Coevolutionary Feature Selection and Reconstruction in Neuro-Evolution 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. Feature reconstruction of time series problems produces reconstructed state-space vectors that are used for training machine learning methods such as neural networks. Recently, much consideration has been given to employing competitive methods in improving cooperative neuro-evolution of neural networks for time series predictions. This paper presents a competitive feature selection and reconstruction method that enforces competition in cooperative neuro-evolution using two different reconstructed feature vectors generated from single time series. Competition and collaboration of the two datasets are done using two different islands that exploit their strengths while eradicating their weaknesses. The proposed approach has improved results for some of the benchmark datasets when compared to standalone methods from the literature.

Keywords: Cooperative coevolution, feedforward networks, problem decomposition, time series

1 Introduction

Cooperative Coevolution (CC) provides an architecture for evolutionary algorithms that breaks down a problem into subcomponents that are implemented as sub-populations [1]. The application of CC for training neural networks is also referred as *cooperative neuro-evolution* [2]. In cooperative neuro-evolution, problem decomposition is defined by the structural properties of the network that contains interdependencies and dependent on the architecture and the type of training problem [2].

Chaotic time series problems are highly sensitive to noise and initial conditions [3]. Neural networks have been successfully used to tackle chaotic time series problems [4,5]. Time series prediction can be improved by exploring different features of the time series data and by selecting optimal values of the associated variables that are used for pre-processing [6].

Takens theorem [7] is one of the techniques for reconstructing the original time series into a phase space that is used for training neural networks [4]. The time lag defines the interval at which the data points are to be picked and the

embedding dimension specifies the size of the sliding window. These parameters are essential for building robust prediction systems that have been the focus of recent work where a quantum-inspired hybrid method was used for financial time series [6]. A multi-objective cooperative coevolution method was also introduced using time-lag as a parameter for reconstruction of the original data into different state space vector dataset as different objectives for financial prediction [8]. A similar approach was used in [9].

Competitive island cooperative coevolution algorithm (CICC) was introduced for training recurrent neural networks for time series prediction [10]. The method used different problem decomposition methods as islands and ensured that their features are used during evolution. It was later applied for global optimization problems [11].

Previous work focused on employing competitive methods that feature problem decomposition methods in neural networks. There has not been much work done that exploited the different parameters used for reconstructed state space vectors the original time series.

This paper presents a cooperative neuro-evolution method that enforces competition and collaboration using two different reconstructed feature vectors generated from a single time series. The method is called *co-evolutionary feature selection and reconstruction* which employs feedforward neural networks for time series prediction. Taken's theorem for state-space feature reconstruction.

The remainder of this paper is organized as follows. In Section 2, the proposed method is introduced. In Section 3, experiments, results and discussion are highlighted. Section 4 concludes the paper with plans for future work.

2 Co-evolutionary Feature Selection and Reconstruction

This section provides details of co-evolutionary feature selection and reconstruction (CSFR) for training feedforward network for time series prediction.

CSFR follows the same principle as the competitive island cooperative coevolution for problem decomposition methods where the exchange of best individuals takes place between the islands after competition [10].

In the proposed method, an island is defined by different reconstructed state space vectors generated from a single time series along with sub-populations that evolve using cooperative coevolution.

The proposed method has two different islands that are created using neuron level decomposition as seen in Figure 1. Each island is evaluated using feedfoward networks with a unique reconstructed dataset as seen in Figure 2. The reconstructed dataset is generated using Taken's embedding theorem with the two conditions that are *time delay (T)* and *embedding dimension (D)* [7]. The embedding dimension is used to determine the number of input neurons in feedforward network. The embedding dimension is fixed while time delay is varied as shown in Figure 3.

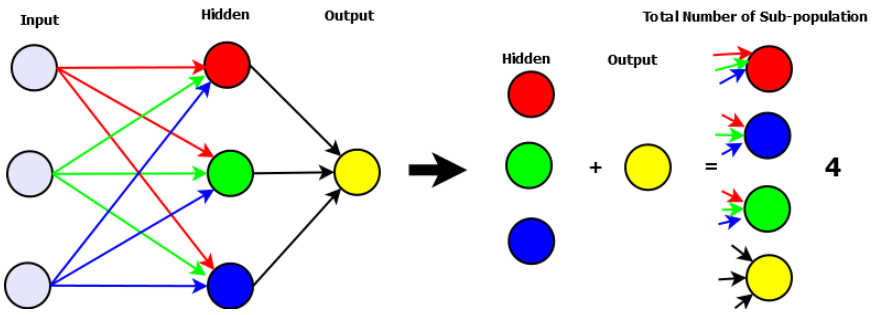


Fig. 1: Neuron level decomposition showing number of sub-components.

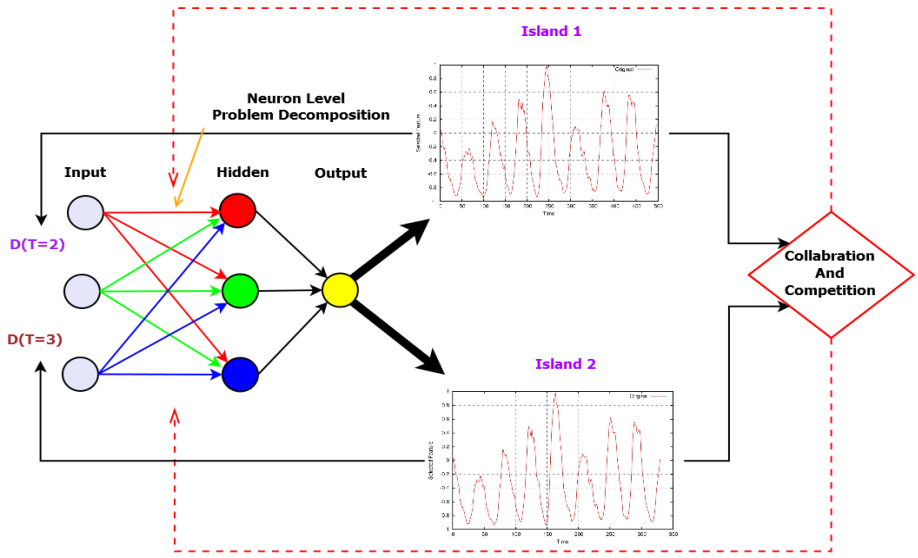


Fig. 2: The islands with different reconstructed datasets compete and collaborate with each other for Sunspot time series. Note that the same problem decomposition is used in both islands, hence the transfer of best solution is done without complications.

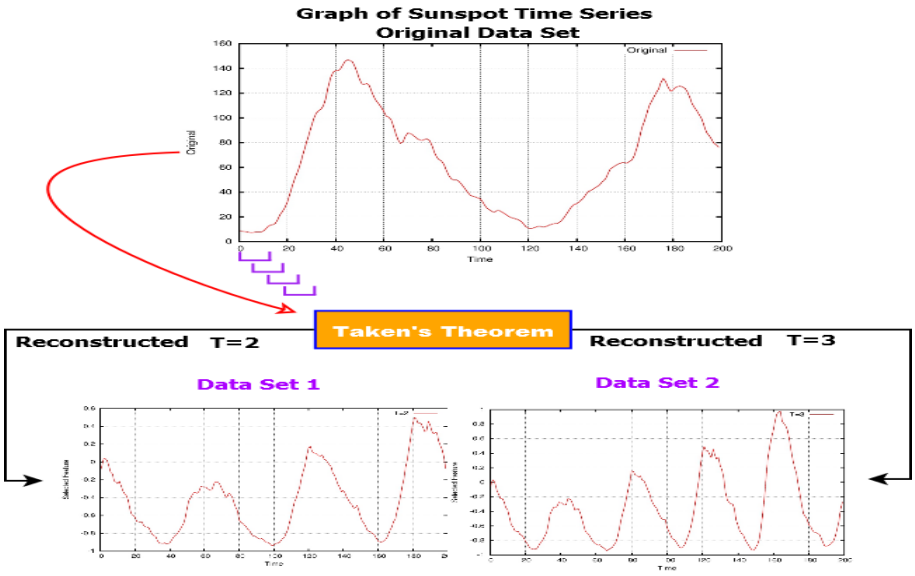


Fig. 3: Reconstruction of Sunspot data set using Taken's Theorem.

Details of the CSFR using feedforward network is given in Algorithm 1. Neuron level problem decomposition is used where the network is broken down into subcomponents that are based on hidden and output neurons [2].

In Step 1, the sub-populations of both islands are initialized with random real values in a suitable range. In Step 2, the evolution of both the islands takes place where each network is evolved for a predefined time based on the number of fitness evaluations (FE) which is called *local island evolution time*. In Step 3, the competition takes place where the algorithm checks if the best solution of the particular island is better than the other island. In Step 4, the solution that is marked as best is copied to the other island which helps the other island evolve. The best solution from both the islands is used to test for generalization.

3 Experiments and Results

This section presents the experiments and results for co-evolutionary feature selection and reconstruction (CSFR) using feedforward network for time series prediction.

3.1 Experimental Setup

The proposed method is evaluated with four different chaotic time series data sets. They include the Mackey-Glass [12] and Lorenz time series [13] that are the two simulated time series. Sunspot [14] and ACI Worldwide Inc. [15] are the two real-world problems.

The data sets used is based on the same configuration as used in past work [4]. The Mackey-Glass and ACI Worldwide Inc. time series are scaled in the range of [0,1], whereas the Sunspot and Lorenz are scaled in the range of [-1,1]. Mackey-Glass and ACI Worldwide Inc., employs feedforward network with sigmoid units in the hidden and the output layer. Lorenz and Sunspot use the hyperbolic tangent unit in the output layer.

The neuron level (NL) decomposition method was used in each of the islands [2]. Standalone cooperative coevolution methods are used for comparison of the results with different time delays. The performance and results of the method were evaluated by using three different numbers of hidden neurons (3, 5 and 7), and compared with standalone methods. The maximum evolution time used is 50 000 for standalone methods. In the proposed method, both islands have 50 000 function evaluations, each similar to the approach used in [10].

The generalized generation gap algorithm with parent-centric crossover (G3-PCX) evolutionary algorithm is used to evolve the sub-populations [16]. *Depth of search* for each sub-population is 1 with pool size of 2 parents and 2 offspring [4]. We have used the population size of 300 from the literature [10]. The root

Algorithm 1: CSFR for Feedforward Networks

```
Step 1: Create Island-One and Island-Two with Sub-populations based on
neuron level problem decomposition:
i. Cooperatively Evaluate Island-One FNN using Reconstructed Dataset-One
ii. Cooperatively Evaluate Island-Two FNN using Reconstructed Dataset-Two
Step 2: Evolution:
while Total-FE ≤ Max-FE do
    while Local-FE ≤ Island-Evolution-Time do
        foreach Sub-population at Island-One do
            foreach Depth of n Generations do
                Create new individuals using genetic operators
                Cooperative Evaluation
            end
        end
    end
    while Local-FE ≤ Island-Evolution-Time do
        foreach Sub-population at Island-Two do
            foreach Depth of n Generations do
                Create new individuals using genetic operators
                Cooperative Evaluation
            end
        end
    end
    Step 3: Competition: Compare the best solutions from both islands
    Step 4: Collaboration: Exchange the best fitness individuals from the
    winning island into the other island. Evaluate the other island.
end
```

mean squared error (RMSE) and normalized mean squared error (NMSE) are used to measure the performance of the network as given in Equations 1 and 2.

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

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

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 length of the observed data. These two performance measures are used in order to compare the results with the literature.

3.2 Results

Tables 1 - 4 report the results for different number of hidden neurons using co-evolutionary feature selection and reconstruction feedforward neural network (CSFR-FNN) with time lags (T=2, T=3). The two different time lags are also used for standalone methods which are cooperative coevolutionary feedforward neural networks (CCFNN) that use different time delays (T=2 and T=3), but with same dimension (D).

The results report RMSE and 95 percent confidence interval from different numbers of hidden neurons, where each case executed 50 independent experimental runs. Note that the best results are those with the least value of RMSE for each case.

In Table 1, in the Mackey-Glass problem, it was observed that CSFR was able to beat both standalone methods (T=2, T=3), and the best result was given by 5 hidden neurons. The overall performance in terms of generalization increased as the number of hidden neurons increased.

In Table 2, Lorenz problem shows that the CSFR has been able to outperform both the standalone methods. The best result was seen in the case of 3 hidden neurons for CSFR and standalone methods.

In Table 3, the Sunspot problem shows that CSFR method has not been able to outperform the one of the standalone methods (T=3). Five hidden neurons have given good results for CSFR methods.

In Table 4, the ACI Worldwide Inc. problem shows that the CSFR method gives competitive results when compared to the standalone methods (T=2, T=3). The five hidden neurons have given best result of CSFR method. It has also been observed that the generalization performance of the CSFR and the other two methods does not deteriorate as the number of the hidden neuron increases as it does for other problems.

Figures 4 - 5 show typical prediction given by the proposed method. It shows that CSFR has been able to give a good prediction performance. CSFR has been able to cope with the noise in the Sunspot time series given in Figure 4 and ACI time series given in Figure 5.

Table 1: The prediction training and generalisation performance (RMSE) of standalone and CSFR on the Mackey-Glass time series

Prob.	H	Training	Generalisation	Best
CCFNN(T=2)	3	0.0107 ± 0.00131	0.0107 ± 0.00131	0.0050
	5	0.0089 ± 0.00097	0.0088 ± 0.00097	0.0038
	7	0.0078 ± 0.00079	0.0078 ± 0.00079	0.0040
CCFNN(T=3)	3	0.0112 ± 0.00149	0.0112 ± 0.00149	0.0039
	5	0.0081 ± 0.00063	0.0080 ± 0.00063	0.0041
	7	0.0080 ± 0.00070	0.0078 ± 0.00070	0.0047
CSFR-FNN	3	0.0090 ± 0.00109	0.00090± 0.001103	0.0041
(T=2,T=3)	5	0.0065 ± 0.00068	0.0065 ± 0.00069	0.0029
	7	0.0072 ± 0.00086	0.0072 ± 0.00086	0.0041

Table 2: The prediction training and generalisation performance (RMSE) of Standalone and CSFR on the Lorenz time series

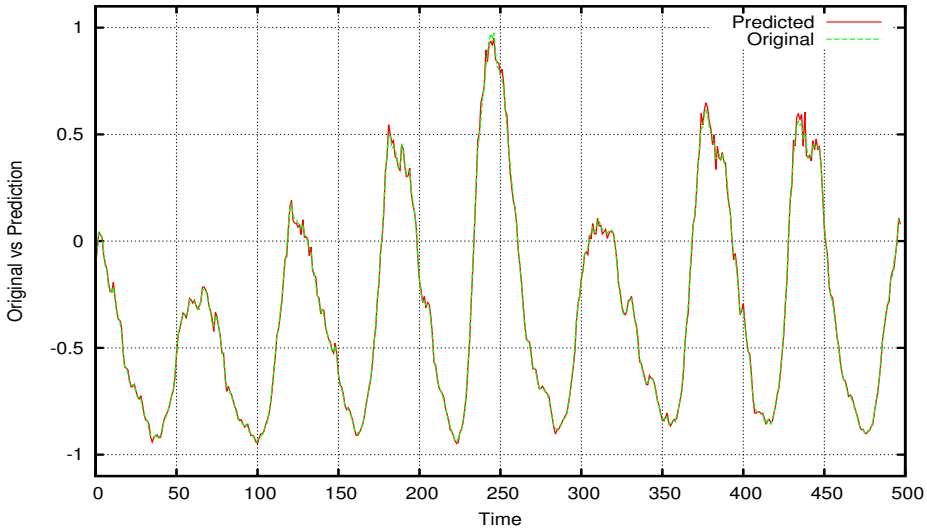
Prob.	H	Training	Generalisation	Best
CCFNN(T=2)	3	0.0170 ± 0.0031	0.0176 ± 0.0031	0.0043
	5	0.0249 ± 0.0062	0.0271 ± 0.0067	0.0021
	7	0.0379 ± 0.0093	0.0416 ± 0.0092	0.0024
CCFNN(T=3)	3	0.0165 ± 0.0028	0.0167 ± 0.0028	0.0030
	5	0.0278 ± 0.00830	0.0292± 0.00829	0.0022
	7	0.0419 ± 0.00982	0.0425 ± 0.0104	0.0031
CSFR-FNN	3	0.0159 ± 0.0037	0.0163 ± 0.0040	0.0027
(T=2,T=3)	5	0.0149 ± 0.0033	0.0162 ± 0.0039	0.0023
	7	0.0293 ± 0.0079	0.0321 ± 0.0083	0.0035

Table 3: The prediction training and generalisation performance (RMSE) of standalone and CSFR on the Sunspot time series

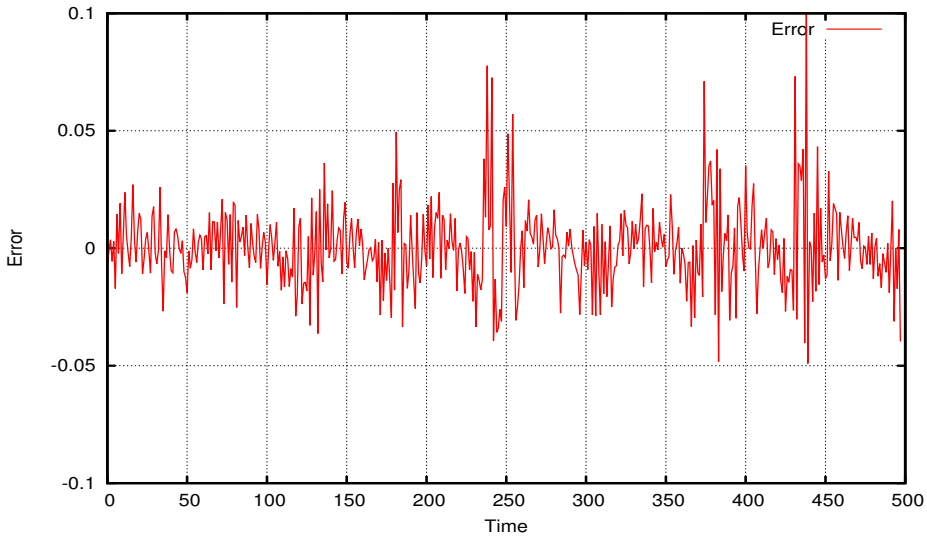
Prob.	H	Training	Generalisation	Best
CCFNN(T=2)	3	0.0207 ± 0.0035	0.0538 ± 0.0091	0.015
	5	0.0289 ± 0.0039	0.0645 ± 0.0093	0.017
	7	0.0353 ± 0.0048	0.0676 ± 0.0086	0.021
CCFNN(T=3)	3	0.0189 ± 0.0145	0.0538 ± 0.0108	0.016
	5	0.0291 ± 0.0143	0.0690 ± 0.0091	0.017
	7	0.0302 ± 0.0174	0.0849 ± 0.00859	0.015
CSFR-FNN	3	0.0211 ± 0.00034	0.0180 ± 0.00072	0.015
(T=2,T=3)	5	0.0205 ± 0.00044	0.0187 ± 0.0036	0.014
	7	0.0209± 0.00035	0.0181 ± 0.00077	0.015

Table 4: The prediction training and generalisation performance (RMSE) of standalone and CSFR on the ACI Worldwide Inc. time series

Prob.	H	Training	Generalisation	Best
CCFNN(T=2)	3	0.0246 ± 0.00348	0.0247 ± 0.00348	0.015
	5	0.0231 ± 0.00588	0.0284 ± 0.00570	0.016
	7	0.0204 ± 0.00159	0.0194 ± 0.00157	0.015
CCFNN(T=3)	3	0.0204 ± 0.0014	0.0170 ± 0.00110	0.014
	5	0.0202 ± 0.00116	0.0164 ± 0.00046	0.014
	7	0.0202 ± 0.00383	0.0199 ± 0.00383	0.014
CSFR-FNN	3	0.0206 ± 0.00054	0.0187 ± 0.00131	0.015
(T=2,T=3)	5	0.0196 ± 0.00020	0.0166 ± 0.00058	0.013
	7	0.0194 ± 0.00023	0.0183 ± 0.00274	0.014



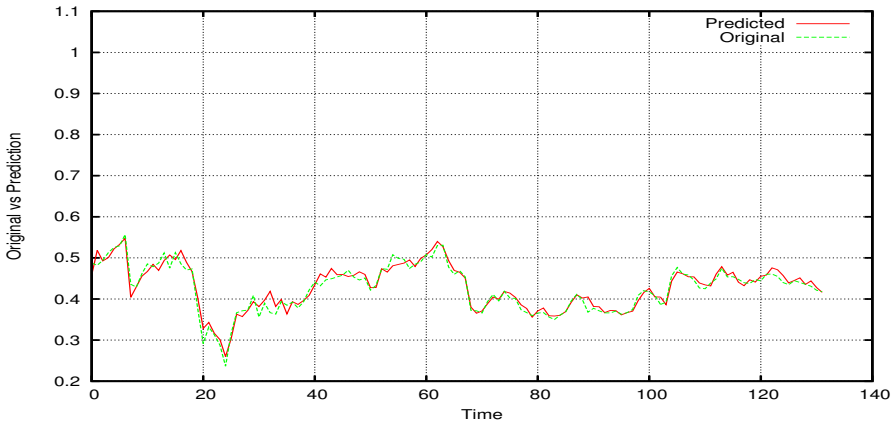
(a) Performance given by CSFR on the testing set for Sunspot.



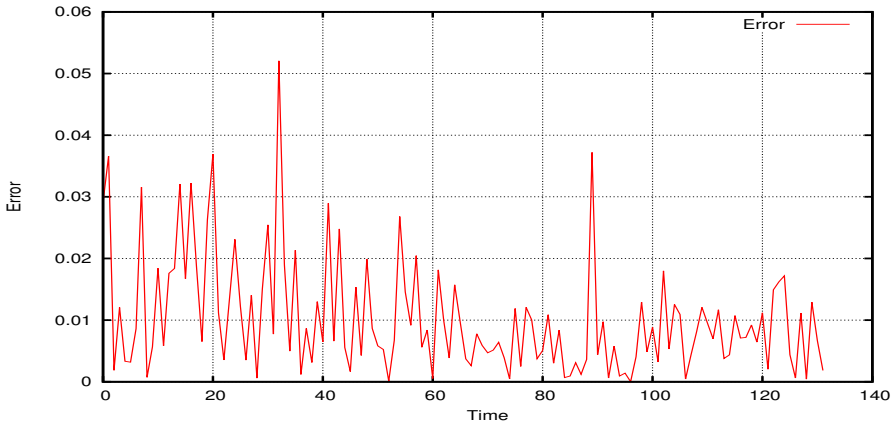
(b) Error on the test data set given by CSFR for the Sunspot time series.

Fig. 4: Typical prediction given by CSFR for Sunspot time series.

The Table 5 compares the best results of CSFR given in Tables 1 - 4 with some of the closely related methods from the literature. The best results are used for the comparison. CSFR has given better performance when compared to



(a) Performance given by CSFR on the testing set for ACI Worldwide.



(b) Error on the test data set given by CSFR for the ACI Worldwide time series.

Fig. 5: Typical prediction given by CSFR for ACI Worldwide time series.

the majority of the methods in the literature. However, there are specific cases that need further improvement.

In Table 5, for Mackey-Glass, the proposed method outperformed all the methods except for locally linear neuro-fuzzy model (LLNF) and radial basis network (RBF). Due to competition and collaboration, CSFR has outperformed them.

In Table 5, for problem Lorenz, it shows the best result on Lorenz time series problem that is compared with some of the related methods from the literature. CFSR outperformed all the methods for Lorenz, except for multi-

Table 5: A comparison with the results from literature for all data sets

Problem	Prediction Method	RMSE	NMSE
Mackey	Radial basis network (RBF-OLS)(2006) [5]	1.02E-03	
	Locally linear neuro-fuzzy model (LLNF-LoLiMot) (2006) [5]	9.61E-04	
	Neuro-fuzzy system with time delay coordinates (2008) [17]		1.26E-03
	Neural fuzzy network (PS0) (2009) [18]	2.10E-02	
	Neural fuzzy network (CPS0) (2009) [18]	1.76E-02	
	Neural fuzzy network and DE (2009) [18]	1.62E-02	
	Neural fuzzy network and GA (2009)[18]	1.63E-02	
	Synapse Level-CCRNN (SL-CCRNN) (2012) [4]	6.33E-03	2.79E-04
	Neuron Level-CCRNN (NL-CCRNN) (2012) [4]	8.28E-03	4.77E-04
	Competitive Island Cooperative Coevolution (CICC-RNN) (2014) [10]	3.99E-03	1.11E-04
	MOCCFNN with 2-objectives (T=2)(MO-CCFNN-T=2) (2014)[8]	3.84E-03	2.80E-05
	MOCCFNN with 2-objectives (T=3)(MO-CCFNN-T=3) (2014) [8]	3.77E-03	2.70E-05
	Proposed CCFNN-CSFR	2.90E-03	1.60E-06
Lorenz	Radial basis network (RBF-OLS)(2006) [5]		1.41E-09
	Locally linear neuro-fuzzy model (LLNF-LoLiMot) (2006) [5]		9.80E-10
	Auto regressive moving average (ARMA-ANN)(2008) [19]	8.76E-02	
	Backpropagation neural network and GA (2011) [20]	2.96E-02	
	Synapse Level-CCRNN (SL-CCRNN) (2012) [4]	6.36E-03	7.72E-04
	Neuron Level-CCRNN (NL-CCRNN) (2012) [4]	8.20E-03	1.28E-03
	Competitive Island Cooperative Coevolution (CICC-RNN) (2014) [10]	3.55E-03	2.41E-04
	MOCCFNN with 2-objectives (T=2)(MO-CCFNN-T=2) (2014)[8]	2.19E-03	2.53E-05
	MOCCFNN with 2-objectives (T=3)(MO-CCFNN-T=3) (2014)[8]	2.18E-03	2.54E-05
	Proposed CCFNN-CSFR	2.32E-03	2.85E-05
Sunspot	Radial basis network (RBF-OLS)(2006) [5]		4.60E-02
	Locally linear neuro-fuzzy model (LLNF-LoLiMot) (2006) [5]		3.20E-02
	Synapse Level-CCRNN (SL-CCRNN) (2012) [4]	1.66E-02	1.47E-03
	Neuron Level-CCRNN (NL-CCRNN) (2012) [4]	2.60E-02	3.62E-03
	Competitive Island Cooperative Coevolution (CICC-RNN) (2014) [10]	1.57E-02	1.31E-03
	MOCCFNN with 2-objectives (T=2)(MO-CCFNN-T=2) (2014)[8]	1.84E-02	1.02E-03
	MOCCFNN with 2-objectives (T=3)(MO-CCFNN-T=3) (2014)[8]	1.81E-02	9.98E-04
	Proposed CCFNN-CSFR	1.58E-02	7.56E-04
ACI	Competitive Island Cooperative Coevolution (CICC-RNN) (2014) [10]	1.92E-02	
	MOCCFNN with 2-objectives (T=2)(MO-CCFNN-T=2) (2014)[8]	1.94E-02	
	MOCCFNN with 2-objectives (T=3)(MO-CCFNN-T=3) (2014)[8]	1.47E-02	
	Proposed CCFNN-CSFR	1.34E-02	9.95E-04

objective cooperative coevolution (MO-CCFNN), radial basis network (RBF) and locally linear neuro-fuzzy model (LLNF).

In Sunspot problem, the performance of the proposed method on the Sunspot time series problem is compared to the methods in the literature. CFSR outperformed all the methods except for CICC-RNN.

In ACI Worldwide Inc. problem, CFSR was able to outperform all the methods in the literature. This shows that the proposed method can handle the noise and the regions which are very chaotic in dataset as it is real world application problem.

3.3 Discussion

The main strength of the proposed method allows to explore and learn from the regions within the data set which are missed given that it is difficult to find the optimal value for the time delay parameter. CFSR was able to perform better due to information sharing during evolution via the neural network weights from two diverse features extracted datasets implemented as islands.

The proposed method gave exceptional results for generalization performance when compared to standalone methods for the Sunspot problem. This shows that the proposed method can perform very well in real world applications that contain noise. This is also the best results when compared to other methods in the literature as shown in Table 5.

One of the major advantages of the proposed method is that it can be implemented in a multi-threaded environment that will speed up the computation time. Neuro-evolution methods have limitations in terms of time when compared to gradient based methods. In a multi-threaded implementation, each island can run on a separate thread and speed up the evolutionary process. Note that when only one neural network is used to evaluate both islands, there can be problems in multi-threaded environment. Appropriate mutex locks as used in multi-thread programming needs to be implemented. One solution is to use two different neural networks that mirror each other in terms of topology one for each island.

4 Conclusion

We proposed a co-evolutionary feature selection and reconstruction method that used different reconstructed features of the separate data sets generated from single time series. It has shown good results on all the different time series problems and has outperformed majority of the methods in the literature.

Future work can employ different problem decomposition methods in the islands and be extended to three of more islands. The proposed framework can employ other neural network architectures such as recurrent neural networks where both the dimension and time lag can be varied to generate different data sets that provide competition. Different methods of feature extraction for time series can be used to enforce the competition. The analysis about the strength of the different islands at different stages of evolution can also be done, i.e. to check which island wins the competition in different time series problems. The proposed approach can also be extended for pattern classification problems where feature selection has been extensively studied.

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. Frean, and M. Zhang, "On the issue of separability for problem decomposition in cooperative neuro-evolution," *Neurocomputing*, vol. 87, pp. 33–40, 2012.
3. H. K. Stephen, *In the Wake of Chaos: Unpredictable Order in Dynamical Systems*. University of Chicago Press, 1993.
4. R. Chandra and M. Zhang, "Cooperative coevolution of Elman recurrent neural networks for chaotic time series prediction," *Neurocomputing*, vol. 186, pp. 116 – 123, 2012.

5. 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.
6. R. de A Araujo, A. de Oliveira, and S. Soares, "A quantum-inspired hybrid methodology for financial time series prediction," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, Barcelona, Spain, Jul. 2010, pp. 1–8.
7. F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980*, ser. Lecture Notes in Mathematics, 1981, pp. 366–381.
8. 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.
9. C. Smith and Y. Jin, "Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction," *Neurocomputing*, vol. 143, pp. 302 – 311, 2014.
10. 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.
11. R. Chandra and K. Bali, "Competitive two island cooperative coevolution for real parameter global optimization," in *IEEE Congress on Evolutionary Computation*, Sendai, Japan, May 2015, p. In Press.
12. M. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.
13. E. Lorenz, *The Essence of Chaos.* University of Washington Press, 1993.
14. 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/
15. "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
16. 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.
17. J. Zhang, H. Shu-Hung Chung, and W.-L. Lo, "Chaotic time series prediction using a neuro-fuzzy system with time-delay coordinates," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 20, no. 7, pp. 956 –964, july 2008.
18. C.-J. Lin, C.-H. Chen, and C.-T. Lin, "A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39, no. 1, pp. 55–68, Jan. 2009.
19. I. Rojas, O. Valenzuela, F. Rojas, A. Guillen, L. Herrera, H. Pomares, L. Marquez, and M. Pasadas, "Soft-computing techniques and arma model for time series prediction," *Neurocomputing*, vol. 71, no. 4-6, pp. 519 – 537, 2008.
20. M. Ardalani-Farsa and S. Zolfaghari, "Residual analysis and combination of embedding theorem and artificial intelligence in chaotic time series forecasting," *Appl. Artif. Intell.*, vol. 25, pp. 45–73, 2011.