Combinational problem decomposition method for Cooperative Coevolution of Recurrent Networks for Time Series Prediction

*

Ravneil Nand
FSTE
The University of the South Pacific
Suva, Fiji
ravneiln@yahoo.com

Mohammed Naseem
SCIMS, FSTE
The University of the South Pacific
Suva, Fiji
mohammed.naseem@usp.ac.fj

Emmenual Reddy
SCIMS, FSTE
The University of the South Pacific
Suva, Fiji
emmenual.reddy@usp.ac.fj

Bibhya Nand Sharma
SCIMS, FSTE
The University of the South Pacific
Suva, Fiji
bibhya.sharma@usp.ac.fj

***Abstract*—The breaking down of a particular problem through problem decomposition has enabled complex problems to be solved efficiently. The two major problem decomposition methods used in cooperative coevolution are synapse and neuron level. The combination of both the problem decomposition as a hybrid problem decomposition has been seen applied in time series prediction. The different problem decomposition methods applied at particular area of a network can share its strengths to solve the problem better, which forms the major motivation. In this paper, we are proposing a combination utilization of two hybrid problem decomposition method for Elman recurrent neural networks and applied to time series prediction. The results reveal that the proposed method has got better results in some datasets when compared to its standalone methods. The results are better in selected cases for proposed method when compared to several other approaches from the literature.**

***Index Terms*—Cooperative coevolution, problem decomposition, recurrent network**

I. Introduction

Coevolutionary algorithms are becoming a prevalent practice in solving computationally difficult problems. The dynamic nature of many real world problems and applications have inspired many researchers to develop algorithms that search for optimal solutions [1]. The cooperative coevolutionary algorithms (CCAs) are evolutionary architectures which solve large problems by disintegrating them into smaller subcomponents and then solving these subcomponents individually to find solution to the larger problem [2], [3]. Here a number of species are evolved together to achieve optimal solutions where individuals are rewarded based on how well they cooperate with one another [4]. Applications of cooperative coevolutionary algorithms have shown encouraging results in training recurrent neural networks [3], [5]–[8].

An issue of cooperative coevolution is its sensitivity towards problem decomposition [9]. According to [10], [11], decomposition strategies have pronounced impact on the performance of cooperative coevolutionary architectures. As such, extensive experimentation is required to identify an ideal decomposition strategy.

Recently, various problem decomposition strategies have been utilized in the field of cooperative neuro-evolution such as the neuron and synapse level decomposition [9] in time series prediction [3], [5], [7]. The adaptive modularity cooperative coevolution framework (AMCC) introduced in [7] uses the strength of different problem decomposition methods at different stages of evolution to solve chaotic time series problem. The work in [3] combines the neuron and synapse decomposition technique to form a hybrid problem decomposition called the Neuron-Synapse Level decomposition (NSL) which produced better results in selected benchmark data sets. Another hybrid problem decomposition was introduced in [12], where neuron and network level problem decomposition were used. Even here, the results produced seemed better in selected datasets.

The results of the Competitive Island based Cooperative Coevolution (CICC) used in [10] shows that competition and collaboration can produce better solutions than the regular decomposition strategies. The combination of two problem decomposition has also produced good results on certain number of datasets.

In this paper, we utilize a combination of two hybrid problem decomposition namely neuron-synapse level problem decomposition and neuron-network level problem decomposition with Elman recurrent network for time series prediction. One problem decomposition is used for certain number of iterations and then the other is run for the same number of iteration, subsequently the best one is selected for training and testings of the network.

The rest of the paper is organized as follows. In Section 2, the proposed problem decomposition is discussed in detail. The Section 3 shows the experimental setup. In Section 4 and 5, the results and discussion are given respectively. Section 6 concludes the paper with a discussion of future extensions of the research.

II. Proposed Method

There is clearly an important role that hybridization or combination plays in everyday life, and the same applies to time series prediction. Using combination of problem decomposition in time series prediction, better results can be sought from each decomposition techniques by refining the training and testing processes.

In this paper, inspired by the research done in [13] where competitive model was used a combinational model is proposed where hybrid problem decomposition algorithms are used to solve a particular problem. In [13], the author used standalone problem decomposition method's in his competitive model, which motivated the use of multiple problem decomposition methods where elimination and selection are done. The model we propose is called Combinational Hybrid Model (CHM) and is shown in Fig.1. CHM breaks down the problem into a lower level using Neuron-Synapse and Neuron-Network level problem decomposition.

In the proposed model, the best solutions are not changed with different methods as done in [13], rather best solutions are just used to train and test the dataset. Initially, the model used one decomposition for certain number of iterations and the other for left over iterations. This way the network was not fully utilizing the potential of both the problem decomposition. Therefore, both were run one after the other for full iteration allowing for best to proceed further and since both decompositions were best for certain times, the combinational result was sorted.

Algorithm 1 shows the proposed method used for training the recurrent network. In the first step of the algorithm, all the subpopulations of both the algorithms are initialized and cooperatively evaluated according to the problem decomposition technique applied.

In the second step, the evolution of the different decomposition method takes place. Here, each method is evolved for a predefined time given by the number of fitness evaluations in a round-robin fashion. This is called *decomposition evolution time* which is given by the number of function evaluations in the respective method. As soon as both decomposition methods have been evolved for the evolution time, the algorithm proceeds to step 3.

In step 3,the two solutions are compared during every run and one solution is eliminated while the best gets selected. This phase is called the *Elimination and Selection*. If both have same results then first decomposition solution is regarded as best and used for training and testing.

Once the best solution is selected, it is then used for training and testing of the dataset, and it continues to run until the number of runs are completed. Apparently, both decompositions collaborate as a team to give a better solution; both decomposition techniques combines as a result.

Algorithm 1: Combinational Hybrid Model for training Recurrent Neural Networks

```
Step 1: Initialisation:
i. Cooperatively evaluate network according to NSL method
ii. Cooperatively evaluate network according to NNL method
Step 2: Evolution:
while FuncEval ≤ GlobalEvolutionTime do
    while FuncEval ≤ Decomposition-Evolution-Time do
        foreach Sub-population at NSL do
            foreach Depth of n Generations do
                Create new individuals using genetic operators
                Cooperative Evaluation
            end
        end
    end
    while FuncEval ≤ Decomposition-Evolution-Time do
        foreach Sub-population at NNL do
            foreach Depth of n Generations do
                Create new individuals using genetic operators
                Cooperative Evaluation
            end
        end
    end
    Step 3: Elimination and selection: Compare the two results. Best solution selected and used for training and testing.
end
```

III. Experimental Setup

In this section, the experimental setup conducted using cooperative coevolution to train recurrent networks is presented.

Taken's embedding theorem [14] allows reconstruction of the dataset before it can be used. The two important conditions of reconstruction of the chaotic time series data into a state space vector are *time delay (T)* and *embedding dimension (D)* [14]. For training and testing the proposed method, the five different datasets are used which are initially split into half. The cooperative coevolution algorithm used in this research is part of the algorithms provided in Smart Bilo Computational Intelligence Framework [15] which provides better comparison with the literature.

The first two datasets used are the *Mackey-Glass time series* dataset [16] and *Lorenz time series* [17] which are simulated

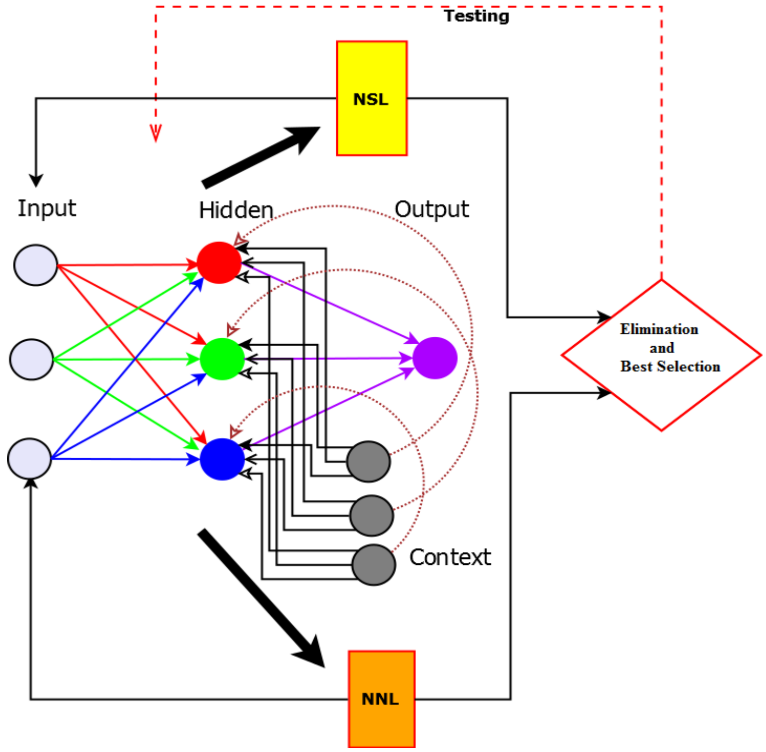

Fig. 1. Combinational Hybrid Model Problem Decomposition method (CHM) breaks down the neural network into separate subpopulation based on the problem decomposition method applied and eliminates and selects best solution for training and testing.

datasets. The third dataset used is *Sunspot time series* [18]. This dataset gives an indication of the solar activities for solar cycles which impact Earth's climate change and is a real world problem [18]. The *ACI Worldwide Inc. time series* is the fourth data set that is used [19]. To obtain the ACI Worldwide Inc. financial time series data set, the NASDAQ stock exchange is used [19]. The Seagate Technology PLC is the last dataset that is used [19]. It is also a financial time series dataset and contains daily closing stock prices.

With reference to literature, the scaling of the five time series dataset is done in the range of [0,1] and [-1,1] in order to provide a fair comparison. The embedding dimensions are kept same as done in literature [3], [12]. Sigmoid units are employed by the recurrent neural network for the Mackey-Glass, Seagate, and ACI Worldwide Inc. time series whereas the hyperbolic tangent unit is used for Lorenz and Sunspot time series.

The algorithm is run 50 times. To terminate each run of the algorithm, the maximum number of function evaluations was set at 50,000 for each decomposition technique. With reference to literature [8], a pool size of 2 parents and 2 offsprings are put in the G3-PCX algorithm. For evolution of all the sup-populations in the proposed method, the G3-PCX evolutionary model which uses the *generation gap model* [20] for selection is used since it has shown good results with cooperative neuro-evolution [21].

The population size was kept at 300 and the number of generations for each sub-population was kept at 1. To compute the prediction performance of the proposed method, the Normalized Mean Squared Error (NMSE) and Root Mean Squared Error (RMSE) are used, similar to that of [8], [13].

$$NMSE = \left(\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} \right) \tag{1}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{2}$$

,where y_i is observed data, $\hat{y}_i$ is predicted data and $\bar{y}_i$ is average of observed data, and N is the observed data's length. These two performance measures are used to compare the results from that of the literature.

IV. Results

The experimental results based on the performance of proposed method are given in this section:

Tables I - V showcase the results for different number of hidden neurons on the proposed method CHM compared to two standalone (NSL and NNL) methods. The results given in the Tables I - V are based on 95 percent confidence interval on RMSE and the best results for each algorithm are highlighted in bold. The *Training* shows the train average with train error sum while *Generalization* is based on test average with test error sum and *Best* shows the best test RMSE.

The Mackey-Glass time series problem is evaluated in the Table I. It was seen that CHM has similar performance as

TABLE I
THE PREDICTION TRAINING AND GENERALIZATION PERFORMANCE (RMSE) BASED ON MACKEY-GLASS TIME SERIES

Method	H	Training	Generalization	Best
RNN-NSL	3	0.0237 ± 0.0021	0.0237 ± 0.0021	0.007
	5	0.0188 ± 0.0018	0.0188 ± 0.0019	0.006
	7	**0.0175 ± 0.0012**	**0.0175 ± 0.0013**	**0.006**
RNN-NNL	3	0.0275 ± 0.0026	0.0275 ± 0.0026	0.011
	5	0.0378 ± 0.0083	0.0378 ± 0.0083	0.014
	7	**0.0164 ± 0.0012**	**0.0164± 0.0012**	**0.009**
RNN-CHM	3	0.0210 ± 0.0046	0.0211 ± 0.0046	0.006
	5	**0.0164 ± 0.0028**	**0.0168 ± 0.0028**	0.007
	7	0.0232 ± 0.0039	0.0233± 0.0039	**0.005**

TABLE II
THE PREDICTION TRAINING AND GENERALIZATION PERFORMANCE (RMSE) BASED ON THE LORENZ TIME SERIES

Method	H	Training	Generalization	Best
RNN-NSL	3	0.0348 ± 0.0169	0.0358 ± 0.0168	0.010
	5	0.0200 ± **0.0029**	**0.0202** ± **0.0030**	0.009
	7	**0.0199** ± 0.0031	0.0213 ± 0.0034	**0.008**
RNN-NNL	3	0.0369 ± 0.0041	0.0374 ± 0.0041	0.015
	5	0.0357 ± 0.0100	0.0358 ± 0.0099	0.009
	7	**0.0223 ± 0.0023**	**0.0226 ± 0.0024**	**0.007**
RNN-CHM	3	**0.0238 ± 0.0050**	**0.0244 ± 0.0051**	**0.006**
	5	0.0358 ± 0.0119	0.0365 ± 0.0120	0.008
	7	0.0406 ± 0.0114	0.0416 ± 0.0116	0.007

NNL method. It has got the lowest best value in comparison to other two methods. The proposed method recorded better generalization performance and best training value with five hidden neurons.

In Table II, the Lorenz time series problem has been evaluated. It was observed that the CHM has poor performance than NSL method and had similar generalization as NNL method. The generalization performance and training of the CHM decreases as the number of the hidden neuron increases. Three hidden neurons for CHM gave the best result.

Table III illustrates the evaluation of the Sunspot time series problem that has presence of noise since it is real-world data. The proposed CHM method was unable to outperform both of the methods (NSL and NNL) in terms of training. The proposed method was better than both the methods in terms of generalization and best value. The best result for CHM was given by three hidden neurons.

In Table IV, the ACI time series problem results are reported. This time series problem also has presence of noise like the Sunspot time series problem. For the given problem, the CHM had similar performance as other methods in terms of training and generalization. It had better best value in comparison to the other two methods. Seven hidden neurons have given the best result for the proposed method.

The Seagate time series problem is evaluated in Table V. For this time series, the CHM method outperformed the other two methods in terms of training, generalization and best value. For CHM, seven hidden neurons gave best results.

TABLE III
THE PREDICTION TRAINING AND GENERALIZATION PERFORMANCE (RMSE) BASED ON THE SUNSPOT TIME SERIES

Method	H	Training	Generalization	Best
RNN-NSL	3	0.0284 ± 0.0044	**0.0612 ± 0.0153**	**0.016**
	5	**0.0250 ± 0.0027**	0.0703 ± 0.0221	0.020
	7	0.0273 ± 0.0112	0.1334 ± 0.0400	0.025
RNN-NNL	3	0.0332 ± 0.0064	0.0682 ± 0.0132	0.026
	5	0.0246 ± 0.0024	0.0772 ± 0.0184	0.021
	7	**0.0203 ± 0.0022**	**0.077 ± 0.0238**	**0.019**
RNN-CHM	3	0.0270 ± 0.0071	**0.0520 ± 0.0124**	**0.015**
	5	0.0269 ± 0.0066	0.0756 ± 0.0196	0.018
	7	**0.0269 ± 0.0046**	0.0793 ± 0.0230	0.017

TABLE IV
THE PREDICTION TRAINING AND GENERALIZATION PERFORMANCE (RMSE) BASED ON THE ACI WORLDWIDE INC. TIME SERIES

Method	H	Training	Generalization	Best
RNN-NSL	3	0.0247 ± 0.0010	0.0198 ± 0.0013	0.015
	5	0.0238 ± 0.0018	0.0187 ± 0.0017	0.015
	7	**0.0225 ± 0.0007**	**0.0173 ± 0.0008**	**0.015**
RNN-NNL	3	0.0255 ± 0.0009	0.0212 ± 0.0014	0.015
	5	0.0297 ± 0.0027	0.0222 ± 0.0020	**0.014**
	7	**0.0220 ± 0.0008**	**0.0171 ± 0.0009**	0.015
RNN-CHM	3	0.0227 ± 0.0019	0.0224 ± 0.0030	0.015
	5	0.0285 ± 0.0032	0.0285 ± 0.0059	0.015
	7	**0.0222 ± 0.0011**	**0.0211 ± 0.0020**	**0.014**

Figures 2 and 3 shows predicted results and error graph of RNN-CHM method with original results on Sunspot dataset for a typical run.

The best results from Table I - V with some of the related methods in literature are given in Table VI. The RMSE best run together with NMSE are given for comparison purposes. The proposed CHM method has shown good performance in nearly all the datasets when compared to other methods in the literature.

The best result on Mackey-Glass time series problem is being compared to works in literature under problem Mackey-

TABLE V
THE PREDICTION TRAINING AND GENERALIZATION PERFORMANCE (RMSE) OF NL, SL AND NNL FOR THE SEAGATE TIME SERIES

Method	H	Training	Generalization	Best
RNN-NSL	3	0.0209 ± 0.0009	**0.1850 ± 0.0359**	**0.028**
	5	**0.0193 ± 0.0004**	0.2293 ± 0.0519	0.055
	7	0.0195 ± 0.0004	0.1987 ± 0.0470	0.034
RNN-NNL	3	0.0234 ± 0.0015	0.1841 ± 0.0395	0.030
	5	0.0241 ± 0.0038	0.2121 ± 0.0495	0.049
	7	**0.0199±0.0004**	**0.1764±0.0333**	**0.021**
RNN-CHM	3	0.0208 ± 0.0033	0.2158 ± 0.0329	**0.023**
	5	**0.0190 ± 0.0008**	0.2081 ± 0.0377	0.036
	7	0.0197 ± 0.0010	**0.1745 ± 0.0310**	0.026

TABLE VI
A COMPARISON WITH THE RESULTS FROM LITERATURE ON DIFFERENT TIME SERIES DATASETS

Problem	Prediction Method	RMSE	NMSE
Mackey Glass	AMCC-RNN [7]	7.53E-03	3.90E-04
	Locally linear neuro-fuzzy model (2006) [22]	9.61E-04	
	SL-CCRNN [8]	6.33E-03	2.79E-04
	NL-CCRNN [8]	8.28E-03	4.77E-04
	CICC-RNN [13]	3.99E-03	1.11E-04
	Proposed CHM	5.48E-03	2.09E-04
Lorenz	RBF with orthogonal least squares (2006) [22]		1.41E-09
	Locally linear neuro-fuzzy model (2006) [22]		9.80E-10
	SL-CCRNN [8]	6.36E-03	7.72E-04
	NL-CCRNN [8]	8.20E-03	1.28E-03
	CICC-RNN [13]	3.55E-03	2.41E-04
	Proposed CHM	6.22E-03	7.37E-04
Sunspot	RBF with orthogonal least squares (2006) [22]		4.60E-02
	Locally linear neuro-fuzzy model (2006) [22]		3.20E-02
	SL-CCRNN [8]	1.66E-02	1.47E-03
	NL-CCRNN [8]	2.60E-02	3.62E-03
	CICC-RNN [13]	1.57E-02	1.31E-03
	Proposed CHM	1.47E-02	1.15E-03
ACI Worldwide	FNN-SL [11]	1.92E-02	
	FNN-NL [11]	1.91E-02	
	MO-CCFNN-T=2 [23]	1.94E-02	
	MO-CCFNN-T=3 [23]	1.47E-02	
	Neuron-Synapse Level method FNN-NSL [3]	1.51E-02	1.24E-03
	Proposed CHM	1.46E-02	2.04E-03
Seagate	FNN-SL [11]	3.74E-02	
	FNN-NL [11]	2.24E-02	
	Neuron-Synapse Level method FNN-NSL [3]	2.45E-02	3.56E-03
	Proposed CHM	2.32E-02	5.62E-03

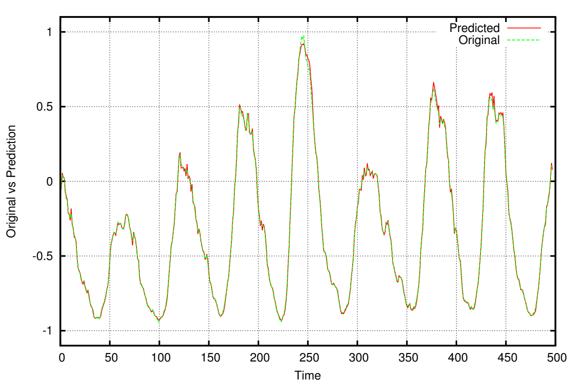

Fig. 2. Performance given by CHM on the testing set for Sunspot dataset.

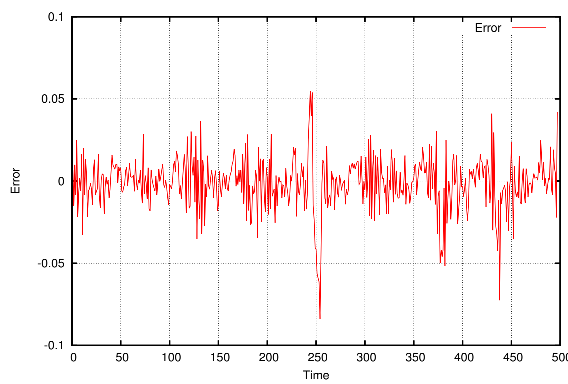

Fig. 3. Error on the test dataset given by CHM for Sunspot dataset .

Glass in Table VI. The proposed method was able to only beat AMCC-RNN, SL-CCRNN and NL-CCRNN in terms of both RMSE and NMSE. Under problem Lorenz, it shows the best result of Lorenz time series problem being compared to works in literature. It has been seen that the proposed method outperformed all the methods expect for CICC model in terms of RMSE.

Again in Table VI, under problem Sunspot, 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 it has been compared to in terms of RMSE. The method has given competitive results.

The best result of the ACI Worldwide Inc. time series problem is compared to works in literature under problem ACI Worldwide. The proposed hybrid method has outperformed all the methods expect for FNN-NSL methods in terms of NMSE. Better and stable performance has been achieved by the CHM.

Again in Table VI, the best result of the Seagate time series problem is compared with results in the literature under problem Seagate. The proposed method has outperformed all the methods expect for FNN-NL in terms of RMSE and FNN-

NSL methods in terms of NMSE. The method had similar performance as in ACI Worldwide Inc. dataset.

V. Discussion

The results obtained for the proposed method are competitive when compared to works from literature involving five different data sets. The application of different decomposition method at different stages of network helps in the prediction.

The proposed hybrid model has given better performances in some of the datasets used when compared to CICC and FNN-NSL. The results for CHM method on Sunspot and financial datasets are better in terms of RMSE. In some cases, CHM gave better performance than standalone methods based on cooperative coevolution (CCRNN-Synapse Level and CCRNN-Neuron Level). The proposed method has also given better performance in comparison to adaptive modularity cooperative coevolution (AMCC), where the problem decomposition method varied with given time.

One of the advantages of the proposed competitive hybrid method (CHM) is the use of two different hybrid problem decomposition. The combination of two hybrid problem decomposition (NSL and NNL) in CHM, allows NSL to be used for decision making and NNL for diversity in the search. Therefore, CHM performs better than other methods in some of the cases. The cases where the method was unable to perform is due to either over training or over fitting.

VI. Conclusions

This paper has applied a new hybrid model called the Combination Hybrid Model (CHM) which was formed by using Neuron-Synapse level problem decomposition method (NSL) and Neuron-Network level problem decomposition method (NNL). CHM was used with Elman recurrent neural network for time series prediction.

The research began with the testing of the proposed hybrid method with benchmark datasets and later on the financial datasets. The application of different decomposition method at different stages of network helped in decision making and assisted in diversity of search.

The results obtained for the proposed method were competitive when compared to works from literature involving the five different datasets. In general, CHM has shown better optimization performance in time and success rate than other methods on financial datasets. The chaotic nature of the dataset was best suited with the proposed method.

In future work, the proposed method can be applied to feedforward network, pattern classification problems and global optimization problems as well as on different datasets. Some applications of the method could be in economics and education as well as to tackle climate change issues.

References

[1] I. Fister Jr, X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," *arXiv preprint arXiv:1307.4186*, 2013.

[2] 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.

[3] R. Nand and R. Chandra, "Neuron-synapse level problem decomposition method for cooperative neuro-evolution of feedforward networks for time series prediction," in *Neural Information Processing*. Springer, 2015, pp. 90–100.

[4] N. García-Pedrajas, E. Sanz-Tapia, D. Ortiz-Boyer, and C. Hervás-Martínez, "Introducing multi-objective optimization in cooperative coevolution of neural networks," in *Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence*. Springer, 2001, pp. 645–652.

[5] R. Nand, "Neuron-synapse level problem decomposition method for cooperative coevolution of recurrent networks for time series prediction," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, July 2016, pp. 3102–3109.

[6] R. Chandra and M. Zhang, "Cooperative coevolution of elman recurrent neural networks for chaotic time series prediction," *Neurocomputing*, vol. 86, pp. 116–123, 2012.

[7] R. Chandra, "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.

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

[9] R. Chandra, "Problem decomposition and adaptation in cooperative neuro-evolution," 2012.

[10] K. K. Bali, R. Chandra, and M. N. Omidvar, "Competitive island-based cooperative coevolution for efficient optimization of large-scale fully-separable continuous functions," in *International Conference on Neural Information Processing*. Springer, 2015, pp. 137–147.

[11] 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.

[12] R. Nand, E. Reddy, and M. Naseem, "Neuron-network level problem decomposition method for cooperative coevolution of recurrent networks for time series prediction," in *International Conference on Neural Information Processing*. Springer, 2016, pp. 38–48.

[13] 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.

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

[15] Smart bilo: An open source computational intelligence framework. 2015. [Online]. Available: http://smartbilo.aicrg.softwarefoundationfiji.org/

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

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

[18] S. S., "Solar cycle forecasting: A nonlinear dynamics approach," *Astronomy and Astrophysics*, vol. 377, pp. 312–320, 2001.

[19] "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

[20] 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.

[21] R. Chandra, "Problem decomposition and adaptation in cooperative neuro-evolution," 2012.

[22] 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.

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