Assistant Professor Sohel RANA, PhD
E-mail: srana_stat@yahoo.com
Waleed DHHAN[*], PhD Candidate
E-mail: w.dhhan@gmail.com
Professor Habshah MIDI, PhD
E-mail: habshahmidi@ gmail.com
Faculty of Science and Institute for Mathematical Research
University Putra, Serdang, Selangor, Malaysia
[*]Department of Business Administration
Al Mustaqbal University College, Babylon, Iraq

A HYBRID TECHNIQUE FOR SELECTING SUPPORT VECTOR REGRESSION PARAMETERS BASED ON A PRACTICAL SELECTION METHOD AND GRID SEARCH PROCEDURE

Abstract. *In order to enhance the generalization ability of the practical selection (PLSN) method for choosing the optimal parameters of the support vector regression (SVR) model that was proposed by Cherkassky and Ma (2004), we investigate a new hybrid technique that combines the PLSN method and the grid search procedure. We explore this and find it to be suitable for different types of additive noise including Laplacian noise density. We show that the proposed parameter selection for SVR achieves a good generalization performance by testing several regression problems (low- and high-dimensional data). Moreover, the proposed method is effective for finding the optimal parameters of SVR for all kinds of noise, including Laplacian noise. The generalization performance of the proposed method is compared with that of the PLSN method, with some numerical studies for Gaussian noise as well as non-Gaussian noise. The results show that the proposed method is superior to the PLSN method for various types of noise.*

Keywords*: machine learning, support vector regression, parameter selection, grid search.*

JEL Classification: 62G05; 62G08

1. Introduction

The support vector machine (SVM) is a powerful approach in the machine learning community, and has become one of the fundamental fields in artificial intelligence during recent years (Frohlich and Zell 2005). It has been successfully applied to regression problems as well as to classification problems. However, the generalization performance of SVM depends heavily on the right selection of the

hyper-parameters C and ε, so the major issue for practitioners attempting to apply SVM is how to set these parameter values (to guarantee a good generalization performance) for a training data set (Wang et al. 2003). Since the set of parameters of SVR should be defined by the user, this tends to make it of limited practical use (Üstün et al. 2005). Hence it is very important to focus on selecting the tuning parameters of SVR.

In the literature there are some practical methods of choosing the parameters C and ε. Cherkassky and Mulier (1998), and Vapnik (1998) proposed that users select the parameters C and ε based on their experience and their prior knowledge. However, this approach is not suitable for non-expert users, since we are dealing with an unknown underlying distribution of the data. Mattera and Haykin (1999) proposed that the value of the parameter C be set equal to the range of the dependent variable. But this approach is not appropriate, given the possible existence of outliers in the training data. At the same time, they recommended choosing the value of the parameter ε so that the proportion of SVs in the SV regression model is around 50% of the number of samples. However, an SV regression model that is built on the basis of a few support vectors is not always applicable (it may result in large prediction errors) (Üstün et al. 2005). Kwok (2001) suggested an asymptotically optimal ε value that is proportional to the variance of the noise. According to Cherkassky and Ma (2004), this technique has a drawback, as it does not reflect the sample size. They strongly recommended that the value of the parameter ε should be smaller when the data for larger sample sizes has the same noise level. Trevor et al. (2001) proposed estimating the parameter C using cross-validation, whereas the parameter ε can be optimally tuned based on noise density.

The standard and optimal method to choose the optimal parameters C and ε is a grid search (GS) (Ceperic et al. 2014; Liang et al. 2011). This approach was initially applied to find the near optimal parameters C and ε for classification problems based on parameter ranges (Hsu et al. 2003), but soon this grid search technique was adopted by other researchers and used additionally for regression problems (Bao et al. 2004). The main downside with the grid search strategy is that it is extremely time-consuming (Ceperic et al. 2014). Cherkassky and Ma (2004) proposed a new analytical method for the selection of the value of the parameter ε as a function of sample size, and the parameter C as a function of the dependent variable directly from the training data, rather than following the cross-validation approach commonly used in support vector machine applications. This approach takes into account the sample size and the possibility of the existence of outliers in the training data. It has good generalization performance for various types of additive noise; however, it is not suitable for Laplacian noise density (Cherkassky and Ma 2004). The Huber's loss function (least-modulus), which is a special form of ε-insensitive loss function when $\varepsilon = 0$, should be used for a Laplacian noise density model (Cherkassky and Ma 2004). Üstün et al. (2005) proposed a new

approach using genetic algorithms and simplex optimization. Frohlich and Zell (2005) proposed a new technique for the selection of parameters for regression as well as classification tasks, based on the algorithm of global optimization (EGO). Zong et al. (2006) introduced a new technique based on a particle swarm optimization (PSO) method for the selection of SVR parameters, but with this approach it is easy to fall into local optima, which would lead to a low optimization performance (Liang et al. 2011). Lahiri and Ghanta (2009) suggested a new method for the optimal tuning of the SVR parameters. This technique blends a hybrid SVR method and a differential evolution method (SVR-DE). Liang et al. (2011) proposed a new technique that they named the chaos differential evolution algorithm (CDE), which merges differential evolution (DE) with the chaotic searching algorithm.

Cherkassky and Ma (2004) concluded that there is no general agreement on the optimal method to set the SV regression parameters. Indeed, until now there has been no completely general way for selecting the parameters. Thus, in this paper we investigate a practical approach that combines the practical selection (PLSN) methods that were introduced by Cherkassky and Ma (2004) and the standard method (grid search), to enhance the generalization ability of the SV regression model. We also try to find an appropriate method for all types of additive noise including Laplacian noise density.

Section 2 contains a short introduction to SV regression. In section 3, a brief explanation of the proposed and the PLSN approaches for selecting SVR parameters is given. Section 4 shows empirical comparisons for nonlinear target functions (low- and high-dimensional) that are corrupted by two types of additive noise, Gaussian noise and non-Gaussian noise. Finally, our conclusions are given in section 5.

2. SVM for regression

The support vector machine (SVM) is a new machine learning approach that was derived from statistical learning theory and was established on the principle of structural risk minimization (SRM) (Cortes and Vapnik 1995; Dhhan et al. 2015). In the SVM the relationship between the input x and the output y is learned directly from the data, without any assumptions about the underlying probability distribution (Tezcan and Cheng 2012). Consequently, it has some obvious advantages such as being globally optimal, having good generalization ability, having a small sample size and being resistant to the over-fitting problem (Vapnik 1999).

2.1 ε-insensitive SV regression

In the support vector regression, the input x is first mapped onto a high-dimensional feature space, which is nonlinearly related to the input space, using a kernel function that transforms the input space into to a high-dimensional feature space in which non-linear relationships can be represented in a linear form (Smola and Schölkopf 2004; Vapnik 2000). The linear model (in the high-dimensional feature space) is given by

$$f(x,w) = w,\Phi(x) + b \qquad (1)$$

Where $\Phi(x)$ denotes a nonlinear function that maps x_i into a higher dimensional feature space, and w and b are the slope and bias term respectively.

The idea behind an SV regression (Vapnik 2000) is to estimate the coefficient values w and b that optimize the generalization ability (predicted risk) by minimizing the following ε-insensitive loss function

$$L_\varepsilon(y_i) = \begin{cases} 0 & \text{if } |y_i - f(x,w)| \leq \varepsilon \\ |y_i - f(x,w)| - \varepsilon & otherwise \end{cases} \qquad (2)$$

In order to find a function $f(x,w)$ that is as flat as possible and gives a deviation ε from the output (y), a smallest w would need to be found. This can be done by minimizing the Euclidean norm $\|w\|^2$ (Samui 2008; Smola and Schölkopf 2004), by introducing some positive slack variables (ξ_i, ξ_i^*), to measure deviations of the training vectors outside the ε-insensitive zone. Thus, the problem can be formulated as a convex optimization problem

$$\begin{aligned} &minimize && \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ &subject\ to && \begin{cases} y_i - f(x,w) - b \leq \varepsilon + \xi_i \\ f(x,w) + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, i = 1,2\ldots,n \end{cases} \end{aligned} \qquad (3)$$

The parameter C is used to specify the trade-off between smoothness (model complexity) and the number of deviations larger than the parameter ε that are tolerated (Smola and Schölkopf 2004; Vapnik 2000). For instance, if the value of parameter C is too large, then the optimization problem is only to minimize the empirical risk (training error), without regard to minimizing the prediction risk

(testing error), which leads to the possibility of an over-fitting phenomenon appearing (Ceperic et al. 2014; Cherkassky and Ma 2004). The parameter ε controls the width of the ε-tube, which is used to fit the training data (Vapnik 2000). If the value of ε is too large, it results in few support vectors (most data points will fit inside the ε-tube) and, consequently, in a less complex (smoother) regression function. Unfortunately, the resulting SV regression model is not continually applicable (it may result in large prediction errors) (Üstün et al. 2005). The final SVR function can be written as follows:

$$f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) k(x_i.x\ ) + b \qquad (4)$$

Where $k(x_i.x\ )$ is the kernel function that satisfies Mercer's conditions (Smola and Schölkopf 2004).

It is well known that the generalization performance of SVR (the prediction accuracy of unseen data) depends strongly on a proper setting of the hyper-parameters C and ε and the kernel parameter h. Consequently, we focus on the choice of these parameters. A radial basis function (RBF) kernel is applied, since this is the most frequently used and could provide better performance than other kernel functions (Cherkassky and Ma 2004; Liu and Hu 2013), with a width parameter h that reflects the input range (x) of the training/testing data sets (Cherkassky and Ma 2004; Schölkopf et al. 1999). The RBF kernel can be stated as in the next equation:

$$K(x\ ,x_j) = exp\left[-\frac{\|x - x_j\|^2}{2\gamma^2}\right] \qquad (5)$$

$$h = (0.1 - 0.5) * range\ (x) \qquad (6)$$

3. Proposed method for parameter selection

The proposed method selects the values of the parameters C and ε using a combination of the practical selection method (Cherkassky and Ma 2004), and the grid search method (Hsu et al. 2003). It provides high generalization accuracy with a short domain for the grid of parameters, and at the same time it is suitable for different additive noise densities.

3.1 Selection of parameter C:

Cherkassky and Ma (2004) proposed a practical approach (Eq. (7)), based on the response values (output), to choose the value for the parameter C.

$$C = |\bar{y}| + 3\sigma_y \tag{7}$$

Where $\bar{y}$ and σ_y are, respectively, the mean and standard deviation of the output (y) values of the training samples.

We propose to modify the prescription above by using its confidence interval as a short grid for the C parameter domain, using the following formula.

$$LL \leq C \leq UL \tag{8}$$

$$\begin{aligned} LL &= C - z_{\alpha l2}\sigma_c/\sqrt{n} \\ UL &= C + z_{\alpha l2}\sigma_c/\sqrt{n} \end{aligned} \tag{9}$$

The proposed modified selection of C that is given by Eq. (8) can yield a better C value since it takes several values including the value that is calculated in Eq. (7)

3.2 Selection of parameter ε:

According to Cherkassky and Ma, (2004), Cherkassky & Mulier (1998), Kwok (2001), the value of ε should be proportional to the input (additive) noise level, that is, $\varepsilon \propto \sigma$. Cherkassky and Ma (2004) proposed the following formula for the optimal ε value:

$$\varepsilon = \frac{3\,\sigma}{\sqrt{n}} \tag{10}$$

The standard deviation of the additive noise σ is known or can be estimated directly from the data (Cherkassky and Ma 2004). These authors noted that this works well for small samples, based on several empirical comparisons, and that it yields ε values that are too small (they are practically equal to zero) for a large sample size n. However, they replace the formula in Eq. (10) by a new formulation:

$$\varepsilon = 3\,\sigma\sqrt{\frac{\ln n}{n}} \tag{11}$$

However, to find short grid domain for the parameter ε, we propose using both formulations (Eqs. (10) and (11)) as lower and upper bounds. Since Eq. (10) is practically equal to zero, we propose to replace it by zero (ε = 0) to include

Huber's loss function (least-modulus) that is convenient for Laplacian noise density (Cherkassky and Ma 2004; Huber 1964).

$$0 \leq \varepsilon \leq 3\,\sigma\sqrt{\frac{\ln n}{n}} \qquad (12)$$

This is what makes the proposed method suitable for all kinds of additive noise densities.

4. Numerical studies for non-linear target functions

This section illustrates some empirical comparisons for a nonlinear regression, with Gaussian and non-Gaussian noise. The performance is measured by looking at the prediction risk (Eq. 13), which is defined as the mean squared error (MSE) between SVR estimates and the true values of the dependent variable for test inputs.

$$\mathrm{MSE} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \qquad (13)$$

We consider the prediction risk to compare the results more confidently: the mean squared testing error (MSET). The prediction risk is averaged over 100 replications of random data sets. Comparison results are obtained by applying the proposed method and the practical selection (PLSN) method for the same data sets.

The performances of the two methods (the proposed method and the PLSN method) are tested on three different sample sizes. Space limitations mean that we are confined to illustrate this by a single sample per example. The other results can be obtained by E-mail from the author. All calculations were carried out using R software.

4.1 Numerical studies for Gaussian noise

We investigate three different regression models. The first example is for a univariate regression, and the next two examples are for multivariate regressions (two- and five-dimensional target functions). The x-values of the data are sampled according to a uniform distribution, and the y-values are generated based on different statistical models. The y-values of the data are corrupted by additive noise (r_i) with zero mean and standard deviation (σ). For the multivariate d-dimensional problems, all d input variables are scaled to the [0-1] range. This procedure yields good SVR performance for various data sets.

4.1.1 Example 1

The first example compares the results based on the univariate target function in Eq. (14) with different noise levels (σ). Four data sets are generated for a small training sample ($n = 30$), with testing sample ($n_t = 146$). The RBF kernel is used with width parameter $h = 4$.

$$y = \frac{sin(x)}{x} + \ r_i \quad with \qquad x \in [-10,10] \qquad (14)$$

The results of applying the proposed method and the PLSN method for these data sets are shown in Table 1 and Figure 1, respectively.

Table 1: Results for univariate target function (*n*=30, *nt*=146)

Data set	Methods	Noise level	C-selection	ε -selection	MSET	%SV
1	PLSN	0.02	1.21	0.02	0.2821	77
	Proposed		1.09 - 1.34	0 - 0.02	**0.2681**	77
2	PLSN	0.2	1.30	0.2	0.9347	50
	Proposed		1.16 - 1.44	0 – 0.2	**0.5356**	67
3	PLSN	0.5	1.77	0.5	2.7876	**30**
	Proposed		1.57 - 1.96	0 – 0.5	**2.0587**	100
4	PLSN	1	2.89	1.01	9.4689	**30**
	Proposed		2.57 - 3.22	0 – 1.01	**6.3164**	57

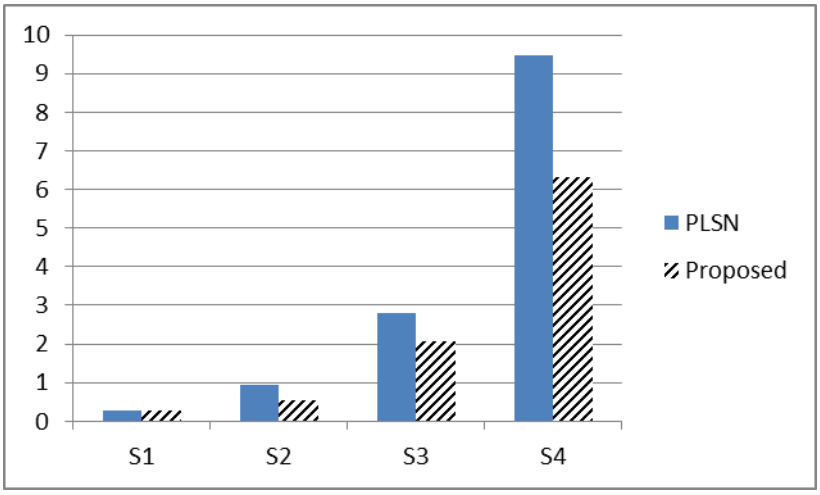

Figure 1: The MSET for univariate target function (*n*=30, *nt*=146)

From Table 1, we see that the proposed technique for selecting C and ε is better than PLSN, as it yields lower MSET (providing smallest prediction risk). On the other hand, it gives support vectors for around 50% of the training data, whereas the PLSN has a reduced SV percentage far below 50% when the noise level is increased. In this case, the resulting SV regression model is not always applicable (Üstün et al. 20005).

Figure 1 clearly shows that the proposed method gives better accuracy than the PLSN method. The criterion of prediction risk shows the superiority of the proposed method over the PLSN method. In the case of low noise level, the difference between the two methods is not great, but it grows when the noise level is increased.

4.1.2 Example 2

This example demonstrates the results of the SV regression parameter selection for multivariate problems. Different noise levels (σ) are used to generate data sets for a two-dimensional target function (Eq. 15) with training and testing samples ($n = 100, n_t = 200$), respectively. The RBF kernel is utilized with width parameter $h = 2$.

$$y = \frac{\sin\sqrt{x_1^2 + x_2^2}}{\sqrt{x_1^2 + x_2^2}} + r_i \qquad with \quad x \in [-5,5] \qquad (15)$$

Table 2 compares the results of the proposed method with those using the PLSN method. The prediction risk that is calculated based on the different noise levels for the two methods (proposed and PLSN) are represented in Table 2 and graphically in Figure 2.

Table 2: Results for two-dimensional target function (*n*=100, *nt*=200)

Data set	Methods	Noise level	C-selection	ε -selection	MSET	%SV
1	PLSN	0.1	1.29	0.064	0.0692	63
	Proposed		1.25 - 1.32	0 - 0.064	**0.0512**	69
2	PLSN	0.2	1.47	0.13	0.2446	61
	Proposed		1.42 - 1.52	0 - 0.13	**0.1815**	64
3	PLSN	0.3	1.71	0.19	0.5395	61
	Proposed		1.65 - 1.78	0 - 0.19	**0.3866**	61
4	PLSN	0.4	1.98	0.26	0.9352	61
	Proposed		1.88 - 2.07	0 - 0.26	**0.5341**	66

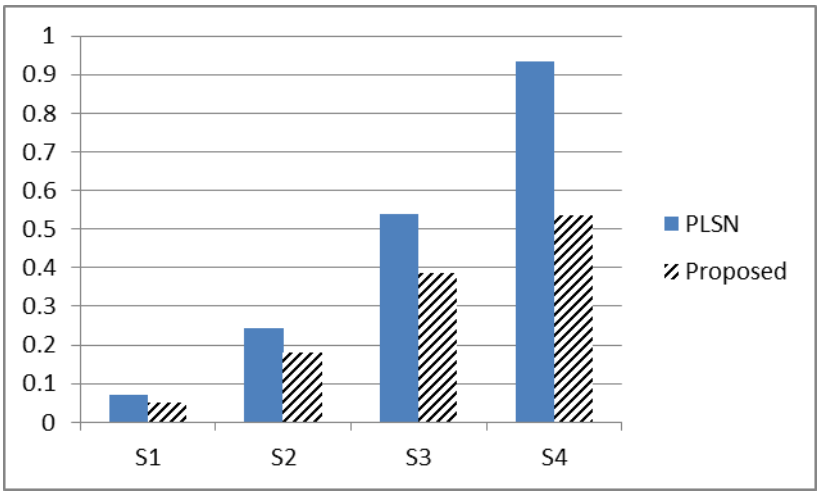

Figure 2: The MSET for two-dimensional target function (*n*=100, *nt*=200)

From the empirical results in Table 2, the proposed method has better performance than the PLSN method. Figure 2 shows that the proposed method has better generalization ability (prediction risk) than the PLSN method, in the case of testing error. One can see that the difference between the two methods increases with increasing noise levels.

4.1.1 Example 3

In order to show the performance of the proposed approach, we consider a higher-dimensional target function (Eq. 16) with x-values generated in the interval [-2, 2]. The response values are corrupted by different additive noise levels (σ). The results are obtained by applying SV regression for parameter selection with training sample ($n = 250$) and testing sample ($n_t = 500$).

$$y = \sin(x_1 x_2) + (x_3 - 0.5)^2 + x_4 + 0.5x_5 + r_i \qquad (16)$$

The results between the estimates for the proposed method and the estimates obtained by utilizing the PLSN approach are compared in Table 3 in terms of prediction risk (MSET) and the percentage of support vectors. Clearly, the parameter selection based on the proposed method is more effective in terms of prediction risk. The same conclusion, that the proposed method is more effective than the PLSN method, can be drawn from Figure 3.

Table 3: Results for higher-dimensional target function (*n*=250, *nt*=500)

Data set	Methods	Noise level	C-selection	ε -selection	MSET	%SV
1	PLSN	0.1	2.26	0.045	0.8041	95
	Proposed		2.17 - 2.35	0 - 0.045	**0.7898**	98
2	PLSN	0.2	2.33	0.09	0.8848	88
	Proposed		2.24 - 2.42	0 – 0.09	**0.8107**	97
3	PLSN	0.5	2.74	0.22	1.4630	78
	Proposed		2.63 - 2.85	0 – 0.22	**1.2840**	96
4	PLSN	0.6	2.92	0.27	1.7729	75
	Proposed		2.81 – 3.04	0 – 0.27	**1.6845**	90

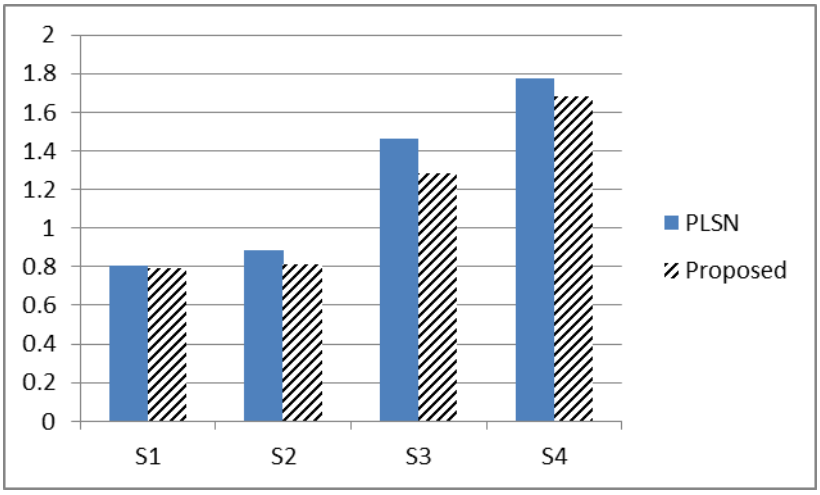

Figure 3: The MSET for higher-dimensional target function (*n*=250, *nt*=500)

4.2 Numerical studies for non-Gaussian noise

In this section, the proposed technique is compared with the PLSN method for parameter selection when the response variable is contaminated by non-Gaussian noise. The x-values of the data are sampled based on a uniform distribution, and the y-values are generated based on two statistical models (univariate and bivariate target functions). The output values (response) of the data are corrupted by additive non-Gaussian noise (Student's t-distribution noise) with different degrees of freedom (DOF), 50 and 100.

4.2.1 Example 4

In this example the univariate target function (Eq. 17) with different noise levels (σ) is used to generate four data sets for a small training sample ($n = 30$), with testing sample ($n_t = 60$) and DOF = 50. The RBF kernel is used with width parameter $h = 2$.

$$y = \frac{1}{3}x^2 + r_i \qquad with \qquad x \in [-2,2] \tag{17}$$

Table 4 demonstrates the comparison of the regression estimates that are obtained using the proposed method and the PLSN method. To read the results more clearly, the prediction risk (MSET) for the two methods (proposed and PLSN) are displayed graphically in Figure 4. It can be seen from Table 4 that, for the PLSN method, the percentage of support vectors decreases when the noise level is increased, which leads to weak generalization ability (high prediction errors). On the other hand, the superior performance of the proposed method over the PLSN method can be observed in Figure 4. From the prediction risks and the percentage of support vectors, the proposed method is more effective than the PLSN approach.

Table 4: Results for univariate target function (*n*=30, *nt*=60, *DOF*=50)

Data set	Methods	Noise level	C-selection	ε -selection	MSET	%SV
1	PLSN	0.1	2.53	0.1	0.1324	43
	Proposed		2.38 - 2.68	0 - 0.1	**0.1086**	60
2	PLSN	0.2	2.57	0.2	0.3127	33
	Proposed		2.41 - 2.73	0 – 0.2	**0.2529**	57
3	PLSN	0.4	2.78	0.4	0.8849	30
	Proposed		2.59 - 2.97	0 – 0.4	**0.6939**	77
4	PLSN	0.6	3.11	0.6	1.8209	27
	Proposed		2.88 - 3.33	0 – 0.6	**1.6343**	63

4.2.2 Example 5

The second comparison, in this section, is for a two-dimensional target function (Eq. 18). Different noise levels (σ) are used to generate four data sets for a training sample and testing sample ($n = 100, n_t = 200$), respectively, with DOF equal to 100. The comparison of the two methods (proposed and PLSN) is explained in Table 5 and Figure 5. It can be seen that the proposed approach shows better performance than the PLSN method, in terms of mean square testing error.

$$y = sin(\pi x_1 x_2) + r_i \qquad with \qquad x \in [-12,12] \tag{18}$$

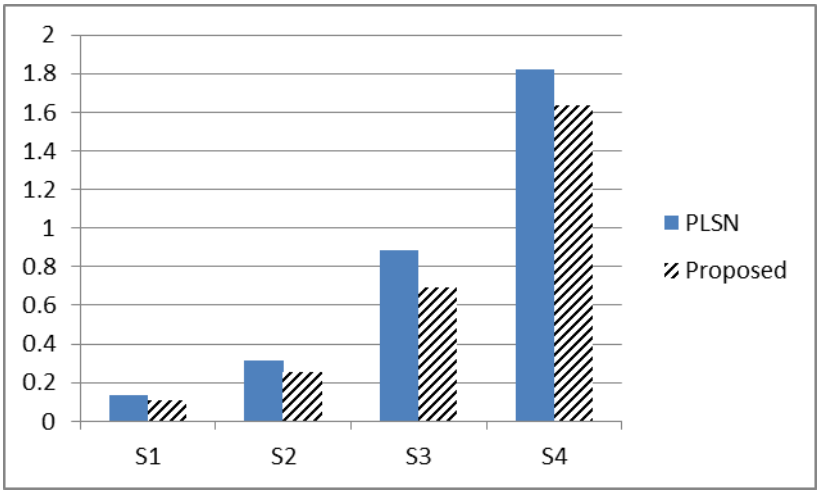

Figure 4: The MSET for univariate target function (*n*=30, *nt*=60, *DOF*=50)

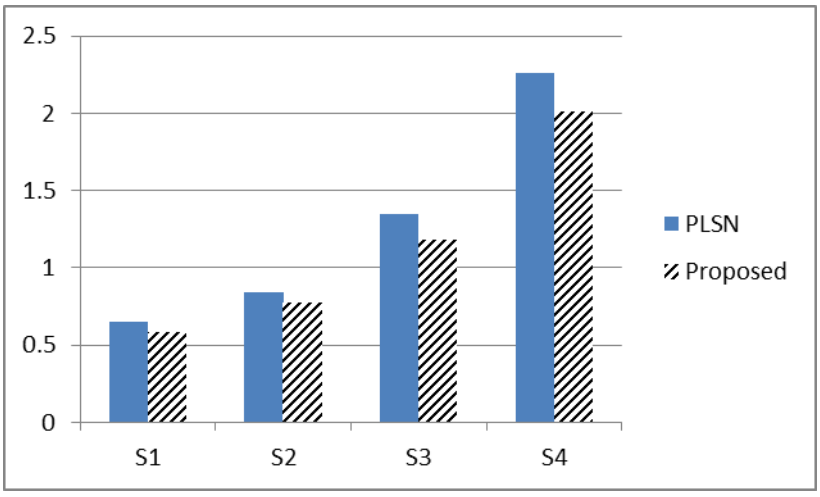

Figure 5: The MSET for two-dimensional target function (*n*=100, *nt*=200, *DOF*=100)

Table 5: Results for two-dimensional target function (*n*=100, *nt*=200, *DOF*=100)

Data set	Methods	Noise level	C-selection	ε -selection	MSET	%SV
1	PLSN	0.1	2.46	0.064	0.6539	81
	Proposed		2.32 - 2.59	0 – 0.064	**0.5807**	99
2	PLSN	0.2	2.55	0.13	0.8418	75
	Proposed		2.40 - 2.69	0 – 0.13	**0.7745**	82
3	PLSN	0.4	2.80	0.26	1.3440	64
	Proposed		2.64 - 2.96	0 – 0.26	**1.1829**	92
4	PLSN	0.6	3.14	0.39	2.2639	63
	Proposed		2.95 - 2.32	0 – 0.39	**2.0114**	66

5. Conclusion

This article describes a hybrid technique that combines the PLSN method and the grid search procedure for setting the free parameters for an SV regression. This combination is to minimize the time duration of the perform the grid search technique and to enhance the generalization ability of the practical selection of Cherkassky and Ma (2004).The proposed approach finds the optimal values of the parameters C and ε based on a short domain (grid) of parameters. A number of empirical comparisons demonstrate that the proposed approach for selecting the parameters yields better generalization performance of SVR than the PLSN for various target functions, noise levels, types of noise and sample sizes.

REFERENCES

[1] **Bao, Y., Lu, Y. & Zhang, J. (2004),** ***Forecasting stock price by SVMs regression;*** Artificial intelligence: Methodology, systems and applications, AIMSA 2004, Varna, Bulgaria, (pp. 295-303), Berlin: *Springer-Verlag;*
[2] **Ceperic, V., Gielen, G. & Baric, A. (2014),** ***Sparse\ Varepsilon-tube Support Vector Regression by Active Learning*****;** Soft Computing, 18:1113-1126;
[3] **Cherkassky, V. & Ma, Y. (2004),** ***Practical Selection of SVM Parameters and Noise Estimation for SVM Regression;*** Neural Networks, 17:113-126;
[4] **Cherkassky, V. & Mulier, F. M. (1998),** ***Learning from Data: Concepts, Theory and Methods*****;** Hoboken, New Jersey: *John Wiley & Sons*;
[5] **Cortes, C. & Vapnik, V. (1995),** ***Support-vector Networks*****;** Machine Learning, 20:273-297;

[6] **Dhhan,W., Rana, S. and H. Midi (2015),** ***Non-sparse ε-Insensitive Support Vector Regression for Outlier Detection*****;** Journal of Applied Statistics, 42(8) (2015), 1723-1739;
[7] **Frohlich, H. & Zell, A. (2005),** ***Efficient Parameter Selection for Support Vector Machines in Classification and Regression via Model-Based Global Optimization*****;** (pp. 1431-1436). Proceedings. 2005 IEEE International Joint Conference Neural Networks, IJCNN'05, Montreal, Canada;
[8] **Hsu, C., Chang, C. & Lin, C. (2003), A** ***Practical Guide to Support Vector Classification;*** Department of Computer Science and Information Engineering. National Taiwan University, Taipei, Technical report;
[9] **Huber, P. J. (1964),** ***Robust Estimation of a Location Parameter*****;** The Annals of Mathematical Statistics, 35:73-101;
[10] **Kwok, J. T. (2001),** ***Linear Dependency between ε and the Input Noise in ε-Support Vector Regression*****;** Artificial Neural Networks—ICANN 2001, (pp. 405-410), Vienna: *Springer*;
[11] **Lahiri, S. K. & Ghanta, K. C. (2009),** ***Support Vector Regression with Parameter Tuning Assisted by Differential Evolution Technique: Study on Pressure Drop of Slurry Flow in Pipeline;*** Korean Journal of Chemical Engineering, 26:1175-1185;
[12] **Liang, W., Zhang, L. & Wang, M. (2011),** ***The Chaos Differential Evolution Optimization Algorithm and Its Application to Support Vector Regression Machine*****;** Journal of Software, 6:1297-1304;
[13] **Liu, J. N. & Hu, Y. (2013),** ***Support Vector Regression with Kernel Mahalanobis Measure for Financial Forecast.*** In: Pedrycz, Witold, Chen, Shyi-Ming, eds. Time series analysis, Modeling and applications, (vol. 47), (pp. 215-227). Berlin: *Springer-Verlag*;
[14] **Mattera, D. & Haykin, S. (1999),** ***Support Vector Machines for Dynamic Reconstruction of a Chaotic System.*** In: Schölkopf, B., Burges, C., Smola, A. eds. *Advances in Kernel Methods: Support vector learning*, (pp. 211-241), Cambridge, MA: *MIT Press*;
[15] **Samui, P. (2008),** ***Slope Stability Analysis: A Support Vector Machine Approach;*** Environmental Geology, 56: 255-267;
[16] **Schölkopf, B., Burges, C. J. & Smola, A. J. (1999),** ***Advances in Kernel Methods: Support Vector Learning;*** Cambridge, MA: *MIT Press*;
[17] **Smola and Schölkopf, B. (2004),** ***A Tutorial on Support Vector Regression;*** Statistics and Computing, 14:199-222;
[18] **Tezcan, J. & Cheng, Q. (2012),** ***Support Vector Regression for Estimating Earthquake Response Spectra;*** Bulletin of Earthquake Engineering, 10:1205-1219;

[19] **Trevor, H., Robert, T. & Jerome, F. (2001),** ***The Elements of Statistical Learning: Data Mining, Inference and Prediction;*** New York: *Springer*;
[20] **Üstün, B., Melssen, W., Oudenhuijzen, M. & Buydens, L. (2005),** ***Determination of Optimal Support Vector Regression Parameters by Genetic Algorithms and Simplex Optimization;*** Analytica Chimica Acta, 544: 292-305;
[21] **Vapnik, V. (1999),** ***An Overview of Statistical Learning Theory*****;** IEEE Transactions on Neural Networks, 10: 988-999;
[22] **Vapnik, V. (1998),** ***Statistical Learning Theory*****;** New York: *Wiley*;
[23] **Vapnik, V. (2000),** ***The Nature of Statistical Learning Theory*****, 2nd ed.;** New York: *Springer*;
[24] **Wang, W., Xu, Z., Lu, W. & Zhang, X. (2003),** ***Determination of the Spread Parameter in the Gaussian Kernel for Classification and Regression*****;** Neurocomputing, 55: 643-663;
[25] **Zong, Q., Liu, W. & Dou, L. (2006),** ***Parameters Selection for SVR Based on PSO;*** Intelligent Control and Automation, 2006. WCICA 2006. The 6th World Congress on, pp. (2811-2814) Dalian, China, IEEE .