24th COBEM - 2017

Associação Brasileira de Engenharia e Ciências Mecânicas

24th ABCM International Congress of Mechanical Engineering
December 3-8, 2017, Curitiba, PR, Brazil

COBEM-2017-1264
EVOLUTIONARY SUPPORT VECTOR REGRESSION APPROACH APPLIED TO BACKSTROKE START PERFORMANCE MODELLING

Karla de Jesus
Centre of Research, Education, Innovation and Intervention in Sport, Faculty of Sport, University of Porto, Porto, Portugal; Porto Biomechanics Laboratory (LABIOMEP), University of Porto, Porto, Portugal; Human Performance Studies Laboratory, Faculty of Physical Education and Physiotherapy, Federal University of Amazon, Manaus, Amazon, Brazil; Human Motor Behavior Studies Laboratory, Faculty of Physical Education and Physiotherapy, Federal University of Amazon, Manaus, Amazon, Brazil
karladejesus@ufam.edu.br

Carlos Eduardo Klein
Leandro dos Santos Coelho
Industrial and Systems Engineering Graduate Program, Pontifical Catholic University of Paraná; Electrical Engineering Graduate Program, Federal University of Paraná, Curitiba, Paraná, Brazil
carlos.klein@outlook.com.br, leandro.coelho@pucpr.br

Kelly de Jesus
Centre of Research, Education, Innovation and Intervention in Sport, Faculty of Sport, University of Porto, Porto, Portugal; Porto Biomechanics Laboratory (LABIOMEP), University of Porto, Porto, Portugal; Human Performance Studies Laboratory, Faculty of Physical Education and Physiotherapy, Federal University of Amazon, Manaus, Amazon, Brazil; Human Motor Behavior Studies Laboratory, Faculty of Physical Education and Physiotherapy, Federal University of Amazon, Manaus, Amazon, Brazil
kellydejesus@ufam.edu.br

Viviana Cocco Mariani
Mechanical Engineering Graduate Program, Pontifical Catholic University of Paraná; Electrical Engineering Graduate Program, Federal University of Paraná, Curitiba, Paraná, Brazil
viviana.mariani@pucpr.br

Leandro José Rodrigues Machado
Centre of Research, Education, Innovation and Intervention in Sport, Faculty of Sport, University of Porto, Porto, Portugal; Porto Biomechanics Laboratory (LABIOMEP), University of Porto, Porto, Portugal
lmachado@fade.up.pt

Mário Augusto Pires Vaz
Porto Biomechanics Laboratory (LABIOMEP), University of Porto, Porto, Portugal; Institute of Mechanical Engineering and Industrial Management, Faculty of Engineering, University of Porto, Porto, Portugal
gmavaz@fe.up.pt

Ricardo Jorge Pinto Fernandes
João Paulo Vilas-Boas
Centre of Research, Education, Innovation and Intervention in Sport, Faculty of Sport, University of Porto, Porto, Portugal; Porto Biomechanics Laboratory (LABIOMEP), University of Porto, Porto, Portugal
ricfer@fade.up.pt , jpvb@fade.up.pt

Abstract. *Support Vector Machine (SVM) is one of the fastest growing methods of machine learning due to its good generalization ability and good convergence performance. SVM is a maximum margin model, which is based on structural risk minimization rather than empirical risk minimization. Originally, SVM developed for solving the classification problems but latter, Support Vector Regression (SVR) evolved from the SVM for doing regression tasks. A SVR model is, in essence, a machine learning method of non-parametric estimation especially aiming at samples with limited sizes. The principle of structural risk minimization makes SVR to have stronger generalization ability. Despite their advantages, SVR models require an accurate selection of the configuration parameters in order to achieve good generalization performance. To overcome this limitation, a hyperparameter selection method based on differential evolution (DE) optimizer was developed. The aim of this study is to develop an evolutionary SVR (E-SVR) approach based on DE selection method to model the backstroke start performance. SVR and the proposed E-SVR approach were applied and compared with other regression methods to predict 5 m backstroke start time using kinematic and kinetic variables and to determine the accuracy of the mean absolute percentage error. The results presented an excellent performance in terms of the prediction errors of both SVR and E-SVR approaches.*

Keywords: *support vector regression, evolutionary algorithm, differential evolution, nonlinear regression, kinematics, kinetics, competitive swimming.*

1. INTRODUCTION

Support vector machine (SVM) related to the Vapnik's statistical learning theory (Vapnik, 1995; Vapnik, 1997) was originally developed for classification problems and later extended to regression problems using a modified SVM version called support vector regression (SVR).

SVR has been successfully applied in various fields, such as smart grid load (Vrablecová et al., 2017) and carbon price forecasting (Zhu et al., 2017), traffic flow prediction (Cheng et al., 2017), urban growth modeling (Shafizadeh-Maghadam et al., 2017), and image processing (Zhang et al., 2016).

The classical SVR model is initially formulated as a linear technique and is subsequently generalized to the non-linear case via a transformation of the input space to another of greater size. It estimates a function by nonlinearly mapping the input space into a high dimensional hidden space and then running the linear regression in the output space. Thus, the linear regression in the output space corresponds to a nonlinear regression in the low dimensional input space. In addition, the theory denotes that if the dimensions of feature space (or hidden space) are high enough, SVR may approximate any nonlinear mapping relations. As the name implies, the design of the SVR hinges upon the extraction of a subset of the training data that serves as support vectors, which represent a stable characteristic of the data.

The structure of a SVR model is very similar to a three-layer artificial neural network with a hidden layer, the number and function of which equal to those of the support vectors. In other words, the number of hidden nodes equals to that of support vectors. It should be noticed that the whole model structure of SVR is adaptively generated directly, while in the former only the weights can be obtained automatically (Ruas et al., 2008).

SVR performance depends on the design parameters. Unfortunately, obtaining SVR parameter values is usually a tedious and lengthy process, particularly if the designer is unfamiliar with the parameter interactions within the chosen SVR model. Recently, as an alternative to the conventional mathematical approaches, modern heuristic optimization techniques, such as evolutionary algorithms (Simon, 2013) have been obtaining much attention due to their ability to find appropriate SVR designs (Zhang et al., 2011) that perform satisfactorily for a given nonlinear regression task.

Evolutionary algorithms are optimization metaheuristics that have yielded promising results for solving nonlinear, non-differentiable and multi-modal optimization problems. Due to its population-based nature, they can avoid being trapped in a local optimum, and, consequently, have the ability to find global optimal solutions. In this context, a promising paradigm is the differential evolution (DE) (Storn and Price, 1995, Storn and Price, 1997) that uses a rather greedy and less stochastic approach to problem solving compared to other evolutionary algorithms. It has been proven to be a powerful tool for solving the global optimization problems, especially with a no smooth objective function because its algorithm does not need the derivative information about the objective function. It has several potentialities: simple structure, easy use, convergence property, quality of solution and robustness. Due to these good features, the effectiveness of differential evolution optimizer has been successfully demonstrated in a variety of continuous optimization problems in many fields (see details in Das et al., 2016). In this work, an SVR model was then optimized through DE (E-SVR) to maximize its generalization performance in the backstroke swimming start performance modelling task.

Predicting backstroke start performance is crucial, since placing in short distance competitive swimming events has been decided by margins as small as hundredths of a second (de Jesus et al., 2014). It is well known that most of the researcher's attention has been given to the biomechanical study of the ventral start techniques. However, non-linear modeling techniques were only applied to the backstroke start technique and respective variants, evidencing its superiority when compared to the traditional linear models (de Jesus et al., 2018).

The remainder of this paper is organized as follows. Section 2 provides the fundamentals of the experimental procedure. The description of SVR and E-SVR is introduced in Section 3. Modelling results based on SVR and E-SVR are reported and discussed in Section 4 followed by conclusions and direction for the future research in Section 5.

2. EXPERIMENTAL PROCEDURE FOR THE BACKSTROKE START MODELLING

In terms of the backstroke start modelling, the adopted case study was experimentally obtained with 10 swimmers who randomly completed 8x15 m maximum backstroke start trials with feet over the wedge and hands on the highest horizontal and vertical handgrip as descripted by De Jesus et al. (2016). Swimmers were videotaped using a dual media camera set-up, as shown in Figure 1, with the starts being performed over an instrumented block with four force plates.

Six kinematic and seven kinetic variables were inputted for modeling and predicting the 5 m backstroke start time (output variable), namely: (i) horizontal force at starting signal, (ii) horizontal force and time before swimmer's hands-off, (iii) horizontal force and time at swimmer's hands-off, (vi) horizontal force and time before swimmer's take-off, (v) horizontal and vertical center of mass position at starting signal, (vi) horizontal and vertical center of mass position at first swimmer's hands water contact, (vii) entry angle, (viii) resultant velocity at swimmer's take-off. The backstroke start time was defined between the acoustic signal until swimmer's vertex achieves the 5 m mark.

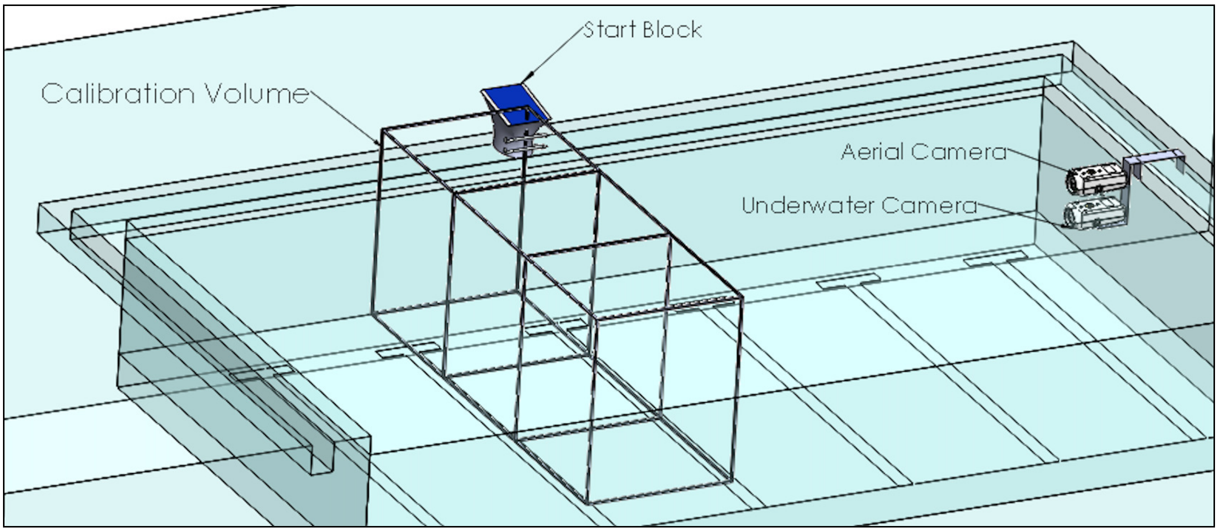


Figure 1. The experimental backstroke start set up.

3. FUNDAMENTALS OF SUPPORT VECTOR REGRESSION

SVR, which has been successfully applied to a variety of real-world problems, simultaneously minimizes the regularization error and empirical risk with a suitable penalty factor. Based on the theory of SVM, the SVR become a well-established method for design black-box models. In the following sub-sections, the procedures of the SVR and E-SVR are described.

3.1 The classical SVR approach

The SVR method evolved from the support vector classification, and the regression case is achieved by the introduction of the ε-insensitive loss function (Li at al., 2018). SVR uses the SVM methodology to regression analysis and can be applied to construct non-linear models by applying a "kernel trick" along with the SVM. The main idea of nonlinear regression of SVR is "kernel trick", which maps the input patterns (W denotes the input space) into a higher-dimensional space F by a function Φ: $W \rightarrow F$. Then, a simple multiple linear regression can be performed in the higher dimensional feature space while the results can be mapped down again into the input space. Once the full process is based on mapping into the higher dimensional space, linear regression and return to the input space, one is aware that it is a data-driven machine learning method.

The procedure of mapping the input space into the higher dimensional one can be made by applying a kernel function. Any function that accomplishes the Mercer condition can be used as a kernel function. There are some well-known base functions and in this paper the Gaussian radial basis function (RBF) is applied.

The adopted Gaussian RBF is given by

$$RBF\ Kernel, K(x, x_k) = e^{\left[\frac{-\|x - x_k\|^2}{2a^2}\right]}, \tag{1}$$

where σ is determines the width (variance) of the basis function and it is considered as design variable. Its graphical representation is according to the Fig. 2.

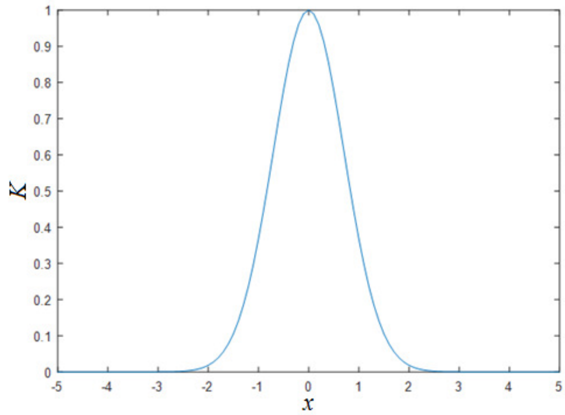


Figure 2. Radial basis function.

Taking the assumption that a set D = {(x_i , y_i)} represents the learning data to be used, where x_i stands for the input data and y_i are the corresponding output data for any i = 1, . . . , N where N is the total of available samples. All regression tasks try to find a matching function $f(x)$ that corresponds to the transfer function in between of x and y. If this relationship is linear then it is expected that:

$$f(x) = a.x + b \tag{2}$$

or in this specific case

$$f(x) =< w.x > +b, \tag{3}$$

where w stands for the weight coefficient and b is the constant coefficient.

Once the input space shall be mapped into a feature space, then a mapping function is introduced and represented by $\Phi(x)$:

$$f(x) =< w, \Phi(x) > +b. \tag{4}$$

Two slack variables are introduced to the equation in order to handle infeasible constraints and taking in assumption that there is a function, which can approximate every pair (x_i, y_i) with an acceptable precision, then the problem can be formulated as a convex optimization problem:

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\left(\xi_i + \xi_i^*\right). \tag{5}$$

$$\text{subject to} \begin{cases} \langle w, \Phi(x)\rangle + b - y_i \le \varepsilon + \xi_i, \\ y_i - \langle w, \Phi(x)\rangle - b \le \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \ge 0, \end{cases} \tag{6}$$

where C is a penalty factor and ξ_i and ξ_i^* are elements showing the difference between the target and estimated values.

The convex optimization problem dual formulation gives an easier way to solve, for that the saddle point condition is substituted and changing the representation $\Phi(.)$ by $K(x_i,x_j) = \Phi^T(x_i)\,\Phi(x_j)$ one shall have:

$$\min \frac{1}{2}\sum_{i,j=1}^{l}\left(\alpha_i^* - \alpha_i\right)\left(\alpha_j^* - \alpha_j\right)K(x_i, x_j) + \varepsilon\sum_{i=1}^{l}(\alpha_i^* + \alpha_i) - \sum_{i=1}^{l} y_i(\alpha_i^* - \alpha_i), \tag{7}$$

$$\text{subject to}\begin{cases}\sum_{\mathrm{i}=1}^{\mathrm{l}}(\alpha_{\mathrm{i}}^* - \alpha_{\mathrm{i}}) = 0,\\ 0 \le \alpha_{\mathrm{i}}, \alpha_{\mathrm{i}}^* \le C,\end{cases} \tag{8}$$

where α_i and α_i^* are Lagrange multipliers and the positive non-zero ones are called supporting vectors.

3.2 The proposed E-SVR approach

The model selection of SVR is concerned with two key issues: how to select an appropriate kernel function, and how to determine the optimal parameters of SVR. For the former, Gaussian radial basis function (RBF) is selected to build the SVR model, because RBF can yield good results in general. However, there are various types of kernel functions that could be used always depending on the aspects of the input training data such as sigmoid, spline, and polynomial. For the latter, we use the DE algorithm to seek the optimal parameters of SVR including the regularization parameter (C), the size of the ε-tube (insensitive loss function parameter) and the width (variance) of the basis function given by σ. The effectiveness and generalization of the SVR model highly depend on the accurate selection of these three control parameters (or hyper-parameters) (Duan et al., 2003).

The parameter $C \geq 0$ is the regularization parameter (or error penalty parameter) that adjusts the trade-off between the empirical error (approximation error) and the complexity (flatness) of the output function. Low values of C will result in simple (or flat) functions while higher values can lead to over-fitting on the input training data.

Another SVR hyper-parameter is ε, which defines the size of the ε-tube around the regression function, which influences the number of support vectors. All these parameters are user-defined parameters. Various methods are proposed for selecting these parameters until now (Sapankevych and Sankar, 2009; Yang et al., 2017). It can be said that the most popular method is the cross-validation method. In this study DE is applied to select the optimal values of the SVR parameters. In this case, the proposed method is called E-SVR here.

Mean k-fold (k=10) cross validated R^2 (coefficient of determination) is used as the fitness function to be maximized. Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it.

K-fold cross-validation is one way to improve over the holdout method. In *k*-fold cross-validation, the original sample is randomly partitioned into *k* equal size subsets (subsamples), and the holdout method is repeated *k* times. Each time, one of the *k* subsets is used as the test set and the other *k-1* subsets are put together to form a training set. Then the average error across all *k* trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set *k-1* times. The variance of the resulting estimate is reduced as *k* is increased. The disadvantage of this method is that the training algorithm has to be rerun from scratch *k* times, which means it takes *k* times as much computation to make an evaluation. A variant of this method is to randomly divide the data into a test and training set *k* different times. The advantage of doing this is that you can independently choose how large each test set is and how many trials you average over.

In terms of the E-SVR procedure, details about the DE optimizer in the E-SVR design are mentioned in the next sub-section.

3.2.1 Background about the DE optimizer

DE is a population-based stochastic function minimizer (or maximizer) relating to EAs, whose simple yet powerful and straightforward features make it very attractive for numerical optimization. Using a few control parameters, DE exhibits a fast convergence for a wide range of benchmark functions.

Various schemes are typically used for DE when creating the trial vectors. Each scheme generates trial vectors by adding the weighted difference between other randomly selected members of the population. Meanwhile, each strategy is dependent on three factors; the solution to be perturbed, number of different solutions considered for perturbation and the type of recombination used. The different schemes of DE are classified using the following notation: DE/α/β/δ, where α indicates the method for selecting the parent chromosome that will form the base of the mutated vector, β indicates the number of difference vectors used to perturb the base chromosome, and δ indicates the recombination mechanism used to create the offspring population. The *bin* acronym indicates that the recombination is controlled by a series of independent binomial experiments.

The particular version subject to our investigation is the *DE/rand/1/bin*-version, which appears to be the most frequently used variant, and is often considered as the "basic" version of the DE-algorithm. The DE/*rand*/1/*bin* involves the following steps and procedures (Ayala et al., 2015):

Step 1: *Initialization of the parameter setup*: The user must choose the key parameters that control DE, i.e., population size (N), boundary constraints of optimization variables, mutation factor (f_m), crossover rate (CR), and the stopping criterion (t_{max}).

Step 2: *Initialize the initial population of individuals*: Initialize the generation's counter t = 0 and also initialize a population of individuals (solution vectors) $y(t)$ in upper and lower bounds of each decision variable with random values generated according to a uniform probability distribution in the n-dimensional problem space.

Step 3: *Evaluate the objective function value*: For each individual, evaluate its objective function value. The objective function will also be referred to as the *fitness function*.

Step 4: *Mutation operation (or differential operation)*: Mutate individuals according to the following equation:

$$z_i(t+1) = y_{i_1}(t) + f_m \cdot [y_{i_2}(t) - y_{i_3}(t)] \tag{9}$$

In the above equations, i=1,2,...,N is the individual's index of population; t is the time (generation); $y_i(t) = \left[y_{i_1}(t), y_{i_2}(t), ..., y_{i_n}(t)\right]^{\mathrm{T}}$ stands for the position of the i-th individual of population of N real-valued n-dimensional vectors; $z_i(t) = \left[z_{i_1}(t), z_{i_2}(t), ..., z_{i_n}(t)\right]^{\mathrm{T}}$ stands for the position of the i-th individual of a *mutant vector*; f_m > 0 is a real parameter, called *mutation factor*, which controls the amplification of the difference between two individuals so as to avoid search stagnation. The mutation operation randomly selects the target vector $y_{i_1}(t)$, with $i \neq i_1$. Then, two individuals $y_{i_2}(t)$ and $y_{i_3}(t)$ are randomly selected with $i_1 \neq i_2 \neq i_3 \neq i$, and the difference vector y_{i_2} - y_{i_3} is calculated.

Step 5: *Crossover (recombination) operation*: Following the mutation operation, crossover is applied in the population. For each mutant vector, $z_i(t+1)$, an index $rnbr(i) \in \{1, 2, \cdots, n\}$ is randomly chosen using a uniform distribution, and a *trial vector*, $u_i(t+1) = \left[u_{i_1}(t+1), u_{i_2}(t+1), ..., u_{i_n}(t+1)\right]^T$, is generated via

$$u_{i_j}(t+1) = \begin{cases} z_{i_j}(t+1) \text{ if } randb(j) \leq CR \text{ or } j = rnbr(i), \\ y_{i_j}(t) \text{ otherwise}, \end{cases} \tag{10}$$

where j=1,2,..., n is the parameter index; $y_{ij}(t)$ stands for the i-th individual of *j-th* real-valued vector; $z_{ij}(t)$ stands for the i-th individual of *j-th* real-valued vector of a *mutant vector*; $u_{ij}(t)$ stands for the i-th individual of *j-th* real-valued vector after crossover operation; *randb(j)* is the j-th evaluation of a uniform random number generation with [0, 1]; CR is a *crossover rate* in the range [0, 1].

To decide whether or not the vector $u_i(t + 1)$ should be a member of the population comprising the next generation, it is compared to the corresponding vector $x_i(t)$. Thus, if f denotes the objective function under minimization, then

$$x_i(t+1) = \begin{cases} u_i(t+1), \text{ if } f(u_i(t+1)) < f(x_i(t)), \\ y_i(t), \text{ otherwise}. \end{cases} \tag{11}$$

Step 6: *Update the generation's counter*: $t = t + 1$;

Step 7: *Verification of the stopping criterion*: Loop to **Step 2** until a stopping criterion is met, usually a maximum number of iterations (generations), t_{max}.

4. RESULTS ANALYSIS

All computation models based on SVR and E-SVR and the DE optimizer was carried out in MATLAB environment on a PC (Personal Computer) using Windows 7 operational system with a Intel(R) Core(TM) i7-5820K CPU (Central Processing Unit), 3.30 GHz and 128 GB of RAM (random access memory). In experiments, 10-fold cross-validation was required in SVR and E-SVR to evaluate the generalization ability of models. SVR and the proposed E-SVR approach were applied and compared with other regression methods to predict 5 m backstroke start time using kinematic and kinetic variables and to determine the accuracy of the mean absolute percentage error.

For the optimization procedure of the E-SVR, we used the following parameters in design of DE/*rand*/1/*bin*: population size equal to 30 individuals and the stopping criterion t_{max} = 50 generations, i. e., each simulation using DE was allowed to run for 1,500 evaluations of the objective function. The setup of DE/*rand*/1/*bin* was using a constant mutation factor given by f_m = 0.5 and a crossover rate of CR = 0.9. Table 1 shows the modelling results obtained by SVR and E-SVR approaches.

Table 1. Modelling results for the backstroke start performance using SVR and E-SVR approaches.

SVR approach	C	ε	σ	10-fold cross validated R^2	MSE	Error (mean)	Error (maximum)	Error (std[(1)])
SVR(1)	50	0.001	0.2	0.9472	0.0013	0.0083	0.1185	0.0359
SVR(2)	100	0.001	0.2	0.9808	4.80E-4	-0.0037	0.0678	0.0219
SVR(3)	150	0.001	0.2	0.9878	3.06E-4	-0.0078	0.0452	0.0159
SVR(4)	50	0.01	0.2	0.9433	0.0014	0.0128	0.1230	0.0359
SVR(5)	100	0.01	0.2	0.9813	4.67E-4	7.86E-4	0.0723	0.0219
SVR(6)	150	0.01	0.2	0.9898	2.56E-4	-0.0033	0.0497	0.0159
SVR(7)	50	0.1	0.2	0.8163	0.0046	0.0578	0.1680	0.0359
SVR(8)	100	0.1	0.2	0.8976	0.0026	0.0458	0.1173	0.0219
SVR(9)	150	0.1	0.2	0.9206	0.0020	0.0417	0.0947	0.0159
SVR(10)	50	0.001	0.3	0.9650	8.76e-4	0.0123	0.0933	0.0273
SVR(11)	100	0.001	0.3	0.9904	2.40E-4	-0.0017	0.0479	0.0156
SVR(12)	150	0.001	0.3	0.9936	1.59E-4	-0.0065	0.0294	0.0110
SVR(13)	50	0.01	0.3	0.9597	0.0010	0.0168	0.0978	0.0273
SVR(14)	100	0.01	0.3	0.9902	2.45E-4	0.0028	0.0524	0.0156
SVR(15)	150	0.01	0.3	0.9951	1.21E-4	-0.0020	0.0339	0.0110
SVR(16)	50	0.1	0.3	0.8183	0.0045	0.0618	0.1428	0.0273
SVR(17)	100	0.1	0.3	0.8993	0.0025	0.0478	0.0974	0.0156
SVR(18)	150	0.1	0.3	0.9213	0.0020	0.0430	0.0789	0.0110
E-SVR	185.43	0.0065	0.398	**0.9999**	3.34E-6	-8.25E-4	0.0048	0.0017

[(1)] std: standard deviation

Figure 3 illustrates the best results (represented in Table 1) of the SVR (15) and E-SVR application to a swimmer start with mean 10-fold cross validated R^2 equal to 0.9951 and 0.9999, respectively.

Inappropriate parameters in SVR can lead to overfitting or underfitting problems. How to properly set the control hyperparameters is a major task, which has a significant impact on the optimal generalization performance and the SVR regression accuracy, especially when it comes to predicting the backstroke swimming start performance modelling task.

As short distance swimming events can be won by margins as small as 0.01 s (De Jesus et al., 2016), prediction-modeling errors should be minimum.

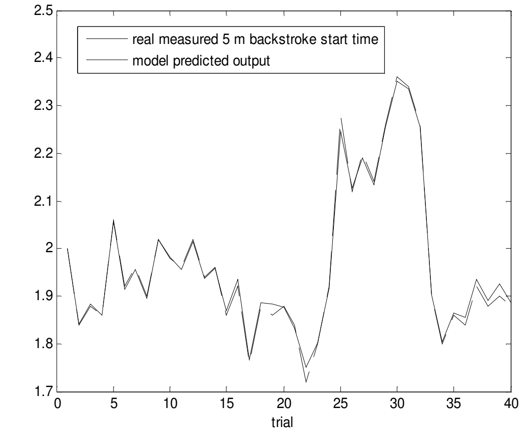
5. CONCLUSION AND FUTURE RESEARCH

SVR is a well-known and computationally powerful machine learning technique for regression and nonlinear modelling problems, which has been successfully applied to solve many practical problems in a wide variety of fields. Compared with tools based on experience risk minimization, such as artificial neural networks and least square methods, it requires less samples and yields better generalization ability.

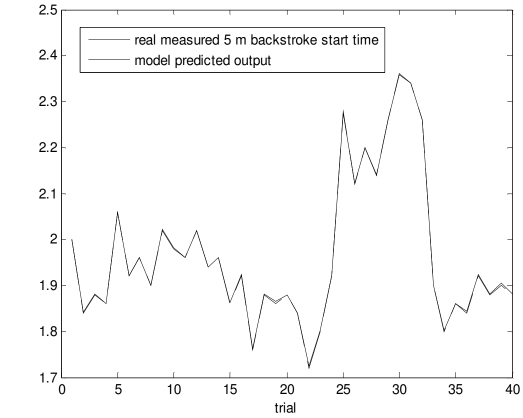
SVR models performance is very sensitive to the selection of the hyperparameters and there is no mathematical based procedure for deriving the exact desired values. Thus, the selection of those parameters is a crucial part of the research on SVR models. The aim of this study was to develop an SVR-based DE model to backstroke swimming start performance modelling task.

On the other hand, in the fields of competitive swimming, research questions are usually complex, with interrelations between different variables being mostly of a nonlinear profile. Results for both mathematical approaches confirm that accurate model fit can be obtained by means of SVR and E-SVR. However, E-SVR should be used rather than SVR for 5 m backstroke start time prediction due to the quite small differences separating the elite level swimmers.

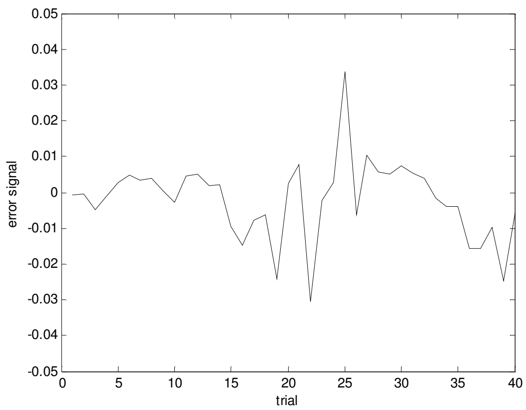
As a future work, we are planning to investigate the effect of the proposed E-SVR approach in single and multi-objective optimization design conceptions on new application fields related to competitive swimming and biomechanics.



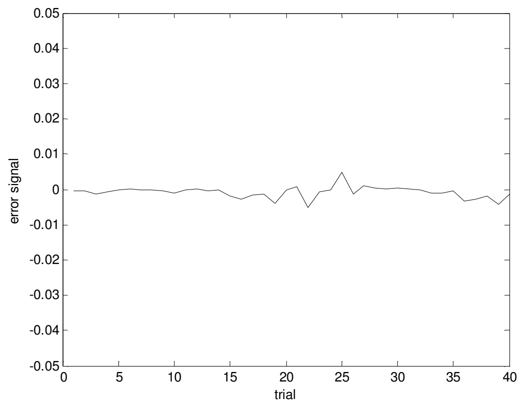
(a) SVR(15)



(b) E-SVR



(c) error signal for the SVR(15)



(d) error signal for the E-SVR

Figure 3. Results of modelling using SVR and E-SVR.

6. ACKNOWLEDGEMENTS

This research was supported by the Coordination for the Improvement of Higher Education Personnel – CAPES (grant number, BEX 0761/12-5/2012-2015), Foundation for Science and Technology (grant number, EXPL/DTP-DES/2481/2013- FCOMP-01-0124-FEDER-041981) and CAPES FCT 039-2014 (grant number 99999.008578/2014-01). Furthermore, Mariani, Coelho and Klein would like to thank National Council of Scientific and Technologic Development of Brazil - CNPq (grants: 303908/2015-7-PQ, 303906/2015-4-PQ and 150501/2017-0) and "Fundação Araucária" (grant: FA 117/2014) for its financial support of this work.

7. REFERENCES

Ayala, H.V.H., Santos, F.M., Mariani, V.C., Coelho, L.S., 2015. "Image thresholding segmentation based on a novel beta differential evolution approach". *Expert Systems with Applications*, Vol. 42, p. 2136-2142.

Cheng, A., Jiang, X., Li, X., Zhang, C. and Zhu, H., 2017. "Multiple sources and multiple measures based traffic flow prediction using the chaos theory and support vector regression method". *Physica A: Statistical Mechanics and its Applications*, Vol. 466, p. 422-434.

De Jesus, K., Ayala, H.V.H., de Jesus, K., Coelho, L.S., Medeiros, A.I.A., Abraldes, J.A., Vaz, M.A.P., Fernandes, R.J., Vilas-Boas., J.P., 2018. Modelling and predicting backstroke start performance using non-liear and linear models. *Journal of Human Kinetics* (in press).

De Jesus, K., De Jesus, K., Abraldes, J.A., Medeiros, A.I., Fernandes, R.J. and Vilas-Boas, J. P., 2016. "Are the new starting block facilities beneficial for backstroke start performance?". *Journal of Sports Sciences*, Vol. 34, p. 871-877.

Das, S., Mullick, S. S. and Suganthan, P.N., 2016. "Recent advances in differential evolution – An updated survey". *Swarm and Evolutionary Computation*, Vol. 27, p. 1-30.

Duan, K., Keerthi, S.S. and Poo, A.N., 2003, “Evaluation of simple performance measures for tuning SVM hyperparameters”. *Neurocomputing*, Vol. 51, p. 41-59.

Ruas, G.I.S., Bragatto, Lamar, M.V., Aoki, A.R. and de Roco, S.M., 2008. “Electrical energy demand prediction using artificial neural networks and support vector regression”. In: *3rd IEEE International Symposium on Communications, Control and Signal Processing* - *ISCCSP*, St. Julian’s, Malta, 2008.

Sapankevych, N. I. and Sankar, R., 2009. “Time series prediction using support vector machines: a survey”. *IEEE Computational Intelligence Magazine*, Vol. 4, p. 24-38.

Shafizadeh-Moghadam, H., Tayyebi, A., Ahmadlou, M., Delavar, M.R. and Hasanlou, M., 2017. “Integration of genetic algorithm and multiple kernel support vector regression for modeling urban growth”. *Computers, Environment and Urban Systems*, Vol. 65, p. 28-40.

Simon, D., 2013. *Evolutionary Optimization Algorithm*, Wiley, New York, USA, 2013.

Storn, R. Price, K., 1995. “Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces”. *Technical Report TR-95-012*, International Computer Science Institute, Berkeley, CA, USA.

Storn, R. Price, K., 1997. “Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces”. *Journal of Global Optimization*, Vol. 11, p. 341-359.

Vapnik, V., 1995. *The Nature of Statistical Learning Theory*. Springer, New York, USA.

Vapnik, V., 1997. *Statistical Learning Theory*. Wiley, New York, USA.

Vrablecová, P., Ezzeddine, A.B., Rozinajová, V., Šárik, S. and Sangaiah, A.K., 2017. “Smart grid load forecasting using online support vector regression”. *Computers & Electrical Engineering*, (in press).

Yang, C., Liu, J., Zeng, Y. and Xie, G., 2017. “Prediction of components degradation using support vector regression with optimized parameters”. *Energy Procedia*, Vol. 127, p. 284-290.

Zhang, J., Zhan, Z.-h., Lin, Y., Chen, N., Gong, Y.-j., Zhong, J.-h., Chung, H.S.H., Li, Y. and Shi, Y.-h., 2011. Evolutionary computation meets machine learning: a survey. *IEEE Computational Intelligence Magazine*, Vol. 6, p. 68-75.

Zhang, Y., Xu, S., Chen, K., Liu, Z. and Chen, C.L. P., 2016. “Fuzzy density weight-based support vector regression for image denoising”. *Information Sciences*, Vol. 339, p. 175-188.

Zhu, B., Han, D., Wang, P., Wu, Z., Zhang, T. and Wei, Y.-M., 2017. “Forecasting carbon price using empirical mode decomposition and evolutionary least squares support vector regression”. *Applied Energy*, Vol. 191, p. 521-530.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.