

Normalize Time Series and Forecast using Evolutionary Neural Network

Sibarama Panigrahi¹, Y. Karali², H. S. Behera³

Department of Computer Science and Engineering, MITS, Rayagada, Odisha, India¹

Department of Computer Science and Engineering, VSSUT Burla, Odisha, India^{2,3}

Abstract

Efficient time series forecasting (TSF) plays a vital role in making better social, organizational, economical and individual strategic decision making under uncertainty. Over the last two decades, application of artificial neural networks (ANNs) to time series forecasting have shown some promise. However, due to several factors, to date, a consistent ANN performance in TSF over different studies has not been achieved. One such factor is normalization of time series before it is fed into any ANN model. Normalization is a pre-processing strategy which has a significant impact on forecast accuracy. Despite its great importance, there has been no general consensus on how to normalize the time series data for ANN models. This paper systematically investigates how to best normalize the univariate time series for ANN models especially, the multilayer perceptron (MLP) network. Five different normalization techniques (Min-Max, Decimal Scaling, Median, Vector and Z-Score) are used to normalize three univariate time series and corresponding forecast accuracy are measured using an evolutionary MLP network. Results show that single-step-ahead and multiple-step-ahead forecast accuracy of ANN depends on the normalization technique being used. It is also observed that with MLP, vector normalization techniques provide better forecast accuracy compared to other normalization techniques considered.

1. Introduction

Time series forecasting (TSF) is the process of predicting the future outcomes based solely on past observations. Traditionally, TSF has been performed predominantly using statistical-based methods [1]. However, over the past few decades artificial neural network (ANN) models were widely used due to its several advantageous features including: 1) data-driven self-adaptive nonlinear methods 2) universal approximators 3) black box in nature. Considering these advantages, more than thousands of papers using ANN based models have been published to forecast time series. Although numerous publications and

several empirical studies have shown superior performance of neural network forecasters over statistical methods [2-3]; but reports were also made on its inferior performance [4-5]. Many factors contribute to this inconsistent performance of neural network. One of the important factors is data normalization before it is fed to any neural network model. Data normalization has a significant impact on the performance of any model because the sole purpose of data normalization is to guarantee the quality of the data before it is fed to any model.

In literature, few studies were done for evaluating the sensitivity of model's performance to different normalization techniques. Mustaffa et al. [6] evaluated the sensitivity of using min-max, decimal scaling and Z-Score normalization techniques in predicting future dengue outbreak using LS-SVM and neural network model (NNM). They suggested that both the models achieve better accuracy using decimal point normalization. Similarly, Eftekhary et al. [7] conducted a study on ranking five normalization techniques based on improved accuracy of support vector machine (SVM) and concluded that non-monotonic normalization method out performs other methods. Jayalakshmi et al. [8] suggested that: data classification using neural networks was dependent on the normalization methods and min-max normalization provides better classification accuracy than other methods. Nayak et al. [9] suggested that neuro-genetic models are sensitive to different normalization techniques after using five normalization techniques for predicting stock index of Bombay stock exchange.

Literature revealed that normalization techniques have a significant impact on performance of a model and normalization technique should be chosen based on the problem and model in hand. Despite the great importance of normalization technique and better performance of ANN models in forecasting across various disciplines, there has been no study to evaluate the sensitivity of normalization techniques to forecasting accuracy using ANNs. Motivated by this need, this paper attempts to evaluate the effect of

various normalization techniques on univariate TSF using an ANN model.

The rest of the paper is organized as follows. Section 2 briefly describes different normalization techniques and DE-ANNT+ method. Section 3 explains the methodology used for univariate TSF using evolutionary neural network. Simulation results are presented in Section 4. Finally, conclusions are drawn in Section 5.

2. Preliminaries

2.1. Normalization

In this study five most popular normalization techniques (such as: decimal scaling, median, min-max, vector and z-score) are considered. All the techniques are discussed briefly using the following notation.

Consider a time series $T = T_1, T_2, T_3, \dots, T_k$ and the normalized series $N = N_1, N_2, N_3, \dots, N_k$.

2.1.1. Decimal Scaling

In this method the decimal point of every data point moves 'P' number of places towards left, where 'P' is the number of digits of maximum absolute value of the dataset.

Mathematically

$$N_i = \frac{T_i}{10^P}$$

where $i = 1, 2, \dots, k$ and $P = \text{length}(\max(|T|))$

2.1.2. Median

In this method all data points are normalized by the median of the original series.

Mathematically

$$N_i = \frac{T_i}{\text{median}(T)}$$

2.1.3. Min-Max

This method linearly transforms data values from a range $[\text{Min}_T, \text{Max}_T]$ to a range $[\text{Min}_N, \text{Max}_N]$ based on the maximum (Max_T) and minimum (Min_T) value of original series (dataset).

Mathematically

$$N_i = \text{Min}_N + \frac{T_i - \text{Min}_T}{\text{Max}_T - \text{Min}_T} \times (\text{Max}_N - \text{Min}_N)$$

2.1.4. Vector Normalization

In this method the time series is considered as a single vector and normalization is carried out by dividing each data value by the root sum squared value of the original series.

Mathematically

$$N_i = \frac{T_i}{\sqrt{\sum_{j=1}^k T_j^2}}$$

2.1.5. Z-Score Normalization

The data values are normalized using the mean (μ_T) and standard deviation (σ_T) of the original data values (series). This method is also called Zero-Mean Normalization because after this normalization the mean of normalized series becomes zero.

Mathematically

$$N_i = \frac{T_i - \mu_T}{\sigma_T}$$

2.2. DE-ANNT+ Method

The objective function of ANN training is a multimodal search problem since it depends on a number of parameters. Therefore the gradient based training algorithms have several shortcomings such as: it can easily get trapped in local minima, have slow convergence properties, training performance is sensitive to initial values of its parameters etc. Therefore, to overcome this problems global optimization technique such as differential evolution (DE) algorithm [10], genetic algorithm [11], particle swarm optimization [12], ant colony optimization [13], a bee colony optimization algorithm [14] or an evolutionary strategy [15] can be used. The DE algorithm is a simple and efficient stochastic direct search method which was introduced several years ago (1997) [10]. Since then it has been developed intensively in recent years [16]. It has various advantages such as: Ability to find global minimum of a non-differentiable, nonlinear and multimodal function irrespective of initial values of its parameters, Parallelizability to cope with computation intensive cost functions, Ease of use and good convergence properties. Thus, in the literature we have found several applications of DE algorithm to ANN training [17-19], more recently [20]. In [17-19], the DE algorithm without adaptive selection of control parameters was used for ANN training where as in DE-ANNT+ [20], a DE algorithm with multiple trial vectors and adaptive selection of control parameters was used for ANN training. In DE-ANNT+ algorithm multiple mutant vectors are generated and the best mutant vector after crossover with target vector produces the trial vector which is used for selection. It was applied to classify the parity-p problem more efficiently than that of DE-ANNT, extended back propagation (EBP), EA-ANNT

and results are comparable to that of Levenberg-Marquardt (LM) algorithm. It was also found that DE-ANNT+ takes less memory than LM algorithm. Hence, in this paper DE-ANNT+ method is used for ANN training. Interested reader may go through [20] to have a detail description regarding DE-ANNT+ method.

In short the DE-ANNT+ operate in following steps:

Step-1: Randomly Initialize the population with each chromosome representing a weight-set of ANN (NOIN input neurons, NOHN hidden neurons, NOON output neurons) having length $([NOIN + 1+NOON] \times NOHN)$, and each gene representing a weight of ANN.

Step 2: Calculate the fitness (mean square on train set) of each chromosome.

Step 3: Generate multiple scale factors to produce multiple mutant vectors after Mutation

Step 4: Select the best mutant vector.

Step 5: Apply binary crossover between target vector and best mutant vector to generate the trial vector.

Step 6: Perform Selection between trial vector and target vector

Step 7: Termination criteria check if satisfied go to step-8 otherwise go to step-3

Step 8: Select the fittest individual as optimal weight set of ANN

3. Method

The main goal of this paper is find the best data normalization technique to forecast univariate time series using an evolutionary artificial neural network (ANN). For this the we fully Connected MLP with only a hidden layer is chosen as computational model because it can be trained faster than two or more hidden layer MLP and still have good approximation capability. The pseudo-code of the methodology used in this paper to perform forecasting using different normalization is given below.

The following methodology is applied to perform SSA forecasting (forecasting horizon=1) and multiple five-step-ahead forecasting (forecasting horizon=5). The SSA is relatively easy and widely discussed in the literature. For MSA forecasting two approaches were found in the literature, such as: direct and recursive. In direct MSA approach k-step-ahead forecasting is obtained directly without obtaining the intermediate 'k' forecasts whereas in recursive approach the k-step-ahead forecasting is obtained by recursively performing 'k' SSA forecasts. In this paper direct approach is used for MSA forecasting because several studies [21] have shown better performance of direct MSA approach than recursive MSA approach.

Pseudo-Code of Methodology

-
1. Normalize the time series using a normalization technique.
 2. Transform the normalized time series into patterns using sliding window method (Depends on the number of input neurons and forecasting horizon)
 3. Divide the normalized patterns into three segments Train (70%), Validation (15%) and Test (15%) patterns.
 4. Train the ANN using DE-ANNT+ method considering the train and validation set.
 - a. **Termination Criteria:** Use the fittest chromosome of the population to perform single-step-ahead (SSA) or multiple-step-ahead (MSA) forecasting on validation set. If the forecast accuracy on validation set of present generation best chromosome performs worse than that of previous generation then terminate and go to step-5.
 5. Obtain the output of ANN on the test set using the optimal weight set for the time series
 6. De-normalize the output of ANN to obtain the actual forecasts.
 7. Measure the forecast accuracy on Train, Validation and Test patterns.
-

4. Experimental Setup and Simulation Results

The simulations in this paper were carried out on a system with Intel® core(TM) 2Duo E7500 CPU, 2.93 GHz with 2GB RAM and implemented using SCILAB5.4.1. All ANNs are trained using DE-ANNT+ [20] with population size 50, number of trial vectors is five and initial value of each chromosome (representing a ANN weight-set) is initialized to uniform distributed random values drawn from a range [-1, 1].

4.1. Time Series

For experimental analysis three univariate time series have been considered from the well-known Hyndman's time series data library exported from datamarket.com. These time series are: Monthly average of exchange rate of Australian dollar measured from July 1969 till August 1995, Wisconsin employment time series measured from January 1961 till October 1975 and Monthly interest rates Government Bond Yield 2-year securities Reserve Bank of Australia measured from January 1969 till September 1994.

4.2. Performance Measure

Literature revealed the use of different measures to evaluate forecast. The NN3 competition organizers have chosen the symmetric mean absolute percentage error (SMAPE) for model evaluation. Hence, for evaluating the forecast accuracy using various normalization techniques, SMAPE measure has been used which is defined as follows.

$$\text{SMAPE} = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i| + |F_i|}{(|Y_i| + |F_i|)/2} \times 100$$

Where Y_i and F_i are true and forecasted values respectively at i^{th} time point, N is the number of forecasting points.

4.3. Results and Discussion

In order to evaluate the effectiveness of normalization techniques on univariate time series forecasting using evolutionary ANN, ten independent simulations were carried out using each of the five normalization techniques for each of the three time series and corresponding 1-step-ahead and 5-step-ahead forecast accuracy were measured. The mean and standard deviation of the results for 1-step-ahead and 5-step-ahead forecast measures are represented in Table-1 and Table-2 respectively. Note that for Australian exchange rate time series; Wisconsin employment time series and monthly interest rates government bond yield 2-year securities reserve bank of Australia time series 2-4-1, 6-3-1 and 12-6-1 ANN structures were respectively used.

Table 1 Average 1-step-ahead SMAPE (%) on three time series with ANN

Normalization Technique	1-Step-Ahead Forecast		
	Train Mean± St.D.	Validation Mean± St.D.	Test Mean± St.D.
Decimal Scaling	3.28± 2.47	3.28 ± 1.75	7.79 ± 9.10
Median	2.95± 1.95	3.54 ± 1.90	5.97 ± 4.71
Min-Max	2.55 ± 1.55	3.09 ± 1.66	5.93 ± 4.47
Vector	3.18 ± 2.24	3.11 ± 1.74	5.39 ± 4.32
Z-Score	2.42 ± 1.54	3.84 ± 2.40	9.04 ± 8.72

It can be observed from Table-1 that the Median, Z-Score and Decimal scaling normalization techniques were outperformed by the other two normalization techniques. Though Min-Max normalization technique performs better than decimal scaling on train and validation sets, Vector normalization provides the best

forecast accuracy on test set. One can observe from Table-2 that vector normalization provides best forecast accuracy on validation and test sets than other methods for 5-step-ahead forecasting.

Table 2 Average 5-step-ahead SMAPE (%) on three time series with ANN

Normalization Technique	1-Step-Ahead Forecast		
	Train Mean± St.D.	Validation Mean± St.D.	Test Mean± St.D.
Decimal Scaling	8.21 ± 4.27	6.99 ± 2.75	18.29 ± 8.97
Median	7.01 ± 3.46	6.94 ± 2.54	17.49 ± 8.06
Min-Max	5.58 ± 2.56	6.64 ± 2.54	21.81 ± 11.36
Vector	7.24 ± 3.62	6.02 ± 2.08	15.44 ± 6.77
Z-Score	4.61 ± 1.81	6.14 ± 2.11	17.13 ± 6.04

5. Conclusion

This paper evaluates the effect of normalization techniques on univariate TSF using evolutionary ANN. For this, five popular normalization techniques (Min-Max, Decimal Scaling, Median, Vector and Z-Score) and three univariate time series are considered. The experimental results revealed that normalization techniques have a significant impact on both single and multiple step-ahead forecasting. The vector normalization techniques gave better forecast accuracy compared to decimal scaling, median, min-max and z-score normalization techniques.

6. References

- [1] S.G. Makridakis, S.C. Wheelright, R.J. Hyndman, *Forecasting: Methods and Applications*.
- [2] N. R. Swanson, H. White, "Forecasting economic time series using flexible versus fixed specifications and linear versus nonlinear econometric models", *International Journal of Forecasting*, vol. 13, no. 4, pp. 439–46, 1997.
- [3] T. Hill, M.O. Connor, W. Remus, "Neural network models for times series forecasting", *Manage. Sci.*, vol. 42, no. 7, pp. 1082–1092, 1996.
- [4] S. Heravi, D. R. Osborn, C. R. Birchenhall, "Linear versus neural network forecasts for European industrial production series", *International Journal of Forecasting*, vol. 20, no.3, pp. 435–446, 2004.
- [5] L. J. Callen, C. C. Kwan, P. C. Yip, Y. Yuan, "Neural network forecasting of quarterly accounting earnings", *International Journal of Forecasting*, vol. 12, no. 4, pp. 475–482, 1996.

- [6] Z. Mustaffa, Y. Yusof, "A Comparison of Normalization Techniques in Predicting Dengue Outbreak", *International Conference on Business and Economic Research IACSIT Press*, vol. 1, pp. 345-349, 2011.
- [7] M. Eftekhary, P. Gholami, S. Safari, M. Shojaee, "Ranking Normalization Methods for Improving the Accuracy of SVM Algorithm by DEA Method", *Modern Applied Science*, vol. 6, no.10, pp. 26-36, 2012.
- [8] T. Jayalakshmi, A. Santhakumaran, "Statistical Normalization and Back Propagation for Classification", *International Journal of Computer Theory and Engineering*, vol. 3, no.1, pp. 1793-8201, 2011.
- [9] S. C. Nayak, B. B. Mishra, H. S. Behera, "Evaluation of Normalization Methods on Neuro-Genetic Models for Stock Index Forecasting", *Information and Communication Technologies (WICT)*, pp. 602-607, 2012.
- [10] R. Storn, K. Price, "Differential evolution- A simple and efficient heuristic for global optimization over continuous spaces", *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.
- [11] D. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning. Reading", MA:Addison-Wesley, 1989.
- [12] J. Kennedy, R.c.Eberhart, Y.Shi, "Swarm intelligence", San Francisco, CA: Morgan Kaufmann, 2001.
- [13] K. Socha, M. Dorigo, "Ant colony optimization for continuous domains", *European Journal Operation Research*, vol. 185, no.3, pp. 1155-1173, 2008.
- [14] D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi, "The bees algorithm- A novel tool for complex optimization problems", in *IPROMS Oxford, U.K.: Elsevier* (2006).
- [15] H.G. Beyer, H.P. Schwefel, "Evolutionary Strategies: A Comprehensive introduction", *Nat. Comput.* Vol. 1, no.1, pp. 3-52, 2002.
- [16] S. Das, P. N. Suganthan, "Differential Evolution: A Survey of the state-of-the-Art", *IEEE Transaction on Evolutionary Computation*, vol. 15, no.1, pp. 4-31, 2011.
- [17] J. Ionen, J. K. Kamarainen, J. Lampinen, "Differential evolution training algorithm for feed-forward neural networks", *Neural Processing Letters*, vol. 17, no. 1, pp. 93-105, 2003.
- [18] J. X. Du, D. S. Huang, X. F. Wang, X. Gu, "Shape recognition based on neural networks trained by differential evolution algorithm", *Neurocomputing*, vol. 70, pp. 896-903, 2007.
- [19] A. Slowik, M. Bialko, "Training of artificial neural networks using differential evolution algorithm", in *Proc. IEEE Conf. Human Syst. Interaction, Cracow, Poland*, pp. 60-65, 2008.
- [20] A. Slowik, "Application of an Adaptive Differential Evolution Algorithm with Multiple Trial Vectors To Artificial Neural Network Training", *IEEE Transaction on Industrial Electronics*, vol. 58, no.8, pp. 3160-3167, 2011.
- [21] A. Sorjamaa, H. Jin, N. Reyhania, Y. Jia, A. Lendasse, "Methodology for long term prediction of time series", *Neurocomputing*, vol. 70, pp. 2861-2869, 2007.