

Original Research Paper

An Evolving Autoregressive Predictor for Time Series Forecasting

¹De Z. Li, ²Wilson Wang and ¹Fathy Ismail

¹Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, Canada

²Department of Mechanical Engineering, Lakehead University, Thunder Bay, Canada

Article history

Received: 02-09-2014

Revised: 03-09-2014

Accepted: 21-02-2015

Corresponding Author:

De Z. Li

Department of Mechanical
Engineering, Lakehead University,
Thunder Bay, Canada

Email: d45li@uwaterloo.ca

Abstract: Autoregressive (AR) model is a common predictor that has been extensively used for time series forecasting. Many training methods can be used to update AR model parameters, for instance, least square estimate and maximum likelihood estimate; however, both techniques are sensitive to noisy samples and outliers. To deal with the problems, an evolving AR predictor, EAR, is developed in this study to enhance prediction accuracy and mitigate the effect of noisy samples and outliers. The model parameters of EAR are trained with an Adaptive Least Square Estimate (ALSE) method, which can learn samples characteristics more effectively. In each training epoch, the ALSE weights the samples by their fitting accuracy. The samples with larger fitting errors will be given a larger penalty value in the cost function; however the penalties of difficult-to-predict samples will be adaptively reduced to enhance the prediction accuracy. The effectiveness of the developed EAR predictor is verified by simulation tests. Test results show that the proposed EAR predictor can capture the dynamics of the time series effectively and predict the future trend accurately.

Keywords: Autoregressive Model, Boosting, Adaptive Least Square Estimate, Time Series Forecasting

Introduction

Time series forecasting is a process to extract features from available data that rule the trend of the data and forecast future data based on the extracted features. It has enormous real world applications, such as electric load prediction (Quan *et al.*, 2013; Goude *et al.*, 2014), financial indices forecast (Li *et al.*, 2013a; Yu *et al.*, 2009) and machine health condition monitoring (Wang, 2013).

Some commonly used prediction tools are Autoregressive (AR) models, Autoregressive-Moving-Average (ARMA) models (Brockwell and Davis, 2009), Neural Networks (NNs) (Li *et al.*, 2014a; 2013b; 2014b) and particle filtering (Li *et al.*, 2014c). The NNs can capture data features through a training stage and conduct time series prediction based on the extracted features; however, they suffer from opaque modeling mechanism. AR is more compact than ARMA and it does not have estimation errors that result from the moving average part in ARMA.

The boosting technique is an ensemble learning method, which combines weak learners to improve the training accuracy whereby each weak learner addresses one particular data property (e.g., the data distribution). Boosting techniques are mainly used in pattern classification (Cao *et al.*, 2012; Schapire and Singer, 1999). In time series forecasting, the boosting techniques have also been employed to improve prediction accuracy (Drucker, 1997). A boosting technique can also be used as a training method to optimize model parameters. An Evolving AR (EAR) predictor is proposed in this study for time series forecasting. The EAR has the generic AR model structure; however, it uses an Adaptive Least Square Estimate (ALSE) method to train model parameters. Some samples are hard to learn because they are noisy samples or outliers. If more effects are put to correctly learn these hard-to-learn samples, the prediction accuracy of the already well learnt samples will drop and the training process will suffer from the

“overfitting” problem. The hard-to-learn samples are detected and their penalties are reduced in the proposed EAR to improve generalized prediction performance. The effectiveness of the proposed EAR predictor is verified here by simulations.

The remainder of this paper is organized as follows: Section 2 presents the theoretical foundation of the proposed EAR predictor. In section 3, the effectiveness of the proposed EAR predictor is examined by simulation tests. Some concluding remarks of the study are given in section 4.

The Proposed Evolving AR Predictor

The EAR predictor assigns penalties to samples at each training epoch according to their fitting errors. The model parameters can be updated adaptively with respect to different samples penalties so as to improve prediction accuracy. Those hard-to-learn samples may distort the training process, so their penalties will be reduced in EAR. The proposed EAR technique is given as follows.

EAR Model Structure

Consider the training data sets $u(k)$:

$$k = 1, 2, \dots, K$$

where, K is the number of samples in the training data set. For s -step-ahead prediction, the training data set can be re-arranged as input vector $x(i) = [u(i), u(i+1), \dots, u(i+d-1)]$ and the output $y(i) = u(i+d+s-1)$; $i = 1, 2, \dots, N$, where $N = (K-d-s+1)$. d is the dimension of the input vector $x(i)$.

The EAR model has the form of:

$$p_t(i) = \Theta_1 u(i-1) + \Theta_2 u(i-2) + \dots + \Theta_{r-1} u(i-r+1) \quad (1)$$

where, Θ_i are linear parameters; $i = 1, 2, \dots, r-1$. Equation 1 can also be written in the following matrix form Equation 2, 3 and 5:

$$Y = X\Theta \quad (2)$$

Where:

$$X = \begin{bmatrix} u(1) & u(2) & \dots & u(r-1) \\ u(2) & u(3) & \dots & u(r) \\ \vdots & \vdots & \ddots & \vdots \\ u(N_i-r-s+1) & u(N_i-r-s+2) & \dots & u(N_i-s-1) \end{bmatrix} \quad (3)$$

$$\Theta = \begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_{r-1} \end{bmatrix} \quad (4)$$

$$Y = \begin{bmatrix} u(r+s) \\ u(r+s+1) \\ \vdots \\ u(N_i) \end{bmatrix} \quad (5)$$

where, N_i is the number of system state u .

Parameter Estimation Using ALSE

The linear parameters increment θ_t of the EAR predictor at t^{th} training epoch can be derived using the weighted least square estimate, WLSE:

$$\theta_t = (X^T W X)^{-1} X^T W Y \quad (6)$$

The weight matrix W is represented by Equation 7:

$$W = \begin{bmatrix} V_t(1) & \dots & 0 & 0 \\ \vdots & V_t(2) & \dots & 0 \\ 0 & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & V_t(N) \end{bmatrix} \quad (7)$$

where, $N = N_i - r - s + 1$ and $V_t(i)$ represent the penalties of sample i at training epoch t .

Formulation of Sample Penalties

In the proposed EAR, the penalties of sample i at training epoch one is set as $V_1(i) = \frac{1}{N}$, where N is the number of samples. Given the penalties V_t , the update of the penalties at step $t+1$ will be performed by Equation 9:

$$V_{t+1}(i) = \frac{V_t(i) \exp(-\beta_t [|y_d(i) - p_t(i)| - \omega_t(i)])}{Z_t} \quad (8)$$

$$= \frac{\exp(-\sum_{t=1}^T (\beta_t [|y_d(i) - p_t(i)| - \omega_t(i)]))}{N \prod_{t=1}^T Z_t} \quad (9)$$

where, y_d are the desired values; p_t are the predicted values using Equation 1; β_t is the learning rate of the t th parameter update epoch; $\omega_t(i)$ is the weight regulator to reduce the penalties of hard-to-learn samples; $Z_t = \sum_{i=1}^N V_t(i) \exp(-\beta_t [|y_d(i) - p_t(i)| - \omega_t(i)])$ is a normalization factor.

The EAR model parameters in Equation 4 at step T are obtained from Equation 10:

$$\Theta_T = \sum_{t=1}^T \beta_t^* \theta_t \quad (10)$$

where, θ_t is the linear parameter increment. The predicted values of EAR at step T are formulated as Equation 11:

$$\begin{aligned} P_T &= X\Theta_T \\ &= X \sum_{t=1}^T \beta'_t \theta_t \\ &= \sum_{t=1}^T \beta'_t p_t \end{aligned} \quad (11)$$

where, $\beta'_t = \frac{\beta_t}{\sum_{t=1}^T \beta_t}$ are the normalized learning rates.

Calculation of β_t

Let $v_t = |y_d(i) - p_t(i)| - \omega_t(i)$ with $v_t \in [0, M_t]$, where M_t is the maximum value of $|y_d(i) - p_t(i)|$ and $\lambda_t = \sum_{i=1}^N V_t(i) v_t(i)$. The upper bound of Z_t can be derived as:

$$\begin{aligned} Z_t &= \sum_{i=1}^N V_t(i) \exp(-\beta_t v_t(i)) \\ &\leq \sum_{i=1}^N V_t(i) \left(\begin{aligned} &\frac{M_t + v_t(i)}{2M_t} \exp(-\beta_t M_t) \\ &+ \frac{M_t - v_t(i)}{2M_t} \exp(\beta_t M_t) \end{aligned} \right) = U \end{aligned} \quad (12)$$

Let $\frac{\partial U}{\partial \beta_t} = 0$ and the learning rate at step t will be:

$$\beta_t = \frac{1}{2M_t} \ln \left(\frac{M_t + \lambda_t}{M_t - \lambda_t} \right) \quad (13)$$

Substituting Equation 13 into Equation 12, the minimum upper bound of Z_t can be derived as Equation 14:

$$Z_t \leq \sqrt{1 - \left(\frac{\lambda_t}{M_t} \right)^2} \quad (14)$$

Since $M_t \geq \lambda_t \geq 0$, then $\sqrt{1 - \left(\frac{\lambda_t}{M_t} \right)^2} \leq 1$.

Mean Absolute Training Error

Since $\beta_t \geq 0$, $\sum_{t=1}^T \beta'_t = 1$, $\beta'_t \geq 0$, $\sum_{i=1}^N V_{T+1}(i) = 1$, $\lambda_t \geq 0$ and the final ensemble prediction $P(i) = \sum_{t=1}^T \beta'_t p_t(i)$, the Mean Absolute Error (MAE) of the training data can be determined by Equation 15-18, 21, 23 and 24:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_d(i) - P(i)| \quad (15)$$

$$= \frac{1}{N} \sum_{i=1}^N \left| y_d(i) - \sum_{t=1}^T \beta'_t p_t(i) \right| \quad (16)$$

$$= \frac{1}{N} \sum_{i=1}^N \left| \sum_{t=1}^T \beta'_t (y_d(i) - p_t(i)) \right| \quad (17)$$

$$\leq \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T |\beta'_t (y_d(i) - p_t(i))| \quad (18)$$

$$\leq \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T |\beta_t (y_d(i) - p_t(i))| \quad (19)$$

$$\leq \frac{1}{N} \sum_{i=1}^N \exp \left(- \sum_{t=1}^T |\beta_t (y_d(i) - p_t(i))| \right) \quad (20)$$

$$= \frac{1}{N} \sum_{i=1}^N \exp \left(- \sum_{t=1}^T \beta_t |y_d(i) - p_t(i)| \right) \quad (21)$$

$$\leq \frac{1}{N} \sum_{i=1}^N \exp \left(- \sum_{t=1}^T \beta_t (|y_d(i) - p_t(i)| - \omega_t(i)) \right) \quad (22)$$

$$= \sum_{i=1}^N V_{T+1}(i) \prod_{t=1}^T Z_t \quad (23)$$

$$= \prod_{t=1}^T Z_t \quad (24)$$

To satisfy Equation 19, $\beta'_t \leq \beta_t$ or $\sum_{t=1}^T \beta_t \geq 1$ should be satisfied. If T is sufficiently large, then $\sum_{t=1}^T \beta_t \geq 1$. Equation 20 can be satisfied when $\sum_{t=1}^T \beta_t |y_d(i) - p_t(i)| \in [0, 0.567]$. This condition can be satisfied by properly scaling the training data. To satisfy Equation 22, $\omega_t(i) \geq 0$, which will be shown in the following subsection. Therefore, the minimization of the training MAE is equivalent to minimizing $\prod_{t=1}^T Z_t$, or minimizing Z_t at each step t .

Given $\sqrt{1 - \left(\frac{\lambda_t}{M_t} \right)^2} \leq 1$, the upper bound of training MAE can be given as:

$$MAE \leq \prod_{t=1}^T Z_t \leq \prod_{t=1}^T \sqrt{1 - \left(\frac{\lambda_t}{M_t} \right)^2} \quad (25)$$

Therefore, as more training epochs are used, the upper bound of the training MAE decreases.

Weight Regulator

Some samples may be noisy samples or outliers, which may mislead the training process and degrade the

prediction accuracy. The following equation can be used to detect these irregular samples Equation 26:

$$\pi_t(i) = \sum_{t=1}^t \beta'_t |y_d(i) - p_t(i)| \quad (26)$$

By summing up the previous weighted errors, the samples can be ranked according to their difficulty levels in learning. Then the weight regulator can be computed from Equation 27:

$$\omega_t(i) = \frac{1}{\gamma} \exp\left(\pi_t(i) \frac{M_t}{\eta_t}\right) \quad (27)$$

where, $\eta_t = \sup(\pi_t(i))$. It can be shown that $\omega_t(i) \geq 0$.

By applying the weight regulator to the sample penalty update as shown in Equation 8, the hard-to-learn samples can be identified and their penalties will be reduced to improve generalized prediction accuracy.

Implementation of the EAR Predictor

The proposed EAR predictor is implemented using the following steps:

- Normalize the data over a proper range (e.g., [0, 1]), so that constraints can be satisfied
- Initialize the penalties of the training data set $V_1(i) = \frac{1}{N}; i = 1, 2, \dots, N$
- Derive the parameter increment θ_t using Equation 6 with the penalties V_t
- Compute the sum of weighted absolute error $\lambda_t = \sum_{i=1}^N V_t(i) (|y_d(i) - p_t(i)| - \omega_t(i))$, where y_d are the desired values and p_t are the predicted values using in Equation 1
- Calculate the learning rate of the t^{th} training epoch, $\beta_t = \frac{1}{2M_t} \ln\left(\frac{M_t + \lambda_t}{M_t - \lambda_t}\right)$
- Update the penalties of the training samples, $V_{t+1}(i) = \frac{V_t(i) \exp(-\beta_t (|y_d(i) - p_t(i)| - \omega_t(i)))}{Z_t}$, where $Z_t = \sum_{i=1}^N V_t(i) \exp(-\beta_t (|y_d(i) - p_t(i)| - \omega_t(i)))$ is a normalization factor
- Repeat steps 3 to 6 at $t = 1, 2, \dots, T$
- Formulate the final EAR model parameters $\Theta_T = \sum_{t=1}^T \beta'_t \theta_t$, where $\beta'_t = \frac{\beta_t}{\sum_{t=1}^T \beta_t}$ are normalized learning rates

Performance Evaluation

To verify the effectiveness of the proposed EAR predictor, simulation tests are conducted to examine its prediction performance. The AR model with the same structure as EAR, but trained by Kalman-filter-based Maximum Likelihood Estimate (MLE) (Hevia, 2008), AR-MLE, is used for comparisons. To satisfy constraints, the data sets used in this section are normalized over the range of [0, 1]. Test results, however, are shown in their original scales.

The Mackey-Glass data set (Farmer, 1982; Li and Wang, 2011; Wang *et al.*, 2012) is a commonly used simulation data set in the field of time series forecasting to compare the performance of predictors, due to its specific properties such as chaotic, non-periodic and non-convergence, it is given by:

$$\frac{dz(t)}{dt} = \frac{0.2z(t - \varpi)}{1 + z^{10}(t - \varpi)} - 0.1z(t) \quad (28)$$

In this simulation test, the data set is obtained from Equation 28 with the initial conditions of $\tau = 30, x(0) = 1.2, dt = 1$ and $x(t) = 0$ for $t < 0$. About 500 samples are selected for training and 50 samples for testing. One-step-ahead forecasting is conducted in the Mackey-Glass data prediction tests. To test the noise tolerance of the two predictors, noisy samples are intentionally added to the Mackey Glass training data; the red circled samples at time step 20, 100, 135, 215, 345, are shown in Fig. 1. γ is given the value of 2; α is given the value of 1.

The training MAE convergences of EAR(3), EAR(6) and EAR(9) are shown in Fig. 2. It is seen from Fig. 2 that the training MAEs decrease as more training epochs are used, which agrees with Equation 25. Figure 3a demonstrates the training data fitting and Fig. 3b shows the prediction performance. From Fig. 3b, it is seen that the EAR predictor outperforms AR-MLE, because EAR predictor can detect and process noisy samples to alleviate the noisy sample misleading effect.

The training MAEs and test MAEs of the two predictors with respect to model orders of 3, 6 and 9 are listed in Table 1; their corresponding training RMSEs and test RMSEs are given in Table 2. 100 training epochs are used in the EAR. From Table 1 and 2, it is seen that the training errors of EAR decrease as the model order increases, because a larger model order indicates more information is input to the predictor for processing and the predictions become more accurate.

From Table 2, the training RMSEs of the proposed EAR are larger than those of AR-MLE, because EAR reduces the penalties of noisy samples to improve the generalization capability of the predictor. Consequently, training errors at noisy samples are large, which leads to larger EAR training RMSE.

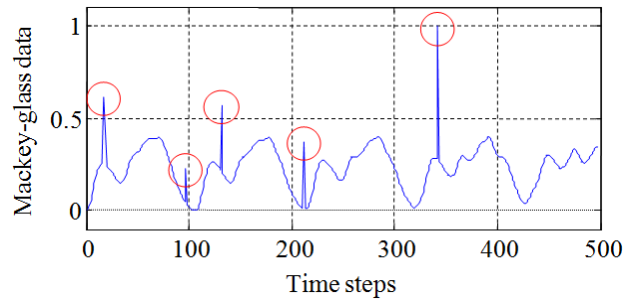


Fig. 1. The Mackey Glass training data with artificial noisy samples (red circled samples)

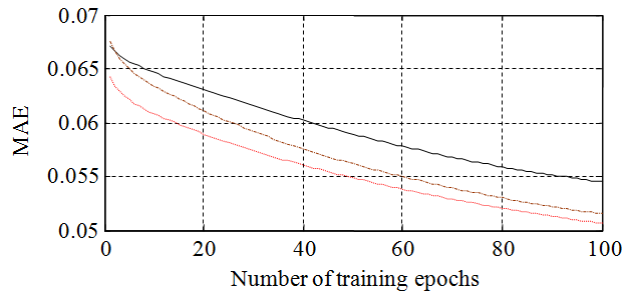


Fig. 2. The training MAE of EAR(3) (black solid line), EAR(6) (brown dash-dotted line) and EAR(9) (red dotted line), corresponding to different number of training epochs

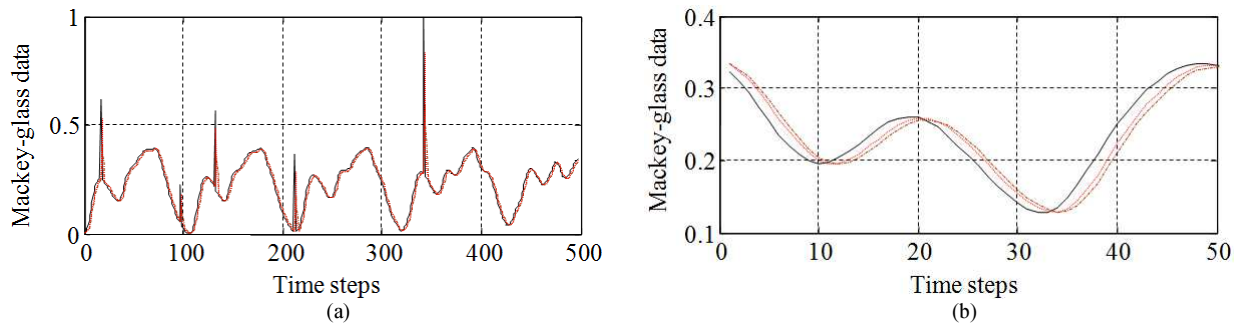


Fig. 3. The performance of (a) training data fitting and (b) test data prediction. The black solid line is the real data; the brown dash-dotted line represents AR(5)-MLE; the red dotted line represents EAR(5) with 100 training epochs

Table 1. MAEs of AR-MLE and EAR predictors in terms of Mackey-glass data

Predictor model order	AR-MLE		EAR	
	Training MAE	Test MAE	Training MAE	Test MAE
3	0.0670	0.0449	0.0545	0.0327
6	0.0657	0.0437	0.0515	0.0279
9	0.0632	0.0386	0.0506	0.0269

Table 2. RMSEs of AR-MLE and EAR predictors in terms of Mackey-glass data

Predictor model order	AR-MLE		EAR	
	Training RMSE	Test RMSE	Training RMSE	Test RMSE
3	0.1505	0.0542	0.1589	0.0393
6	0.1497	0.0534	0.1576	0.0342
9	0.1478	0.0494	0.1553	0.0340

In terms of test MAEs and test RMSEs, Tables 1 and 2 show that the EAR predictor outperforms AR-MLE at all three model orders. This better performance of EAR is attributed to its improved training technique and effective noise tolerance mechanism.

Conclusion

An evolving AR predictor, EAR, has been developed in this study for time series forecasting. The EAR can gradually learn the training data characteristics with more training epochs and accurately forecast the future states of a dynamic system. The noisy samples are addressed using a weight regulator to reduce their misleading effect in training. The effectiveness of the proposed EAR predictor is verified using Mackey-Glass simulation examples. Test results have shown that the EAR predictor can effectively capture the dynamic behaviour of a time series and predict its future states accurately.

Author's Contributions

All authors equally contributed in this work.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

- Brockwell, P.J. and R.A. Davis, 2009. *Time Series: Theory and Methods*. Springer, New York, ISBN-10: 1441903194, pp: 577.
- Cao, J., S. Kwong and R. Wang, 2012. A noise-detection based AdaBoost algorithm for mislabeled data. *Pattern Recognit.*, 45: 4451-4465. DOI: 10.1016/j.patcog.2012.05.002
- Drucker, H., 1997. Improved boosting algorithms using confidence-rated predictions. *Proceedings of the 14th International Conference on Machine Learning, (CML' 97)*, pp: 107-115.
- Farmer, J.D., 1982. Chaotic attractors of an infinite-dimensional dynamical system. *Phys. D: Nonlinear Phenomena*, 4: 366-393. DOI: 10.1016/0167-2789(82)90042-2
- Goude, Y., R. Nedellec and N. Kong, 2014. Local short and middle term electricity load forecasting with semi-parametric additive models. *IEEE Trans. Smart Grid*, 5: 440-446. DOI: 10.1109/TSG.2013.2278425
- Hevia, C., 2008. Maximum likelihood estimation of an ARMA(p,q) model. The World Bank, DECRG.
- Li, D.Z. and W. Wang, 2011. An enhanced GA technique for system training and prognostics. *Adv. Eng. Software*, 42: 452-462. DOI: 10.1016/j.advengsoft.2011.03.014
- Li, D., W. Wang and F. Ismail, 2013a. Enhanced fuzzy-filtered neural networks for material fatigue prognosis. *Applied Soft Comput.*, 13: 283-291. DOI: 10.1016/j.asoc.2012.08.031
- Li, D., W. Wang and F. Ismail, 2013b. Fuzzy neural network technique for system state forecasting. *IEEE Trans. Cybernet.*, 43: 1484-1494. DOI: 10.1109/TCYB.2013.2259229
- Li, D.Z., W. Wang and F. Ismail, 2014a. An evolving fuzzy neural predictor for multi-dimensional system state forecasting. *Neurocomputing*, 145: 381-391. DOI: 10.1016/j.neucom.2014.05.014
- Li, D., W. Wang and F. Ismail, 2014b. A fuzzy-filtered grey network technique for system state forecasting. *Soft Comput.* DOI: 10.1007/s00500-014-1281-1
- Li, D.Z., W. Wang and F. Ismail, 2014c. A mutated particle filter technique for system state estimation and battery life prediction. *IEEE Trans. Instrument. Measur.*, 63: 2034-2043. DOI: 10.1109/TIM.2014.2303534
- Quan, H., D. Srinivasan and A. Khosravi, 2013. Short-term load and wind power forecasting using neural network-based prediction intervals. *IEEE Trans. Neural Netw. Learn. Syst.*, 25: 303-315. DOI: 10.1109/TNNLS.2013.2276053
- Schapire, R.E. and Y. Singer, 1999. Improved boosting algorithms using confidence-rated predictions. *Machine Learn.*, 37: 297-336. DOI: 10.1023/A:1007614523901
- Solomatine, D.P. and D.L. Shrestha, 2004. AdaBoost.RT: A boosting algorithm for regression problems. *Proceedings of the International Joint Conference on Neural Networks*, Jul. 25-29, IEEE Xplore Press, pp: 1163-1168. DOI: 10.1109/IJCNN.2004.1380102
- Wang, W., 2013. An enhanced diagnostic system for gear system monitoring. *IEEE Trans. Syst. Man Cybernet.*, 38: 283-291. DOI: 10.1109/TSMCB.2007.908864
- Wang, W., D. Li and J. Vrbanek, 2012. An evolving neuro-fuzzy technique for system state forecasting. *Neurocomputing*, 87: 111-119. DOI: 10.1016/j.neucom.2012.02.006
- Yu, L., H. Chen, S. Wang and K.K. Lai, 2009. Evolving least squares support vector machines for stock market trend mining. *IEEE Trans. Evolut. Comput.*, 13: 87-102. DOI: 10.1109/TEVC.2008.928176