

Research Article

Machine Learning of the Reactor Core Loading Pattern Critical Parameters

Krešimir Trontl,¹ Dubravko Pevec,¹ and Tomislav Šmuc²

¹Department of Applied Physics, Faculty of Electrical Engineering and Computing, Unska 3, 10000 Zagreb, Croatia

²Division of Electronics, Ruđer Bošković Institute, Bijenička 54, 10002 Zagreb, Croatia

Correspondence should be addressed to Krešimir Trontl, kresimir.trontl@fer.hr

Received 11 March 2008; Accepted 23 June 2008

Recommended by Igor Jencic

The usual approach to loading pattern optimization involves high degree of engineering judgment, a set of heuristic rules, an optimization algorithm, and a computer code used for evaluating proposed loading patterns. The speed of the optimization process is highly dependent on the computer code used for the evaluation. In this paper, we investigate the applicability of a machine learning model which could be used for fast loading pattern evaluation. We employ a recently introduced machine learning technique, support vector regression (SVR), which is a data driven, kernel based, nonlinear modeling paradigm, in which model parameters are automatically determined by solving a quadratic optimization problem. The main objective of the work reported in this paper was to evaluate the possibility of applying SVR method for reactor core loading pattern modeling. We illustrate the performance of the solution and discuss its applicability, that is, complexity, speed, and accuracy.

Copyright © 2008 Krešimir Trontl et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Decrease of the fuel cycle costs is an important factor in nuclear power plant management. The economics of the fuel cycle can strongly benefit from the optimization of the reactor core loading pattern, that is, minimization of the amount of enriched uranium and burnable absorbers placed in the core, while maintaining nuclear power plant operational and safety characteristics.

The usual approach to loading pattern optimization involves high degree of engineering judgment, a set of heuristic rules, an optimization algorithm, and a reactor physics computer code used for evaluating proposed loading patterns. Since the loading pattern optimization problem is of combinatorial nature and involves heuristics requiring large numbers of core modeling calculations (e.g., genetic algorithms or simulated annealing algorithms), the time needed for one full optimization run is essentially determined by the complexity of the code that evaluates the core loading pattern.

The aim of the work reported in this paper was to investigate the applicability of a machine learning modeling for fast loading pattern evaluation. We employed a recently

introduced machine learning technique, support vector regression (SVR), which has a strong theoretical background in statistical learning theory. SVR is a supervised learning method in which model parameters are automatically determined by solving a quadratic optimization problem.

This paper reports on the possibility of applying SVR method for reactor core loading pattern modeling. Required size of the learning data set, as a function of targeted accuracy, influence of SVR free parameters, as well as input vector definition were studied.

In Section 2, the support vector regression method is discussed. Basics of fuel loading pattern development and optimization as well as the methodology applied for the investigation of applicability of the SVR method for fuel loading pattern evaluation are presented in Section 3. Results and discussion are given in Section 4, while in Section 5 the conclusions based on this work are drawn.

2. SUPPORT VECTOR REGRESSION

Machine learning is, by its definition, a study of computer algorithms that improve automatically through experience.

One of machine learning techniques is the support vector machines (SVMs) method, which has a strong theoretical background in statistical learning theory [1]. The method proved to be a very robust technique for complex classification and regression problems. Although, historically speaking, the first implementation of SVM was for classification problems [2, 3], in the last decade, the application of SVM for nonlinear regression modeling is noticeable in different fields of science and technology [4–10], the main reason being robustness and good generalization properties of the method.

In the upcoming paragraphs, we will give a short introduction into the support vector regression method, stressing only the most important theoretical and practical aspects of the technique. Additional information can be found in referenced literature.

In general, the starting point of the machine learning problem is a collection of samples, that is, points, to learn the model (training set) and a separate set to test the learned model (test set). Since we are interested in developing a regression model, we will consider a training data set, as well as testing data set, comprised of a number of input/output pairs, representing the experimental relationship between input variables (\vec{x}_i) and corresponding scalar output value (y_i):

$$\{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_n, y_n)\} \subset \Re^d \times \Re. \quad (1)$$

In our case, the input vector defines the characteristics of the loading pattern, while the output value, also referred to as a target value, denotes the parameter of interest.

The modeling objective is to find a function $y = f(\vec{x})$ such that it accurately predicts (with ε tolerance) the output value (y) corresponding to a new input vector (\vec{x}), yet unseen by the model (the model has not been trained on that particular input vector) [11].

Due to the high complexity of underlying physical process that we are modeling, the required function can be expected to have high nonlinear properties. In the support vector regression approach, the input data vector \vec{x} is mapped into a higher dimensional feature space F using a nonlinear mapping function Φ , and a linear regression is performed in that space. Therefore, a problem of nonlinear regression in low-dimensional input space is solved by linear regression in high-dimensional feature space.

The SVR technique considers the following linear estimation function:

$$f(\vec{x}) = \langle \vec{w}, \Phi(\vec{x}) \rangle + b, \quad (2)$$

where \vec{w} denotes the weight vector, b is a constant known as bias, $\Phi(\vec{x})$ is the mapping function, and $\langle \vec{w}, \Phi(\vec{x}) \rangle$ is the dot product in feature space, such that $\Phi : \vec{x} \rightarrow F$, $\vec{w} \in F$ [12]. The unknown parameters w and b are estimated using the data points in the training set. To avoid overfitting and maximize generalization capability of the model, a regularized form of the functional, following principles of structural risk minimization (SRM), is minimized:

$$R_{\text{reg}}[f] = \sum_{i=1}^M C(f(\vec{x}_i) - y_i) + \lambda \|\vec{w}\|^2, \quad (3)$$

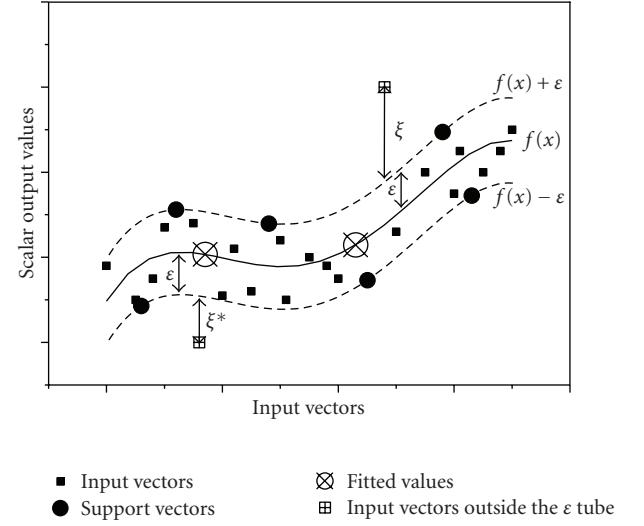


FIGURE 1: The schematic illustration of the SVR using ε -insensitive cost function (tube).

where $R_{\text{reg}}[f]$ denotes regression risk (possible test set error), based on empirical risk which is expressed through the cost function C determined on the points of the training set, and a term reflecting the complexity of the regression model. Minimization task thus involves simultaneous minimization of the empirical risk and minimization of structural complexity of the model. Most commonly used cost function (loss functional) related to empirical risk is the so called “ ε insensitive loss function”:

$$C(f(\vec{x}_i) - y_i) = \begin{cases} \|f(\vec{x}) - y\| - \varepsilon, & \text{for } \|f(\vec{x}) - y\| \geq \varepsilon, \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

where ε is a parameter representing radius of the tube around regression function. The SVR algorithm attempts to position the tube around the data, as depicted in Figure 1 [7], and according to (4) does not penalize data points for which calculated values (y) lie inside this tube. The deviations of points that lie more than ε away from the regression function are penalized in the optimization through their positive and negative deviations ξ and ξ^* , called “slack” variables.

It was shown that the following function minimizes the regularized functional given by (3) [1]:

$$f(\vec{x}, \vec{w}) = f(\vec{x}, \vec{\alpha}, \vec{\alpha}^*) = \sum_{i=1}^n (\alpha_i^* - \alpha_i) K(\vec{x}_i, \vec{x}) + b, \quad (5)$$

where α_i^*, α_i are Lagrange multipliers describing \vec{w} , and are estimated, as well as parameter b , using an appropriate quadratic programming algorithm, and $K(\vec{x}_i, \vec{x})$ is a so called *kernel* function describing the dot product $\langle \vec{w}, \Phi(\vec{x}) \rangle$ in the feature space. A number of kernel functions exist [13].

Kernel functions used in this work are described in more details in the following section.

Due to the character of the quadratic optimization, only some of the coefficients $\alpha_i^* - \alpha_i$ are nonzero, and the corresponding input vectors \vec{x} are called *support vectors* (SVs). Input vectors matching zero $\alpha_i^* - \alpha_i$ coefficients are positioned inside the ϵ tolerance tube and are therefore, not interesting for the process of model generation. Support vectors that are determined in the training (optimization) phase are the “most informative” points, that compress the information content of the training set. In most of the SVR formulations, there are two free parameters to be set by the user: C -cost of the penalty for data-model deviation, and ϵ -insensitive zone. These two free parameters and the chosen form of the kernel function and its corresponding parameters control the accuracy and generalization performance of the regression model.

3. METHODOLOGY

One of the key processes of both, safe and economical operations of nuclear reactor, is in-core fuel management, or to be more precise, fuel loading pattern determination and optimization. Every method and technique used for fuel loading pattern determination and optimization tasks, whether based on engineering judgement, heuristic rules, genetic algorithms, or a combination of stated approaches, requires a large number of potential fuel loading patterns evaluation. The evaluation is normally performed using a more or less sophisticated reactor physics code. Usage of such codes is time consuming. Therefore, in this work, we are investigating the possibility of SVR method being used as a fast tool for loading pattern evaluation.

However, taking into account that the SVR method is to be used, a number of factors have to be addressed prior to creating a model. The first is the setting of the loading pattern that is to be investigated, including the method by which the experimental data points are to be generated, the definition of the input space and parameters used as target values. The second is the choice of the kernel function and appropriate free parameters used in the SVR model. Finally, SVR modeling tools have to be addressed.

3.1. Computational experiment setup

Taking into account the preliminary and inquiring characteristics of the study, we decided to use limited fuel assembly inventory for a single loading pattern optimization as a basis for the development of our regression models. NPP Krško Cycle 22 loading pattern has been used as a reference one. 121 fuel assemblies, grouped in 7 batches that were used for core loading in Cycle 22 have been used for generating a large number of randomly generated fuel loading patterns, which were then divided into training and testing data sets and employed in SVR model development process. The global core calculations of each of the trial loading patterns have been conducted using MCRAC code of the FUMACS code package, which also includes the LEOPARD code for

two-group cross-section preparation [14]. The calculation is based on quarter core symmetry, fixed cycle length, and fixed soluble boron concentration curve.

The generation phase, that is, the definition of the loading patterns, has been based on a semirandom algorithm. In order to narrow the investigated input space as much as possible, as well as to stay within the limits of the numbers of available fuel assemblies per batch, we introduced a limitation for every fuel assembly regarding the position where it can be placed: fuel assemblies originally placed on axes positions could be randomly placed only on axes positions, and vice versa. The central location fuel assembly was fixed for every loading pattern.

The most important issue in the regression model development is the definition of the input space to be used for SVR model development. Since in a quarter core symmetry setup, the NPP Krško core is defined by 37 fuel assemblies, and having in mind the inquiring nature of the work, we decided to simplify the problem by the assumption of the 1/8 core symmetry, resulting in 21 fuel assemblies defining the core. Fuel assembly (position) is defined by initial enrichment, number of IFBAs, and reactor history, or at least burnup accumulated in previous cycles. Therefore, the number of potential parameters defining the input space is 63. The high dimensionality of the input space generally increases the number of training points and time required for the development of the SVR of certain generalization properties. Therefore, we decided to reduce the number of parameters by introducing k -inf at the beginning of the cycle as a new parameter and representing fuel assembly only by k -inf and number of IFBAs (0 for old fuel, and 32, 64, 92, and 116 for fresh fuel). Thus, the final number of parameters defining the input space was 42.

The SVR model would eventually be used in an optimization algorithm as a fast tool for loading pattern evaluation. Therefore, the target parameters which we want to model should be the most important parameters on which such an evaluation is based. In this work, we used the global core effective multiplication factors at the beginning and at the end of the cycle (k_{effBOC} and k_{effEOC}), as well as power peaking factor ($F_{\Delta H}^N$) as target parameters for which separate SVR models were built.

3.2. Kernel functions

The idea of the kernel function is to enable mathematical operations to be taken in the input space, rather than in the high-dimensional feature space [15]. The theory is based upon reproducing kernel Hilbert spaces (RKHSs) [16].

A number of kernel functions have been proposed in the literature. The particular choice of the kernel that is going to be used for mapping nonlinear input data into a linear feature space is highly dependent on the nature of the data representing the problem. It is up to the modeller to select the appropriate kernel function. In this paper, the focus is placed on two widely used kernel functions, namely, radial basis

function (RBF), also called Gaussian and the polynomial function (PF), which are defined by (6)

$$\begin{aligned} K_{\text{RBF}}(\vec{x}_i, \vec{x}_j) &= \exp\left(\frac{-\|\vec{x}_i - \vec{x}_j\|^2}{2\sigma^2}\right), \\ K_{\text{PF}}(\vec{x}_i, \vec{x}_j) &= (\vec{x}_i^T \vec{x}_j + 1)^d. \end{aligned} \quad (6)$$

In the case of RBF kernel, parameter σ represents the radius of the Gaussian kernel, while d in the case of PF kernel represents the degree of the polynomial kernel.

As already mentioned, the behaviour of the SVR technique strongly depends on the selection of the kernel function, its corresponding parameters, and general SVR “free” parameters (C and ϵ). All the parameters used in this study were determined by a combination of engineering judgement and optimization procedure based on the application of genetic algorithms [17].

3.3. SVR modeling tools

Excellent results in SVR application to a wide range of classification and regression problems in different fields of science and technology, initiated creation of a number of implementations of the support vector machines algorithm, some of which are freely available software packages. In this work, we decided to test three often used packages: SVMTorch [18], LIBSVM [19], and WEKA [20].

As stated in the previous subsection, RBF and PF kernel functions have been used. The general form of the kernels is given in (6). However, practical parameterisation of the functions, that is, their representation, is somewhat different from code to code. For example, parameter g in LIBSVM notation for RBF represents $1/(2\sigma)^2$. Whenever, a direct comparison of codes has been performed, general kernel parameters have been set (see (6)), and code specific parameters were modified to reflect on these values.

4. RESULTS AND DISCUSSION

4.1. Comparison of code packages

The comparison of three code packages for SVR modeling, namely, SVMTorch, LIBSVM, and WEKA, has been conducted using a maximum training set size of 15 000 data points while the test set consisted of 5000 data points. The number of data points for learning models is typically enlarged until satisfactory results regarding the accuracy are achieved. In this subsection, only the results of final models comparison are presented.

Preliminary analyses revealed that preprocessing of the input data is required in order to allow normal and reasonably fast operation of all SVR code packages. Mainly, due to the fact that input variables span extremely different ranges, scaling of the input data has been performed, including the scaling of the target values (all in the range 0 to 1), using one of LIBSVM codes: SVMSCALE.

Models for three target values (k_{effBOC} , k_{effEOC} and $F_{\Delta H}^N$) were compared for the model accuracy, learning and

implementation times (Pentium 4 Mobile CPU 1.7 GHz, 256 MB RAM, Windows XP SP2), and the relative number of support vectors as the measure of model generalization characteristics. The implementation time has been measured on 5000 data points. The accuracy of the model was determined using root mean square error (RMSE) and relative average deviation (RAD) defined as

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{\sum_{i=1}^n (y_i - f_i)^2}{n}}, \\ \text{RAD} &= \frac{\sum_{i=1}^n (|y_i - f_i|/y_i) \times 100\%}{n}, \end{aligned} \quad (7)$$

where f_i stands for predicted value corresponding to the target value y_i . The metric of interest was also the percentage of tested data points which had the predicted value deviate from the target value by more than 20%:

$$\frac{|y_i - f_i|}{y_i} \times 100\% > 20\%. \quad (8)$$

In the case of RBF kernel function, the initial values of free parameters were estimated using a genetic algorithm (GA) on the LIBSVM code. The ranges for every parameter (C , ϵ , and σ) were set, based on engineering judgement, from 1 to 1000 for C and 0.001 to 2.0, and 1 to $7.07(\sqrt{50})$ for ϵ and σ , respectively. The GA was characterized by 20 populations each consisting of 100 members. The training set consisted of 4500 data points, while the test set had 500 data points. The best result was obtained for $C = 371.725$, $\epsilon = 0.05154$, and $\sigma = 6.4697$.

In the case of the PF kernel function, we decided to set the d parameter to the commonly used value of 3, while for simplicity reasons C and ϵ were set to 371.725 and 0.05154, respectively. Comparison results for RBF kernel function are given in Table 1 while in Table 2 comparison results for PF kernel function are presented.

The results of preliminary tests suggest that appropriate regression models using SVM method can be developed for all target values regardless of the applied code package. The only difference is the learning time required for the model to be developed. The implementation or deployment time for the execution of the model (maximum of 30 seconds for 5000 calculations) is not the issue. The accuracy for the k_{effBOC} and k_{effEOC} target values is satisfactory, while additional effort has to be placed on developing the $F_{\Delta H}^N$ model by adjusting SVR parameters and increasing the training set size.

4.2. Training set size influence on SVR model quality

SVR model quality can be interpreted as the time required for the model learning, accuracy of the model, and generalization characteristics of the model. As shown in the previous subsection, model implementation/deployment time is not the key issue.

As discussed previously, the size of the training set influences all factors of the model quality, and generally thorough analysis of that influence is necessary. Here, we present the results of preliminary tests conducted for

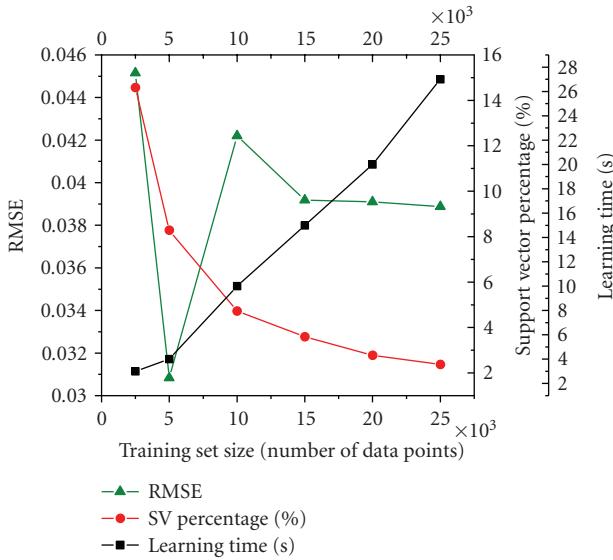
TABLE 1: Comparison of results for RBF kernel function.

Target value	Code package	Accuracy			Learning/Implen. time [s]	SV [%]
		RMSE	RAD [%]	>20%[%]		
k_{effBOC}	SVMTorch	0.029	6.793	3.44	120/14	5.27
	LIBSVM	0.029	7.179	3.96	18/3	3.59
	WEKA	0.028	6.621	3.20	2250/6	3.77
k_{effEOC}	SVMTorch	0.050	5.048	1.96	10800/30	16.69
	LIBSVM	0.045	4.550	1.76	1260/15	18.37
	WEKA	0.045	4.570	1.98	28160/30	18.22
$F_{\Delta H}^N$	SVMTorch	0.040	15.060	20.42	13080/13	16.91
	LIBSVM	0.039	14.810	19.64	1080/14	17.97
	WEKA	0.039	14.801	19.58	33362/14	17.86

TABLE 2: Comparison results for PF kernel function.

Target value	Code package	Accuracy			Learning/Implen. time [s]	SV [%]
		RMSE	RAD [%]	>20% [%]		
k_{effBOC}	SVMTorch	0.030	6.418	3.76	50/11	6.88
	LIBSVM	0.030	7.610	5.62	9/3	4.83
	WEKA*	0.030	6.259	3.46	4027/10	2.43
k_{effEOC}	SVMTorch	0.072	7.147	4.92	840/20	19.33
	LIBSVM	0.058	5.856	3.12	2113/11	30.21
	WEKA*	0.056	6.095	3.34	31120/45	30.02
$F_{\Delta H}^N$	SVMTorch	0.044	16.057	22.92	420/18	18.50
	LIBSVM	0.039	14.992	20.50	325/8	18.43
	WEKA*	0.042	15.701	22.40	7000/30	17.17

* PF kernel in the form $K_{\text{PF}}(\vec{x}_i, \vec{x}_j) = (\vec{x}_i^T \vec{x}_j)^d$.

FIGURE 2: Training set size influence on model quality for k_{effBOC} preliminary tests.

k_{effBOC} model development using LIBSVM code package (see Figure 2). The characteristics of applying other code packages on all target values are qualitatively very similar.

Apart from the anomaly observed for the RMSE curve at the training set size of 5000 data points originating in statistical and random characteristic of the training and testing data sets, the accuracy (RMSE) and the generalization properties (low SV percentage) of the models increase with the increase of the training set size. The learning time is also increased exhibiting a nearly linear trend.

5. CONCLUSIONS

This work introduces a novel concept for fast evaluation of reactor core loading pattern, based on general robust regression model relying on the state of the art research in the field of machine learning.

Preliminary tests were conducted on the NPP Krško reactor core, using the MCRAC code for the calculation of reference data. Three support vector regression code packages were employed (SVMTorch, LIBSVM, and WEKA) for creating regression models of effective multiplication factor at the beginning of the cycle (k_{effBOC}), effective multiplication factor at the end of the cycle (k_{effEOC}), and power peaking factor ($F_{\Delta H}^N$).

The preliminary tests revealed a great potential of the SVR method application for fast and accurate reactor core loading pattern evaluation. However, prior to the final conclusion and incorporation of SVR models in optimization

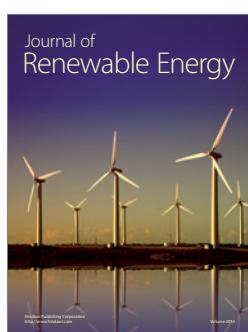
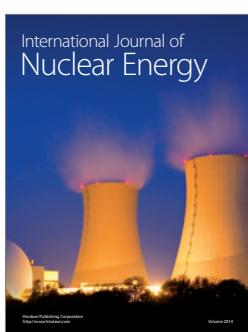
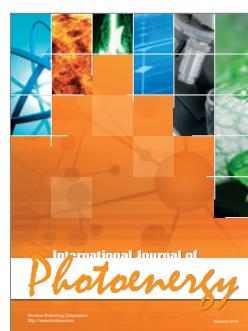
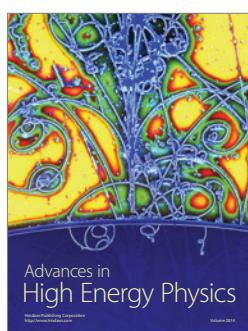
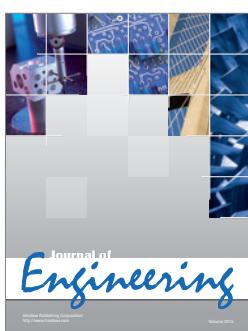
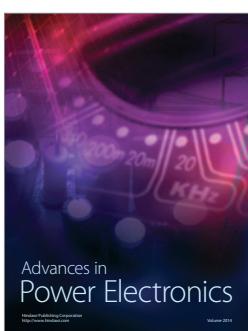
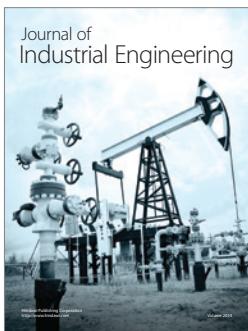
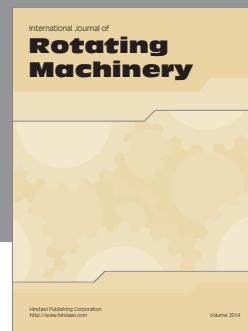
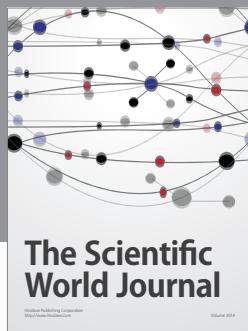
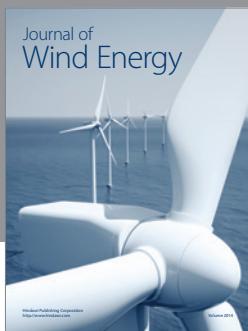
codes, additional tests and analyses are required, mainly focused on the parameters defining input vector, thus influencing its size, the required size of the training set and parameters defining kernel functions.

In the case of the scenario involving machine learning from the results of more accurate and time consuming 3D code, we do not anticipate any major changes in the learning stage of SVR model development, as well as its implementation. However, generation of training and testing data sets would be more demanding (time consuming and requiring more hardware resources).

These are the issues that are within the scope of our future research.

REFERENCES

- [1] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.
- [2] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pp. 130–136, San Juan, Puerto Rico, USA, June 1997.
- [3] B. Schölkopf, *Support vector learning*, Ph.D. thesis, R. Oldenbourg, Munich, Germany, 1997.
- [4] S. M. Clarke, J. H. Griebisch, and T. W. Simpson, "Analysis of support vector regression for approximation of complex engineering analyses," in *Proceedings of the Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC '03)*, pp. 535–543, Chicago, Ill, USA, September 2003.
- [5] T. Gu, W. Lu, X. Bao, and N. Chen, "Using support vector regression for the prediction of the band gap and melting point of binary and ternary compound semiconductors," *Solid State Sciences*, vol. 8, no. 2, pp. 129–136, 2006.
- [6] E. Myasnikova, A. Samsonova, M. Samsonova, and J. Reinitz, "Support vector regression applied to the determination of the developmental age of a *Drosophila* embryo from its segmentation gene expression patterns," *Bioinformatics*, vol. 18, supplement 1, pp. S87–S95, 2002.
- [7] S. Nandi, Y. Badhe, J. Lonari, et al., "Hybrid process modeling and optimization strategies integrating neural networks/support vector regression and genetic algorithms: study of benzene isopropylation on Hbeta catalyst," *Chemical Engineering Journal*, vol. 97, no. 2-3, pp. 115–129, 2004.
- [8] D. J. Strauß, G. Steidl, and U. Welzel, "Parameter detection of thin films from their X-ray reflectivity by support vector machines," *Applied Numerical Mathematics*, vol. 48, no. 2, pp. 223–236, 2004.
- [9] D. O. Whiteson and N. A. Naumann, "Support vector regression as a signal discriminator in high energy physics," *Neurocomputing*, vol. 55, no. 1-2, pp. 251–264, 2003.
- [10] K. Trontl, T. Šmuc, and D. Pevec, "Support vector regression model for the estimation of γ -ray buildup factors for multi-layer shields," *Annals of Nuclear Energy*, vol. 34, no. 12, pp. 939–952, 2007.
- [11] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge, UK, 2005.
- [12] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [13] A. J. Smola, *Learning with kernels*, Ph.D. thesis, Technische Universität Berlin, Berlin, Germany, 1998.
- [14] B. Petrović, D. Pevec, T. Šmuc, and N. Urli, "FUMACS (FUel MAnagement Code System)," Rudjer Bošković Institute, Zagreb, Croatia, 1991.
- [15] S. R. Gunn, "Support vector machines for classification and regression," Tech. Rep., Faculty of Engineering, Science and Mathematics, University of Southampton, Southampton, UK, May 1998.
- [16] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.
- [17] B. Üstün, W. J. Melissen, M. Oudenhuijzen, and L. M. C. Buydens, "Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization," *Analytica Chimica Acta*, vol. 544, no. 1-2, pp. 292–305, 2005.
- [18] R. Collobert and S. Bengio, "SVM Torch: support vector machines for large-scale regression problems," *The Journal of Machine Learning Research*, vol. 1, no. 2, pp. 143–160, 2001.
- [19] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," Manual, 2001.
- [20] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, San Francisco, Calif, USA, 2nd edition, 2005.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>