Archana Thakran. et al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 4, Issue 2, June 2017, pp. 139-148

Analysis and Prediction of Nonstationary Financial Time Series using OSELM

Archana Thakran[1], Ram Pal Singh[2], and Akshay Bhasin[3]

[1]Department of Computer Science, ARSD, University of Delhi, Delhi, INDIA.
[2]Department of Computer Science, DDUC, University of Delhi, Delhi, INDIA.
rprana@gmail.com
[3]Department of Computer Science and Engg,BIT, Mesra, INDIA.
akshay8595@yahoo.com

Abstract. In this paper, we study the application of Online Sequential Extreme Learning Machine(OSELM) for forecasting time series which are chaotic, nonlinear complex and nonstationary inherently. OSELM is a learning algorithm for single layer feed forward neural networks(SLFNs) which can learn the data one-by-one or chunk-by-chunk with xed or varying chunk size with much faster rate than the conventional learning algorithms. It has been shown in literature that OSELM computation-ally faster with better generalization ability than other existing popular machine learning methods. The experiments are performed on di erent time series like IBM and SantaFe-A with di erent activation functions namely sigmoid, sin, hardlim and rbf. It is observed that experimental results are very competitive and close to the actual values of real life nonstationary such as IBM and SantaFe-A. Simulation results demon-strate the e ectiveness and e ciency of method in dealing with complex nonlinear time series forecasting.

Keywords: ELM, Financial Time Series,OSELM,Regression

1 Introduction

For practical situation in dynamic complex system modelling, nonlinear time series prediction and signal processing play very important role. Arti cial neural networks (ANNs) were widely used for pattern recognition, classi cation and regression problems. Despite of various advantages of (ANNs) such as better approximation capabilities, simple network structure still su ers from disadvan-tages like presence of local minima's, imprecise learning rate, selection of hid-den neurons and over tting etc. Therefore, in order address existing problems of ANNs, OSELM has been implementedand applied to predict nonstationary (where statistical parameters like mean and standard deviation alter over a pe-riod of time)time series like nancial time series IBM and SantaFe-A(laser gener-ated time series data were recorded from a Far-Infrared-Laser in a chaotic state). In order to handle this type of time series, the optimal number of hidden nodes and activation functions to be determined beforehand by users. Financial time series forecasting is a challenging real time regression problem. For past decade, gradient descent method [1] has been used in various learning algorithms. How-ever, the gradient descent method su ers from major problems like very slow learning rate, may converge to local minima's, therefore requires large number of iteration steps to obtain better learning performance. Another technique pro-posed by Vapnik et al. was Support Vector Machine(SVM) [2] which maps data set from input space to a high dimensional feature space through some prede-termined non linear mapping.

Latest technique Extreme Learning Machine (ELM) suggested by Huang et al. [3] which is a learning algorithm for Single Hidden layer Feed forward net-works (SLFNs). In ELM formulation, input weights and hidden layer biases are arbitrarily generated and xed. Later we solve the system of linear equations in terms of unknown weights connecting hidden layer to output layer. The solution of these equations is obtained using Moore-Penrose generalised Pseudo Inverse [5]. Since output weights are analytically determined using the least squares method thereby allowing a great reduction in training time. Another variants of ELM introduced by Liang et al.[6][7] is Online Sequential Learning Machine (OS-ELM) which is simple and e cient learning algorithm that can handle training data one by one or chunk by chunk of varying length and discard data samples for which training has already been done and it uses recursive least squares so-lution for algorithm.

In this paper, Section 2 discusses brie y OSELM algorithm and study its appli-cability for given time series. Section 3 covers the preprocessing of data sets by accumulated generating operation (AGO). It also brie y discusses various acti-vation functions that are used. Section 4 gives experimental results and Section 5 discusses conclusion.

2 Online Sequential ELM (OSELM)

In this research paper, OSELM has been used to analyse the given time series as training process and predict the future values by training OSELM using training samples.Online sequential ELM introduced by Liang et al.[6] was an improve-ment and it can handle the real time data more e ciently. In certain situations the entire data set may not be available initially but keeps arriving at di erent intervals of time which OSELM can handle such data e ectively. OSELM can learn from the training samples by taking one element at a time or by grouping the samples into varying size chunks[6].

Salient Features of OSELM are:

{ It can learn from the recently arrived training samples instead of entire historic training samples. OSELM discards the samples for which the training is complete.

{ Training samples are clubbed into chunks of varying or xed size and the training samples are fed sequentially to the learning algorithm.

{ OSELM has no former knowledge about the number of training samples presented to it.

In OSELM the input weights and biases are randomly generated and xed and based on this the output weight matrix is determined using the recursive least squares algorithm [8].

$$\sim = Z^{y}Y \quad (1)$$

where rank(Z) = P , P is the number of hidden neurons. Sequential solution of Eqn.(1) results in implementing OSELM.

Learning Algorithm of OSELM: Given training sample N = f(xi; yi)kxi $<^{M}$; yi $<^{N}$; i = 1; ; Ng

where N training data samples are available initially[3][4][6].

1. Initialization Phase:
   Take a small chunk of training data N_0 = f(xi; yi)ki = 1; ; N_0g from the given training set N where N_0 P
   (a) Arbitrarily generate the hidden neuron parameters(wi; bi)i = 1; ; P .
   (b) Compute initial hidden layer output matrix Z_0

$$Z_0 = \begin{bmatrix} G(w_1 b_1 x_1) & \cdots & G(w_P b_P x_1) \\ \vdots & \ddots & \vdots \\ G(w_1 b_1 x_{N_0}) & \cdots & G(w_P b_P x_{N_0}) \end{bmatrix}_{N_0 \, P} \quad (2)$$

   (c) Approximate the initial output weight matrix (0) where it is given

$$^{(0)} = Z_0^{y} Y_0 \quad (3)$$

   where Z_0 and Y_0 is de ned as $Z_0^{y} = (Z_0^{T} Z_0)^{1} Z_0^{T}$, G_0 = $Z_0^{T} Z_0$ and
   Y_0 = fy_1; ; y_{N_0} g^{T}
   (d) Set k = 0.
2. Sequential Learning Phase:
   (a) Take (k + 1)th chunk of training data

$$N_{k+1} = f(x_i; y_i)ji = (\sum_{j=0}^{k} N_j) + 1; \; ; (\sum_{j=0}^{k+1} N_j) \quad (4)$$

   where each chunk is of varying sizes N_0,N_1, ,N_{k+1}.
   (b) Set $Y_{(k+1)}$ as given by array

$$Y_{(k+1)} = \begin{bmatrix} y^{T}_{(\sum_{j=0}^{k} N_j+1)} \\ \vdots \\ y^{T}_{(\sum_{j=0}^{k+1} N_j)} \end{bmatrix}_{N_{(k+1)} \, N} \quad (5)$$

(c) Compute $Z_{(k+1)}$ as given at bottom of the page[1]

(d) Compute $(k + 1)^{th}$ output weight matrix $^{(k+1)}$ sequentially using the following equation

$$^{(k+1)} = {}^{(k)} + G_{(k}{}^{1}{}_{+1)} Z_{(}{}^{T}{}_{k+1)} (Y_{(k+1)} \quad Z_{(k+1)} \; {}^{(k)}) \qquad (7)$$

where $G_{k+1} = G_k + Z^T_{k+1} Z_{k+1}$

(e) Set k=k+1. Goto step 2(a)

3 Experimental Results

3.1 Data Sets

The data sets available for experiment are inherently noisy, chaotic, nonstation-ary and highly random. We have used IBM as nancial time series[2] and laser generated time series data were recorded from a Far-Infrared-Laser in a chaotic state SantaFe-A[3] in the experiments. A data plot was made of a SantaFe-A se-ries and shown in Fig.1. The data size and their ranges have been described in the Table 1.

Data Sets	Time Period	Training Samples	Testing Samples
IBM	**1978-2014**	**9871**	**3290**
SantaFe-A	**-**	**750**	**250**

Table 1: Details of the Data Sets used in Prediction Analysis.

3.2 Preprocessing of Data Sets with AGO

The rst order accumulated generating operation(AGO) on original time se-ries for reducing stochastic volatility and provides smoothness[9]. The 1^{st} order (AGO) may be summarised as follows. Let the original data be $X^{(0)}$ having N samples are given by

$$x^{(0)} = fx_1^{(0)}; x_2^{(0)}; \quad ; x_i^{(0)}; \quad ; x_N^{(0)} g \qquad (8)$$

1

$$Z_{(k+1)} = \begin{bmatrix} G(w_1; b_1; x(\sum_{j=0}^{k} N_j+1)) & \cdots & G(w_P; b_P; x(\sum_{j=0}^{k} N_j+1)) \\ \vdots & \ddots & \vdots \\ G(w_1; b_1; x(\sum_{j=0}^{k} N_j+1)) & \cdots & G(w_P; b_P; x(\sum_{j=0}^{k} N_j+1)) \end{bmatrix}_{N_0 \times P} \qquad (6)$$

2 http:// nance.yahoo.com/home

3 www-psych.stanford.edu/ andreas/Time-Series/SantaFe.html

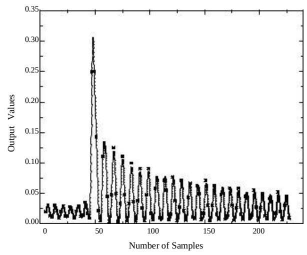

Fig. 1: Time Series of a SantaFe-A

where $x^{(0)}_i$ is the time series sample in original series $x^{(0)}$ at time step of i. For our analysis all data values to be positive. Eqn.(8) is transformed into a new sequence

$$x^{(1)} = \{x_1^{(1)}; x_2^{(1)}; \ ; x_i^{(1)}; \ ; x_N^{(1)}\} \qquad (9)$$

by the 1st order AGO where

$$x_k^{(1)} = \sum_{i=1}^{k} x_i(0)$$

$$x_k^{(1)} = \{x_{(k-1)}^{(1)} + x_k^{(0)}\}; \ k = 2; \ ; N$$

with $x_1^{(1)} = x_1^{(0)}$

Further, inverse accumulated generation operation (IAGO) where series in learn-ing model is converted back to its original to achieve the predicted values. AGO prediction model based on OSELM can be discussed in detailed. Liang et al. based on ELM proposed online sequential ELM (OSELM)which has improve-ment of batch learning based ELM to sequential learning. Simulation results are in favoured of OSELM with better generalization ability and fast process-ing compared to other existing methods. Particularly with OSELM, the pro-cesses of learning and prediction of time series are sequentially performed. In OSELM based model, the learning process of k^{th} step, the input and output of OSELM are set as $x^{(1)}(k \ i)(i = 1; 2; \ ; n)$ and $x^{(1)}(k)$, which are output AGO operator applied on the original series $x^{(0)}(k \ i)(i = 1; 2; \ ; n)$ and x (k),respectively and thereby input to learning model. By learning mapping process from $x^{(1)}(k \ i)(i = 1; 2; \ ; n)$ and $x^{(1)}(k)$, the connecting weight pa-rameters are iteratively adjusted by OSELM method and prediction can be made immediately.

In this paper, online adjusted learning is done by another model OSELM for di erent activation functions in hidden layer nodes and results are summarized in the given Tables III-XI. The blocks that represent both AGO and IAGO op-erators give accumulated generation and inverse accumulated generation in this scheme. At $k^{(th)}$ step, learning model output $x^{(1)}(k + 1)$ is obtained by imple-menting $x^{(1)}(k \ i)(i = 1; 2; \ ; n)$ as inputs to the learning machine, where

$x^{(1)}$(k i) are the AGO results of original $x^{(0)}$(k i)(i = 1; 2; ; n). The ul-timate prediction result $x^{(0)}$(k + 1) is then achieved by implementing IAGO: $x^{(0)}$(k + 1) = $x^{(1)}$(k + 1) $x^{(1)}$(k).

3.3 Normalization

In all our experiments the original data sets are normalised with zero error and standard deviation equals to 1 and data is adjusted into the range [0,1]. To perform normalization the following expression is used as

$$x\hat{}_1 = \frac{X_1 \quad X_{min}}{X_{max} \quad X_{min}} \tag{10}$$

3.4 Performance Analysis

To analyse the prediction performance results obtained by using the learn-ing algorithm OSELM with statistical metrics, namely, root mean square er-ror (RMSE), normalized mean square error (NMSE), mean absolute di erence (MAD) and directional symmetry (DS). The de nitions of these criteria are listed in the Table 2. The values of NMSE and RMSE give the measures of the deviation between predicted values and actual values and smaller the values of NMSE and RMSE, the closer are the actual values to the predicted values of time series. The directional symmetry (DS) metrics provides the correctness of predicted direction having higher value percentage indicates better predictor. In this experimental investigation, OSELM has been used alongwith di erent activation functions provide better performance on di erent time series used for prediction under AGO and normalization smoothness parameter. The machine learning algorithm for on said time series are implemented in the experimental results using MATLAB code [4]. For the sake of comparison of di erent activation functions like sigmoid,sin,hardlim and rbf are used in experiments.

3.5 Simulation Results

The e ectiveness of proposed prediction system is determined by employing non-linear complex IBM and SantaFe-A time series data set. All results are simulated and calculated for OSELM with AGO operator with MATLAB 10 environment with 2.80GHz (CPU) and with 4GB memory (RAM). OSELM based method shows better prediction performance than that of other existing methods with smaller prediction errors and prediction accuracies of OSELM is more or less at the same level with faster processing speed but better than support vector re-gression based methods. Results obtained from OSELM compared with existing methods as given in Tables II-V.

In order to further demonstrate the e ectiveness of OSELM with existing learn-ing methods, nancial time series IBM and SantaFe-A time series and their re-spective ranges for training and testing data set size are shown in Table I. In

[4] http://www.ntu.edu.sg/home/egbhuang

the typical examples, OSELM is tested one-by-one and chunk-by-chunk learning with 20 sample chunk size. As given in Table I, IBM data samples size starting from year 1978 to 2014 has been used. Out of which 9871 data sample is used for training process and the remaining 3290 data samples used for testing pro-cess. Second time series in Table I is SantaFe-A laser generated time series data were recorded from a Far-Infrared-Laser in a chaotic state. For this time series training process data samples size is 750 and testing process samples size is 250 and performance based on determined criteria is explicitly given in Eqns.(9-12). Performance results on learning algorithms with activation functions sigmoid (sig) and sine (sin) with respective time series is given in Tables II-V.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(A_i - P_i)^2} \quad (11)$$

$$NMSE = \frac{1}{\sigma^2 n}\sum_{i=1}^{n}(A_i - P_i)^2 \quad (12)$$

where $\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(P_i - \bar{P})^2$

$$MAD = \sum_{i=1}^{n}\frac{|(A_i - P_i)^2|}{n} \quad (13)$$

$$DS = \frac{100}{n}\sum_{i=1}^{n}(d_i) \quad (14)$$

It should be noted that A and P represent actual and predicted values, n is the total number of data points

AF	PD	LA	LM	Time fsecg	RMSE		] Nodes
					Training	Testing	
sigmoid	Normalisation	ELM	Batch	0.0625	0.0189	0.0727	3
		OSELM	20-by-20	0.1875	0.0194	0.0496	5
	AGO	ELM	Batch	0.9531	0.0029	0.0050	139
		OSELM	20-by-20	0.1094	0.0061	0.0004	11
sin	Normalisation	ELM	Batch	0.0313	0.0199	0.0567	3
		OSELM	20-by-20	0.1250	0.0183	0.1487	3
	AGO	ELM	Batch	0.1875	0.0054	0.0004	31
		OSELM	20-by-20	0.1712	0.0058	0.0004	13
hardlim	Normalisation	ELM	Batch	2.8437	0.0157	0.2125	183
		OSELM	20-by-20	2.7187	0.0354	0.1495	153
	AGO	ELM	Batch	1.4688	0.0178	0.0026	111
		OSELM	20-by-20	5.1406	0.0239	0.0031	181
rbf	Normalisation	ELM	Batch	0.0313	0.0179	0.1723	7
		OSELM	20-by-20	0.2187	0.0181	0.2462	3
	AGO	ELM	Batch	0.3750	0.0053	0.0004	41
		OSELM	20-by-20	0.2187	0.1613	0.0106	3

Table 2: Regression Analysis of IBM nancial time series AF:Activation Function, PD:Preprocessing of Data, LA: Learning Algorithm, LM:Learning Mode.

AF	PD	LA	LM	Time fsecg	RMSE		] Nodes
					Training	Testing	
sigmoid	**Normalisation**	**ELM**	**Batch**	**0.0006**	**0.1412**	**0.1892**	**35**
		OSELM	**20-by-20**	**0.0313**	**0.1592**	**0.1914**	**13**
	AGO	**ELM**	**Batch**	**0.0313**	**0.0171**	**0.0299**	**31**
		OSELM	**20-by-20**	**0.0313**	**0.0200**	**0.0060**	**25**
sin	**Normalisation**	**ELM**	**Batch**	**0.0312**	**0.1411**	**0.1914**	**35**
		OSELM	**20-by-20**	**0.0312**	**0.01693**	**0.1414**	**37**
	AGO	**ELM**	**Batch**	**0.0312**	**0.0203**	**0.0294**	**19**
		OSELM	**20-by-20**	**0.0781**	**0.0310**	**0.0308**	**3**
hardlim	**Normalisation**	**ELM**	**Batch**	**0.0625**	**0.1462**	**0.1864**	**81**
		OSELM	**20-by-20**	**0.3437**	**0.1384**	**0.1888**	**187**
	AGO	**ELM**	**Batch**	**0.0468**	**0.0234**	**0.0287**	**81**
		OSELM	**20-by-20**	**0.3125**	**0.0265**	**0.0087**	**189**
rbf	**Normalisation**	**ELM**	**Batch**	**0.0312**	**0.1384**	**0.1829**	**37**
		OSELM	**20-by-20**	**0.0312**	**0.2012**	**0.1324**	**37**
	AGO	**ELM**	**Batch**	**0.0625**	**0.0197**	**0.0047**	**27**
		OSELM	**20-by-20**	**0.0625**	**0.0011**	**0.0047**	**27**

Table 3: Regression Analysis of SantaFe-A laser generated time series AF:Activation Function,PD:Preprocessing of Data, LA: Learning Algorithm, LM:Learning Mode.

			IBM			SantaFe-A		
AF	PD	Models	Metrics			Metrics		
			NMSE	**MAD**	**DS**	**NMSE**	**MAD**	**DS**
sig	**AGO**	**OSELM**	**0.0162**	**0.00032**	**85.58%**	**0.03212**	**0.0259**	**46.53 %**
	NORM	**OSELM**	**0.0693**	**0.0408**	**49.45%**	**0.0471**	**0.0159**	**62.37 %**
sin	**AGO**	**OSELM**	**0.1387**	**0.0011**	**89.64%**	**0.01387**	**0.0031**	**61.64%**
	NORM	**OSELM**	**0.2344**	**0.0594**	**50.94%**	**0.01344**	**0.0594**	**68.94%**

Table 4: Performance Analysis of IBM and SantaFe-A time series using OSELM learning algorithm where PD:Preprocessing of Data.

Data Set	Converged NMSE				Computation Time			
	RLFSVR[10]	**SVR[10]**	**OSELM**	**ELM**	**RLFSVR[10]**	**SVR[10]**	**OSELM**	**ELM**
IBM	**2.658**	**3.714**	**0.0162**	**0.4709**	**0.235**	**6.6000**	**0.1094**	**0.9531**

Table 5: Comparative analysis of the OSELM and ELM learning models with other models RLFSVR and SVR

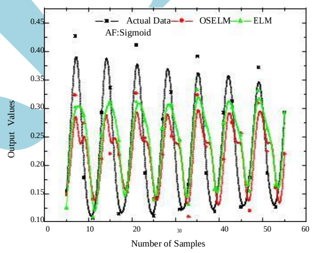

Fig. 2: Predicted Result of SantaFe-A time series using normalised data and learning algorithms OSELM and ELM.

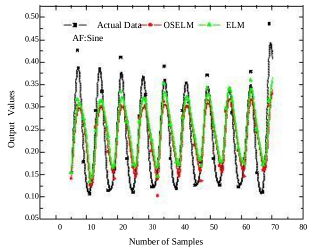

Fig. 3: Predicted Result of SantaFe-A time series using normalised data and learning algorithm OSELM and ELM.

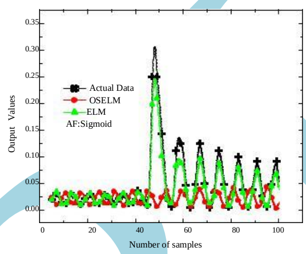

Fig. 4: Predicted Result of SantaFe-A time series using AGO data and learning algorithm OSELM and ELM.

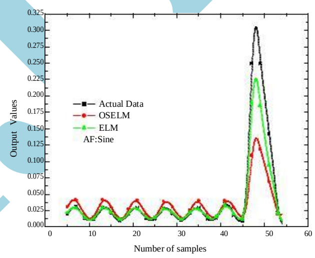

Fig. 5: Predicted Result of SantaFe-A time series using AGO data and learning algorithm OSELM and ELM.

4 Conclusion

The use of OSELM for complex nonlinear time series forecasting is studied in the paper. It is concluded by observing that in most of the cases OSELM per-forms better than SVR and RLFSVR in terms of NMSE and computational time in predicting the future values of time series and other category of time series like SantaFe-A. The learning algorithm provide promising alternative to BP neural network and support vector machines for time series forecasting. As demonstrated in the experiment, OSELM forecast signi cantly better with AGO and normalization preprocessing ways for data set than support vector regression based forecasting model. Moreover, several activation functions such as sigmoid, sin, hardlim and radial bias function are used to determine better performance of forecasting model. The performance of learning model has been evaluated by carrying out a series of experiments on various time series. Finally, extreme learning based method has very few parameters need to tune and optimize for obtaining optimum results and there is very minimal chances of either over-tting or under- tting of the training data and leads to no signi cance impact on models performance. The proposed implementation of online sequential ELM has unique advantage that it can adaptively tune weights when some new chunk of data samples are added or deleted.

References

1. Rumelhart. D.E., Hinton. G.E., Williams. R.J., Learning Representing by Back Prop-agation Errors, Nature, 323, pp.533-536,1986.
2. Vapnik. V.N., The Nature of Statistical Learning, 2nd ed, New York: Springer, 2000.
3. Huang. G-B., Wang. D-H. and Lan. Y., Extreme Learning Machine : A Survey, In-ternational Journal of Machine Learning and Cybernitics,vol.2, pp.107-122, 2011.
4. Huang. G-B., Zhu. Q-Y.and Siew. C-K., Extreme Learning Machine : A new Learning Scheme of Feedforward Neural Network,Int. Joint Conf. on Neural Netw. (IJCNN 2004) Budapest,Hungray, pp.25-29, 2004.
5. Rao. C.R. and Mitra. S.K., Generalised Inverse of Matrices and its Applications, Wiley, N.Y, 1971.
6. Liang. N-Y.,.Huang. G-B, Saratchandran. P. And Sundararajan. N, A fast and accurate Online Sequential learning algorithm for feed forward network, IEEE Trans. for Neural Network, 17(6), pp.1411-1423, 2006.
7. Yibin Ye, Squartini. S. and Piazza. F., Online Sequential Extreme Learning Machine in nonstationary environments, Neurocomputing, vol.116, pp.94-101,2013.
8. Chong. E.K.P. and Zak. S.H., An introduction to optimization, Wiley, New York,2001.
9. Luo Y, Zeng. B and Liao. D, Non-equidistant GRM(1,1) Generated by Accumulated Generating Operation of Reciprocal Number and its Application,
International Journal of Computer Science Issues, vol. 10, issue 2, no 2, pp.119-123, 2013.
10. Khemchandani. R, Jayadeva and Chandra. S, Regularised least squares fuzzy support vector regression for nancial time series forecasting, Expert Systems and applications, vol.36, pp.132-138, 2009.

