1998; Smola and Schölkopf 2004)

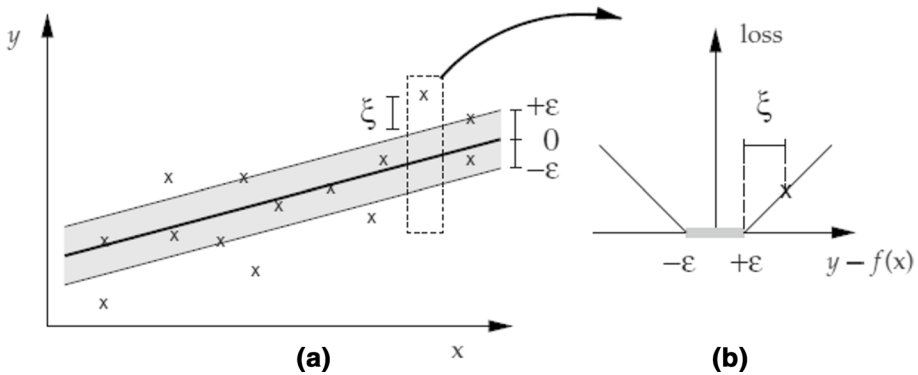


Fig. 1 SVR model. (a) ε -tube (shaded region) and slack variables ξ_i , (b) ε -insensitive loss function (Schölkopf and Smola 2002)

$$\begin{aligned} & \text{maximize} \begin{cases} -\frac{1}{2} \sum_{n,j=1}^N (\alpha_n - \alpha_n^*) (\alpha_j - \alpha_j^*) k(x_n, x_j) \\ -\varepsilon \sum_{n=1}^N (\alpha_n + \alpha_n^*) + \sum_{n=1}^N y_n (\alpha_n - \alpha_n^*) \end{cases} & (4) \\ & \text{subject to} \sum_{n=1}^N (\alpha_n - \alpha_n^*) = 0 \text{ and } \alpha_n, \alpha_n^* \in [0, C] \end{aligned}$$

$$f(x) = \sum_{n=1}^N (\alpha_n - \alpha_n^*) k(x_n, x) + b \tag{5}$$

where α_n and α_n^* are Lagrange multipliers, $k(x_n, x)$ is a kernel function and b is the bias. The support vectors (SVs) are the points that appear with nonzero coefficients in Eq. (5). Therefore, SVR has a sparse solution (Schölkopf and Smola 2002).

The kernel function $k(x_n, x)$ in Eq. (5) is a symmetric function satisfying Mercer’s conditions and is defined as a linear dot product of the nonlinear mapping (Vapnik 1998). A nonlinear function is learned by a linear learning machine in the kernel-induced feature space, while the capacity of the system is controlled by a parameter that does not depend on the dimensionality of the space.

Table 1 shows the kernel functions and their parameters used in this study. u is the parameter needed for polynomial and sigmoid-type kernels, d is the degree of the

Table 1 Kernel functions

Kernel	Equation	Kernel parameters
Linear	$k(x_n, x) = x_n^T x$	–
Polynomial	$k(x_n, x) = (\gamma x_n^T x + u)^d$	γ, u, d
Laplacian	$k(x_n, x) = \exp\left(-\frac{1}{2r^2} \ x_n - x\ \right) = \exp(-\gamma \ x_n - x\)$	γ
Gaussian radial basis function (RBF)	$k(x_n, x) = \exp\left(-\frac{1}{2r^2} \ x_n - x\ ^2\right) = \exp(-\gamma \ x_n - x\ ^2)$	γ
Sigmoid	$k(x_n, x) = \tanh(\gamma x_n^T x + u)$	γ, u

polynomial, $\gamma = 1/2r^2$ and $r > 0$ is a parameter that defines the kernel width. The kernel parameters are determined during the training phase.

The performance of SVR generalization depends on the correct specification of the free hyperparameters, namely, the value of the ϵ -insensitivity, the regularization parameter C , and the kernel parameters. Usually, the kernel type is first selected by the user based on the properties of the application data, and then the SVR hyperparameters are selected using some computational or analytic approaches.

Cherkassky and Ma (2004) summarized many practical approaches for setting the values of the regularization parameter C and the ϵ -insensitivity. They proposed the analytical selection of the C parameter directly from the training data, the analytical selection of the ϵ parameter based on the (known or estimated) level of noise in the training data and the (known) number of training samples, and the selection of the RBF kernel width parameter to reflect the input range of the training/test data.

2.2 Relevance vector regression

In this section, the fundamental theory of relevance vector regression is introduced. For more details on RVM, readers can refer to Tipping (2000, 2001), Schölkopf and Smola (2002), Tipping and Faul (2003), and Bishop (2006).

The approach uses a dataset of input and output (target) pairs $\{x_n, t_n\}_{n=1}^N$ follows a probabilistic formulation and assumes $p(t_n|x) = \mathcal{N}(t_n|f(x_n), \sigma^2)$, where the notation specifies a Gaussian distribution over t_n with mean $f(x_n)$ and variance σ^2 . The approach considers functions similar in type to those implemented by SVM, i.e., (Tipping 2000),

$$f(x) = \sum_{n=1}^N \omega_n k(x_n, x) + \omega_0 \tag{6}$$

where ω_n are the model weights, $k(x_n, x)$ is a kernel function and ω_0 is the bias. In this study, we use the kernel functions given in Table 1.

The RVR is a Bayesian treatment of Eq. (6). RVR adopts a fully probabilistic framework and introduces a prior on the model weights governed by a set of hyperparameters, each of which is associated with a weight and whose most probable values are iteratively estimated from the data. The likelihood estimation of the dataset can then be written as (Tipping 2001),

$$p(\mathbf{t}|\boldsymbol{\omega}, \sigma^2) = (2\pi\sigma^2)^{-N} \exp\left\{-\frac{1}{2\sigma^2} \|\mathbf{t} - \boldsymbol{\Phi}\boldsymbol{\omega}\|^2\right\} \tag{7}$$

where $\mathbf{t} = (t_1, \dots, t_N)^T$, $\boldsymbol{\omega} = (\omega_0, \dots, \omega_N)^T$ and $\boldsymbol{\Phi}$ is the $N \times (N + 1)$ ‘design’ matrix with $\boldsymbol{\Phi} = [\phi(x_1), \phi(x_2), \dots, \phi(x_N)]^T$, wherein $\phi(x_n) = [1, k(x_n, x_1), k(x_n, x_2), \dots, k(x_n, x_N)]^T$.

According to Tipping (2000), the maximum-likelihood estimation of $\boldsymbol{\omega}$ and σ^2 from Eq. (7) will result in severe overfitting. Here, he prefer to use smoother (less complex) functions by defining a zero-mean Gaussian prior distribution over the weights. The introduction of an individual hyperparameter α_n for each weight parameter ω_n is the key feature of RVR. Thus, the weight prior takes the form of

$$p(\boldsymbol{\omega}|\boldsymbol{\alpha}) = \prod_{n=0}^N \mathcal{N}(\omega_n|0, \alpha_n^{-1}) \tag{8}$$

where α is a vector of $N + 1$ hyperparameters and α_n represents the precision of the corresponding parameter ω_n (Bishop 2006). The marginal likelihood for the hyperparameters is obtained by integrating the weights (Tipping 2001)

$$p(\mathbf{t}|\alpha, \sigma^2) = (2\pi)^{-N/2} \left| \sigma^2 \mathbf{I} + \phi \mathbf{A}^{-1} \phi^T \right|^{-1/2} \exp \left\{ -\frac{1}{2} \mathbf{t}^T (\sigma^2 \mathbf{I} + \phi \mathbf{A}^{-1} \phi^T)^{-1} \mathbf{t} \right\} \quad (9)$$

where $\mathbf{A} = \text{diag}(\alpha_0, \alpha_1, \dots, \alpha_N)$.

The values of α and σ^2 are determined using type-II maximization likelihood, in which the marginal likelihood function is maximized by integrating out the weight parameters. In the RVR method, a proportion of the hyperparameters $\{\alpha_n\}$ is driven to large values. The weight parameters ω_n corresponding to these hyperparameters thus have posterior distributions with means and variances both equal to zero (Bishop 2006). Thus, these parameters are removed from the model, and sparsity is realized. In the case of models with the form of Eq. (6), the inputs x_n corresponding to the remaining nonzero weights are called the relevance vectors (RVs) and are analogous to the support vectors (SVs) of a SVR.

According to Tipping (2000), some advantages of RVRs over the SVRs are: (i) they can produce probabilistic output, (ii) there is no need to define the regularization parameter C and the insensitivity parameter ε , and (iii) non-Mercer kernel functions can be used. The most compelling feature of the RVR is that it is capable of generalization performance comparable to that of an equivalent SVR using, in most cases, significantly smaller number of RVs than the number of SVs used by an SVR to solve the same problem. More significantly, in RVR, the parameters governing complexity and noise variance (α 's and σ^2) are automatically estimated by the learning procedure, whereas in SVR, it is necessary to tune the hyperparameters C and ε (Tipping 2001; Tipping and Faul 2003; Bishop 2006).

2.3 Differential evolution

The differential evolution (DE) algorithm, proposed by Storn and Price (1997), is an evolutionary algorithm (EA) for global optimization, which has been widely applied in many scientific and engineering fields (Qin et al. 2009).

The DE algorithm involves the three main operations of mutation, crossover, and selection (Storn and Price 1997). DE is a scheme for generating trial parameter vectors. Mutation and crossover are used to generate new vectors (trial vectors), and selection then determines which of the vectors will survive into the next generation.

The original version of DE can be defined by the following constituents (Storn 2008):

- Population: DE is a population-based optimizer that attacks the starting point problem by sampling the objective function at multiple, randomly chosen initial points. DE aims to evolve the population of Np D -dimensional vectors, which encodes the g th generation candidate solutions, towards the global optimum (Price et al. 2006).
- Once the population is initialized, DE mutates and recombines the population to produce a population of Np trial vectors. The scale factor, $F \in (0, 1+)$, is a positive real number that controls the rate at which the population evolves (Price et al. 2006).
- Following the mutation operation, crossover is applied to the population. The crossover probability, $Cr \in [0, 1]$, is a user-defined value that controls the fraction of parameter values copied from the mutants (Price et al. 2006).

- Selection: If the trial vector has an equal or lower objective function value than that of its target vector it replaces the target vector in the next generation $g+1$; otherwise, the target retains its place in the population for at least one more generation.

The mutation strategies can vary with the type of individual modified to form the donor vector, the number of individuals considered for the disorder and the type of crossing used. The mutation strategy is denoted by $DE/\xi/\beta/\delta$, where (Santos et al. 2012):

- ξ denotes the vector to be disturbed,
- β determines the number of weighted differences,
- δ denotes the crossover type.

The setting of the DE control parameters is crucial for the performance of the algorithm. According to Storn and Price (1997), DE is much more sensitive to the choice of scale factor F than it is to the choice of crossover probability Cr .

According to Eiben et al. (2007), there are two major approaches for setting the parameter values: parameter tuning and parameter control. Parameter tuning is a commonly practiced approach that tries to find good values for the parameters before the algorithm runs and then runs the algorithm using these values, which remain fixed during the run. In parameter control, the values for the parameters are changed during the run. The methods for changing the values of the parameter can be classified into one of three categories: deterministic parameter control, adaptive parameter control, and self-adaptive parameter control.

In self-adaptive parameter control, the parameters to be adapted are encoded into chromosomes and undergo mutation and recombination. Better values of these encoded parameters lead to better individuals, which are in turn more likely to survive and produce offspring and hence propagate these better parameter values.

The algorithm proposed in Brest et al. (2006), the jDE algorithm, employs a self-adaptive scheme to perform the automatic setting of the scale factor F and crossover rate Cr control parameters. The control parameter population size Np does not change during the run. The algorithm implements the $DE/rand/1/bin$ mutation strategy.

In our study, we use the jDE algorithm proposed by Brest et al. (2006) and implemented by Conceição and Mächler (2015). The latter implementation differs from the DE algorithm proposed by Brest et al. (2006) most notably in the use of the $DE/rand/1/either-or$ mutation strategy (Price et al. 2006) and a combination of jitter with dither (Storn 2008), and the immediate replacement of each worse parent in the current population by its newly generated better or equal offspring (Babu and Angira 2006) instead of updating the current population with all the new solutions simultaneously as in classical DE.

3 Proposed modeling strategy

In actual applications, many practitioners select the RVR and SVR hyperparameters empirically by trial and error or by using a grid search (exhaustive search) technique with a cross-validation method. These procedures are computationally intensive and may not result in the best performance. Choosing the optimal values for the RVR and SVR hyperparameters

is important in obtaining accurate modeling results. In this study, we apply a self-adaptive DE algorithm to optimize the RVR and SVR hyperparameters for modeling the phosphorus concentration levels in the steelmaking process.

Fig. 2 shows the flowchart for implementing the RVR and SVR models optimized by the DE algorithm. The procedure is as follows:

Step 1: Collect the database of the process, select the variables, normalize the observations, and divide the dataset into the training and test datasets.

Step 2: Select the training dataset. Select the kernel function and set the initial RVR kernel parameters or free SVR hyperparameters (C , ϵ , and kernel parameters).

Step 3: Set the population size N_p ($N_p = 10 \times n_p$) in the self-adaptive DE algorithm, where n_p is the number of RVR or SVR parameters, and set the stopping criterion: maximum number of iterations ($200 \times n_p$) and tolerance (1×10^{-7}).

Step 4: Train the RVR or SVR and calculate the fitness function value. The fitness function is defined as the root mean square error (RMSE) and the objective is to minimize the RMSE

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} = \sqrt{\frac{1}{n} \sum_{i=1}^n (e_i)^2} \quad (10)$$

where y_i is the observed value measured during the process, \hat{y}_i is the predicted value estimated by the model, e_i is the residual, and n is the number of observations used in fitting the model.

Step 5: If the maximum number of iterations or tolerance for the stopping criterion is reached, the smallest RMSE is selected, and the best parameter estimates of RVR or SVR are output.

Step 6: Train RVR or SVR with the best parameter estimates and obtain the optimized model.

Step 7: Obtain the predicted values for the training and test datasets using the optimized RVR or SVR model.

Step 8: Using the predicted values and residuals, perform a performance analysis of the model.

Step 9: If the model is valid, the RVR or SVR model to predict the phosphorus concentration levels in the steelmaking process is obtained.

We also use the grid search technique and cross-validation method to select the optimal parameters for the RVR and SVR models with the Gaussian RBF and Laplacian kernel functions. Because SVR has three free hyperparameters (C , ϵ , and kernel parameter γ) to tune, the use of a grid search technique with a cross-validation method allows the selection of the optimal parameters that have the smallest mean squared error (MSE). RVR has a kernel parameter γ to be tuned, for which we use a cross-validation method to select the optimal parameter that has the smallest MSE.

Regression models with good fits present little discrepancies between the observed and predicted values. The adequacy of a model is also an essential aspect because the relation between the response and the factors should be significant and independent of the number or type of input variables. The standard regression model assumes that the residuals are

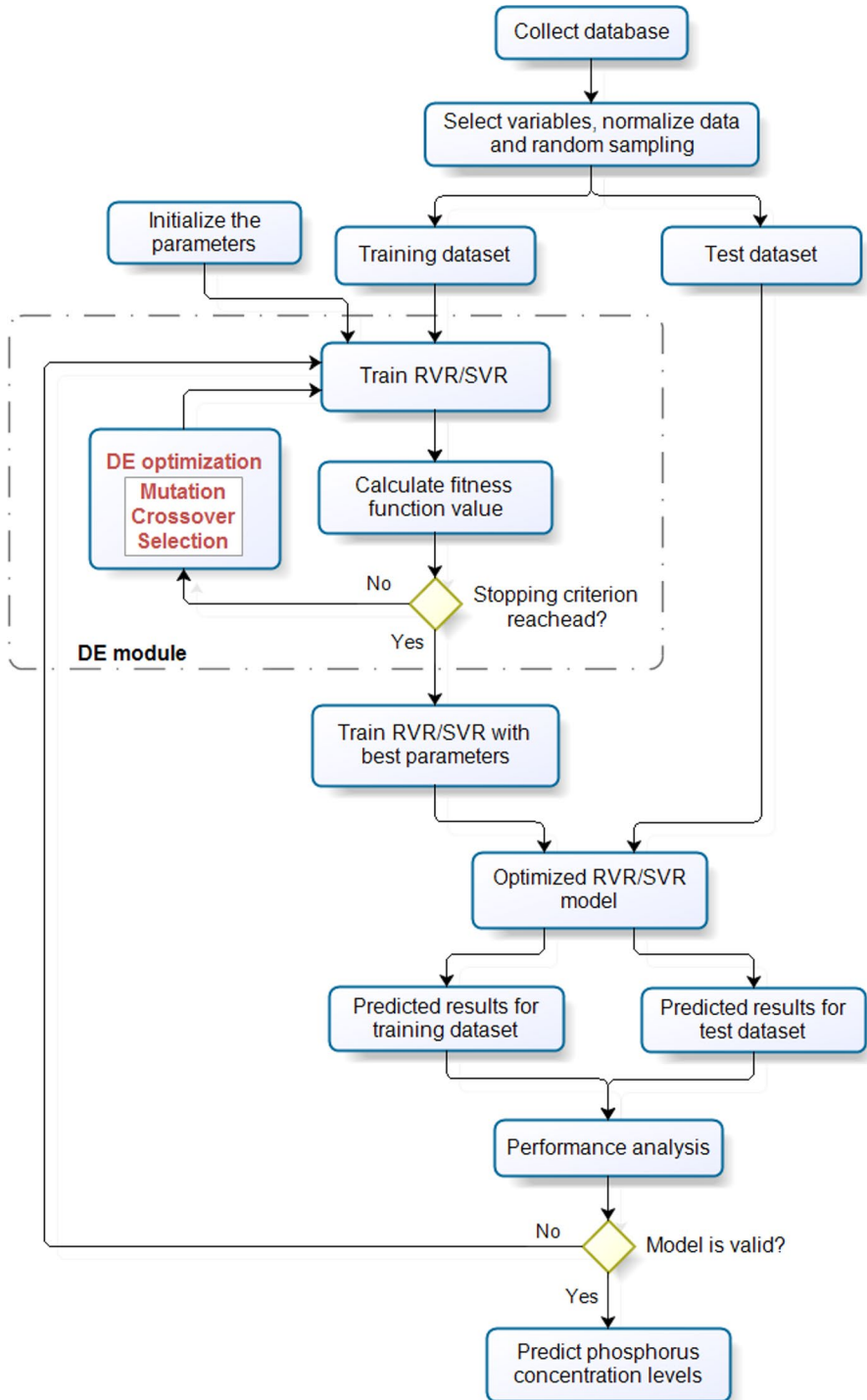


Fig. 2 Flowchart to implement RVR and SVR models optimized by DE algorithm

independent and identically distributed (i.i.d) normal random variables with zero mean and constant variance.

To evaluate the generalization capacity of the models, the following error minimization strategies are used: the RMSE (Eq. 10), the mean squared error (MSE), the mean absolute error (MAE), and the mean absolute percentage error (MAPE). The latter three are given by Eq. (11) to Eq. (13), where y_i is the observed value, \hat{y}_i is the predicted value and n is the number of observations.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (11)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (12)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{\hat{y}_i} \right| \times 100 \quad (13)$$

4 Applied study

This section presents the implementation of the models. First, we briefly describe the case study of the phosphorus concentration levels in the steelmaking process. Next, we present the implementation of the RVR and SVR models described in Section 3. We also compare the RVR and SVR models developed in this study with the ridge regression, MLR, model tree, ANN, and RVFL models. Simulations and calculations were performed with the open-source software R® (R 2018). The SVR implementation used LIBSVM, a library for support vector machines (Chang and Lin 2011). All programs ran on a personal computer with an Intel Core i7-2670QM, 2.2 GHz, 8 GB DDR3-1333 SDRAM, Windows 7 Professional 64-bit.

4.1 Case study

The implementation of the models is illustrated through an applied study for modeling the phosphorus concentration levels in the steelmaking process for Medium-Carbon Ferromanganese (FeMnMC). The study was carried out in a Brazilian steelmaking company. One of the main factors affecting the product quality in steelmaking companies is the existence of contaminants in alloy steel.

The refining process in the study uses high-purity oxygen to reduce the carbon level in High-Carbon Ferromanganese (FeMnHC) originating from FeMnMC, which has a higher market value. During this process, changes occur in the proportion of several elements, including that of phosphorus in the final product.

Phosphorus (P) is one of the main contaminants that interferes with the steelmaking processes. Ferromanganese alloys are the major sources of P contamination during the steelmaking process, which requires limited use of this type of alloy during the process

Table 2 Variables related to the steelmaking process

Group	Specifications	Unit
Alloy composition	Si*, P*, Fe*, Ti*, C*, O ₂	%
Slag composition	MgO, MnO, CaO, SiO ₂ , Al ₂ O ₃ , BaO, K ₂ O, TiO ₂ , FeO	wt-%
Load composition	Initial, Liquid, Dross	Nm ³
Alkalinity levels	BB, BQ, BO	Nm ³
Output	P	%

(Um et al. 2014). Increased phosphorus levels can significantly affect the physical aspects of alloy steel and severely compromise its quality. P-rich steel compounds usually exhibit: (i) increased hardness, (ii) decreased ductility, (iii) ghost lines in carbon-rich alloy steels, and (iv) increased frailty of steel bonds at high and low temperatures (Chaudhary et al. 2001).

The FeMnMC steelmaking process has 21 initial input variables that are relevant for modeling the dephosphorization process. These input variables were grouped as follow: (i) composition of FeMnHC alloys used as raw materials for the converter, (ii) composition of slag, (iii) composition of loads, and (iv) levels of alkalinity: binary, quaternary e optical basicity. The output variable is the proportion of phosphorus (P) in the final process of FeMnMC steelmaking. The selected database covers a sample of 257 observations. Table 2 shows the variables related to the steelmaking process.

Pedrini and Caten (2010) developed seven MLR models to predict the phosphorus concentration level in this process. The refining process of ferromanganese consists of a decarburization reaction between liquid metal and oxygen injected in the metallic bath. To realize the dephosphorization process, CaO is dissolved during decarburization to reduce the proportion of phosphorus in the final product.

Pedrini and Caten (2010) adopted the suggestion of an engineer from the company to developed the MLR model called Model 8. This model uses the natural logarithm of the difference between the phosphorus concentration in the final process for FeMnMC and the phosphorus concentration of FeMnHC, the raw material in the refining process. The MLR (Model 8) model found to predict the phosphorus concentration levels is

$$\ln(P - P^*) = -0.804 \ln(Fe^*) + 0.371 \ln(MnO) - 0.656 \ln(CaO) \quad (14)$$

For the same process, Acosta et al. (2016) have developed a MLP neural network to predict the phosphorus concentration levels. For the ANN model, they used an MLP network with 11 neurons in one hidden layer, the logistic activation function, the learning rate of 0.01 and the momentum rate of 0.1. The ANN gave a RMSE of 0.0151986 on the training dataset.

4.2 Implementation of the models

To develop this study, we used a database created from the information system of the company. This database contains all the variables related to the FeMnMC steelmaking process, as shown in Table 2.

The data preprocessing phase to for the RVR and SVR models consisted of correlation analysis between the process input variables and the phosphorus concentration in the final

process by applying the lasso method (Tibshirani 1996). The input variables selected are as follows: percentage of initial phosphorus in the alloy composition (P^*), percentage of initial carbon in the alloy composition (C^*), percentage of manganese oxide in the slag composition (MnO), percentage of calcium oxide in the slag composition (CaO), and liquid volume in the load composition (Liquid). The output variable is the phosphorus (P) concentration in the final process.

The 257 observations were normalized into the interval $[0, 1]$. The observation set was then randomly divided into two parts: a training dataset composed of 205 (80%) observations and a test dataset composed of the remaining 52 (20%) observations. The training dataset was used to estimate the regression models representing the phosphorus concentration in the actual steelmaking process, and the test dataset was used to evaluate and compare the predictive power of the regression models.

For the RVR and SVR models, we used the kernel functions in Table 1. A flowchart of the RVR and SVR parameter selection is presented in Fig. 2. For the optimization tasks using the self-adaptive DE algorithm (Conceição and Mächler 2015), we used the RMSE as the fitness function of the training dataset, Eq. (10).

The search space of the control parameters for the RVR is $\gamma \in [0.001;1]$, $u \in [0;10]$ and $d \in [1;5]$. To tune the SVR parameters with the RBF kernel using the training data, we first used the procedure proposed by Cherkassky and Ma (2004) to obtain the three free hyperparameters (C , ε and RBF kernel parameter γ) and identify the best search region. The search space of the control parameters for the SVR is: $C \in [1;50]$, $\varepsilon \in [0.001;1]$, $\gamma \in [0.001;1]$, $u \in [0;10]$ and $d \in [1;5]$.

Table 3 shows the best RVR and SVR model parameters for the kernel functions, where DE represents optimization by the self-adaptive DE algorithm and GS represents selection by the 10-fold cross-validation method. We used the CPU running time (seconds) to evaluate the speed to select the hyperparameters in the RVR and SVR models. According to results listed in Table 3, the CPU time was reduced when we used a DE algorithm to optimize the RVR and SVR hyperparameters. The tuning of RVR involves only the kernel parameters, whereas SVR has more parameters for tuning (C , ε and kernel parameters). Because of this, the CPU times of RVR models have smaller values of the CPU times of SVR models.

From Table 3, the DE-RVR Laplacian kernel has a smaller RMSE value than the other RVR models and DE-SVR RBF kernel has a smaller value of RMSE than those of other SVR models. We observe that the DE-RVR Laplacian kernel performed slightly better than the DE-SVR RBF kernel, but the DE-RVR Laplacian kernel produced a smaller number of RVs (10) compared to the number of SVs (135) in the DE-SVR RBF kernel. The number of SVs is 65.8% of the training dataset, which can be considered as an indication of the goodness of fit and the adequacy of the model because a large number of SVs can cause overfitting of the model. The number of RVs is 4.9% of the training dataset, and the number of SVs is nearly thirteen times greater than the number of RVs.

From Table 3, it can be seen that the: (i) DE-RVR Laplacian and DE-RVR RBF kernels have smaller values of RMSE than the GS-RVR Laplacian and GS-RVR RBF kernels, (ii) the DE-SVR Laplacian and DE-SVR RBF kernels have smaller RMSE values than the GS-SVR Laplacian and GS-SVR RBF kernels, (iii) GS-RVR Laplacian and GS-RVR RBF kernels performed slightly better than the GS-SVR Laplacian and GS-SVR RBF kernels, and (iv) RVR Linear has the greatest value of RMSE.

Based on the error indices (Table 3), the selected RVR model is the DE-RVR Laplacian kernel with the optimal kernel parameter $\gamma = 0.002224$. The number of RVs is 10, and the RMSE on the training dataset is 0.0137368. The SVR model selected is the DE-SVR RBF

Table 3 Best RVR and SVR model parameters

	Model	Kernel parameters						CPU time (s)	
		γ	d	u	C	ϵ	RVs/ SVs(%)		RMSE
RVR	RVR Linear	–	–	–	–	–	7 (3.4)	0.0292070	1.66
	DE-RVR Poly-nomial	0.019960	1	5.6122	–	–	3 (1.5)	0.0144930	147.72
	DE-RVR Laplacian	0.002224	–	–	–	–	10 (4.9)	0.0137368	83.10
	DE-RVR RBF	0.003713	–	–	–	–	7 (3.4)	0.0142890	119.97
	DE-RVR Sigmoid	0.045197	–	0.4746	–	–	4 (1.9)	0.0150930	95.21
	GS-RVR Laplacian	0.002824	–	–	–	–	9 (4.4)	0.0143510	739.41
	GS-RVR RBF	0.002510	–	–	–	–	19 (9.2)	0.0144207	945.26
SVR	DE-SVR Linear	–	–	–	1.6382	0.2537	135 (65.8)	0.0151627	57.63
	DE-SVR Poly-nomial	0.02227	1	3.2336	44.6386	0.2529	136 (66.3)	0.0151633	1402.60
	DE-SVR Laplacian	0.01854	–	–	6.7758	0.2656	138 (67.3)	0.0142609	700.58
	DE-SVR RBF	0.04175	–	–	3.3651	0.3449	135 (65.8)	0.0140385	726.60
	DE-SVR Sigmoid	0.00551	–	0.0014	15.8328	0.3422	114 (55.6)	0.0151339	969.14
	GS-SVR Laplacian	0.00503	–	–	7.7224	0.2445	144 (70.2)	0.0148802	1559.25
	GS-SVR RBF	0.01050	–	–	4.0025	0.0300	138 (67.3)	0.0147601	1229.04

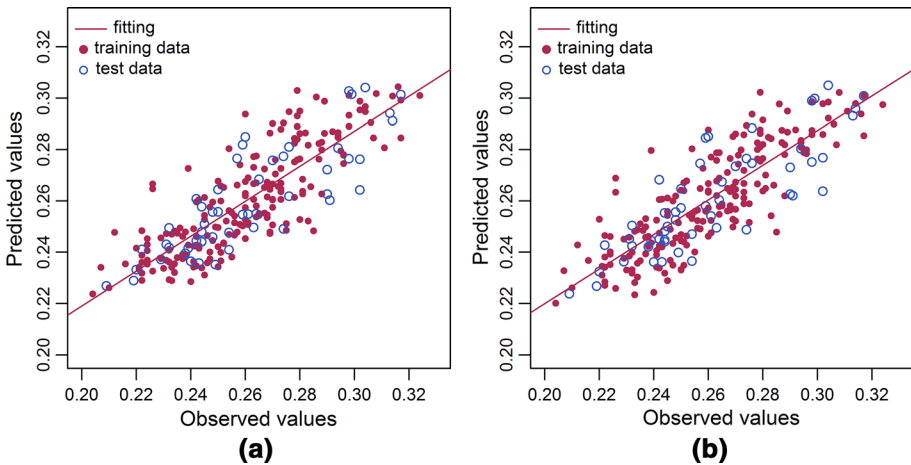
kernel with the optimal parameters $C = 3.3651$, $\epsilon = 0.3449$ and $\gamma = 0.04175$. The number of SVs is 135, and the RMSE on the training dataset is 0.0140385. These optimized RVR and SVR models were used to model the phosphorus concentration levels in the steelmaking process.

In this study, the performance of regression models was evaluated using both residual analysis and error minimization strategies. We tested the normality of the residuals of the fitted models using the Shapiro–Wilk test for the training data, and obtained a p -value higher than 0.4 for the two models, which indicates that the residuals are normally distributed. We examined the autocorrelation of the residuals using the Durbin-Watson test for the training data, and the results indicate no significant correlations in the residuals of the models. We used the Levene test to check homoscedasticity and obtained a p -value higher than 0.5, which means that residuals can be considered as having constant variances. After these tests, we concluded that the residuals of the RVR and SVR models are independent and identically distributed (i.i.d) normal random variables with constant variances. This is an evidence for the goodness of the fits, and shows that the models are appropriate for the observations. Therefore, the models can be utilized to predict the phosphorus concentration levels in the final process.

Table 4 shows the statistical properties of the phosphorus concentration levels obtained by applying the models on the training and test datasets. The statistical properties obtained from the models were found to be similar to those obtained experimentally. Fig. 3 shows

Table 4 Statistical properties of the phosphorus concentration levels predicted from models

	Phosphorus concentration levels		
	Experimental	RVR	SVR
Maximum	0.3240	0.3044	0.3045
Minimum	0.2040	0.2237	0.2186
Mean	0.2599	0.2597	0.2596
Median	0.2590	0.2554	0.2582
Standard deviation	0.02546	0.02045	0.02031
Variance	0.0006483	0.0004185	0.0004125

**Fig. 3** Predicted values against observed values for the training and test datasets: (a) RVR model, (b) SVR model

the predicted values against the observed values for the training and test datasets for these models. These goodness of fit graphs confirm the good predictive performance of the models.

Compared with traditional ANNs, SVR possesses some advantages: it has high generalization capability and avoids local minima, it always has a solution, does not need the network topology to be determined in advance, and it has a simple geometric interpretation and provides a sparse solution. SVR provides good performance when the model parameters are well tuned. The disadvantages of SVR are that it requires the tuning of many model parameters, and the results obtained are not probabilistic (Wang et al. 2003; Tipping 2000).

There are some advantages associated with RVR. The generalization performance of RVR is comparable to an equivalent SVR. Furthermore, RVR produces probabilistic output, and there is no need to tune the regularization parameter C and the insensitivity parameter ϵ necessary in SVR. RVR yields sparse models with fewer relevance vectors (Tipping 2000).

Ridge regression (RR) is one of the methods to shrink the coefficients of correlated predictors towards each other (Marquardt and Snee 1975). The lambda (λ) parameter is the regularization penalty, and a cross-validation method can be used to select λ (Friedman,

Hastie and Tibshirani 2010). We also used RR to model the phosphorus concentration levels using the training dataset. The λ is 0.001811864, and the RMSE on the training dataset is 0.0151184. The test dataset was used with the RR model to predict the future values of the phosphorus concentration levels.

Pao et al. (1994) proposed a random vector functional link (RVFL) neural network. The RVFL is an extension of single layer feedforward neural (SLFN) networks with additional direct connections from the input layer to the output layer (Qiu et al. 2018). The RVFL network has a set of nodes called enhancement nodes, which are equivalent to the neurons in the hidden layer in the conventional SLFN. In RVFL the actual values of the weights from the input layer to hidden layer are randomly generated in a suitable domain and kept fixed in the learning stage (Zhang and Suganthan 2015). The number of enhancement nodes (hidden neurons) were determined by a cross-validation method in order to avoid overfitting. The number of enhancement nodes is 8, and the RMSE on the training dataset is 0.0146759.

Model trees (MT) use recursive partitioning to build a piecewise linear model in the form of a model tree (Quinlan 1992). The idea is to split the training cases in much the same way as when growing a decision tree, using a criterion of minimizing intra-subset variation of class values rather than maximizing information gain. M5 (Quinlan 1992) builds tree-based models but, whereas regression trees (Breiman et al. 1984) have values at their leaves, the trees constructed by M5 can have multivariate linear models. In this work, we used the M5 rule based model with boosting and corrections based on nearest neighbors in the training dataset (Quinlan 1993, Fernández-Delgado et al. 2019). The M5 tunable hyperparameters were selected by a cross-validation method. The number of training committees is 2, the number of neighbors for prediction is 0, and the RMSE on the training dataset is 0.0152490.

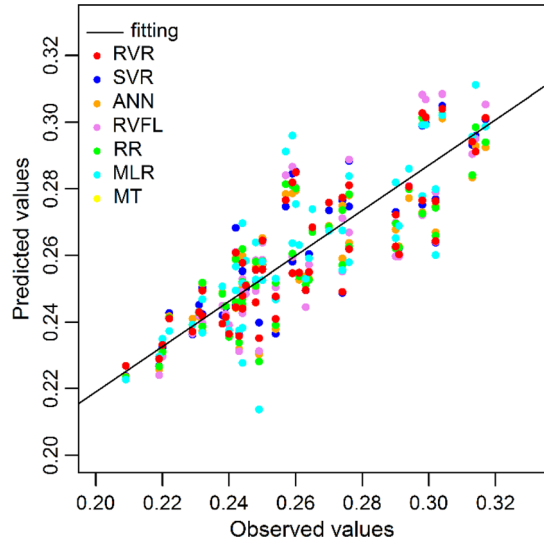
In order to compare the models with the MLR model proposed by Pedrini and Caten (2010), the test dataset was used to predict future values of the phosphorus concentration levels. The coefficient models were estimated by the least square method based on the t-student statistical test at 5%, Eq. (14). The RMSE of the training dataset is equal to 0.0161804.

Table 5 summarizes the statistical measures of the error minimization results. Fig. 4 shows the predicted values against the observed values on the test dataset for the RVR, SVR, ANN, RVFL, RR, MLR, and MT models. Fig. 4 confirms that the RVR and SVR models have better performance than the other models.

Table 5 Statistical measures of error minimization results

Test dataset				
Model	MSE	RMSE	MAE	MAPE(%)
RVR	0.000222690	0.0149220	0.0121490	4.614
SVR	0.000223762	0.0149594	0.0118265	4.522
ANN	0.000238135	0.0152254	0.0122523	4.670
RVFL	0.000236267	0.0153710	0.0123570	4.682
RR	0.000236665	0.0153834	0.0125313	4.811
MLR	0.000247714	0.0157389	0.0127920	4.851
MT	0.000259726	0.0161160	0.0131768	5.061

Fig. 4 Predicted values against observed values for the test dataset for the models



We analyzed the results from Table 3, Table 4, Table 5, Fig. 3, and Fig. 4. We can observe that the RVR, SVR, ANN, RVFL, RR, MLR, and MT models achieved good performance in the predicting the phosphorus concentration levels. In Fig. 3, we note that there is a substantial agreement between the training results and the test results, indicating that there are no overfitting problems with the RVR and SVR models.

Analyzing the results in Table 5 from the test dataset, it can be seen that the:

- (i) RVR model has smaller values of MSE, and RMSE than the other models,
- (ii) SVR model has smaller values of MAE, and MAPE than the other models;
- (iii) The ascending order of RMSE is:

$$\text{RVR} < \text{SVR} < \text{ANN} < \text{RVFL} < \text{RR} < \text{MLR} < \text{MT}$$
- (iv) The ascending order of MAE is:

$$\text{SVR} < \text{RVR} < \text{ANN} < \text{RVFL} < \text{RR} < \text{MLR} < \text{MT}$$
- (v) the machine learning techniques (RVR and SVR) have better performance than the statistical methods (RR and MLR).

Statistical tests are employed to give a detail analysis about the performance differences among all the regression models. Parametric tests assume a series of hypotheses on the data on which they are applied (independence, normality, and homoscedasticity). If such assumptions do not hold, the reliability of the tests is not guaranteed. Nonparametric tests do not assume particular characteristics for the underlying data distribution. Nonparametric tests can perform two classes of analysis: pairwise comparisons and multiple comparison (Derrac et al. 2011, Latorre et al. 2020).

The Friedman test is a nonparametric test analogue of the parametric two-way analysis of variance (Garcia et al. 2010). To calculate the statistic, the Friedman test ranks the model performance for each problem and compute the average of each model between problems (Carrasco et al. 2020). The null-hypothesis states that all the models have the same performance. Once Friedman's test rejects the null hypothesis, we can proceed

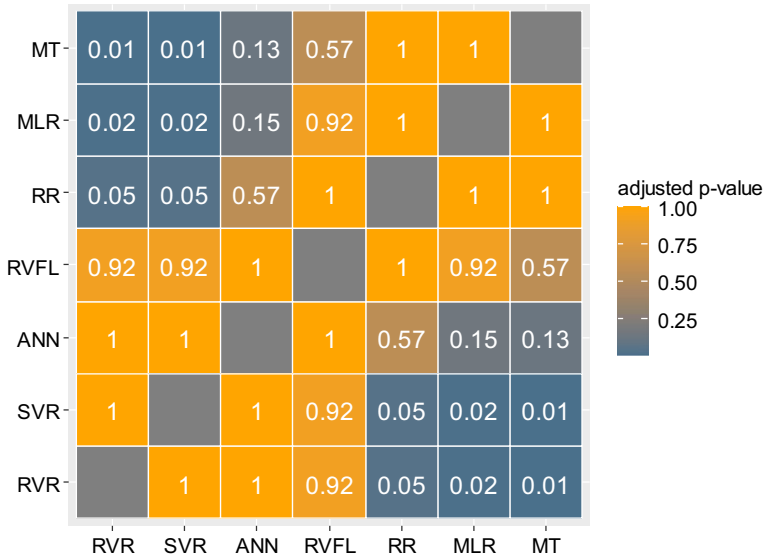


Fig. 5 Adjusted p-values using the Bergmann-Hommel post-hoc test for multiple comparisons

with the Bergmann-Hommel post-hoc test in order to find the pairs of models which produce differences ($N \times N$ comparisons) (Derrac et al. 2011).

We used the Friedman test with the seven models (RVR, SVR, ANN, RVFL, RR, MLR, and MT) and the p-value reported by this test is 0.0033, which is significant at the significance level ($\alpha = 0.05$). Then, we proceed to perform the Bergmann-Hommel post-hoc test in order to determine the location of the differences between these models.

Fig. 5 shows the adjusted p-values using the Bergmann-Hommel post-hoc test for multiple comparisons. The null hypothesis is rejected if the adjusted p-value is less than the significance level ($\alpha = 0.05$). The p-values below 0.05 indicate that the respective models differ significantly in prediction errors. We can observe that are significant differences between the RVR and MLR, RVR and MT, SVR and MLR, SVR and ML. The difference is not significant between RVR, SVR, ANN and RVFL.

5 Conclusions

The impurities in the metal alloys interfere with the steelmaking process. High levels of phosphorus can severely affect the physical integrity of steel bonds and threaten the quality of the final product.

In this work, we applied relevance vector machine for regression (RVR) and support vector machine for regression (SVR) optimized by a self-adaptive differential evolution algorithm to the predictive modeling of phosphorus concentration levels in a steelmaking process based on actual data.

In the past decade, relevance vector machines have gained the attention of many researchers. Relevance vector machine (RVM) is a Bayesian sparse kernel technique for regression and classification of identical functional form to the support vector machine (SVM). The RVR and SVR generalization performance depends on the correct

specification of the hyperparameters. One of the most widely used approaches to select the RVR and SVR hyperparameters is the grid search technique with a cross-validation method. Differential evolution (DE) has also been used to optimize the RVR and SVR hyperparameters. It is essential to choose the best control parameters for DE to achieve the optimal algorithm performance. Thus, we used a self-adaptive scheme to tune the DE parameters automatically.

We used five kernel functions and applied a self-adaptive DE algorithm to optimize the RVR and SVR hyperparameters. Based on the error indices, the RVR model selected is the DE-RVR Laplacian kernel and the SVR model selected is the DE-SVR RBF kernel.

We compared the performance of the RVR and SVR models with the RR, MLR, ANN, MT, and RVFL models. The comparative analysis shows that RVR and SVR have better performance than the RR, MLR, ANN, MT, and RVFL models in the predicting the phosphorus concentration levels in the steelmaking process.

We used the Friedman test and Bergmann-Hommel post-hoc test. We can observe that there are significant differences between the RVR and MLR, RVR and MT, SVR and MLR, SVR and ML. The difference is not significant between RVR, SVR, ANN and RVFL.

RVR has slightly better performance than the other models. RVR has nearly the same performance as SVR, but RVR produced nearly thirteen times fewer RVs than the SVs produced by SVR. Furthermore, the tuning of RVR involves only the kernel parameters, whereas SVR has more parameters for tuning (C , ϵ and kernel parameters).

The results of this study indicate that the RVR and SVR models are adequate tools for predicting the phosphorus concentration levels in the steelmaking process. The proposed approach provides an effective strategy to support practitioners in modeling other chemical and industrial processes.

Declarations

Conflict of interest None.

References

- Acosta, S. M., & Sant'Anna, A.M.O., & Canciglieri Jr., O. (2016). Forecasting modeling for energetic efficiency in an industrial process. *Chemical Engineering Transactions.*, 52, 1081–1086. <https://doi.org/10.3303/CET1652181>.
- Babu, B. V., & Angira, R. (2006). Modified differential evolution (MDE) for optimization of nonlinear chemical processes. *Computers and Chemical Engineering*, 30, 989–1002. <https://doi.org/10.1016/j.compchemeng.2005.12.020>.
- Barella, S., Mapelli, C., Mombelli, D., Gruttadauria, A., Laghi, E., Ancona, V., & Valentino, G. (2017). Model for the final decarburization of the steel bath through a self-bubbling effect. *Ironmaking & Steelmaking*. <https://doi.org/10.1080/03019233.2017.1405179>.
- Basak, D., Pal, S., & Patranabis, D. C. (2007). Support vector regression. *Neural Information Processing – Letters and Reviews*, 11(10), 203–224.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
- Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Wadsworth.
- Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10, 646–657. <https://doi.org/10.1109/TEVC.2006.872133>.

- Camci, F., Chinnam, R. B., & Ellis, R. D. (2008). Robust kernel distance multivariate control chart using support vector principles. *International Journal of Production Research*, 46(18), 5075–5095. <https://doi.org/10.1080/00207540500543265>.
- Carrasco, J., García, S., Ruedab, M. M., Dasc, S., & Herrera, F. (2020). Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: practical guidelines and a critical review. *Swarm and Evolutionary Computation*, 54, 1–20. <https://doi.org/10.1016/j.swevo.2020.100665>.
- Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(27), 1–27. <https://doi.org/10.1145/1961189.1961199>.
- Chaudhary, P. N., Goel, R. P., & Roy, G. G. (2001). Dephosphorisation of high carbon ferromanganese using BaCO₃ based fluxes. *Ironmaking and Steelmaking*, 28(5), 396–403. <https://doi.org/10.1179/irs.2001.28.5.396>.
- Cheng, W. D., Cai, C. Z., Luo, Y., Li, Y. H., & Zhao, C. J. (2016). Modeling and predicting the tensile strength of poly (lactic acid)/graphene nanocomposites by using support vector regression. *International Journal of Modern Physics B*, 30(10), 1–12. <https://doi.org/10.1142/S0217979216500521>.
- Cherkassky, V., & Ma, Y. (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, 17, 113–126. [https://doi.org/10.1016/S0893-6080\(03\)00169-2](https://doi.org/10.1016/S0893-6080(03)00169-2).
- Cherkassky, V., & Mulier, F. (2007). *Learning from data: concepts, theory, and methods*. (2nd ed.). John Wiley & Sons.
- Conceição, E., & Mächler, M. (2015). DEoptimR: differential evolution optimization in pure R. R package version 1.0-8.
- Das, S., Mullick, S. S., & Suganthan, P. N. (2016). Recent advances in differential evolution – An updated survey. *Swarm and Evolutionary Computation*, 27, 1–30. <https://doi.org/10.1016/j.swevo.2016.01.004>.
- Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1), 3–18. <https://doi.org/10.1016/j.swevo.2011.02.002>.
- Eiben, A., Michalewicz, Z., Schoenauer, M., & Smith, J. (2007). Parameter control in evolutionary algorithms. In Lobo, F. G., Lima, C. F. and Michalewicz, Z. (eds), *Parameter setting in evolutionary algorithms*, 54(54), Springer Verlag, 19–46. <https://dx.doi.org/https://doi.org/10.1007/978-3-540-69432-8>.
- Fernández-Delgado, M., Sirsat, M. S., Cernadas, E., Alawadi, S., Barro, S., & Febrero-Bande, M. (2019). An extensive experimental survey of regression methods. *Neural Networks*, 111, 11–34. <https://doi.org/10.1016/j.neunet.2018.12.010>.
- Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 1–22.
- Garcia, S., Fernández, A., Luengo, J., & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. *Information Sciences*, 180(10), 2044–2064. <https://doi.org/10.1016/j.ins.2009.12.010>.
- Ghaedi, A. M., & Vafaei, A. (2017). Applications of artificial neural networks for adsorption removal of dyes from aqueous solution: a review. *Advances in Colloid and Interface Science*, 245, 20–39. <https://doi.org/10.1016/j.cis.2017.04.015>.
- Ghaedi, M., Ghaedi, A. M., Hossainpour, M., Ansari, A., Habibi, M. H., & Asghari, A. R. (2014). Least square-support vector (LS-SVM) method for modeling of methylene blue dye adsorption using copper oxide loaded on activated carbon: Kinetic and isotherm study. *Journal of Industrial and Engineering Chemistry*, 20(4), 1641–1649. <https://doi.org/10.1016/j.jiec.2013.08.011>.
- Ghaedi, M., Rahimi, M. R., Ghaedi, A. M., Shilpi Agarwal, I. J., & Gupta, V. K. (2016). Application of least squares support vector regression and linear multiple regression for modeling removal of methyl orange onto tin oxide nanoparticles loaded on activated carbon and activated carbon prepared from Pistacia atlantica wood. *Journal of Colloid and Interface Science*, 461, 425–434. <https://doi.org/10.1016/j.jcis.2015.09.024>.
- Ghaedi, M., Dashtian, K., Ghaedi, A. M., & Dehghanian, N. (2016). A hybrid model of support vector regression with genetic algorithm for forecasting adsorption of malachite green onto multi-walled carbon nanotubes: central composite design optimization. *Physical Chemistry Chemical Physics*, 18, 13310–13321. <https://doi.org/10.1039/c6cp01531j>.
- Ghugare, S. B., Tiwary, S., & Tambe, S. S. (2017). Computational intelligence based models for prediction of elemental composition of solid biomass fuels from proximate analysis. *International Journal of System Assurance Engineering and Management*, 8(4), 2083–2096. <https://doi.org/10.1007/s13198-014-0324-4>.

- Haykin, S. (2009). *Neural networks and learning machines*. (3rd ed.). Prentice Hall.
- He, S., Xiao, L., Wang, Y., Liu, X., Yang, C., Lu, J., Gui, W., & Sun, Y. (2017). A novel fault diagnosis method based on optimal relevance vector machine. *Neurocomputing*, 267, 651–663. <https://doi.org/10.1016/j.neucom.2017.06.024>.
- Imani, M., Kao, H.-C., Lan, W.-H., & Kuo, C.-Y. (2018). Daily sea level prediction at Chiayi coast, Taiwan using extreme learning machine and relevance vector machine. *Global and Planetary Change*, 161, 211–221. <https://doi.org/10.1016/j.gloplacha.2017.12.018>.
- Jia, S., Qian, X., & Yuan, X. (2017). Optimal design for dividing wall column using support vector machine and particle swarm optimization. *Chemical Engineering Research and Design*, 125, 422–432. <https://doi.org/10.1016/j.cherd.2017.07.028>.
- Kecman, V. (2001). *Learning and soft computing: support vector machines, neural networks, and fuzzy logic models*. MIT Press.
- Latorre, A., Molina, D., Osaba, E., Del Ser, J., & Herrera, F. (2020). Fairness in bio-inspired optimization research: a prescription of methodological guidelines for comparing meta-heuristics. *Neural and Evolutionary Computing*.
- Liu, Y. (2017). Adaptive just-in-time and relevant vector machine based soft-sensors with adaptive differential evolution algorithms for parameter optimization. *Chemical Engineering Science*, 172, 571–584. <https://doi.org/10.1016/j.ces.2017.07.006>.
- Marquardt, D. W., & Snee, R. D. (1975). Ridge regression in practice. *The American Statistician*, 29(1), 3–20. <https://doi.org/10.2307/2683673>.
- Mazumdar, D., & Evans, J. W. (2009). *Modeling of Steelmaking Processes*, first ed., CRC Press.
- Pao, Y.-H., Park, G.-H., & Sobajic, D. J. (1994). Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing*, 6(2), 163–180. [https://doi.org/10.1016/0925-2312\(94\)90053-1](https://doi.org/10.1016/0925-2312(94)90053-1).
- Pedrin, D. C., & Caten, C. S. (2010). Modelagem estatística para a previsão do teor de fósforo em ligas de ferromanganês. *Revista Ingepro*, 2, 14–25.
- Peng, H., & Ling, X. (2015). Predicting thermal-hydraulic performances in compact heat exchangers by support vector regression. *International Journal of Heat and Mass Transfer*, 84, 203–213. <https://doi.org/10.1016/j.ijheatmasstransfer.2015.01.017>.
- Price, K., Storn, R., & Lampinen, J. (2006). *Differential evolution: a practical approach to global optimization*. Springer-Verlag.
- Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 13, 398–417. <https://doi.org/10.1109/TEVC.2008.927706>.
- Qiu, X., Suganthan, P. N., & Amaratunga, & G.A.J. (2018). Ensemble incremental learning Random Vector Functional Link network for short-term electric load forecasting. *Knowledge-Based Systems*, 145, 182–196. <https://doi.org/10.1016/j.knosys.2018.01.015>.
- Quinlan, J. R. (1992). Learning with continuous classes. *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, 343–348.
- Quinlan, J. R. (1993). Combining instance-based and model-based learning. *Proceedings of the Tenth International Conference on Machine Learning*, 236–243.
- R (2018). R: a language and environment for statistical computing. R Foundation for statistical computing, ISBN 3-900051-07-0. Available at <http://www.r-project.org>.
- Santos, G. S., Luvizotto, L. G. J., Mariani, V. C., & Coelho, L. S. (2012). Least squares support vector machines with tuning based on chaotic differential evolution approach applied to the identification of a thermal process. *Expert Systems with Applications*, 39, 4805–4812. <https://doi.org/10.1016/j.eswa.2011.09.137>.
- Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels - support vector machines, regularization, optimization and beyond*. The MIT Press.
- Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14(3), 199–222. <https://doi.org/10.1023/B:STCO.0000035301.49549.88>.
- Storn, R. (2008). Differential evolution research - trends and open questions. In Chakraborty, U. K. (Eds), *Advances in differential evolution*. *Studies in Computational Intelligence*, 143 (pp. 11–12). https://doi.org/https://doi.org/10.1007/978-3-540-68830-3_1.
- Storn, R., & Price, K. (1997). Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4), 341–359. <https://doi.org/10.1023/A:1008202821328>.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B (Methodological)*, 267–288.

- Tipping, M. (2000). The relevance vector machine. In S. A. Solla, T. K. Leen, & K. -R. Müller (Eds.), *Advances in neural information processing systems*, 12 (pp. 652–658). MIT Press.
- Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1, 211–244. <https://doi.org/10.1162/15324430152748236>.
- Tipping, M. E., & Faul, A. C. (2003). Fast marginal likelihood maximization for sparse Bayesian models. In C. M. Bishop & B.J. Frey (Eds.), *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics* (pp. 3–6).
- Um, H., Lee, K., Kim, K.-Y., Shin, G., & Chung, Y. (2014). Effect of carbon content of ferromanganese alloy on corrosion behavior of MgO-C refractory. *Ironmaking and Steelmaking*, 41(1), 31–27. <https://doi.org/10.1179/1743281212Y.0000000098>.
- Vapnik, V. N. (1998). *Statistical learning theory*. . John Wiley & Sons.
- Verma, M., Thirumalaiselvi, A., & Rajasankar, J. (2017). Kernel-based models for prediction of cement compressive strength. *Neural Computing & Applications*, 28(1), 1083–1100. <https://doi.org/10.1007/s00521-016-2419-0>.
- Wang, W., Xu, Z., Lu, W., & Zhang, X. (2003). Determination of the spread parameter in the Gaussian kernel for classification and regression. *Neurocomputing*, 55(3–4), 643–663. [https://doi.org/10.1016/S0925-2312\(02\)00632-X](https://doi.org/10.1016/S0925-2312(02)00632-X).
- Zaidi, S. (2015). Novel application of support vector machines to model the two-phase boiling heat transfer coefficient in a vertical tube thermosiphon reboiler. *Chemical Engineering Research and Design*, 98, 44–58. <https://doi.org/10.1016/j.cherd.2015.04.002>.
- Zhang, C., He, Y., Yuan, L., Xiang, S., & Wang, J. (2015). Prognostics of lithium-ion batteries based on wavelet denoising and DE-RVM. *Computational Intelligence and Neuroscience*. <https://doi.org/10.1155/2015/918305>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.