Neuron-Synapse Level problem decomposition method for Cooperative Neuro-Evolution of Feedforward Networks for Time Series Prediction

Ravneil Nand and Rohitash Chandra

School of Computing Information and Mathematical Sciences
University of South Pacific, Suva, Fiji.
ravneiln@yahoo.com, c.rohitash@gmail.com

Abstract. A major concern in cooperative coevolution for neuro-evolution is the appropriate problem decomposition method that takes into account the architectural properties of the neural network. Decomposition to the synapse and neuron level has been proposed in the past that have their own strengths and limitations depending on the application problem. In this paper, a new problem decomposition method that combines neuron and synapse level is proposed for feedfoward networks and applied to time series prediction. The results show that the proposed approach has improved the results in selected benchmark data sets when compared to related methods. It also has promising performance when compared to other computational intelligence methods from the literature.

1 Introduction

Cooperative coevolution evolutionary algorithm which divides a larger problem into smaller counterparts called sub-components are evolved in isolation and cooperatively evaluated [1]. Cooperative neuro-evolution is a form of machine learning that employs cooperative coevolution (CC) for training [2, 3, 4, 5]. Cooperative neuro-evolution has been seen in solving real world problems such as pattern classification [6, 3], time series prediction [7, 8, 9, 6] and control [10].

It can be generalized that the two established problem decomposition methods for cooperative neuro-evolution are synapse level (SL) [11, 9] and neuron level (NL) [10, 12, 6]. In SL, the network is decomposed to its lowest level of granularity where the number of sub-components depends on the number of weight-links in neural network. In NL problem decomposition, the number of subcomponents consists of the total number of hidden and output neurons. In the case of time series prediction, both NL and SL give very competitive performance [9] while in the case of pattern classification, SL is unable to perform [6]. SL views the network training as a fully separable problem and performs well in problems involving neural network applications where the problem contain less inter-dependencies. It failed in the case of pattern classification as it has

difficulty to decompose the problem and therefore NL performed better as it appeals to partially-separable problems [6].

In recent developments, the combination of NL and SL through a competitive island cooperative coevolution method (CICC) gave better results [7, 13]. In CICC, a problem decomposition method is seen as an island that competes with other islands or problem decomposition methods in different phases of evolution. The winner island at each phase injects its solution to the losing island. This essentially means that the evolutionary processes takes advantage of features of both problem decomposition methods.

There is potential for incorporation of different problem decomposition methods that can share its strengths to solve the problem. CICC used the different problem decomposition features to guide its search. Another way to incorporate them is to use architectural properties of the neural network, i.e use synapse level in the region where more diversity in search is needed and neuron level where decision making is required.

In this paper, we combine neuron and synapse problem decomposition to form a hybrid problem decomposition called Neuron-Synapse Level (NSL) problem decomposition. NSL is intended for training feedforward networks that are chaotic time series problems. NSL problem decomposition enables more subpopulations than NL method and lower number than SL method. The performance of the proposed approach is compared with standalone neuron and synapse level [7].

The rest of the paper is organized as follows. In Section 2, the proposed method is presented while in Section 3, experiments and results are given with discussion. Section 4 concludes the paper with a discussion of future extensions of the paper.

2 Neuron-Synapse Level Problem Decomposition

In cooperative coevolution, problem decomposition is based on the architectural properties of the neural network. Synapse level problem decomposition decomposes the neural network having highest number of subcomponents where each interconnected weight becomes a subcomponent [11]. Whereas, neuron level problem decomposition employs hidden and output neurons as reference for decomposition and the number of subcomponents depend on them [14].

In the proposed NSL decomposition method, each subcomponent consists of incoming and outgoing connections associated with neurons in the hidden layer. It is similar to method to cooperative coevolutionary model for evolving artificial neural networks (COVNET) [15] where all weight connections are treated as incoming weights. The difference lies in the breaking down the network further in the weights connected by hidden-output layer where the decomposition is at synapse level.

The calculation of the actual output is the sum of all the outputs generated as in all the methods mentioned earlier. NSL employs a single subcomponent for each neuron that groups interacting variables (synapses) that are connected to

the hidden neuron. Therefore, each subcomponent for a layer is shown in Fig.1 and composed as follows:
1. Input-hidden layer subcomponents: comprises of all the weight connection from input neuron i to the hidden layer j and the bias. Input-hidden layer weights are decomposed as neuron level [14]:
2. Hidden-output layer subcomponents: comprises of all weight-links from each neuron j in the hidden layer connected to all output layer neurons k and the bias . Hidden-output layer weights are decomposed as synapse level [11].

The total number of sub-components is equal to total number of hidden neurons plus the number of weights and biases within hidden and output neuron in the neural network. The sub-components are implemented as sub-populations.

The proposed method is used in training feedforward network and is shown in Algorithm 1. In Step 1, the problem is broken down in the number of sub-components based on the decomposition technique used. In Step 2, the encoding of the problem takes place based on the number of neurons in the hidden layer.

Once the network has been encoded, the algorithm executes Step 3 where evolution take place using genetic operators for each sub-population. The sub-populations are evolved using genetic operators as defined by the evolutionary algorithm selected. Evaluation of the fitness of each individual for a particular sub-population is done cooperatively with the fittest individuals from the other sub-populations [1].

Cooperative evaluation for an individual in a particular sub-population is done by concatenating the fittest individuals from the rest of the sub-populations. All the sub-populations are evolved for a fixed number of generations. Once the network has been evolved according to the maximum fitness evaluations specified, the generalisation performance is tested.

It was observed for Enforced Sub-population (ESP) [16] that during training, too many or too few hidden units can seriously affect learning and generaliza-

Algorithm 1: NSL for Training Feedforward Networks

Step 1: Decompose the problem into subcomponents according to NSL.
Step 2: Encode each subcomponent in a sub-population according to hidden layer.
Step 3: Initialise and cooperatively evaluate each sub-population.
foreach *Cycle until termination* **do**
  foreach *Sub-population* **do**
    foreach *Depth of n Generations* **do**
      Select and create new offspring using genetic operators
      Cooperative Evaluation the new offspring
      Add new offspring's to the sub-population
    end
  end
end

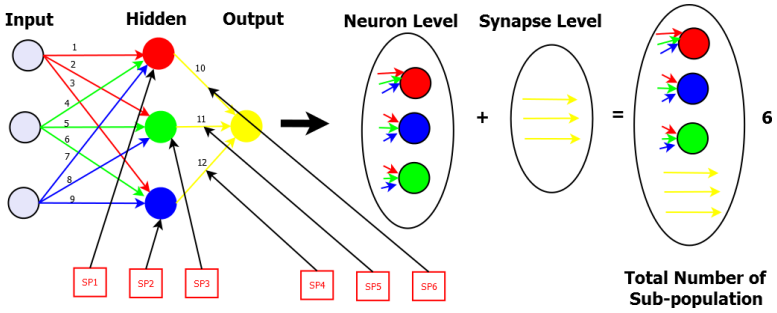

Fig. 1. Neuron-Synapse Level Problem Decomposition (NSL) breaks down the neural network into specific regions and encodes it into the respective sub-populations.

tion. In synapse and neuron level problem decomposition methods, the number of hidden neurons defined the number of sub-populations which affects diversity. Synapse level enabled the most diversity but falls short in applications where the problem is not separable and contain inter-dependencies which affects the training of the weights as separable problem. In the proposed method, the number of sub-populations would be increased when compared to neuron level and hence more diversity is given while the network problem is approached as separable (synapse level) and non-separable (neuron level) training problem [6].

3 Experiments, Results and Discussion

This section reports on the results and discussions based on the experiments conducted on chaotic time series problems using the proposed neuron-synapse level (NSL) problem decomposition method for cooperative neuro-evolution of feedforward networks. Neuron level (NL) and synapse level (SL) problem decomposition methods from the literature have been used for comparison [9].

3.1 Experimental Setup

We used five different time series data sets to train and test the proposed method. Taken's embedding theorem [17] is used to reconstruct the data set before data sets as done previously [9, 7, 18].

Mackey Glass time series [19] is simulated data set [19]. The phase space of this original time series is reconstructed with the embedding dimension $D = 3$ and $T = 2$. The time series is scaled in the range [0,1]. The *Lorenz time series* [20] which is also simulated data set was used. was scaled in the range of [-1,1]. The phase space of this time series is reconstructed with the embedding dimension $D = 3$ and $T = 2$.

The *Sunspot time series* [21] is a real world problem that was scaled in the range [-1,1]. The phase space of this time series is reconstructed with the embedding dimension $D = 5$ and $T = 2$.

The Seagate Technology PLC [22] financial time series data set is composed of daily closing stock prices from December 2006 to February 2010. The phase space of this time series is reconstructed with the embedding dimension $D = 5$ and $T = 2$ and scaled in the range [0,1]. The ACI Worldwide Inc. [22] is also a financial time series dataset composed with daily closing stock prices. The phase space of this time series is reconstructed with the embedding dimension $D = 5$ and $T = 2$. The time series is being scaled in the range [0,1]. All the time series is scaled to give an accurate and fair comparison with literature.

The feedforward neural network employs sigmoid units in the hidden layer for the Mackey Glass, Seagate and ACI Worldwide Inc time series. For Lorenz and Sunspot time series, the hyperbolic tangent unit is used. The Root Mean Squared Error (RMSE) and Normalised Mean Squared Error (NMSE) are used to measure the prediction performance of the proposed method as done in [7, 9].

The maximum number of function evaluations used was 50 000 with 50 independent experimental runs. The G3-PCX algorithm uses the *generation gap model* [23] for selection in which a pool size of 2 parents and 2 offspring is placed as seen in literature [7, 9]. As for the population size, 300 was taken from the literature in order to provide a fair comparison [7].

3.2 Results

In Tables 1 - 5, the results are shown for different number of hidden neurons using the NSL, NL and SL method. The results of NSL is compared with the results of standalone cooperative coevolution methods.

The results in the Tables 1 - 5 are based on 95 percent confidence interval on RMSE and shows the best run from different numbers of hidden neurons based on different methods. The *Training* shows the train average with train error sum while *Generalisation* is based on test average with test error sum and lastly *Best* shows the best test rmse. The best results for each method are highlighted in bold.

In Table 1, the Mackey-Glass time series problem is being evaluated. It was seen that NSL has performed much better than synapse method. The method recorded better generalization performance and best training value with seven hidden neurons. The NSL method was unable to outperform NL method.

In Table 2, the Lorenz time series problem is being evaluated. It also shows that the NSL has performed much better than the SL and gave competitive result with NL. It has been also observed that the generalisation performance of the NSL and the other two methods deteriorates as the number of the hidden neuron increases due to over fitting. The best result was seen for three hidden neuron for NSL.

In Table 3, the sunspot time series problem is being evaluated. The time series is real time series where noise is present. In this time series, the NSL method outperforms one of the standalone methods. The 5 hidden neurons have given best result for NSL whereas 3 hidden neurons for the other two methods.

In Table 4, the ACI time series problem is being evaluated. The time series is real time series where noise is present as in sunspot time series. Even for this time

Table 1. The prediction training and generalisation performance (RMSE) of NL, SL and NSL for the Mackey-Glass time series

Prob.	H	Training	Generalisation	Best
FNN-NL	3	0.0107 ± 0.00131	0.0107 ± 0.00131	0.005
	5	0.0089 ± 0.00097	0.0088 ± 0.00097	**0.0038**
	7	**0.0078 ± 0.00079**	**0.0078 ± 0.00079**	0.0040
FNN-SL	3	0.0237 ± 0.0023	0.0237 ± 0.0023	0.0125
	5	0.0195 ± 0.0012	0.0195 ± 0.0012	0.0124
	7	**0.0177 ± 0.0009**	**0.0178 ± 0.0009**	**0.0121**
FNN-NSL	3	0.0119 ± 0.00089	0.0119 ± 0.00090	**0.0049**
	5	0.0107 ± 0.00081	0.0107 ± 0.00081	0.0056
	7	**0.0100 ± 0.00055**	**0.0100 ± 0.00055**	0.0066

Table 2. The prediction training and generalisation performance (RMSE) of NL, SL and NSL for the Lorenz time series

Prob.	H	Training	Generalisation	Best
FNN-NL	3	**0.0170 ± 0.0031**	**0.0176 ± 0.0031**	0.0043
	5	0.0249 ± 0.0062	0.0271 ± 0.0067	**0.0021**
	7	0.0379 ± 0.0093	0.0416 ± 0.0092	0.0024
FNN-SL	3	0.0680 ± 0.0325	**0.0452** ± 0.0229	0.0153
	5	**0.0526** ± 0.0084	0.0546 ± 0.0084	0.0082
	7	0.0574 ± **0.0075**	0.0605 ± **0.0074**	**0.0079**
FNN-NSL	3	**0.0350 ± 0.00914**	**0.0357 ± 0.0093**	**0.0023**
	5	0.0523 ± 0.00978	0.0547 ± 0.00999	0.0099
	7	0.0459 ± 0.0090	0.0494 ± 0.00923	0.0032

Table 3. The prediction training and generalisation performance (RMSE) of NL, SL and NSL for the Sunspot time series

Prob.	H	Training	Generalisation	Best
FNN-NL	3	**0.0207 ± 0.0035**	**0.0538 ± 0.0091**	**0.015**
	5	0.0289 ± 0.0039	0.0645 ± 0.0093	0.017
	7	0.0353 ± 0.0048	0.0676 ± 0.0086	0.021
FNN-SL	3	**0.5391** ± 0.0261	**0.4998** ± 0.0238	**0.210**
	5	0.5601 ± 0.0208	0.5210 ± 0.0177	0.302
	7	0.5682 ± **0.0178**	0.5250 ± **0.0132**	0.344
FNN-NSL	3	0.0403 ± 0.0088	0.0953 ± 0.01443	**0.013**
	5	**0.0356 ± 0.0074**	**0.0842 ± 0.0098**	0.022
	7	0.0396 ± 0.0080	0.0940 ± 0.00987	0.019

Table 4. The prediction training and generalisation performance (RMSE) of NL, SL and NSL for the ACI Worldwide Inc. time series

Prob.	H	Training	Generalisation	Best
FNN-NL	3	0.0214 ± 0.00039	0.0215 ± 0.00039	0.020
	5	0.0203 ± 0.00047	0.0212 ± 0.00041	0.019
	7	**0.0201 ± 0.00038**	**0.0208 ± 0.00033**	**0.019**
FNN-SL	3	0.4666 ± 0.0399	0.4112 ± 0.0362	0.080
	5	**0.4135** ± 0.0388	**0.3902** ± 0.0378	**0.042**
	7	0.4491 ± **0.0279**	0.4244 ± **0.0270**	0.134
FNN-NSL	3	0.0224 ± 0.00113	0.0197 ± 0.00119	0.015
	5	0.0215 ± 0.000364	**0.0185 ± 0.00092**	**0.015**
	7	**0.0209 ± 0.000364**	0.0192 ± 0.0010	0.015

Table 5. The prediction training and generalisation performance (RMSE) of NL, SL and NSL for the Seagate time series

Prob.	H	Training	Generalisation	Best
FNN-NL	3	**0.02530** ± 0.05582	**0.1809 ± 0.03548**	0.032
	5	0.02313 ± 0.07403	0.2261 ± 0.04811	**0.031**
	7	0.02189 ± **0.08061**	0.2408 ± 0.05306	0.053
FNN-SL	3	0.4129 ± 0.03401	**0.3492 ± 0.02977**	0.089
	5	0.3820 ± **0.03381**	0.3727 ± 0.03371	**0.088**
	7	**0.3816** ± 0.04425	0.4183 ± 0.04306	0.094
FNN-NSL	3	0.02187 ± 0.00078	**0.1988** ± 0.04394	**0.025**
	5	**0.01862** ± 0.00029	0.2562 ± **0.04221**	0.028
	7	0.01896 ± **0.00027**	0.3059 ± 0.04504	0.035

series, the NSL has performed much better than the SL and gave competitive result with NL. Five hidden neurons have given the best result for NSL similar to Lorenz time series problem.

In Table 5, the Seagate time series problem is being evaluated. The time series is real time series where noise is present as in sunspot and ACI time series. For this time series, the NSL method outperforms the other two methods. The 3 hidden neurons have given best result for NSL.

Tables 6 - 10, compares the best results from Table 1 - 5 with some of the related methods in literature. The RMSE best run together with NMSE are used for the comparison. The proposed NSL method has given better performance when compared to some of the methods in the literature.

Table 6. A comparison with the results from literature on the Mackey time series

Prediction Method	RMSE	NMSE
AMCC-RNN [8]	7.53E-03	3.90E-04
Locally linear neuro-fuzzy model (2006) [24]	9.61E-04	
SL-CCRNN [9]	6.33E-03	2.79E-04
NL-CCRNN [9]	8.28E-03	4.77E-04
CICC-RNN [7]	3.99E-03	1.11E-04
Proposed FNN-NSL	4.86E-03	4.48E-05

Table 7. A comparison with the results from literature on the Lorenz time series

Prediction Method	RMSE	NMSE
RBF with orthogonal least squares (2006) [24]		1.41E-09
Locally linear neuro-fuzzy model (2006) [24]		9.80E-10
SL-CCRNN [9]	6.36E-03	7.72E-04
NL-CCRNN [9]	8.20E-03	1.28E-03
CICC-RNN [7]	3.55E-03	2.41E-04
Proposed FNN-NSL	2.34E-03	2.87E-05

Table 8. A comparison with the results from literature on the Sunspot time series

Prediction Method	RMSE	NMSE
RBF with orthogonal least squares (2006) [24]		4.60E-02
Locally linear neuro-fuzzy model (2006) [24]		3.20E-02
SL-CCRNN [9]	1.66E-02	1.47E-03
NL-CCRNN [9]	2.60E-02	3.62E-03
CICC-RNN [7]	1.57E-02	1.31E-03
Proposed FNN-NSL	1.33E-02	5.38E-04

Table 9. A comparison with the results from literature on the ACI time series

Prediction Method	RMSE	NMSE
CICC-RNN [7]	1.92E-02	
FNN-SL [18]	1.92E-02	
FNN-NL [18]	1.91E-02	
MO-CCFNN-T=2 [25]	1.94E-02	
MO-CCFNN-T=3 [25]	1.470E-02	
Proposed FNN-NSL	1.51E-02	1.24E-03

Table 10. A comparison with the results from literature on the Seagate time series

Prediction Method	RMSE	NMSE
FNN-SL [18]	3.74E-02	
FNN-NL [18]	2.24E-02	
Proposed FNN-NSL	2.45E-02	3.56E-03

In Table 6, the best result of Mackey-Glass time series problem is being compared. The proposed method outperformed all the methods except CICC-RNN result. Due to competition and collaboration, the method used in CICC-RNN has performed better than the proposed method.

The Table 7 below shows the best result on Lorenz time series problem that is being compared to other computational intelligence methods in the literature. The proposed method outperformed all the methods in terms of the RMSE but was unable to outperform NMSE of two methods, RBF-OLS and LLNF-LoLiMot. These methods have additional enhancements such as the optimization of the embedding dimensions and strength of architectural properties of hybrid neural networks with residual analysis [24].

In Table 8, the best result of the Sunspot time series problem is compared with results in the literature where the proposed method has shown to outperform the rest of the methods.

In Table 9, the best result of the ACI time series problem is being compared with results in the literature. The proposed method could not outperform multi-objective method having T=3, however, the results are better when compared to other methods from the literature.

In Table 10, the best result of the Seagate time series problem is compared with results in the literature. The proposed method has been not able to outperform NL method.

3.3 Discussion

The results obtained were very promising when compared to other methods from literature involving five different data sets. The proposed method (NSL) has given better performances in nearly all benchmark data sets and financial data set used. It is being compared to similar evolutionary methods namely training neural fuzzy networks [24] and competitive cooperative coevolution methods [7]. It creates lower number of subcomponents for the problem when compared to synapse level (SL) and higher number of subcomponents when compared to neuron level (NL). The increase in the number of sub-populations when compared to NL seems to provide more diversity in cooperative coevolution and improve the prediction performance.

NSL incorporates two problem decomposition (NL and SL) to use architectural properties of the neural network, where NL appeals to partially-separable problems where decision making is required and SL views the network training as a fully separable problem and is applied to region where more diversity in search is needed. Therefore, NSL performs better than other methods.

4 Conclusions

In this paper, neuron-synapse level problem decomposition method has been proposed for feedforward networks with application to time series prediction. The proposed method uses the strength of both the problem decomposition methods.

It fulfills the limitations faced by a single problem decomposition method. It can be further enhanced by involving competition in it as done in competitive island cooperative coevolution method where different problem decomposition methods compete and collaborate during evolution.

The method has given promising performance on the different benchmark problems and has outperformed several methods from the literature. The method performs better for real work time series problems when compared to simulated ones. This is an advantage as real work time series problems contains noise that makes prediction models difficult to train and generalise.

In future work, the proposed method can be extended to feedforward and recurrent neural networks for pattern classification problems.

References

[1] M. Potter and K. De Jong, "A cooperative coevolutionary approach to function optimization," in *Parallel Problem Solving from Nature PPSN III*, ser. Lecture Notes in Computer Science, Y. Davidor, H.-P. Schwefel, and R. Mnner, Eds. Springer Berlin Heidelberg, 1994, vol. 866, pp. 249–257.

[2] R. Chandra, M. R. Frean, and M. Zhang, "Crossover-based local search in cooperative co-evolutionary feedforward neural networks," *Appl. Soft Comput.*, vol. 12, no. 9, pp. 2924–2932, 2012.

[3] N. García-Pedrajas and D. Ortiz-Boyer, "A cooperative constructive method for neural networks for pattern recognition," *Pattern Recogn.*, vol. 40, no. 1, pp. 80–98, 2007.

[4] J. Lehman and R. Miikkulainen, "Neuroevolution," vol. 8, no. 6, p. 30977, 2013.

[5] M. A. Potter and K. A. De Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evol. Comput.*, vol. 8, no. 1, pp. 1–29, 2000.

[6] R. Chandra, M. Frean, and M. Zhang, "On the issue of separability for problem decomposition in cooperative neuro-evolution," *Neurocomputing*, vol. 87, pp. 33–40, 2012.

[7] R. Chandra, "Competitive two-island cooperative coevolution for training Elman recurrent networks for time series prediction," in *International Joint Conference on Neural Networks (IJCNN)*, Beijing, China, Jul. 2014, pp. 565 – 572.

[8] ——, "Adaptive problem decomposition in cooperative coevolution of recurrent networks for time series prediction," in *International Joint Conference on Neural Networks (IJCNN)*, Dallas, TX, USA, Aug. 2013, pp. 1–8.

[9] R. Chandra and M. Zhang, "Cooperative coevolution of Elman recurrent neural networks for chaotic time series prediction," *Neurocomputing*, vol. 186, pp. 116 – 123, 2012.

[10] F. Gomez and R. Mikkulainen, "Incremental evolution of complex general behavior," *Adapt. Behav.*, vol. 5, no. 3-4, pp. 317–342, 1997.

[11] F. Gomez, J. Schmidhuber, and R. Miikkulainen, "Accelerated neural evolution through cooperatively coevolved synapses," *J. Mach. Learn. Res.*, vol. 9, pp. 937–965, 2008.

[12] R. Chandra, M. Frean, and M. Zhang, "An encoding scheme for cooperative coevolutionary neural networks," in *23rd Australian Joint Conference on Artificial Intelligence*, ser. Lecture Notes in Artificial Intelligence. Adelaide, Australia: Springer-Verlag, 2010, pp. 253–262.

[13] R. Chandra, "Competition and collaboration in cooperative coevolution of Elman recurrent neural networks for time-series prediction," *Neural Networks and Learning Systems, IEEE Transactions on*, p. In Press, 2015.

[14] R. Chandra, M. Frean, M. Zhang, and C. W. Omlin, "Encoding subcomponents in cooperative co-evolutionary recurrent neural networks," *Neurocomputing*, vol. 74, no. 17, pp. 3223 – 3234, 2011.

[15] N. Garcia-Pedrajas, C. Hervas-Martinez, and J. Munoz-Perez, "COVNET: a cooperative coevolutionary model for evolving artificial neural networks," *IEEE Transactions on Neural Networks*, vol. 14, no. 3, pp. 575–596, 2003.

[16] F. J. Gomez, "Robust non-linear control through neuroevolution," PhD Thesis, Department of Computer Science, The University of Texas at Austin, Technical Report AI-TR-03-303, 2003.

[17] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980*, ser. Lecture Notes in Mathematics, 1981, pp. 366–381.

[18] S. Chand and R. Chandra, "Cooperative coevolution of feed forward neural networks for financial time series problem," in *International Joint Conference on Neural Networks (IJCNN)*, Beijing, China, July 2014, pp. 202–209.

[19] M. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.

[20] E. Lorenz, "Deterministic non-periodic flows," *Journal of Atmospheric Science*, vol. 20, pp. 267 – 285, 1963.

[21] SILSO World Data Center, "The International Sunspot Number (1834-2001), International Sunspot Number Monthly Bulletin and Online Catalogue," Royal Observatory of Belgium, Avenue Circulaire 3, 1180 Brussels, Belgium, accessed: 02-02-2015. [Online]. Available: http://www.sidc.be/silso/

[22] "NASDAQ Exchange Daily: 1970-2010 Open, Close, High, Low and Volume," accessed: 02-02-2015. [Online]. Available: http://www.nasdaq.com/symbol/aciw/stock-chart

[23] K. Deb, A. Anand, and D. Joshi, "A computationally efficient evolutionary algorithm for real-parameter optimization," *Evol. Comput.*, vol. 10, no. 4, pp. 371–395, 2002.

[24] A. Gholipour, B. N. Araabi, and C. Lucas, "Predicting chaotic time series using neural and neurofuzzy models: A comparative study," *Neural Process. Lett.*, vol. 24, pp. 217–239, 2006.

[25] S. Chand and R. Chandra, "Multi-objective cooperative coevolution of neural networks for time series prediction," in *International Joint Conference on Neural Networks (IJCNN)*, Beijing, China, July 2014, pp. 190–197.