Multi-Objective Cooperative Coevolution of Neural Networks for Time Series Prediction

Shelvin Chand and Rohitash Chandra

Abstract— **The use of neural networks for time series prediction has been an important focus of recent research. Multi-objective optimization techniques have been used for training neural networks for time series prediction. Cooperative coevolution is an evolutionary computation method that decomposes the problem into subcomponents and has shown promising results for training neural networks. This paper presents a multi-objective cooperative coevolutionary method for training neural networks where the training data set is processed to obtain the different objectives for multi-objective evolutionary training of the neural network. We use different time lags as multi-objective criterion. The trained multi-objective neural network can give prediction of the original time series for preprocessed data sets distinguished by their time lags. The proposed method is able to outperform the conventional cooperative coevolutionary methods for training neural networks and also other methods from the literature on benchmark problems.**

I. Introduction

A multi-objective optimization problem is one in which there are multiple objectives, each of which have different optimal solutions [1]. These objectives are usually in conflict with one another, if there are sufficient differences between the optimal solutions for each of the objective function [1]. Time-series prediction involves the use of past and present time series data to make future predictions [2], [3]. Time series prediction has been applied in many different areas that range from weather [4] to financial prediction [5].

In the past, time series prediction research has explored the importance for the time lag where a simple, deterministic method was proposed for the selection of optimal time lags for non-uniform embedding [6]. The proposed method was able to handle optimization problems in a multi-parameter space of arguments while improving time series prediction. Quantum-inspired hybrid methods have also been explored in order to determine the best possible time-lag to represent the original time series, with good results on financial prediction [7]. A hybrid model that combined neural networks with a modified genetic algorithm was proposed to perform an evolutionary search for the minimum necessary time-lags for determining the phase space that generates the time series [8]. A morphological rank linear time-lag added evolutionary forecasting method was also proposed that carries out an evolutionary search for the lowest number of relevant time lags necessary to efficiently represent the patterns and characteristics of a complex time series [9]. A meta-evolutionary algorithm simultaneously evolved both the neural networks and the set of lags needed to predict the time series [10]. The approach showed good results on a number of time series problems in which it was able to reconstruct the data set very efficiently and accurately.

Multi-objective time series prediction using computational intelligence methods have been used to improve the prediction accuracy in the past [11]. Multi-objective evolutionary algorithms have been used to optimize radial-basis networks for time series prediction which incorporated heuristics that were able to detect and remove networks which did not contribute much to the net output while preserving those that produced good results [12]. The use of multi-objective evolutionary neural networks for time series prediction employed training and validation accuracy as the two different objectives [11]. Multiple error measures have also been used as the different objectives in training evolutionary neural networks with multi-objective optimization [13].

Cooperative Coevolution is a biologically inspired evolutionary computation method that divides a large problem into smaller subcomponents [14]. Cooperative coevolution has been applied to several areas which include training [15] neural networks in solving a wide range of problems that include pattern classification and control problems [16]. Multi-objective cooperative coevolution has also been explored recently. Iorio and Li developed Non-dominated Sorting Cooperative Coevolutionary Genetic Algorithm (NSCCGA) which was able to outperform NSGA-II [17] on some of the important bench mark functions. The authors of [18] explored multi-objective cooperative coevolution using a special niching mechanism and an extending operator to maintain diversity. Their method showed good results in finding more distributed non-dominated solutions. Multi-objective cooperative coevolution has also been used for large scale optimization [19].

A way to improve time series prediction is to explore the different features of the time series data and to choose optimal values for the associated variables that are used for pre-processing. Taken's theorem is used for reconstructing the original time series into a phase space that is used by prediction models for training [20]. The *time lag* defines the interval at which the data points are picked and the embedding dimension specifies the size of the sliding window that is used to capture points to make a reconstructed phase space [2]. The appropriate time lag and embedding dimension is important as the data set has to be reconstructed in such a way that it retains the main features of the time series while minimizing the level of noise from the original data set. The time lags are uniform across the entire data set.

Shelvin Chand and Rohitash Chandra are with the School of Computing, Information and Mathematical Sciences, University of the South Pacific, Laucala Campus, Suva, Fiji. (email: {chand.shelvin}@gmail.com), {c.rohitash}@gmail.com).

This paper employs multi-objective cooperative coevolution to train a feed-forward network for a time series problem using Neuron Level problem decomposition method. We test the proposed method with three benchmark time series data sets which include the Lorenz, Mackey Glass and Sunspot time series problems. We also apply the proposed method to the ACI Worldwide financial time series. The aim is to see if multi-objective optimization using data sets with different *time lag* can help improve time series prediction. It will also be important to test whether having multiple optimal solutions has any significant contribution towards improving generalization performance. Moreover, it would be interesting to see whether cooperative coevolution and the selected decomposition method are able adapt well to a multi-objective environment when dealing with neural network training.

The rest of the paper is as follows. Details on the proposed method is discussed in Section 2. Section 3 presents the results and analysis of the experiment. Section 4 concludes the paper with a discussion on future research.

II. Multi-Objective Cooperative Coeveolutionary Neural Network for Time Series Prediction

Alg. 1 Multi-Objective Cooperative Neuro-Evolution

Step 1: Decompose the problem into k subcomponents using Neuron level decomposition
Step 2: Initialize and cooperatively evaluate each sub-population for each objective
Step 3: Rank and Identify Non-Dominated Individuals
for each *cycle* until termination **do**
  for each Sub-population **do**
    for n Generations **do**
      i) Select and create new offspring using Parent-Centric Crossover
      ii) Cooperatively evaluate the new offspring for each objective
      iii) Update sub-population with the best individuals
      iv) Rank the sub-population and identify non-dominated individuals
      v) Assign Pareto Front
    end for
  end for
end for

A. The Different Set of Objectives

A multi-objective problem gives rise to a set of optimal solutions (known as pareto-optimal) [17]. The solutions within the pareto-optimal front are known as non-dominated solutions as no solution can claim to be better than any other with respect to all the objective functions [21]. They can also be called non-inferior, admissible or efficient solutions [22]. Any single objective component of a non-dominated solution within the pareto-optimal set can only be improved by degrading at least one of its other objective components [22].

Features of time series data can be used as different objectives in an attempt to improve the prediction accuracy. We use Taken's embedding theorem [20] to reconstruct the data before it is used for prediction. Taken's theorem expresses that the vector series reproduces many important characteristics of the original time series [20]. The theorem allows for chaotic time series data to be reconstructed into a d-dimensional vector with the two conditions of *time lag* and *embedding dimension* [20]. The value of embedding dimension and time lag must be carefully chosen in order for the vector to be able to reproduce important characteristics of the original data set [23].

We used time lag values as the different objectives in this paper. Different time lag values reproduce the original data set in different ways which in effect means dealing with two different data sets. Training with different time lag values can also allow the neural network to generalize better and explore different aspects and patterns within the time series. Noise within the data set is also a major issue and the time lag determines how much of noise to be present.

B. Multi-Objective Cooperative Coevolution for Feed-Forward Networks

Cooperative coevolution divides a large problem into subcomponents. Each sub-component is represented by different sub-populations. One of the main reasons for using cooperative coevolution is because it promises more diverse solutions in comparison to other single-population based evolutionary algorithms [15]. Problem decomposition is a major issue in cooperative coevolution as it is important to group interacting variables together to minimize interaction amongst subcomponents [24]. The two major problem decomposition methods are those on the synapse level [16] and neuron level [24]. Different problem decomposition methods have shown strengths and weaknesses in different types of problems and neural network architectures.

We propose a multi-objective cooperative coevolutionary method for training neural networks. With multi-objective cooperative coevolution, each neural network is trained on two objectives. Solutions which are able to perform well on both objectives are carried forward and preserved. In conventional cooperative coevolution, the best individuals are used to represent each sub-population while in our method, the non-dominated individuals are used as representatives for each sub-population. Individuals within the sub-populations are also ranked according to their performance on the multiple objectives.

In Algorithm 1, the feed forward network is decomposed using the Neuron level problem decomposition into k subcomponents. In Neuron level decomposition, k is equal to the total number of all the hidden and output neurons [25]. The subcomponents are implemented as sub-populations. Evaluation of the fitness of each individual for a particular sub-population is done cooperatively with non-dominated individuals from the other sub-populations [14].

Cooperative evaluation for an individual j, in a particular sub-population i, is done by concatenating it with random non-dominated individuals from the rest of the sub-populations. The individual is then evaluated for each given objective which are represented by the different time lags

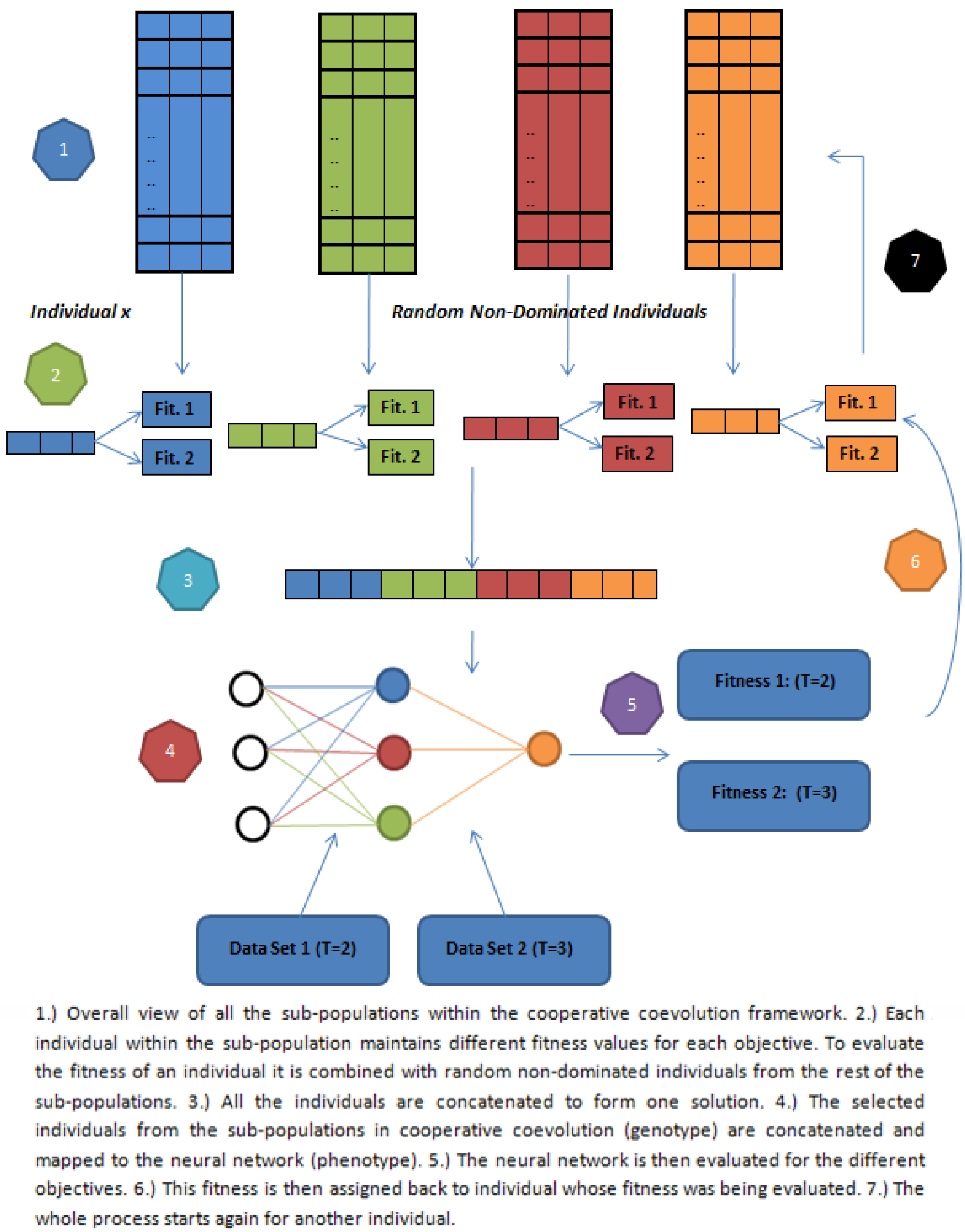

Fig. 1. Multi-objective framework for training neural networks using cooperative coevolution

as shown in Fig. 1. Once all individuals within a sub-population have been evaluated, the entire sub-population is ranked and the non-dominated individuals are identified (similar to NSGA-II [17]). During the first iteration when the non-dominated individuals are not known, an individual is evaluated by selecting random individuals from the other sub-populations. A *cycle* is completed when all the sub-populations are evolved. Once all sub-populations have been evolved, they are re-ranked so that the new individuals added to the population are taken into consideration. The fitness evaluation is further explained in Fig. 1.

C. Identifying Non-Dominated Solutions

We are employing cooperative coevolution to train neural networks where each individual within the sub-populations represents a portion of the overall neural network. When evaluating the fitness of an individual, it is concatenated with other individuals from the rest of the sub-populations to form a complete solution. The fitness for the overall network on the different objectives is then assigned back to the individual whose fitness is being evaluated, even though it is just a subset of the overall solution. This is known as fitness assignment [14]. In a minimization problem, an individual x is considered non-dominated in comparison to another individual y when no fitness value of y is less than x and at least one fitness value of y is greater than x [26].

After each iteration, the individuals within the sub-population are compared and ranked with respect to each others' fitness values in order to identify the non-dominated solutions. The rank of an individual is equal to the number of individuals within the population which dominate that particular individual. If an individual is non-dominated then the rank will be 0. Based on this rank, the individuals are assigned to the different set of pareto fronts. The non-dominated individuals belong to the first pareto front. The main idea is to maintain the individuals within the non-dominated front and reduce the number of dominated individuals within the sub-population. Subsequently, individuals with a rank of 1 are assigned to the second pareto front and so on until the sub-population is fully classified. The overall search is directed towards the non-dominated region as the best individuals are used for evolution.

D. Evolving the Sub-populations

All the sub-populations are evolved for a fixed number of generations. The generalized generation gap model with parent-centric crossover operator (G3-PCX) evolutionary algorithm [27] is the designated evolutionary algorithm in the sub-populations. G3-PCX has been used because it has shown good results in our previous research [28].

The parent-centric crossover operator is used to generate new offsprings. In order to generate a new offspring, n parents are chosen. One of them is randomly chosen from the non-dominated front and the others are randomly chosen from the entire population. The algorithm checks to ensure that we do not have two same parents at any particular point. When a new offspring is generated, it is compared with the parents in terms of fitness on all the given objectives. At this stage, the non-dominated solutions from the pool of parents and offsprings are identified which will eventually replace the original parents in the main sub-population.

The x best individuals are picked and replaced into the main population where x is equal to the number of parents which participated in the evolution process. While identifying multiple non-dominated solutions is essential, the main goal of multi-objective time series prediction is to improve generalization [11].

E. Selecting the Final Solution

The multi-objective algorithm produces multiple non-dominated solutions and therefore, the selection of the final solution is a major issue. It must also be considered that not all solutions that are pareto-optimal for the training set will be pareto optimal for the testing set [13]. Based on experimental trials, the final solution is a random combination of non-dominated individuals from the pareto front. A non-dominated individual at random is picked from the different sub-populations to form the final solution, similar to NSCCGA [29] where the representative for each sub-population is randomly chosen from the non-dominated front. We note that no solution in the non-dominated region is better with respect to another and for this reason a random selection from the non-dominated front was the best option.

III. Experiments and Results

This section presents experimental results on three benchmark time series data sets and a financial time series data set using the proposed method. The performance is evaluated by using different number of hidden neurons that reflect the scalability of the proposed evolutionary method. The results are compared with conventional cooperative coevolution (non-multi-objective) that employs T=2 and T=3. The experimental setup and the results with 95% confidence are given in the following sub-sections.

TABLE VI

Comparison of the best (minimum) results given in individual runs.

Dataset	SO (T=2)	MO (T=2)	SO (T=3)	MO (T=3)
Mackey Glass	3.78E-03	3.84E-03	4.04E-03	**3.77E-03**
Lorenz	2.21E-03	2.19E-03	3.38E-03	**2.18E-03**
Sunspot	1.69E-02	1.84E-02	**1.66E-02**	1.81E-02
ACI Worldwide	1.92E-02	1.94E-02	1.55E-02	**1.47E-02**

A. Experimental Setup

The results for the proposed multi-objective cooperative coevolutionary method is compared with the single-objective cooperative coevolutionary method in order to determine if the results are improved. Root mean squared error (RMSE) and normalized mean squared error (NMSE) are used to evaluate the performance. The *depth of search* is kept at 1 as this number has achieved good results in our past work [28]. This indicates that all the sub-populations in cooperative coevolution are evolved in a round-robin fashion for a single generation.

The maximum number of function evaluations was set at 50 000 which was used as a termination condition for both the single-objective and multi-objective algorithm. Population size for both methods is kept at 300 individuals which gave good results in our trial experiments. Neuron level problem decomposition method was used for dividing the problem into subcomponents.

The G3-PCX evolutionary algorithm is used to evolve all the sup-populations. The G3-PCX genetic algorithm uses the *generation gap model* [27] for selection. We use a pool size of 2 parents and 2 offsprings for the single-objective method and a pool size of 4 parents and 4 offsprings for the multi-objective method. The multi-objective method needs as many non-dominated solutions as possible and a bigger pool size will accommodate more non-dominated solutions.

Each individual within the multi-objective population maintains multiple fitness values for the different objectives. The two objectives for this experiment are the different *time lag (T)* values (T=2 & T=3). In the single-objective method, separate experiments are done for T=2 & T=3.

We used four time series problems to evaluate the performance. This included three benchmark data sets (Mackey

TABLE I

THE TRAINING AND GENERALIZATION PERFORMANCE OF THE PROPOSED APPROACH ON THE MACKEY GLASS TIME SERIES

Prob.	H	Training(T=2)	Training(T=3)	Generalization (T=2)	Generalization (T=3)
MO-CCFNN	3	9.67E-03 ± 1.08E-03	9.70E-03 ± 1.08E-03	9.66E-03 ± 1.08E-03	9.65E-03 ± 1.08E-03
(RMSE)	5	8.54E-03 ± 7.73E-04	8.59E-03 ± 7.73E-04	8.50E-03 ± 7.73E-04	8.47E-03 ± 7.72E-04
	7	**7.81E-03 ± 5.99E-04**	**7.86E-03 ± 6.03E-04**	**7.76E-03 ± 5.99E-04**	**7.73E-03 ± 6.02E-04**
	9	8.76E-03 ± 7.27E-04	8.80E-03 ± 7.30E-04	8.72E-03 ± 7.27E-04	8.68E-03 ± 7.29E-04
MO-CCFNN	3	2.05E-04 ± 5.78E-05	2.07E-04 ± 5.75E-05	2.05E-04 ± 5.81E-05	2.06E-04 ± 5.79E-05
(NMSE)	5	1.52E-04 ± 2.81E-05	1.54E-04 ± 2.78E-05	1.51E-04 ± 2.84E-05	1.51E-04 ± 2.84E-05
	7	**1.24E-04 ± 1.98E-05**	**1.26E-04 ± 1.98E-05**	**1.23E-04 ± 2.00E-05**	**1.22E-04 ± 2.01E-05**
	9	1.58E-04 ± 2.87E-05	1.59E-04 ± 2.84E-05	1.57E-04 ± 2.89E-05	1.56E-04 ± 2.89E-05

TABLE II

THE TRAINING AND GENERALIZATION PERFORMANCE OF THE PROPOSED APPROACH ON THE LORENZ TIME SERIES

Prob.	H	Training(T=2)	Training(T=3)	Generalization (T=2)	Generalization (T=3)
MO-CCFNN	3	**1.69E-02 ± 2.71E-03**	**1.70E-02 ± 2.69E-03**	**1.73E-02 ± 2.88E-03**	**1.73E-02 ± 2.88E-03**
(RMSE)	5	2.49E-03 ± 7.19E-03	2.54E-02 ± 7.08E-03	2.64E-02 ± 7.18E-03	2.61E-02 ± 7.09E-03
	7	4.30E-03 ± 9.96E-03	4.41E-02 ± 9.53E-03	4.61E-02 ± 9.93E-03	4.47E-02 ± 9.53E-03
	9	4.05E-02 ± 8.85E-03	4.14E-02 ± 8.58E-03	4.45E-02 ± 8.78E-03	4.29E-02 ± 8.57E-03
MO-CCFNN	3	**1.92E-03 ± 5.37E-04**	**1.92E-03 ± 5.31E-04**	**2.11E-03 ± 6.08E-04**	**2.12E-03 ± 6.10E-04**
(NMSE)	5	6.18E-03 ± 3.19E-03	6.45E-03 ± 3.36E-03	7.19E-03 ± 3.73E-03	7.08E-03 ± 3.62E-03
	7	1.55E-02 ± 5.29E-03	1.63E-02 ± 5.69E-03	1.79E-02 ± 5.99E-03	1.67E-02 ± 5.70E-03
	9	1.29E-02 ± 4.59E-03	1.34E-02 ± 4.59E-03	1.57E-02 ± 5.21E-03	1.48E-02 ± 5.13E-03

TABLE III

THE TRAINING AND GENERALIZATION PERFORMANCE OF THE PROPOSED APPROACH ON THE SUNSPOT TIME SERIES

Prob.	H	Training(T=2)	Training(T=3)	Generalization (T=2)	Generalization (T=3)
MO-CCFNN	3	**2.45E-02 ± 1.32E-02**	**2.42E-02 ± 1.31E-02**	5.95E-02 ± 9.01E-03	**5.90E-02 ± 8.92E-03**
(RMSE)	5	2.72E-02 ± 1.26E-02	2.69E-02 ± 1.29E-02	**5.93E-02 ± 9.02E-03**	5.95E-02 ± 9.22E-03
	7	2.90E-02 ± 1.52E-02	2.85E-02 ± 1.50E-02	7.38E-02 ± 8.86E-03	7.26E-02 ± 8.72E-03
	9	3.67E-02 ± 1.53E-02	3.59E-02 ± 1.56E-02	8.34E-02 ± 8.28E-03	8.34E-02 ± 8.39E-03
MO-CCFNN	3	**2.07E-03 ± 7.87E-04**	**2.04E-03 ± 7.62E-04**	1.39E-02 ± 4.26E-03	**1.37E-02 ± 4.19E-03**
(NMSE)	5	2.91E-03 ± 2.02E-03	2.95E-03 ± 2.14E-03	**1.38E-02 ± 4.26E-03**	1.41E-02 ± 4.51E-03
	7	2.53E-03 ± 5.89E-04	2.44E-03 ± 5.60E-04	1.96E-02 ± 4.20E-03	1.90E-02 ± 4.11E-03
	9	4.44E-03 ± 2.14E-03	4.32E-03 ± 2.18E-03	2.37E-02 ± 4.46E-03	2.39E-02 ± 4.52E-03

TABLE IV

THE TRAINING AND GENERALIZATION PERFORMANCE OF THE PROPOSED APPROACH ON THE ACI WORLDWIDE TIME SERIES

Prob.	H	Training(T=2)	Training(T=3)	Generalization (T=2)	Generalization (T=3)
MO-CCFNN	3	2.21E-02 ± 5.14E-04	2.29E-02 ± 1.45E-03	2.21E-02 ± 5.14E-04	1.86E-02 ± 8.26E-04
(RMSE)	5	2.14E-02 ± 4.94E-04	2.23E-02 ± 1.50E-03	2.14E-02 ± 4.94E-04	1.74E-02 ± 6.53E-04
	7	2.16E-02 ± 5.60E-04	2.24E-02 ± 1.46E-03	2.17E-02 ± 5.61E-04	1.80E-02 ± 8.09E-04
	9	**2.13E-02 ± 3.71E-04**	**2.22E-02 ± 1.47E-03**	**2.11E-02 ± 3.67E-04**	**1.72E-02 ± 5.25E-04**
MO-CCFNN	3	9.93E-04 ± 5.07E-05	1.07E-03 ± 6.39E-05	2.69E-03 ± 1.33E-04	1.95E-03 ± 1.90E-04
(NMSE)	5	9.29E-04 ± 3.26E-05	1.01E-03 ± 2.34E-05	2.53E-03 ± 1.22E-04	1.70E-03 ± 1.37E-04
	7	9.47E-04 ± 3.54E-05	1.02E-03 ± 3.10E-05	2.59E-03 ± 1.45E-04	1.84E-03 ± 1.90E-04
	9	**9.15E-04 ± 2.96E-05**	**9.97E-04 ± 2.61E-05**	**2.44E-03 ± 8.96E-05**	**1.65E-03 ± 1.06E-04**

TABLE V

SUMMARY OF THE BEST RESULTS FROM SINGLE OBJECTIVE CC-FNN (RMSE)

Dataset	Mean (T=2)	Best	Mean (T=3)	Best
Mackey	7.78E-03 ± 7.86E-04	3.78E-03	7.97E-03 ± 6.92E-04	4.04E-03
Lorenz	1.80E-02 ± 3.16E-03	2.21E-03	1.75E-02 ± 2.85E-03	3.38E-03
Sunspot	5.64E-02 ± 7.68E-03	1.69E-02	5.61E-02 ± 8.50E-03	1.66E-02
ACI Worldwide	2.07E-02 ± 2.59E-04	1.92E-02	1.90E-02 ± 6.27E-04	1.55E-02

Glass, Sunspot & Lorenz) and a financial data set (ACI Worldwide Inc.) from the NASDAQ stock exchange. This included two simulated time series problems (Mackey Glass & Lorenz) and two real world time series problems(Sunspot & ACI Worldwide Inc.) which contained noise. The same experimental setup for data from our previous work was used [28]. In the financial data set, the closing stock prices of ACI Worlwide between the period of December 2006 and February 2010 is used. A total of 1000 data points are used for each of the benchmark data sets while 800 data points were used for the financial data set. There was a 50-50 split between the testing and training set as this will allow for proper training and good test for generalization.

B. Results

This section reports on the performance of the proposed method for training a feedforward neural network on the benchmark time series problem with Neuron level (NL) problem decomposition method. The results for 50 experimental runs with 95% confidence interval are given in Tables I - VI with the best results highlighted in bold.

In the Mackey Glass problem, the proposed multi-objective method was able to give better generalization in comparison to the conventional (single-objective) cooperative coevolutionary method as shown in Table I and V. The proposed method recorded the lowest mean error rate for both T=2 and T=3. A greater improvement in terms of mean error and overall best was noticed on the data set which was reconstructed using a time lapse of 3. The mean error (RMSE) decreased as the number of hidden neurons increased. The difference between the training and testing performance was also quite small, implying that there was no sign of over-fitting. This can be because it is a simulated time-series and does not contain noise.

In the Lorenz time series problem, the proposed method outperformed the single-objective method as it recorded the lowest mean error for both T=2 and T=3 (Table II and V). In this problem, the multi-objective method gave better generalization with a smaller number of hidden neurons and as the number of hidden neurons increased, the overall performance deteriorated. The proposed method also reported improvement in the overall best for T=3. There was very little sign of any form of over-fitting as the training and test accuracy were very similar. This can be because it is a simulated time-series and does not contain noise.

The previous problems were simulated time series problems. The real test for the proposed method was the Sunspot and ACI Worldwide Inc. time series problems. These are real world time series' which contain noise that makes prediction difficult. In the Sunspot time series problem, the proposed approach did not show any improvement in comparison to the single-objective method, as shown in Table III and V. The method produced better results for smaller number of hidden neurons and as the number of hidden neurons increased, the results deteriorated. The proposed method was able to identify a large number of non-dominated solutions for this problem in comparison to the previous data sets. This means that there was a large pool of solutions to choose from when doing the final test prediction. A different combination of individuals may have given better or worse results as our final solution is a combination of random individuals from the non-dominated front. The poor results can also be attributed to over-fitting as the data contains noise.

The proposed approach gave a competitive performance on the ACI Worldwide financial time series as shown in Table IV and V. There was a big improvement noted on the test data set for T=3 but no improvement noted for the test data set reconstructed using T=2. It must be noted that the performance improved as the number of hidden neurons increased. The proposed method, as with the Sunspot problem, was able to identify a very large number of non-dominated solutions for this problem. This was the only problem in which there was a big difference noted in terms of accuracy for the different time lags. This may be due to the noise within the data set and the different amounts inherited by each time lag. Over-fitting was not an issue with this data set.

The multi-objective method reported the overall lowest error in 3 of the 4 problems as seen in table VI. The proposed method also gave competitive performance in comparison to the methods in the literature as shown in tables VII, VIII and IX. It outperformed majority of the methods except for hybrid NARX-Elman recurrent neural networks with residual analysis (2010) [30] and Auto regressive moving average with neural network (ARMA-ANN) (2008) [31]. This is mainly because these two models used different methods for data preprocessing.

C. Discussion

The goal of multi-objective time series prediction is to improve the prediction accuracy. By training on different time lags, the proposed multi-objective cooperative coevolutionary method was able to out-perform conventional cooperative coevolution in training feed-forward neural networks for time series problems. The proposed approach gave good results on majority of the benchmark data sets and also out-performed majority of the methods in the literature.

The main advantage of using different time lag values as the different objectives is that it allows the algorithm to explore regions within the data set which may be missed or ignored with a single time lag. It also allows the algorithm to study the patterns within the data set in more depth. We tested the method on two different test data sets (T=2 and T=3) and found that it showed improvement in both, with a greater improvement noted on T=3. Essentially, we had a single neural network which was able to predict for different time lags with the same or better accuracy in comparison to a single-objective method.

IV. Conclusion

In this paper, we proposed a multi-objective cooperative coevolutionary method for training feed-forward neural networks for time series prediction. The method utilized the

TABLE VII

A COMPARISON WITH THE RESULTS FROM LITERATURE ON THE MACKEY TIME SERIES

Prediction Method	RMSE	NMSE
Auto regressive moving average with neural network (ARMA-ANN)(2008) [31]	2.50E-03	
Boosted recurrent neural networks (2006) [32]		1.60E-04
Neural fuzzy network and hybrid of cultural algorithm and cooperative Particle swarm optimisation (CCPSO) (2009) [33]	8.42E-03	
Neural fuzzy network and particle swarm optimisation (PS0) (2009) [33]	2.10E-02	
Neural fuzzy network and cooperative particle swarm optimisation (CPS0) (2009) [33]	1.76E-02	
Neural fuzzy network and differential evolution (DE) (2009) [33]	1.62E-02	
Neural fuzzy network and genetic algorithm (GA) (2009)[33]	1.63E-02	
Hybrid NARX-Elman RNN with Residual Analysis (2010) [30]	3.72E-05	2.70E-08
Backpropagation neural network and genetic algorithms with residual analysis (2011) [34]	1.30E-03	
HMM-Fuzzy (2006) [35]	5.50E-03	
HMM-Fuzzy with EA (2012) [36]	4.80E-03	
Proposed Multi-Objective CCFNN–T=2	**3.84E-03**	**2.80E-05**
Proposed Multi-Objective CCFNN–T=3	**3.77E-03**	**2.70E-05**

TABLE VIII

A COMPARISON WITH THE RESULTS FROM LITERATURE ON THE LORENZ TIME SERIES

Prediction Method	RMSE	NMSE
Pseudo Gaussian - radial basis neural network (2002)	9.40E-02	
Boosted recurrent neural networks (2006) [32]		3.77E-03
Auto regressive moving average with neural network (ARMA-ANN)(2008) [31]	8.76E-02	
Backpropagation-through-time (BPTT-RNN) (2010) [37]		1.85E-03
Real time recurrent learning (RTRL-RNN) (2010) [37]		1.72E-03
Recursive Bayesian LevenbergMarquardt (RBLM-RNN) (2010) [37]		9.00E-04
Hybrid NARX-Elman RNN with Residual Analysis (2010) [30]	1.08E-04	1.98E-10
Backpropagation neural network and genetic algorithms with residual analysis (2011) [34]	2.96E-02	
Proposed Multi-Objective CCFNN–T=2	**2.19E-03**	**2.53E-05**
Proposed Multi-Objective CCFNN–T=3	**2.18E-03**	**2.54E-05**

TABLE IX

A COMPARISON WITH THE RESULTS FROM LITERATURE ON THE SUNSPOT TIME SERIES

Prediction Method	RMSE	NMSE
McNish-Lincoln [38]		8.00E-02
Multi-layer perceptron (1996) [39]		9.79E-02
Elman RNN (1996) [39]		9.79E-02
Wavelet packet multilayer perceptron (2001)[40]		1.25E-01
Radial basis network with orthogonal least squares (RBF-OLS)(2006) [41]		4.60E-02
Locally linear neuro-fuzzy model - Locally linear model tree (LLNF-LoLiMot) (2006) [41]		3.20E-02
Hybrid NARX-Elman RNN with Residual Analysis (2010) [30]	1.19E-02	5.90E-04
Proposed Multi-Objective CCFNN–T=2	**1.84E-02**	**1.02E-03**
Proposed Multi-Objective CCFNN–T=3	**1.81E-02**	**9.98E-04**

time lag feature of state space reconstructed time series as different objectives.

The proposed method gave a competitive performance in comparison to the single objective approach for training feedforward neural networks. It was also able to identify a good number of non-dominated solutions for each of the given problems which provides us with a wide range of optimal solutions. The results show that the multi-objective approach for training using different time-lags improves the overall generalization performance.

In future research, we would like to apply our proposed method to real world time series problems that include climate change and key problems in health. It will also be interesting to apply multi-objective cooperative coevolution to recurrent neural networks.

REFERENCES

[1] N. Garca-Pedrajas, C. Hervas-Martinez, and J. Munoz-Perez, "Multi-objective cooperative coevolution of artificial neural networks (multi-objective cooperative networks)," *Neural Networks*, vol. 15, pp. 1259–1278, 2002.

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

[3] H. K. Stephen, *In the Wake of Chaos: Unpredictable Order in Dynamical Systems*. University of Chicago Press, 1993.

[4] E. Lorenz, *The Essence of Chaos*. University of Washington Press, 1993.

[5] H. Jiang and W. He, "Grey relational grade in local support vector regression for financial time series prediction," *Expert Systems with Applications*, vol. 83, pp. 136–145, 2012.

[6] M. Ragulskis and K. Lukoseviciute, "Non-uniform attractor embedding for time series forecasting by fuzzy inference systems," *Neurocomputing*, pp. 2618–2626, 2009.

[7] R. de A Araujo, A. de Oliveira, and S. Soares, "A quantum-inspired hybrid methodology for financial time series prediction," *The 2010*

International Joint Conference on Neural Networks (IJCNN), pp. 1–8, 2010.

[8] T. Ferreira, G. Vasconcelos, and P. Adeodato, "A new evolutionary approach for time series forecasting," *2007. IEEE Symposium on Computational Intelligence and Data Mining*, pp. 616–623, 2007.

[9] R. de A.Araujo, R. Aranildo, and T. Ferreira, "Morphological-rank-linear time-lag added evolutionary forecasting method for financial time series forecasting," *IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 1340–1347, 2008.

[10] E. Parras-Gutierrez and V. Rivas, "Time series forecasting: Automatic determination of lags and radial basis neural networks for a changing horizon environment," *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2010.

[11] S. C. Chiam, K. C. Tan, and A. A. Mamun, "Multiobjective evolutionary neural networks for time series forecasting," *4th International Conference on Evolutionary Multi-Criterion Optimization*, pp. 346–360, 2007.

[12] J. Gonzalez, I. Rojas, H. Pomares, and J. Ortega, "Rbf neural networks, multiobjective optimization and time series forecasting," *6th International Work-Conference on Artificial and Natural Neural Networks*, pp. 498–505, 2001.

[13] J. Fieldsend and S. Singh, "Pareto evolutionary neural networks," *IEEE Transactions on Neural Networks*, vol. 16, pp. 338–354, 2005.

[14] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in *PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature*. London, UK: Springer-Verlag, 1994, pp. 249–257.

[15] ——, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evol. Comput.*, vol. 8, no. 1, pp. 1–29, 2000.

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

[17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, vol. 6, pp. 182–197, 2002.

[18] K. C. Tan, T. H. Lee, Y. J. Yang, and D. S. Liu, "A cooperative coevolutionary algorithm for multiobjective optimization," *2004 IEEE International Conference on Systems, Man and Cybernetics*, pp. 1926–1931, 2004.

[19] L. M. Antonio and C. A. Coello Coello, "Use of cooperative coevolution for solving large scale multiobjective optimization problems," *2013 IEEE Congress on Evolutionary Computation*, pp. 2758–2765, 2013.

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

[21] B. Dorronsoro, G. Danoy, A. Nebro, and B. Bouvry, "Achieving super-linear performance in parallel multi-objective evolutionary algorithms by means of cooperative coevolution," *Computers and Operations Research*, vol. 40, pp. 1552–1563, 2013.

[22] C. K. Goh, K. C. Tan, D. S. Liu, and S. C. Chiam, "A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design," *European Journal of Operational Research*, vol. 202, pp. 42–54, 2010.

[23] C. Frazier and K. Kockelman, "Chaos theory and transportation systems: Instructive example," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 20, pp. 9–17, 2004.

[24] 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.

[25] ——, "An encoding scheme for cooperative coevolutionary feedforward neural networks," in *AI 2010: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 6464, pp. 253–262.

[26] K. Deb and R. B. Agrawal, "Simulated Binary Crossover for Continuous Search Space," Tech. Rep., 1994.

[27] 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.

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

[29] A. W. Iorio and X. Li, "A cooperative coevolutionary multiobjective algorithm using non-dominated sorting," *The Genetic and Evolutionary Computation Conference (GECCO-2004)*, pp. 537–548, 2004.

[30] M. Ardalani-Farsa and S. Zolfaghari, "Chaotic time series prediction with residual analysis method using hybrid elman-narx neural networks," *Neurocomputing*, vol. 73, no. 13-15, pp. 2540 – 2553, 2010.

[31] 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.

[32] M. Assaad, R. Bon, and H. Cardot, "Predicting chaotic time series by boosted recurrent neural networks," in *Neural Information Processing*, ser. Lecture Notes in Computer Science, I. King, J. Wang, L.-W. Chan, and D. Wang, Eds. Springer Berlin / Heidelberg, 2006, vol. 4233, pp. 831–840.

[33] 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," *Trans. Sys. Man Cyber Part C*, vol. 39, pp. 55–68, January 2009.

[34] 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, January 2011.

[35] M. Hassan, B. Nath, and M. Kirley, "Hmm based fuzzy model for time series forecasting," *Proceedings of World Congress on Computational Intelligence (WCCI2006)*, pp. 9963–9968, 2006.

[36] M. R. Hassan, B. Nath, M. Kirley, and J. Kamruzzaman, "A hybrid of multiobjective evolutionary algorithm and hmm-fuzzy model for time series prediction," *Neurocomputing*, vol. 81, pp. 1–11, 2012.

[37] D. Mirikitani and N. Nikolaev, "Recursive bayesian recurrent neural networks for time-series modeling," *Neural Networks, IEEE Transactions on*, vol. 21, no. 2, pp. 262 –274, feb. 2010.

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

[39] T. Koskela, M. Lehtokangas, J. Saarinen, and K. Kaski, "Time series prediction with multilayer perceptron, fir and elman neural networks," in *In Proceedings of the World Congress on Neural Networks*, 1996, pp. 491–496.

[40] K. K. Teo, L. Wang, and Z. Lin, "Wavelet packet multi-layer perceptron for chaotic time series prediction: Effects of weight initialization," in *Proceedings of the International Conference on Computational Science-Part II*, ser. ICCS '01, 2001, pp. 310–317.

[41] 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.