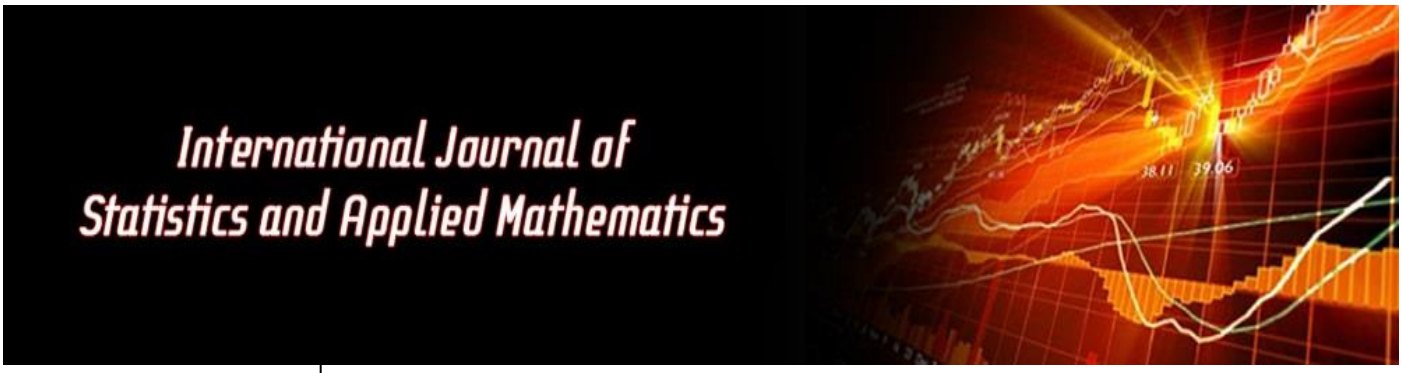

ISSN: 2456-1452
Maths 2024; 9(2): 10-16

https://www.mathsjournal.com
Received: 07-12-2023
Accepted: 10-01-2024

Mohan Kumar TL
Department of Agricultural Statistics, Applied Mathematics and Computer Science, UAS, GKVK, Bengaluru, Karnataka, India

Prajneshu
Indian Agricultural Statistics Research Institute, New Delhi, India

Prathima CM
Department of Agricultural Statistics, Applied Mathematics and Computer Science, UAS, GKVK, Bengaluru, Karnataka, India

Halagunde Gowda GR
The Central Silk Board, Bengaluru, Karnataka, India

Corresponding Author:
Mohan Kumar TL
Department of Agricultural Statistics, Applied Mathematics and Computer Science, UAS, GKVK, Bengaluru, Karnataka, India

Particle swarm optimization algorithm-based nonlinear LS-SVM model for modelling and forecasting time-series data

Mohan Kumar TL, Prajneshu, Prathima CM and Halagunde Gowda GR

DOI: https://doi.org/10.22271/maths.2024.v9.i2a.1638

Abstract
In this article, a novel Nonparametric, Nonlinear Least Squares Support Vector Machine (LS-SVM) methodology is thoroughly studied. The Particle Swarm Optimization (PSO), which is a very efficient population-based global stochastic optimization technique, is employed to estimate the hyper-parameters and time lag of Nonlinear LS-SVM model for time-series modelling. Relevant computer program is written in MATLAB function (m file). The MATLAB and STATISTICA software packages are used for carrying out data analysis. Subsequently, as an illustration, the methodology was applied to all-India annual rainfall time-series data. Superiority of this approach over ANN model is demonstrated using Root Mean Square Error (RMSE) and Mean Absolute Percent Error (MAPE) criteria for data under consideration.

Keywords: Indian monsoon rainfall, nonlinear LS-SVM, PSO algorithm, grid search, ANN, SVM and MATLAB software

1. Introduction
Analysis of time-series data has usually been carried out by employing the well-known Box-Jenkins Autoregressive Integrated Moving Average (ARIMA) family of models. However, the main disadvantage of these approaches is that the underlying models are 'linear'. During the last three decades or so, the area of "Nonlinear time-series modelling" has rapidly been growing. Here, there are basically two approaches, *viz*. Parametric or Nonparametric. However, quite often it is noticed that there is no appropriate parametric form to describe the time-series data under consideration in a satisfactory manner. In such cases, 'Nonparametric regression methodology' may be employed (Chandran and Prajneshu, 2007) [1]. During the last few years, another nonparametric methodology of 'Wavelet analysis' has emerged. The novel idea of wavelets is that these are localized in both time and space, whereas traditional Fourier bases are localized only in frequency but not in time. Wavelet analysis can be carried out either in the Time domain or in the Frequency domain. Ghosh *et al*. (2010) [2] investigated the wavelet approach in the frequency domain for analyzing time-series data. Chattopadhyay and Chattopadhyay, 2013 [3] was reported that in the last decade, with the success of the backpropagation algorithm for computing derivatives, the Artificial Neural Networks (ANN) methodology has been extensively used. However, the main drawbacks of this methodology are that the resultant models generally over fit the data and that the solution gets trapped in local minima.
The methodology, which does not suffer from the above limitations, is the Support Vector Machine/Regression (SVM / SVR) methodology developed by Vapnik (2000) [4]. This Nonparametric methodology is very promising for nonlinear regression analysis. It implements the Structural Risk Minimization principle, which has been shown to be superior to the traditional Empirical Risk Minimization principle implemented in ANN models. An excellent description of various aspects of SVR methodology is given in Smola and Scholkopf (2004) [5]. Later on, the Least Squares version of SVM, known as the Least Squares-Support Vector Machine (LS-SVM) was proposed by Suykens *et al.*, 2002 [6] for nonlinear regression analysis.

A heartening aspect of Nonparametric Nonlinear LS-SVM as compared to standard Nonlinear SVR for regression problems is that it applies linear least squares criterion to the loss function with only equality type constraints to obtain a set of linear equations instead of epsilon insensitive loss function with inequality type constraints to form traditional convex Quadratic Programming problem. This leads to the advantages of fast convergence, high accuracy and low computational efforts.

Despite the superior features of the above methodology, its generalization and efficiency are sensitive to the values of hyper-parameters as well as the dimension of the input vector (the lagged observations). Therefore, the selection of optimal hyper-parameters and time lags is an important step in Nonlinear LS-SVM modelling. To this end, Chen and Wang (2007) [7] discussed the advantages and disadvantages of various efficient estimation procedures and also proposed real-valued Genetic algorithm (GA) technique. However, its main drawback is that the GA-SVR approaches cannot optimize all Nonlinear SVR's hyper-parameters and time lags simultaneously. Moreover, GAs are capable of finding the solution through evolution operators, such as crossover and mutation. Evolution is inductive. In reality, life does not always evolve towards a good solution. Fortunately, a very efficient population-based stochastic optimization algorithm, *viz*. The Particle Swarm Optimization (PSO) technique is capable of rectifying the above limitations (Parsopoulos and Vrahatis, 2010) [8]. In contrast to GA, which exploits the competitive characteristics of biological evolution, PSO simulates cooperative and social behaviour, such as fish schooling, birds flocking, or insects swarming.

In this article, our purpose is to thoroughly study the Nonlinear LS-SVM methodology and PSO-based estimation procedure to estimate its optimal hyper-parameters as well as time lags. The relevant computer program is written in MATLAB function (m file). Finally, as an illustration, all-India annual rainfall time-series data are considered. The superiority of the developed model is demonstrated over the ANN approach.

2. Description of the methodologies

2.1 Nonlinear Least Squares Support Vector Machine (LS-SVM)

Suppose the training data set is $D = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in R^n$ is the i^{th} input data, $y_i \in R$ is the i^{th} target data and N corresponds to the size of the training data. In the primal weight space, the Nonlinear LS-SVM function for regression is given by.

$$y(x) = w^T \varphi(x) + b, \tag{1}$$

Where $\varphi(.): R^n \to R^{n_h}$ is a nonlinear function, which maps the input space into a higher dimensional feature space, $w \in R^{n_h}$ is the weight vector and b is the bias term. The Nonlinear LS-SVM for regression formulation can be described as.

Objective function: $\min_{w,b,e} J_P(w,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{i=1}^{N} e_i^2,$ (2)

Subject to the constraint: $y_i = w^T \varphi(x_i) + b + e_i; \quad i = 1, 2, \ldots N,$ (3)

Where $w \in R^{n_h}$ is the weight vector in primal weight space, $e_i \in R$ are error variables and b is the bias term. Note that the cost function J_P consists of a sum of squared fitting error and regularization term in primal feature space. The relative importance of these two terms is determined by the positive real constant γ. In the case of noisy data, one avoids over-fitting by taking a smaller γ value. The weight vector w can be infinite-dimensional, which makes a calculation of w as impossible, in general. The Lagrangian function can be constructed as.

$$L(w, b, e : \alpha) = J_p(w, e) - \sum_{i=1}^{N} \alpha_i \{w^T \varphi(x_i) + b + e_i - y_i\}, \tag{4}$$

Where α_i, i=1, 2,…,N are Lagrangian multipliers, also called support values. The Karusha-Kuhn-Tucker (KKT) conditions for optimality are given by partially differentiating L.

$$\begin{cases} \frac{\partial L}{\partial w} = 0 \quad \to \quad w = \sum_{i=1}^{N} \alpha_i \varphi(x_i) \\ \frac{\partial L}{\partial b} = 0 \quad \to \quad \sum_{i=1}^{N} \alpha_i = 0 \\ \frac{\partial L}{\partial e_i} = 0 \quad \to \quad \alpha_i = \gamma e_i \; ; \quad i = 1, 2, \ldots N \\ \frac{\partial L}{\partial \alpha_i} = 0 \quad \to \quad w^T \varphi(x_i) + b + e_i - y_i = 0 \; ; \quad i = 1, 2, \ldots N \end{cases} \tag{5}$$

After eliminating 'w' and 'e_i', optimization problem can be transferred into the following linear solution system in dual space:

$$\begin{bmatrix} 0 & 1_v^T \\ 1_v & \Omega + \frac{1}{\gamma} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \quad \text{Solve in } \alpha, b \tag{6}$$

Where $y = [y_1, y_2, \ldots, y_N]^T$, $1_v = [1, \ldots; 1]^T$, $\alpha = [\alpha_1, \ldots, \alpha_N]^T$ and $\Omega = \{ \Omega_{ij} \}$ is given by

$$\Omega_{ij} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j), \quad i, j = 1, 2, \ldots, N. \tag{7}$$

In the above, $K(x_i, x_j)$ is called the inner-product kernel function. Thus, Nonlinear LS-SVM for regression is obtained as

$$y(x) = w^T \varphi(x) + b = \sum_{i=1}^{N} \alpha_i K(x, x_i) + b, \tag{8}$$

Where α_i and b are solutions of the linear system.
For a univariate time-series forecasting problem, the dimension of input vectors is the past lagged observations. The Nonlinear LS-SVM model performs a nonlinear functional mapping from the past observations to the future value as:

$$y_t = f(y_{t-1}, y_{t-2}, \ldots, y_{t-p}) + \varepsilon_t \tag{9}$$

Where y_t is the output value, p is the dimension of the input vector and ε_t is the error term.
Several kernel functions $K(x_i, x_j)$ are available in the literature, like the Polynomial function, Gaussian kernel, and Radial-basis function (RBF). In this article, the RBF kernel function $K(x, x_i) = exp\left\{ - \|x - x_i\|^2 / (2\sigma^2) \right\}$ is used to train the Nonlinear LS-SVM model, since it has only one hyper-parameter that needs to be pre-determined and yields good performance for nonlinear time-series forecasting. Training Nonlinear LS-SVM with RBF kernel function is required to be optimized through two hyper-parameters, *viz*. Regularization parameter gamma (γ) and Kernel bandwidth parameter sigma (σ), which may affect Nonlinear LS-SVM generalization performance. Also, the important task of Nonlinear LS-SVM modelling is to estimate the optimal dimension of input vector p (i.e. the lagged observations). However, in practice, the choice of p is difficult; often trial and error method is conducted. To this end, PSO is employed to optimize Nonlinear LS-SVM hyper-parameters as well as time lag p simultaneously.

2.2 Development of Particle Swarm Optimization Algorithm (PSO) for Nonlinear LS-SVM
The PSO is a population-based stochastic optimization search method, i.e. it exploits a population of potential solutions to probe the search space concurrently, proposed by Kennedy and Eberhart in 1995 (Parsopoulos and Vrahatis, 2010 [8]). The population is called the '*swarm'* and its individuals are called the *'particles'* Let $A \subset R^n$ be the search space and $A \to Y \subseteq R$ be the objective function, then the swarm is defined as a set $S = \{X_1, X_2, \ldots, X_N\}$ of N particles (candidate solution), defined as $X_i = (X_{i1}, X_{i2}, \ldots, X_{in})^T \in A$, i=1, 2,…, N. Indices are arbitrarily assigned to particles, while N is a user-defined parameter of the algorithm. The PSO algorithm investigates solution space using a set of particle vectors that are updated from iteration to iteration. Each particle is a potential solution, characterized by three quantities: Its velocity $V_i = (V_{i1}, V_{i2}, \ldots, V_{in})^T$, current position $X_i = (X_{i1}, X_{i2}, \ldots, X_{in})^T$ and personal best position $pbest_i = (pbest_{i1}, pbest_{i2}, \ldots, pbest_{in})^T$. Let t denote the current iteration and *gbest* denote its global best position achieved so far by any of its particles. Initially, the swarm is randomly dispersed within the search space, and random velocity is assigned to each particle. Particles interact with one another by sharing information to discover the optimal solution. Each particle moves in the direction of its personal best position (*pbest*) and its global best position (*gbest*). To search optimal solution, each particle changes its velocity according to the cognitive and social parts as given by eq. (10) and move to a new potential solution by updating its position based on the eq. (11).

$$V_{ij}(t+1) = \begin{cases} -V_{\min}, & if\ V_{ij}(t+1) < -V_{\min} \\ w(t) V_{ij}(t) + c_1 R_1 [pbest_{ij}(t) - X_{ij}(t)] + c_2 R_2 [gbest_j(t) - X_{ij}(t)]; & if\ -V_{\min} \le V_{ij}(t+1) \le V_{\max} \\ V_{\max}, & if\ V_{ij}(t+1) > V_{\max}; \ i = 1, 2, \ldots, N; \quad j = 1, 2, \ldots, n \end{cases} \tag{10}$$

$$X_{ij}(t+1)=\begin{cases}-X_{\min}, & if\ X_{ij}(t+1)<-X_{\min}\\ X_{ij}(t)+\beta V_{ij}(t+1), & if\ -X_{\min}\le X_{ij}(t+1)\le X_{\max}\\ X_{\max}, & if\ X_{ij}(t+1)>X_{\max};\ i=1,2,\ldots,N;\ j=1,2,\ldots,n\end{cases} \quad (11)$$

Where V_{ij}, X_{ij} and $pbest_{ij}$ are respectively velocity, position and *pbest* position of particle i on the j^{th} dimension and $gbest_j$ is the j^{th} dimension *gbest* position among all particles at iteration t. The R_1 and R_2 are random values, which are mutually independent and uniformly distributed over [0, 1], β is constraint factor used to control the velocity weight, whose value is usually set equal to 1. Positive constants c_1 and c_2 are usually called the "acceleration factors". Factor c_1 is sometimes referred to as a "cognitive" parameter, while c_2 is referred to as a "social" parameter. The $w(t)$ is the inertia weight at iteration t and is used to balance global exploration and local exploitation. This can be determined by.

$$w(t)=w_{up}-(w_{up}-w_{low})\,t/T_{\max}, \quad (12)$$

Where t is current iteration number, w_{up} and w_{low} are desirable lower and upper limits of w, and T_{max} is the maximum number of iterations.

The process of optimizing the Nonlinear LS-SVM model hyper-parameters as well as time lags by PSO is described in the following steps.

1. **Data preparation:** The time-series data points are divided into two parts: (i) Training dataset, which is used only for building the model, and (ii) Testing dataset, which is used for validation of the model.
2. **Initialization and PSO parameter setting:** First, set the time lag one to the maximum number of lags to be considered as the dimension of the input vector to estimate optimal time lag p. Then, Nonlinear LS-SVM model hyper-parameters γ and σ (Here n=2) are directly coded with real values within a given search space to randomly generate N number of initial particles of swarm set S. The search space of hyper-parameters γ and σ are respectively limited to ranges of $[\gamma_{min}, \gamma_{max}]$ and $[\sigma_{min}, \sigma_{max}]$. Also randomly generate initial particle velocity (V_i) limited to the range $[-V_{max}, V_{max}]$.
3. **Evaluate each particle's fitness:** For each candidate particle of initial particles, train the Nonlinear LS-SVM model using k-fold cross-validation to evaluate the minimum value of fitness function, defined as the cross-validation Mean Square Error (CV-MSE):

$$\text{Minimum}\{\text{CV-MSE}(\gamma,\sigma)\}=\frac{1}{k}\sum_{i=1}^{k}\left\{\frac{1}{m}\sum_{j=1}^{m}(y_j-\hat{y}_j)^2\right\}, \quad (13)$$

Where y_j and $\hat{y}_j$ are respectively actual and predicted values of the validation subset, m is the number of samples in the validation subset and k is the number of folds for cross-validation. Then, update the personal best position $pbest_i$ of each particle and the global best position *gbest* achieved so far by any of its particles according to minimum fitness value.

1. Now set the iteration number (t) one to the maximum number of iterations and evaluate inertia weight $w(t)$ at iteration t using eq. (12).
2. Compute and update the velocity of each particle using eq. (10).
3. Compute and update the position of each particle using eq. (11).
4. Termination: Repeat the search process from Step (iii) to Step (vi) until stop conditions, such as maximum iteration, are met.

Finally, optimal hyper-parameters and time lag p are utilized to build the Nonlinear LS-SVM model on the training dataset and the testing dataset is used to validate the fitted model. To perform the above tasks, the LS-SVMLab, Version 1.8 toolbox, developed by De Brabanter *et al.* (2011) [9] for the MATLAB 2012a software platform, is employed for training the Nonlinear LS-SVM model. We have written MATLAB function (m file) code for estimating hyper-parameters of the Nonlinear LS-SVM model using the PSO algorithm, and the same is appended as ANNEXURE-1.

3. An illustration

As an illustration, all-India annual rainfall (in mm) time-series data for the period 1871 to 2011, obtained from the website (www.tropmet.res.in) of the Indian Institute of Tropical Meteorology, Pune, India are considered. The data points for the period 1871 to 1999 are used as a training dataset and the remaining data points for the period 2000 to 2011 are used as a testing dataset. Note that, before training the model, training and testing data points are normalized to the range [0, 1] separately using eq. (14), which can improve the generalization power of Nonlinear LS-SVM.

$$y'=\frac{y_t-Min(y_t)}{Max(y_t)-Min(y_t)} \quad (14)$$

Through the initial experiment, the parameters of the PSO technique were set as follows: 100 initial particles (so that it is enough to cover the search space within the limited iterations based on experimental runs), 500 maximum iterations, inertia weight is initially set as 0.9 and reduced to 0.1 linearly according to eq. (16) and c_1=c_2=2.05. The searching range of Nonlinear LS-SVM model hyper-parameters γ and σ were respectively set in the range [0.001,1000] and [0.001,100] and their velocity bounds were respectively set to be in the range [-499. 99, 499. 99] and [-49.99, 49.99]. For comparison purposes, Grid search, the most

commonly used method for estimating optimal hyper-parameters was also employed. In the training stage, the PSO and Grid search techniques were employed to estimate optimal hyper-parameters and time lag of Nonlinear LS-SVM based on 10-fold cross-validation and the results along with CV-MSE values are reported in Table 1.

Table 1: Optimal hyper-parameters and time lag estimated by Grid Search and PSO techniques.

Model	γ	σ	lag	CV-MSE
LS-SVM	593.751	0.055	10	0.031
PSO+LS-SVM	996.999	0.053	9	0.029

The following Fig. 1 illustrates the CV-MSE estimates over a different lag period by Grid Search and PSO techniques.

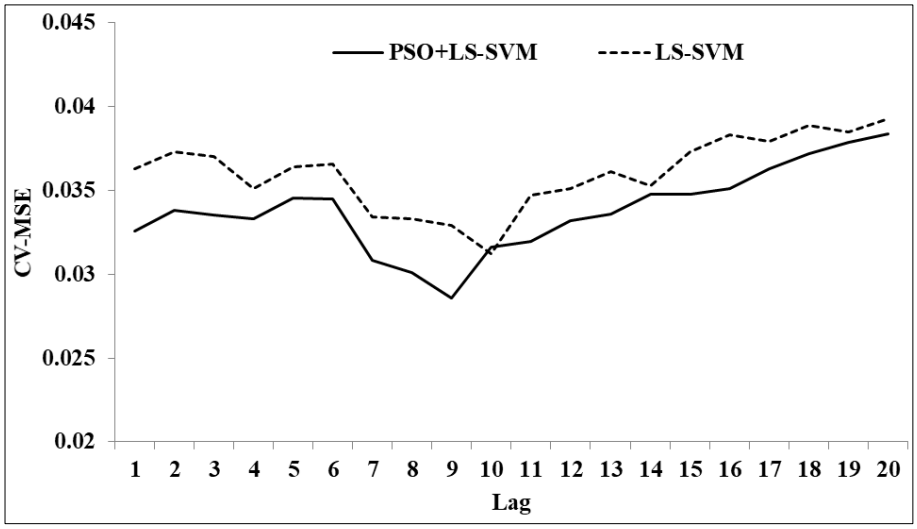

Fig 1: Cross-Validation MSE over different lag periods by Grid Search and PSO techniques.

Then, optimal hyper-parameters values and time lag are utilized to train the Nonlinear LS-SVM model for modeling all-India annual rainfall time-series data.

Further, for comparing the prediction performance of Nonlinear LS-SVM models, we also employed the most widely used Artificial Neural Network as a benchmark model using the STATISTICA Neural Network Version 7 software package. The ANN model performs a nonlinear functional mapping from the past observations to the future value, which is equivalent to a Nonlinear Autoregressive model. Before developing ANN model, training and testing data points are normalized to the range [0, 1] separately using eq. (14). In this article, a typical three-layer feed-forward network with one input layer, one hidden layer and one output layer is used to develop the model. The most commonly used sigmoid activation function for time-series modelling was applied to the input layer to the hidden layer and the hidden layer to the output layer. In the training and testing of the ANN network, the same input lag structure of PSO + Nonlinear LS-SVM model was considered. The network was trained using the Back-Propagation algorithm with a learning rate of 0.001 and the trial and error method was performed to optimize the optimal number of neurons in the hidden layer, which was found to be 9 neurons.

Further, Nonlinear LS-SVM models and ANN model are compared based on Root Mean Square Error (RMSE) and Mean Absolute Percent Error (MAPE) respectively given in eq. (15) and eq. (16).

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \tag{15}$$

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| * 100 \tag{16}$$

Where, y_j is the actual value and $\hat{y}_j$ is the predicted value, n is the number of samples in the data subset. The RMSE and MAPE for the above-fitted models are computed on the training dataset, which are reported in Table 2.

Table 2: Comparison of prediction performance over different models on training dataset.

Accuracy\Models	PSO+LS-SVM	LS-SVM	ANN
RMSE	0.125	0.1930	95.234
MAPE	0.009	0.014	7.028

It may be noted that RMSE and MAPE for the PSO + Nonlinear LS-SVM model on the training dataset are respectively computed as0.125 and 0.009, which are found to be lower than the corresponding Nonlinear LS-SVM model, respectively computed as 0.1930 and 0.014 and ANN model, respectively computed as 95.234 and 7.028. The PSO + Nonlinear LS-SVM model is found to

be the best-fitted model as compared to Nonlinear LS-SVM and ANN models. The predicted value on the training data*set al*ong with actual data points of the fitted PSO + Nonlinear LS-SVM model are plotted as shown in Fig. 2.

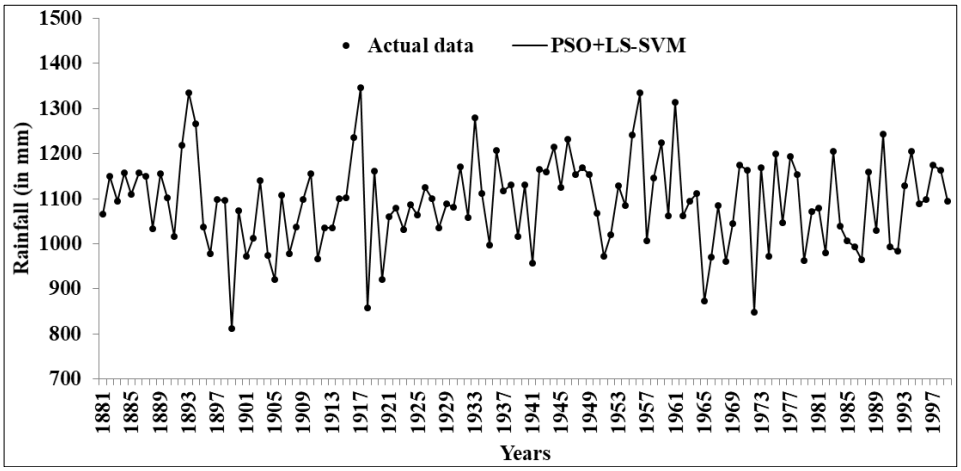

Fig 2: Fitted PSO + Nonlinear LS-SVM model along with data points.

Further, one-step ahead forecasts along with actual values during the period 2000 to 2011 for the above models are reported in Table 3.

Table 3: One-step ahead forecasts of all-India annual rainfall (In mm) data for the fitted models.

Years	Actual data	Predicted		
		PSO+LS-SVM	LS-SVM	ANN
2000	989.7	1039.78	1021.97	1034.28
2001	1059.9	1037.78	1041.88	1037.10
2002	882.4	996.98	1007.50	1014.71
2003	1088.5	1035.40	1023.90	1024.54
2004	1003.4	1077.37	1061.93	1023.44
2005	1141.1	1077.86	1080.46	1034.25
2006	1112.7	1031.02	1036.06	1031.28
2007	1149.7	1056.79	1044.69	1038.60
2008	1118.3	1039.07	1047.79	1044.29
2009	886.4	989.71	1007.99	1003.45
2010	1107.6	1037.82	1026.57	1053.74
2011	1051.8	1042.13	1031.59	1015.04
Accuracy	RMSE	73.957	77.420	80.791
	MAPE	6.609	6.792	7.004

The RMSE and MAPE for the PSO + Nonlinear LS-SVM model on the testing dataset are respectively computed as 73.957 and 6.609, which are also found to be lower than the corresponding Nonlinear LS-SVM model, respectively computed as 77.420 and 6.792, and ANN model, respectively computed as 80.791 and 7.004. For the testing dataset, the PSO + Nonlinear LS-SVM model is found to be the best-fitted model as compared to Nonlinear LS-SVM and ANN.

4. Conclusion

In this article, Nonlinear LS-SVM methodology which is a very promising methodology for nonlinear regression analysis is described. PSO technique, which avoids over-fitting or under-fitting, is employed to optimize the two hyper-parameters as well as the time lag (dimensions of input vector) of the above model. PSO is a powerful optimization method, which not only has strong global search capability but also is very easy to implement. As an illustration, modelling and forecasting of all-India annual rainfall time-series data is considered. It is shown that the PSO + Nonlinear LS-SVM model can achieve greater accuracy than Nonlinear LS-SVM using the Gird search. Also, the superiority of this model over the ANN model is demonstrated using Root Mean Square Error (RMSE) and Mean Absolute Percent Error (MAPE) criteria for the data under consideration.

5. Acknowledgment

The authors are grateful to the referee for valuable comments.

6. References

1. Chandran KP, Prajneshu. Nonparametric regression approach for modelling and forecasting country's meteorological subdivisions rainfall data. J. Agrometeorol. 2007;8(2):254-259.
2. Ghosh H, Paul RK, Prajneshu. Wavelet frequency domain approach for statistical modeling of Indian monsoon rainfall time-series data. J Statist. Theo. Prac. 2010;4(4):813-825.
3. Chattopadhyay S, Chattopadhyay G. A supervised Neural Network model for predicting average summer monsoon rainfall in India. J Ind. Soc. Ag. Statist. 2013;67(1):43-49.

4. Vapnik V. The Nature of Statistical Learning Theory. Springer-Verlag, New York; c2000.
5. Smola AJ, Scholkopf B. A tutorial on support vector regression, Statist. Comput. 2004;14:199-222.
6. Suykens JAK, Van Gestel T, De Brabanter J, De Moor B, Vandewalle J. Least Squares Support Vector Machines, World Scientific, Singapore; c2002.
7. Chen KY, Wang CH. Support vector regression with genetic algorithm in forecasting tourism demand. Tour. Mgmt. 2007;28(1):215-226.
8. Parsopoulos KE, Vrahatis MN. Particle Swarm Optimization and Intelligence: Advanced and Application. Information Science Reference, New York; c2010.
9. De Brabanter J, Karsmakers P, Ojeda F, Alzate C, De Brabanter J, Pelckmans, *et al*. LS-SVMlab: Matlab toolbox for Least Squares Support Vector Machines. (http://www.esat.kuleuven.ac.be/sista/LS-SVMlab); c2002a