Extreme Support Vector Regression

Wentao Zhu [a], Jun Miao [a], Laiyun Qing [b,*]

[a] *Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*

[b] *School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China*

Abstract

Extreme Support Vector Machine (ESVM), a variant of ELM, is a nonlinear SVM algorithm based on regularized least squares optimization. In this paper, a regression algorithm, Extreme Support Vector Regression (ESVR), is proposed based on ESVM. Experiments show that, ESVR has a better generalization ability than the traditional ELM. Furthermore, ESVM can reach comparable accuracy as SVR and LS-SVR, but has much faster learning speed.

Key words: Extreme learning machine, support vector regression, extreme support vector machine, extreme support vector regression, regression.

1 Introduction

Extreme Learning Machine (ELM) is a great successful algorithm for both classification and regression. It has the good generalization ability at an extremely fast learning speed [1]. Moreover, ELM can overcome some challenging issues that other machine learning algorithms face [1]. Some desirable advantages can be found in ELM such as, extremely fast learning speed, less human intervene and great computational scalability. The essence of ELM is that the hidden layer parameters need not be tuned iteratively and the output weights can be simply calculated by least square optimization [2,3]. Extreme Learning Machine (ELM) has attracted a great number of researchers and engineers [4–8] recently.

Extreme Support Vector Machine (ESVM), a kind of single hidden layer feed forward network, has the same extremely fast learning speed, but it has

* Corresponding author
Email address: lyqing@ucas.ac.cn (Laiyun Qing).

a better generalization ability than ELM [9] on classification tasks. ESVM, a special form of Regularization Network (RN) derived from Support Vector Machine (SVM), has the same advantages as ELM such as, that hidden layer parameter can be randomly generated [9]. Due to these ideal properties, many researches have been conducted on ESVM [10–13]. In fact, ESVM is a variant of ELM. However, ESVM in [9] cannot be applied to regression tasks.

In this paper, Extreme Support Vector Regression (ESVR) algorithm was proposed for regression. Our ESVR algorithm is based on the ESVM model and the essential of ELM for regression is utilized. Some comparison experiments show that the ESVR algorithm has quite good generalization ability and the learning speed of ESVR is quite large.

This paper is organized as follows. ELM and ESVM are briefly reviewed in Section 2. The linear ESVR, nonlinear ESVR are proposed in Section 3. Performances of ESVR compared with ELM, SVR and LS-SVR are verified in Section 4.

2 Extreme Support Vector Machine

We here briefly introduce the basic concept of ELM and Extreme Support Vector Machine (ESVM). ELM can reach not only the smallest training errors, but also the best generalization ability [14]. ESVM is based on regularization least squares in the feature space. The performance of ESVM is better than ELM on classification tasks [9].

2.1 Extreme Learning Machine

ELM is a single hidden layer forward network (SLFNs). The parameters of the hidden layer can be randomly generated, and need not be iteratively tuned [2,3]. The least square optimization process tackles the output weight vector [2,3]. Therefore, the learning speed of ELM is extremely fast. Moreover, ELM has the unified algorithm to tackle classification and regression problems.

For N arbitrary distinct samples $(\mathbf{x}_i, \mathbf{t}_i) \in (\mathbf{R}^d \times \mathbf{R}^m)$, where $\mathbf{x}_i$ is the extracted feature vector, and $\mathbf{t}_i$ is the target output. For the SLFNs, the mathematical model with L hidden nodes is

$$\sum_{i=1}^{L} \beta_i g_i(\mathbf{x}_j) = \sum_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}_j) = \widehat{\mathbf{t}}_j, j = 1, ..., N, \tag{1}$$

where $\widehat{\mathbf{t}}_j$ is the output of the SLFNs, and $G(\mathbf{a}_i, b_i, \mathbf{x}_j)$ is the hidden layer feature mapping. According to [3], the hidden layer parameters $(\mathbf{a}_i, b_i)$ can be randomly generated.

The goal of ELM is to approximate the expected targets by the above

predicted targets. That is,

$$\|\mathbf{H}\widehat{\beta} - \mathbf{t}\| = \min_{\beta} \|\mathbf{H}\beta - \mathbf{t}\|, \tag{2}$$

where

$$\mathbf{H} = \begin{bmatrix} G(\mathbf{a}_1, b_1, \mathbf{x}_1) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ G(\mathbf{a}_1, b_1, \mathbf{x}_N) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_N) \end{bmatrix}_{N \times L}, \beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N \times m}.$$

Therefore, the least square method can be used to solve the above optimization problem. That is to say, the output weight β can be obtained by the following equation.

$$\widehat{\beta} = \mathbf{H}^{\dagger}\mathbf{T}, \tag{3}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of matrix $\mathbf{H}$ [15].

From the above discussion, ELM can be implemented by the following steps. First, randomly generate hidden node parameters $(\mathbf{a}_i, b_i), i = 1, ..., L$, where L is the parameter of ELM denoting the number of hidden nodes. Second, calculate the hidden layer mapped feature matrix $\mathbf{H}$ as the above equation. Third, calculate the output weight by the least square optimization.

2.2 Extreme Support Vector Machine

Instead of using kernels to represent data features by SVM, ESVM explicitly utilizes SLFNs to map the input data points into a feature space [9]. ESVM is a variant of ELM [16]. The essential of ESVM is a kind of regularization network. Similar to ELM, ESVM has a number of advantages, such as, fast learning speed, good generalization ability and fewer human intervene.

The model of ESVM can be obtained by replacing the inequality constraint in the traditional SVM with the equality constraint [9].

$$\begin{aligned} \min_{(\mathbf{w},r,\mathbf{y}) \in \mathbf{R}^{\tilde{n}+1+m}} & \frac{\nu}{2}\|\mathbf{y}\|^2 + \frac{1}{2}\left\| \begin{bmatrix} \mathbf{w} \\ r \end{bmatrix} \right\|^2 \\ s.t. \quad & D(\Phi(A)\mathbf{w} - r\mathbf{e}) + \mathbf{y} = \mathbf{e} \end{aligned} \tag{4}$$

In the above equation, $\Phi(\mathbf{x}) : \mathbf{R}^n \to \mathbf{R}^{\tilde{n}}$ is the feature mapping function in the hidden layer of SLFNs. $\mathbf{y}$ is the slack variable of the model. ν is the tradeoff parameter between allowable errors and the minimization of weights, and $\mathbf{e}$ is a vector of size $m \times 1$ which is filled with 1s, where m is the number of the samples. D is the diagonal matrix of the element of 1 or -1 denoting the labels. A is the sample data matrix.

After deduction, the solution of the model is simply equivalent to calculating the following expression according to [9]:

$$\begin{bmatrix} \mathbf{w} \\ r \end{bmatrix} = (\frac{\mathbf{I}}{\nu} + {E_\Phi}^T E_\Phi)^{-1} {E_\Phi}^T D\mathbf{e}, \tag{5}$$

where $E_\Phi = [\Phi(A), -\mathbf{e}] \in \mathbf{R}^{m\times(\tilde{n}+1)}$.

ESVM can reach better generalization ability than ELM almost in all classification tasks [9]. Due to the simple solution, ESVM can learn at an extremely fast speed. Additionally, the activation functions can be explicitly constructed. However, diagonal label matrix D must be constructed in the above ESVM model and D must be with the element of 1 or -1 in the above deduction, which means that the ESVM model cannot be applied to multi-class classification or regression tasks directly.

3 Extreme Support Vector Regression

In this section, we will extend ESVM from classification tasks to regression tasks. The linear and nonlinear extreme support vector regression will be proposed.

3.1 The Linear Extreme Support Vector Regression

Our model is derived from the formulation of ESVM. Similar to ESVM, ESVR also replaces the inequality constraint of the ϵ-SV regression with the equality constraint [17]. But different from ESVM, the diagonal target output matrix need not be constructed. The model of ESVR is constructed as follows.

$$\begin{aligned} \min_{(\mathbf{w},r,\mathbf{y})\in\mathbf{R}^{\tilde{n}+1+m}} & \frac{\nu}{2}\|\mathbf{y}\|^2 + \frac{1}{2}(\mathbf{w}^T\mathbf{w} + r^2) \\ s.t. \quad & A\mathbf{w} - r\mathbf{e} - \mathbf{T} = \mathbf{y} \end{aligned}, \tag{6}$$

where $\mathbf{T}$ is the expected target output of the sample data matrix A.

We will provide the solution of the above ESVR model. If $\mathbf{w}, r$ have been obtained, the test process is to calculate $\mathbf{x}^T\mathbf{w} - r$ to get the output target of the sample. Nonlinear ESVR also will be supplied by introducing a nonlinear feature mapping function in the following section.

3.2 The Nonlinear Extreme Support Vector Regression

Nonlinear ESVR can be obtained by simply replace the original data matrix A by the transformed matrix $\Phi(A)$.

$$\begin{aligned} \min_{(\mathbf{w},r,\mathbf{y})\in\mathbf{R}^{\tilde{n}+1+m}} \frac{\nu}{2}\|\mathbf{y}\|^2 + \frac{1}{2}\left\|\begin{bmatrix}\mathbf{w}\\ r\end{bmatrix}\right\|^2 \\ s.t.\ \ \Phi(A)\mathbf{w} - r\mathbf{e} - \mathbf{T} = \mathbf{y} \end{aligned} \tag{7}$$

After deduction, analytical solution can be obtained.

If $m < \tilde{n} + 1$, we can obtain a simple analytical solution of $\mathbf{w}$ and r.

$$\begin{bmatrix}\mathbf{w}\\ r\end{bmatrix} = {E_\Phi}^T\mathbf{s} = {E_\Phi}^T(\frac{\mathbf{I}}{\nu} + E_\Phi {E_\Phi}^T)^{-1}\mathbf{T} \tag{8}$$

If $m > \tilde{n} + 1$,

$$\begin{bmatrix}\mathbf{w}\\ r\end{bmatrix} = {E_\Phi}^T\mathbf{s} = (\frac{\mathbf{I}}{\nu} + {E_\Phi}^T E_\Phi)^{-1}{E_\Phi}^T\mathbf{T} \tag{9}$$

,where $E_\Phi = [\Phi(A), -\mathbf{e}] \in \mathbf{R}^{m\times(\tilde{n}+1)}$.

From the above discussion, the algorithm of ESVR can be explicitly concluded as follows. Firstly,randomly generate hidden layer parameters and choose an activation function. $\Phi(A)$ can be obtained. Secondly, construct the matrix $E_\Phi = [\Phi(A), -\mathbf{e}]$. Thirdly, choose some positive parameters ν to calculate $\begin{bmatrix}\mathbf{w}\\ r\end{bmatrix}$ by expression (8) or (9). When a new instance $\mathbf{x}$ comes, we can use $\Phi(\mathbf{x})^T\mathbf{w} - r$ to predict it.

3.3 The Essence of ESVR

Inspired by support vector theory in SVM, ESVR is an proximal algorithm of SVR. Intuitively, we replace the inequality constraints in ϵ-SV regression with equality constraints. The following equation is the ϵ-SV regression constraints formula [17,18].

$$\begin{aligned} T_i - \langle w, x_i\rangle + r &\le \epsilon + y_i \\ \langle w, x_i\rangle - r - T_i &\le \epsilon + y_i^\star \\ y_i, y_i^\star &\ge 0 \end{aligned} \tag{10}$$

Actually, the replacement is a proximal method and proximal decision plane is obtained in the ESVR.

After deduction, the analytical solution of ESVR is quite similar to that of ELM. Compared to the algorithm of ELM, ESVR is similar to regularized ELM besides a biased term. However, the generalization performance of ESVR is better than that of ELM, SVR and LS-SVR. The technique used in ESVR is quite important for overcoming ill-pose problems and singular problems that traditional ELM may encounter [19]. Furthermore, ESVR has the desirable features as that of ELM such as, fast learning speed, fewer human intervene. From the computation view, ESVR is a variant of ELM. Such random parameters are utilized in the ESVR. ESVR has the similar form of that of regularized ELM.

4 Performance Verification

In this section, the performance of ESVR is compared with ELM, SVR and LS-SVR on some benchmark regression problems data sets.

4.1 Experimental Conditions

All the simulations for ESVR, ELM, SVR and LS-SVR for regression algorithms were carried out in MATLAB R2010a environment running in a Xeon E7520, 1.87GHZ CPU. The codes used for ELM, SVR and LS-SVR were downloaded from [1], [2], and [3] respectively.

In order to extensively verify the performance of ESVR, ELM, SVR and LS-SVR, twelve data sets of different sizes and dimensions were downloaded from UC Irvine Machine Learning Repository [4] or StatLib library [5] for simulation. These data sets can be divided into three categories according to different sizes and feature dimensions. Baskball, Strike, Cloud, and Autoprice are of small size and low dimensions. Pyrim, Housing, Bodyfat, and Cleveland are of small size and medium dimensions. Balloon, Quake, Space-ga, and Abalone are of large size and low dimensions. Table 1 lists some features of the regression data sets in our simulation.

In the experiments, three fold cross validation was conducted to select parameters. The best parameters ν of ESVR, the cost factor C and kernel parameter γ of SVR, LS-SVR were obtained from the candidate sequence

[1] http://www.ntu.edu.sg/eee/icis/cv/egbhuang.html
[2] http://asi.insarouen.fr/enseignants/ arakotom/toolbox/index.html
[3] http://www.esat.kuleuven.be/sista/lssvmlab/
[4] http://archieve.ics.uci.edu/ml/
[5] http://lib.stat.cmu.edu/

Table 1
Specification of regression problems

Datasets	# Attributes	# Training data	# Testing data
Baskball	4	64	32
Cloud	9	72	36
Autoprice	9	106	53
Strike	6	416	209
Pyrim	27	49	25
Bodyfat	14	168	84
Cleveland	13	202	102
Housing	13	337	169
Balloon	2	1334	667
Quake	3	1452	726
Space-ga	6	2071	1036
Abalone	8	2784	1393

$2^{-25}, 2^{-24}, ..., 2^{23}, 2^{24}, 2^{25}$. The number of hidden layer nodes $\tilde{n}$ in ESVR was obtained from $[10, 300]$ with step 10. The average performance of testing Root Mean Square Errors (RMSE) was conducted as the evaluation metric to select the best parameters. And all the data sets were normalized into $[-1, 1]$ before the regression process. The kernel function used in the experiments was the RBF function. The activation function of ESVR was sigmoidal function.

4.2 Performance comparison on benchmark datasets

Comparisons of generalization performance between ESVR and ELM on the above twelve different benchmark regression data sets were firstly carried out. Nonlinear models with sigmoidal additive feature map function were used for comparison. Ten round experiments of the same parameters were conducted to obtain an average performance evaluation in each fold due to randomly selecting parameters in the hidden layer. Figure 1 is the testing RMSE of ESVR and ELM with different numbers of hidden nodes on six of the twelve real world data sets.

Figure 1 shows the testing RMSE of ESVR is lower than that of ELM. We can observe that the performance of ELM is varied greatly with the number of hidden nodes as well. Moreover, the standard deviation of ELM is much larger than that of ESVR. The result of the experiment reveals that the generalization of ESVR is better than that of ELM. Furthermore, ESVR is more stable than ELM from Figure 1, because the slack variable added can make our model more stable in the ESVR.

The second experiment was conducted to compare the performances of ESVR, SVR and LS-SVR. In this experiment, performances of ESVR algorithm were validated compared with SVR and LS-SVR. The same kernel function (RBF function) was used for SVR and LS-SVR. The activation function

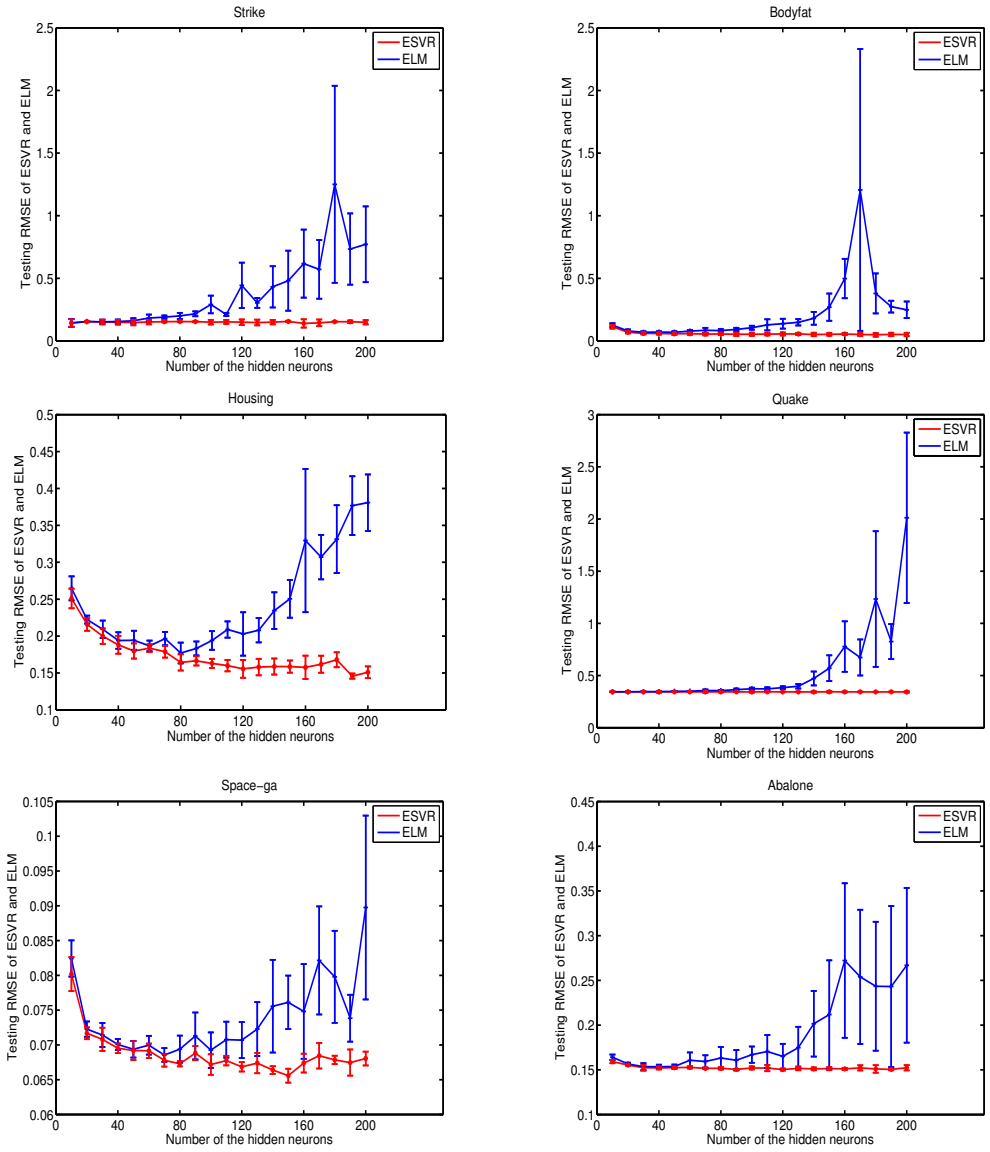

Figure 1. Testing RMSE of ESVR and ELM

of ESVR was sigmoidal function. Through three fold cross validation, the best parameters (C, γ) or $(\nu, \tilde{n})$ were obtained. Table 2 records parameters of different models on different data sets.

Table 3 is the performance results of ESVR, SVR and LS-SVR. Training time and testing RMSE were recoded as the learning speed and generalization ability of the model separately. The best results for different data sets were emphasized into bold face.

Table 3 shows that the testing RMSE of ESVR is the lowest in most of the data sets. The training time of ESVR is much less than that of SVR and LS-SVR especially in the large scale data instances. These results reveal that, ESVR has comparable generalization ability than that of SVR and LS-SVR. Furthermore, the average learning speed of ESVR can reach at least three

Table 2
Parameters of ESVR, SVR and LS-SVR

Algorithms	SVR		LS-SVR		ESVR	
	C	γ	C	γ	ν	$\tilde{n}$
Baskball	2^{10}	2^{2}	2^{0}	2^{2}	2^{4}	250
Cloud	2^{20}	2^{7}	2^{25}	2^{16}	2^{3}	170
Autoprice	2^{19}	2^{5}	2^{9}	2^{7}	2^{5}	290
Strike	2^{-3}	2^{-2}	2^{-1}	2^{-1}	2^{1}	250
Pyrim	2^{0}	2^{2}	2^{3}	2^{3}	2^{-1}	260
Bodyfat	2^{6}	2^{3}	2^{9}	2^{7}	2^{2}	300
Cleveland	2^{22}	2^{13}	2^{22}	2^{25}	2^{-5}	240
Housing	2^{6}	2^{1}	2^{6}	2^{3}	2^{7}	280
Balloon	2^{3}	2^{1}	2^{25}	2^{5}	2^{25}	260
Quake	2^{1}	2^{-12}	2^{-1}	2^{-15}	2^{0}	40
Space-ga	2^{3}	2^{-1}	2^{11}	2^{3}	2^{19}	300
Abalone	2^{-1}	2^{-1}	2^{2}	2^{2}	2^{9}	150

Table 3
Performance comparisons of SVR, LS-SVR and ELM

Algorithms	SVR		LS-SVR		ESVR	
	Testing RMSE	Training Time(s)	Testing RMSE	Training Time(s)	Testing RMSE	Training Time(s)
Baskball	0.2567	0.1029	0.2568	0.0049	**0.2521**	0.0208
Cloud	0.1729	0.0774	0.1810	0.0065	**0.1582**	0.0115
Autoprice	0.1381	0.1328	**0.1359**	0.0072	0.1561	0.0365
Strike	**0.1443**	0.9707	0.1472	0.0541	0.1497	0.0641
Pyrim	**0.2151**	0.0336	0.2159	0.0051	0.2184	0.0240
Bodyfat	0.0514	0.0485	**0.0502**	0.0128	0.0506	0.0458
Cleveland	**0.4267**	0.2690	0.4333	0.0147	0.4279	0.0365
Housing	0.1469	0.7729	0.1458	0.0455	**0.1409**	0.0771
Balloon	0.0242	7.8253	0.0099	1.0798	**0.0098**	0.1932
Quake	0.3438	205.7426	**0.3425**	2.4292	0.3440	0.0146
Space-ga	**0.0654**	92.1705	0.0665	2.5293	0.0661	0.3677
Abalone	0.1519	250.4772	**0.1486**	9.6423	0.1510	0.1875

times of that of LS-SVR, and at least ten times of that of SVR on the above real world benchmark data sets. The reason that ESVR is much faster is the same as that why ELM has an extremely fast learning speed. The solution of ESVR is an analytical equation. The learning process is simply to solve an least square expression.

5 Conclusions

This paper studies the ESVM algorithm and proposes a new regression algorithm ESVR. Similar to ESVM, ESVR is a new nonlinear SVM algorithm

based on regularized least squares and it is also a variant of ELM algorithm. ESVR not only can be used to regression tasks, but also can be applied to classification tasks. Performances of ESVR are compared with that of ELM, SVR and LS-SVR. ESVR has a little better generation ability than ELM. Compared to SVR and LS-SVR, ESVR has a comparable generalization ability, but has the much faster learning speed.

Acknowledgement

The authors would like to thank Mr. Zhiguo Ma and Mr. Fuqiang Chen for their valuable comments. This research is partially sponsored by National Basic Research Program of China (No.2009CB320900), and Natural Science Foundation of China (Nos. 61070116, 61070149, 61001108, 61175115, and 61272320), Beijing Natural Science Foundation (No. 4102013), and Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions.

References

[1] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.

[2] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 2, pp. 985–990, IEEE, 2004.

[3] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[4] W. Zong, H. Zhou, G.-B. Huang, and Z. Lin, "Face recognition based on kernelized extreme learning machine," in *Autonomous and Intelligent Systems*, pp. 263–272, Springer, 2011.

[5] H.-J. Rong, Y.-S. Ong, A.-H. Tan, and Z. Zhu, "A fast pruned-extreme learning machine for classification problem," *Neurocomputing*, vol. 72, no. 1, pp. 359–366, 2008.

[6] M. van Heeswijk, Y. Miche, T. Lindh-Knuutila, P. A. Hilbers, T. Honkela, E. Oja, and A. Lendasse, "Adaptive ensemble models of extreme learning machines for time series prediction," in *Artificial Neural Networks–ICANN 2009*, pp. 305–314, Springer, 2009.

[7] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, no. 16, pp. 3056–3062, 2007.

[8] Q. He, T. Shang, F. Zhuang, and Z. Shi, "Parallel extreme learning machine for regression based on mapreduce," *Neurocomputing*, 2012.

[9] Q. Liu, Q. He, and Z. Shi, "Extreme support vector machine classifier," in *Advances in Knowledge Discovery and Data Mining*, pp. 222–233, Springer, 2008.

[10] Q. He, C. Du, Q. Wang, F. Zhuang, and Z. Shi, "A parallel incremental extreme svm classifier," *Neurocomputing*, vol. 74, no. 16, pp. 2532–2540, 2011.

[11] B. Frénay and M. Verleysen, "Using svms with randomised feature spaces: an extreme learning approach," in *Proceedings of the 18th European symposium on artificial neural networks (ESANN), Bruges, Belgium*, pp. 28–30, 2010.

[12] A. Subasi, "A decision support system for diagnosis of neuromuscular disorders using dwt and evolutionary support vector machines," *Signal, Image and Video Processing*, pp. 1–10, 2013.

[13] P.-F. Pai and M.-F. Hsu, "An enhanced support vector machines model for classification and rule generation," in *Computational Optimization, Methods and Algorithms*, pp. 241–258, Springer, 2011.

[14] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 2, pp. 513–529, 2012.

[15] C. R. Rao and S. K. Mitra, "Generalized inverse of a matrix and its applications," *J. Wiley, New York*, 1971.

[16] G.-B. Huang, X. Ding, and H. Zhou, "Optimization method based extreme learning machine for classification," *Neurocomputing*, vol. 74, no. 1, pp. 155–163, 2010.

[17] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[18] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[19] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *Neural Networks, IEEE Transactions on*, vol. 17, no. 6, pp. 1411–1423, 2006.