# An Architecture for Encoding Two-Dimensional Cyclone Track Prediction Problem in Coevolutionary Recurrent Neural Networks

Rohitash Chandra* Ratneel Deo[†] and Christian W. Omlin[‡]
*[†]Artificial Intelligence and Cybernetics Research Group
Software Foundation, Nausori, Fiji
http://aicrg.softwarefoundationfiji.org
Email: c.rohitash@gmail.com
[†]School of Computing Information and Mathematical Sciences
University of South Pacific, Suva, Fiji.
Email: deo.ratneel@gmail.com
[‡]School of Computing
University of South Africa
Florida 1709
Johannesburg, South Africa
Email: omlinc@unisa.ac.za

***Abstract*—Cyclone track prediction is a two dimensional time series prediction problem that involves latitudes and longitudes which define the position of a cyclone. Recurrent neural networks have been suitable for time series prediction due to their architectural properties in modeling temporal sequences. Coevolutionary recurrent neural networks have been used for time series prediction and also applied to cyclone track prediction. In this paper, we present an architecture for encoding two dimensional time series problem into Elman recurrent neural networks composed of a single input neuron. We use cooperative coevolution and back-propagation through-time algorithms for training. Our experiments show an improvement in the accuracy when compared to previous results using a different recurrent network architecture.**

## I. Introduction

The forecast of the intensity and track of tropical cyclones is considered extremely important for avoiding casualties and mitigating damage to properties [1], [2]. A comprehensive review of tropical cyclone track forecasting techniques can be found in [3]. Computational intelligence methods have established themselves as complementary approaches for tropical cyclone tracking and prediction [4], [5], [6], [7]. Amongst these methods, neural networks trained using an evolutionary learning paradigm have shown great promise [8], [9], [10], [11].

Recurrent neural networks (RNNs) are well-suited for modeling temporal sequences due to their architectural properties [12], [13]. Cooperative coevolution (CC) is an evolutionary algorithm for optimization [14] that has been used to train RNNs with promising results [10]. Cooperative coevolution divides a network architecture into subcomponents determined by the problem decomposition method [13], [15].

Cyclone path and wind intensity prediction is a multidimensional time series problem. In previous work, we have applied cooperative coevolution to train RNNs to predict both the wind intensity [16] and tracks [11] of cyclones. We considered the modeling of a cyclone's track as a two-dimensional time series problem consisting of the latitude and longitude [11]. Although the results were promising, there is room for further improvements in the accuracy of the predictions.

In this paper, we train Elman RNNs with a single input neuron for both latitude and longitude using cooperative coevolution and back-propagation through-time [17]. We reconstruct two-dimensional time series data using Taken's theorem [18] and experiment with different number of hidden neurons in RNNs to test scalability and robustness. The results are compared with our previous work which used a similar architecture for the same prediction problem. The main contribution of this paper is introducing a new architecture for encoding two-dimensional time-series data in Elman RNNs.

The rest of the paper is organized as follows. Section II presents a brief background on neural networks for cyclone modeling and prediction and Section III presents a detailed presentation of the proposed method for cyclone track prediction. Experimental results are given in Section IV followed by conclusion and directions for future research in Section V.

## II. Background and Related Work

### A. Background on Tropical Cyclones

A tropical cyclone is a non-frontal low pressure system with organized convection that forms over warm tropical waters [19]. A cyclone once formed moves over the ocean in the direction away from the equator lasting a few days to sometimes 2-3 weeks [20]. During its lifetime, a cyclone can travel hundreds of kilometers and the actual position of the cyclone's eye recorded every six hours defines the cyclone's track. The forecast of a cyclone comprises cyclone track, intensity, induced storm surges, rainfall and threat to coastal areas [2]. The direction of cyclone movement and wind intensity are the most important features in the forecast as it helps the inhabitants to prepare ahead of time and minimize damage to life and property. For this reason, forecasting cyclone track and intensity is considered extremely important forecast functions by scientists and meteorological agencies around the world [1]. Various combination techniques in the track and intensity prediction models are incorporated to account for the variation in cyclone behavior in different ocean basins and achieve highest possible accuracy and reliability [1], [3].

### B. Neural Networks as Models for Cyclone Prediction

Neural networks have been used for the prediction of the maximum potential intensity of cyclones [21], [16].The error backpropagation learning algorithm was used in a feedforward neural network model with two hidden layers with binary triggers that dynamically triggered the neurons based on the regressions of the inputs. The proposed model provided satisfactory results on Western North Pacific tropical cyclones [21]. A model inspired by the human visual system consisting of a multi-layered neural network architecture with bi-directional connections in the hidden layers was introduced by [22]. The prediction of the direction of movement from previously unseen satellite images showed good performance on the novel images.

A hybrid neural network model that clusters input data using self-organizing maps and feeds data from the different clusters to separate multi-layer perceptrons for training and prediction was proposed in [23]. The method was used for forecasting actual typhoon-rainfall in Taiwan's Tanshui river basin and showed improved performance over the conventional prediction methods.

An investigation was conducted on the impact of varying the number of layers and the number of neurons per layer in a neural network for prediction of the direction and intensity of cyclones over the North Indian Ocean [4]. The study found that an increase in the number of hidden layers improved the accuracy of the forecast while the number of nodes in the hidden layer had no significant effect on performance. An approach combining a multilayer perceptron with a neuro-fuzzy model for the prediction of a cyclone's track and surge height of cyclones for the same cyclone data showed good prediction performance [24]. Chandra et al. [11] proposed track prediction based on cooperative coevolution of Elman RNNs for cyclones in the South Pacific; the cyclone's track given by latitude and longitude were treated as separate dimensions. Although the results were promising in terms of the error, large differences were observed in the predicted position of the cyclone when compared to the actual position.

## III. Recurrent Networks for Cyclone Track Prediction

RNNs are dynamical systems which use states from previous time steps to compute new current state; they are thus well-suited for modeling, classification and prediction of temporal sequences [12].

Elman RNNs use a context layer to compute the new state from the previous state and current inputs. Note that the basic components of an observed dynamical system are clearly represented in an Elman network: the input stands for the control of the system, the contextual hidden layer stands for the state of the system and the outputs stand for the measurement [25]. The network is able to develop representations of unobservable states of a dynamical system in the hidden layer through learning.

The change of the hidden state neurons' activation in Elman style recurrent networks [12] is given by Equation (1).

$$y_i(t) = f\left(\sum_{k=1}^{K} v_{ik}\ y_k(t-1) + \sum_{j=1}^{J} w_{ij}\ x_j(t-1)\right) \qquad (1)$$

where $y_k(t)$ and $x_j(t)$ represent the output of the context state neuron and input neurons respectively, $v_{ik}$ and $w_{ij}$ represent their corresponding weights, and $f(.)$ is a sigmoid transfer function.

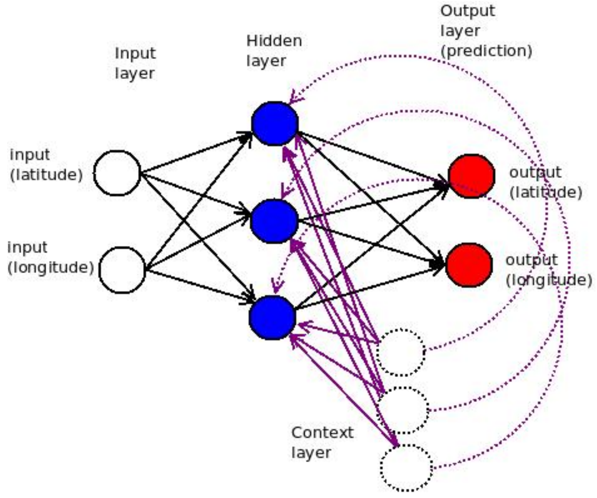


Fig. 1. Previous network architecture: Elman RNN used for prediction of latitude and longitude via separate neurons of the cyclone path with two input and output neurons [11].

### A. Proposed RNN Architecture for Cyclone Tracks

In previous work, the use of two input and output neurons for a cyclone's longitude and latitude, respectively, has been deployed using Elman RNN [11] as shown in Figure 1.

The latitude and longitude are separate time series which are interrelated as they are part of one variable of the cyclone, i.e., the cyclone's track. We propose a new RNN model that combines the two dimensions of latitude and longitude into a single data stream variable in an attempt to represent the direct relationship between the dimensions as shown in Figure 2. This network architecture is similar to that of Figure 1 but uses one single input neuron and one output neuron which predicts both longitude and latitude as depicted. In this model, the single neurons represent both the longitude and latitude, thus preserving some form of correlation between the two time series. The proposed network architecture is called single input-output neural network (SIORNN). It will be trained using the error backpropagation through time (BPTT) and the cooperative coevolution algorithm.

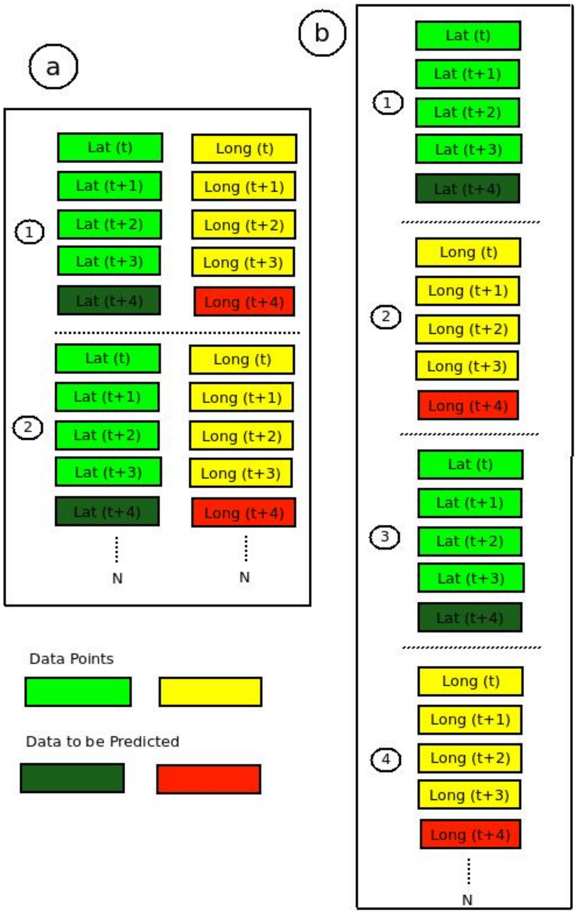


Fig. 3. Embedded data reconstructed using Taken's theorem. Embedding dimension (D) of 4 is depicted. Both (a) and (b) have the two dimensions longitude and latitude.

### *B. Neural Networks Training with Cooperative Coevolution*

Problem decomposition in cooperative coevolutionary learning determines how a problem is broken down into subcomponents. The subcomponents are implemented as subpopulations that are evolved in a *round-robin* fashion. The *search depth* determines the number of generations which each subpopulation will evolve through and must be chosen a priori. In our past work, we have found that a search depth of one generation is suitable for this evolutionary process [26].

The general cooperative coevolutionary method for training Elman recurrent neural networks is given in Algorithm 1. The recurrent neural network is decomposed in $k$ subcomponents using the neural level problem decomposition method [26], where $k$ is equal to the total number of hidden, context and output neurons. Each subcomponent contains all the weight links from the previous layer connecting to a particular neuron. Each hidden neuron also acts as a reference point for the recurrent (state or context) weight links connected to it. Therefore, the subcomponents for a recurrent network with a single hidden layer are composed as follows:

1) Hidden layer subcomponents: weight links from each neuron in the $hidden(t)$ layer connected to all $input(t)$ neurons and the bias of $hidden(t)$, where $t$ is time.
2) State (recurrent) neuron subcomponents: weight links from each neuron in the $hidden(t)$ layer connected to all hidden neurons in previous time step $hidden(t-1)$.
3) Output layer subcomponents: weight links from each neuron in the $output(t)$ layer connected to all $hidden(t)$ neurons and the bias of $output(t)$.

.

**Alg. 1** Cooperative Coevolutionary Training of Elman Recurrent Networks

```
Step 1: Decompose the problem into $k$ subcomponents according to the number of Hidden, State, and Output neurons
Step 2: Encode each subcomponent in a sub-population in the following order:
i) Hidden layer sub-populations
ii) State (recurrent) neuron sub-populations
iii) Output layer sub-populations
Step 3: Initialize and cooperatively evaluate each sub-population
for each cycle until termination do
  for each Sub-population do
    for $n$ Generations do
      i) Select and create new offspring
      ii) Cooperatively evaluate the new offspring
      iii) Add the new offspring to the sub-population
    end for
  end for
end for
```

The subcomponents are implemented as sub-populations that employ the generalized generation gap with parent-centric crossover (G3-PCX) operator genetic algorithm [27]. A *cycle* is completed when all the sub-populations are evolved for a fixed number of generations.

Cooperative evaluation in the evolution phase is shown in Step 3 (ii). This is done by concatenating the chosen individual from a sub-population $k$ with the single best individual from the rest of the sub-populations [14]. The algorithm halts when the termination condition is satisfied: either a specified fitness has been achieved as measured by the root mean squared error on the training data set or the maximum number of function evaluations has been reached.

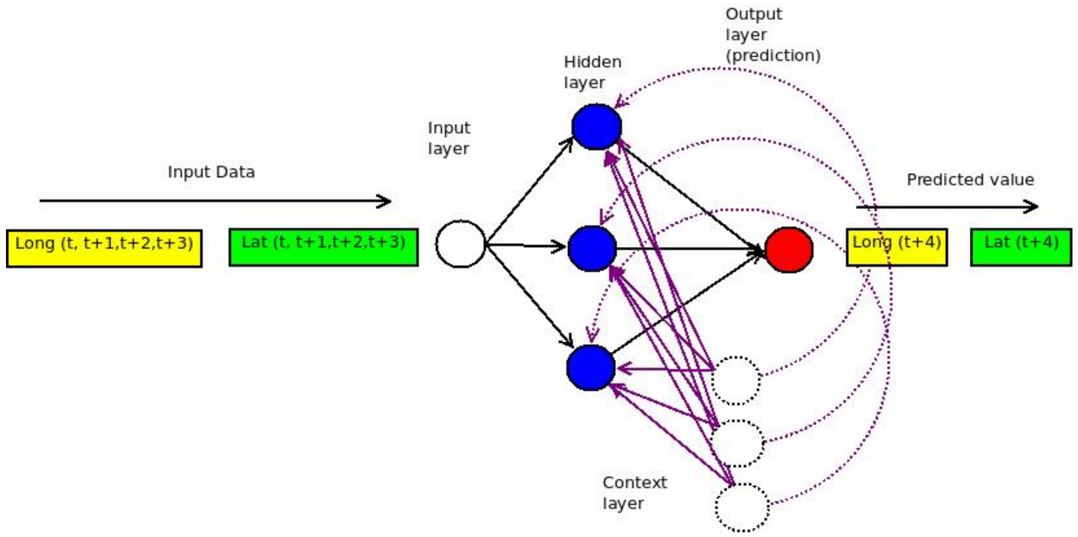


Fig. 2. Proposed RNN architecture: A single input and output neuron Elman recurrent neural network (SIORNN) used for predicting latitude and longitude of the cyclone path.

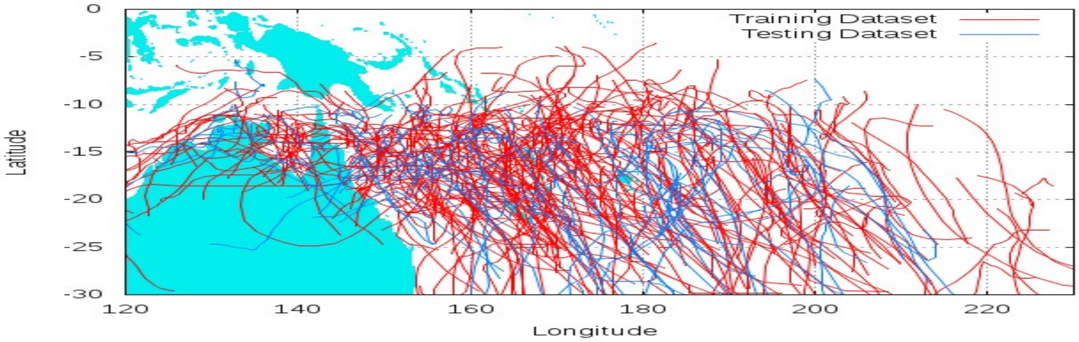


Fig. 4. Tropical cyclones track data in the South Pacific from 1985 to 2013. (Generated using Gnuplot)

## IV. Simulation and Analysis

### A. Data Preprocessing and Reconstruction

We used Taken's theorem [18] to reconstruct the time series data in state space vector. Given an observed time series $x(t)$, an embedded phase space $Y(t) = [(x(t), x(t-T), ..., x(t(D-1)T)]$ can be generated, where, $T$ is the time delay, $D$ is the embedding dimension, $t = 0, 1, 2, ..., N - DT - 1$ and $N$ is the length of the original time series.

In the above case, only one-dimensional time series is considered. We consider two dimensions (latitude and longitude) and hence Taken's theorem is applied to two dimensions.

The reconstructed vector is used to train the RNN for one-step-ahead prediction. In the cooperative coevolutionary recurrent network (CCRNN) architecture, two neurons are used in the input and the output layer to represent the latitude, longitude shown in Figure 1. In the proposed architecture (SIORNN), there is only one neuron in the input and output layer. The processed data was laid out in two layouts as shown in Figure 3 in order to be successfully handled by the different network architectures. The recurrent network unfolds $k$ steps in time which is equal to the embedding dimension $D$ [10], [28], [29].

The time series data contained 6000 points in the training set (Tropical Cyclones from 1985 - 2005). There were 2000 points in the test set (Tropical Cyclones from 2006 - 2013) taken from JTWC data set [30]. The readings were taken every 6 hours during the course of each of the tropical cyclones. Figure 4 shows the actual positions of the tropical cyclones from the dataset.

All the cyclone time series were combined together. Data preprocessing was done by considering the position in the southern hemisphere and converting all into one region. The conversion of latitude was done by multiplying the original latitude by -1 to accommodate for South in the Southern Hemisphere. The longitudes with East (E) coordinates remained unchanged while the West (W) coordinates were subtracted from 360° to define all points in terms of East coordinates for easier plotting of cyclone tracks on the spatial map.

The data set contained time series of the position (latitude and longitude). We used the following combinations of dimension and time lag using Taken's theorem.

- Configuration A: $D = 4$ and $T = 2$, reconstructed dataset contains 3417 samples in training set and 1298 samples in test set.
- Configuration B: $D = 5$ and $T = 3$, reconstructed dataset contains 2278 samples in training set and 865 samples in test set.

### B. Experimental Design

We experimented with different number of hidden neurons in the RNN that employed sigmoid units in the hidden and output layer. We use implementation from *Smart Bilo: An Open Source Computational Intelligence Framework* in our experiments [31].

The CC-SIORNN as well as BPTT-SIORNN were both trained for and their predictions tested for 24-hour and 30-hour advance warning. The termination condition was set at 50,000 function evaluations for CC and 2000 epochs for BPTT. The root mean squared error (RMSE) was used to evaluate the performance of the two architectures for cyclone track prediction given in Equation 2.

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

where $y_i$ and $\hat{y}_i$ are the observed and predicted data, respectively. $N$ is the length of the observed data.

## V. Results and Discussion

The mean and 95% confidence interval is given from 30 experimental runs are shown in Table I and Table II. The best results with the least values of RMSE given by each of the configurations are shown in bold. The test for robustness of the proposed architecture is done using different sets of configuration to show scalability as they contain varied data set sizes.

The comparison of the combination of the single neuron and the multi-neuron CCRNN methods shown by cyclone tracks is given in Tables I and II. The best performance was achieved by CC-SIORNN. It was able to outperform the other methods for all the cases.

The number of hidden neurons did not make any considerable difference to the results although 5 hidden neurons gave the best performance in configuration B and 3 hidden neurons had better performance in configuration A. CCRNN produced best best results with 7 neurons in the hidden layer. It seems that 3 hidden neurons were not sufficient to represent the two-dimensional time series problem when separate neurons handled the two dimensions.

Figure 5 shows the performance of cooperative coevolution using the proposed SIORNN architecture. It shows the path of 6 selected cyclones from those that occurred between 2006 and 2013. Figure 6 shows the typical prediction performance of a single experimental run given by the BPTT-SIORNN for cyclone track test data set (2006 - 2013 tropical cyclones). The errors shown by the prediction is also given in the graphs.

### A. Discussion

Cyclone track is generally viewed as a single dimension but, it is modeled as two separate dimensions of latitude and longitude. Through this research, we found out that although we treat latitude and longitude independently, there is correlation between them and when they are encoded into a recurrent neural network as a single stream of data the prediction performance improves significantly regardless of the training algorithm.

Cooperative coevolution and error backpropagation through-time gave better performance when compared to the two neuron architecture due to the adapted network configuration. The improvement in performance could be due to the combination of the two dimensional time series consisting of a cyclone's longitude and latitude into a single stream of data points as represented in Figure 3 (b). A single data stream increases the chance of preserving the interdependencies between the latitude and longitude while reaffirming the correlation between the two inputs. Therefore, SIORNN outperforms CCRNN. In the traditional approach there is minimal probability of preserving interdependencies within the track attributes as each input is encoded independently therefore, loosing the correlation between latitude and longitude.

The major errors in predictions were seen at locations where there was a transition from one cyclone to another in the data. This is due to the concatenation of the various cyclones into a single data stream. As seen from the results of a typical prediction given in Figure 6, the region where there is a switch from one cyclone to another produces a large error in the prediction. The network was unable to cope with the sudden change in the data due to the occurrence of the cyclone at a different location. The data concatenated the location of the cyclones where the the end of one cyclone is adjacent to the beginning of the other cyclone which are independent events. These have been considered as joint events for training RNNs. Further studies need to be done in order to improve the prediction accuracy at the beginning and end of the cyclones.

## VI. Conclusions and Future Work

In this paper, we investigated how the the two-dimensional time series consisting of longitude and latitude is best represented for superior prediction performance when used for training RNNs. In the first method, the latitude and longitude is presented to a recurrent neural network with two input neurons whereas the second methods combines both variables in a single data stream and employs a network with single input neuron which interleaves successive longitude and latitude values. We trained RNNs with cooperative coevolution and backpropagation-through-time algorithms.

The results show that a single input and output layer neuron network trained with either of the algorithms outperforms networks trained with separate inputs for longitude and latitude. It is evident that it is more difficult train the recurrent neural network for both tasks given by the previous method as the number of dimensions increase along with the noise in the time series and with it the uncertainly. The proposed method

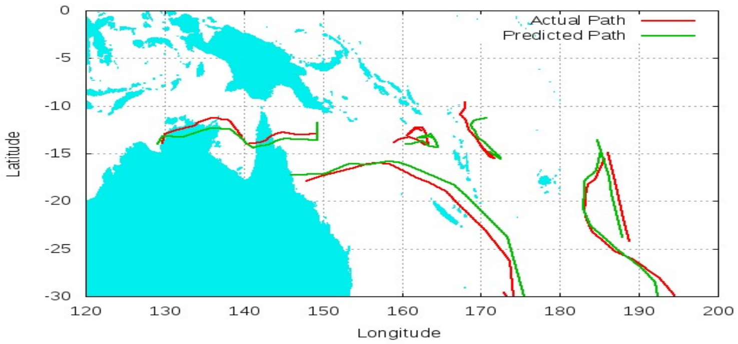


Fig. 5. Performance of SIORNN using cooperative coevolution for 6 random cyclones from the year 2006 to 2013.

TABLE I
RESULTS FOR CONFIGURATION A

| Model | Hidden | RMSE (Train) | RMSE (Test) | Best |
|---|---|---|---|---|
| CCRNN | 3 | 0.0508 ± 0.0010 | 0.0484 ± 0.0010 | 0.0455 |
| CCRNN | 5 | 0.0493 ± 0.0006 | 0.0471 ± 0.0006 | 0.0447 |
| CCRNN | 7 | 0.0492 ± 0.0007 | **0.0471± 0.0006** | **0.0448** |
| CC-SIORNN | 3 | 0.0252 ± 0.0003 | **0.0244** ± **0.0002** | 0.0238 |
| CC-SIORNN | 5 | 0.0252 ± 0.0003 | 0.0245 ± 0.0003 | 0.0238 |
| CC-SIORNN | 7 | 0.0266 ± 0.0033 | 0.0260 ± 0.0034 | **0.0237** |
| BPTT-SIORNN | 3 | 0.0265 ± 0.0002 | 0.0257 ± 0.0002 | 0.0245 |
| BPTT-SIORNN | 5 | 0.0260 ± 0.0003 | 0.0254 ± 0.0002 | 0.0245 |
| BPTT-SIORNN | 7 | 0.0256 ± 0.0003 | **0.0251** ± **0.0003** | **0.0242** |

TABLE II
RESULTS FOR CONFIGURATION B

| Model | Hidden | RMSE (Train) | RMSE (Test) | Best |
|---|---|---|---|---|
| CCRNN | 3 | 0.0526 ± 0.0014 | 0.0481± 0.0013 | 0.0432 |
| CCRNN | 5 | 0.0506± 0.0007 | 0.0462± 0.0008 | 0.0430 |
| CCRNN | 7 | 0.0497± 0.0006 | **0.0456± 0.0006** | **0.0425** |
| CC-SIORNN | 3 | 0.0260 ± 0.0004 | 0.0242 ± 0.0004 | **0.0232** |
| CC-SIORNN | 5 | 0.0254 ± 0.0001 | **0.0237** ± **0.0001** | 0.0233 |
| CC-SIORNN | 7 | 0.0256 ± 0.0003 | 0.0241 ± 0.0003 | **0.0232** |
| BPTT-SIORNN | 3 | 0.0254 ± 0.0002 | 0.0242 ± 0.0002 | **0.0235** |
| BPTT-SIORNN | 5 | 0.0252 ± 0.0001 | **0.0239** ± **0.0001** | **0.0235** |
| BPTT-SIORNN | 7 | 0.0252 ± 0.0001 | **0.0239** ± **0.0001** | **0.0235** |

has alleviated this weakness and produced improved results that motivates real time implementation.

Although the results have been very promising, it may be possible to approach the multidimensional problem as a group of single-dimensional time series problems using a mixture of computational intelligence methods for cyclone track prediction.

There is also motivation for using additional atmospheric conditions that are major attributes in the formation of cyclones such as the sea surface temperature, pressure and humidity and the change of their intensity with time. Another attribute that can be considered is the speed at which the cyclone is moving and the geographical landscape, i.e., sea and land.

## REFERENCES

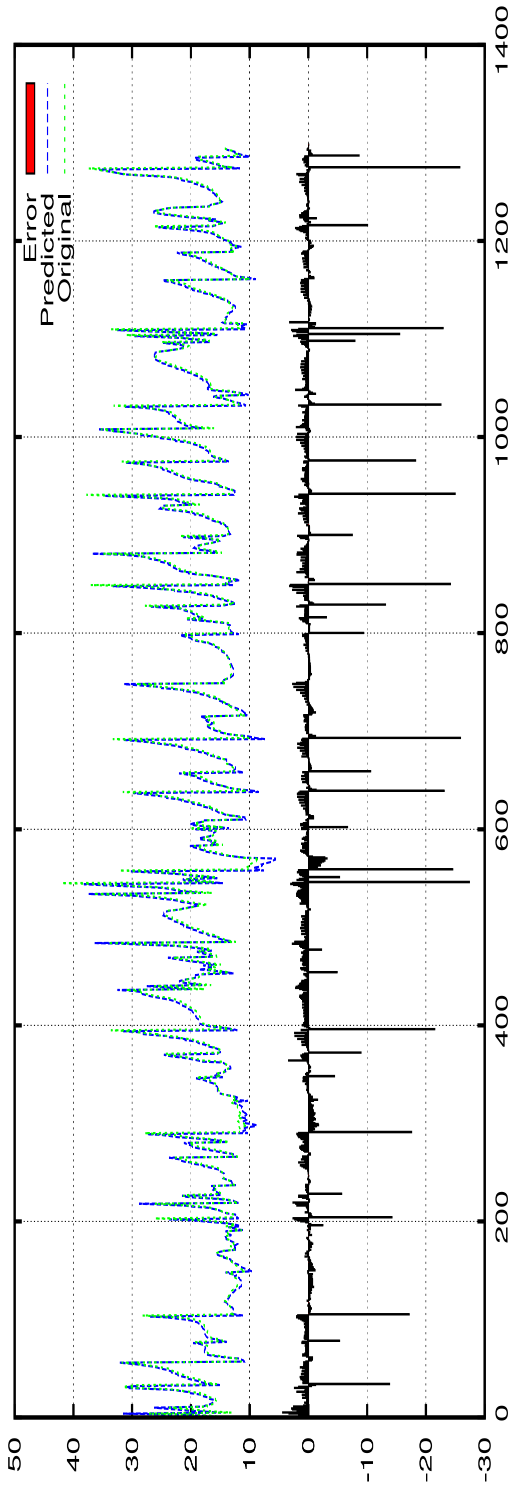
[1] "Tropical-cyclone forecasting: a worldwide summary of techniques and verification statistics," *Bulletin of American Meteorological Society*, vol. 68.

[2] G. Holland, "Global guide to tropical cyclone forecasting. bureau of meteorology research center, melbourne, australia," http://cawcr.gov.au/publications/BMRC_archive/tcguide/globa_guide_intro.htm, 2009, accessed: January 21, 2015.

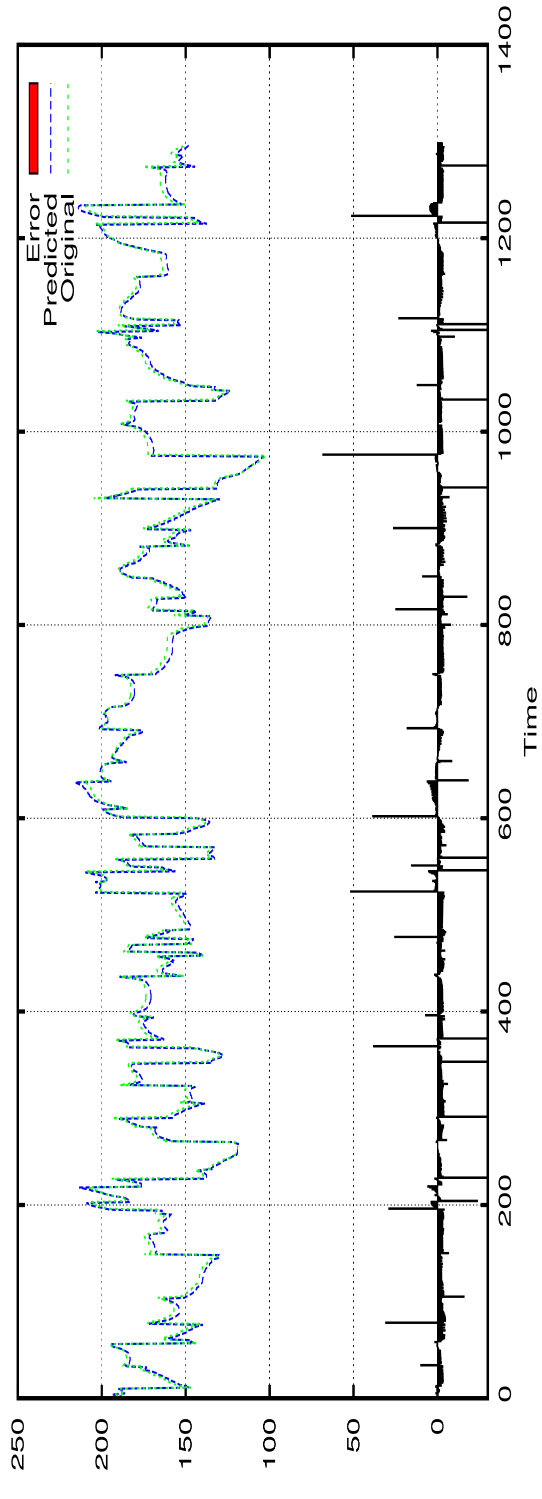
[3] C. Roy and R. Kovordnyi, "Tropical cyclone track forecasting techniques - a review," *Atmospheric Research*, vol. 104-105, pp. 40–69, 2012.

[4] S. Chaudhuri, D. Dutta, S. Goswami, and A. Middey, "Track and intensity forecast of tropical cyclones over the north indian ocean with multilayer feed forward neural nets," *Meteorological Applications*, vol. 22, no. 3, pp. 563–575, 2015. [Online]. Available: http://dx.doi.org/10.1002/met.1488

[5] L. E. Carr III, R. L. Elsberry, and J. E. Peak, "Beta test of the systematic approach expert system prototype as a tropical cyclone track forecasting aid," *Weather and forecasting*, vol. 16, no. 3, pp. 355–368, 2001.



(a) Performance of latitude prediction the test data set



(b) Performance of longitude prediction the test data set

Fig. 6. Typical prediction performance of a single experiment (one-step ahead prediction) given by BPTT-SIORNN for **Cyclone track** test data set (2006 - 2013 tropical cyclones) where *Time* is taken at six hour intervals.

[6] R. Kovordnyi and C. Roy, "Cyclone track forecasting based on satellite images using artificial neural networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 64, pp. 513–521, 2009.

[7] L. Jin, C. Yao, and X.-Y. Huang, "A nonlinear artificial intelligence ensemble prediction model for typhoon intensity," *Monthly Weather Review*, vol. 136, pp. 4541–4554, 2008.

[8] R. Chandra, "Multi-objective cooperative neuro-evolution of recurrent neural networks for time series prediction," in *IEEE Congress on Evolutionary Computation*, Sendai, Japan, May 2015, pp. 101–108.

[9] W. Du, S. Y. S. Leung, and C. K. Kwong, "Time series forecasting by neural networks: A knee point-based multiobjective evolutionary algorithm approach," *Expert Systems with Applications*, vol. 41, no. 18, pp. 8049 – 8061, 2014.

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

[11] R. Chandra, K. Dayal, and N. Rollings, "Application of cooperative neuro-evolution of Elman recurrent networks for a two-dimensional cyclone track prediction for the South Pacific region," in *International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, July 2015, pp. 721–728.

[12] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179–211, 1990.

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

[14] 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.

[15] 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.

[16] R. Chandra and K. Dayal, "Cooperative coevolution of Elman recurrent networks for tropical cyclone wind-intensity prediction in the South Pacific region," in *IEEE Congress on Evolutionary Computation*, Sendai, Japan, May 2015, pp. 1784–1791.

[17] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

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

[19] (2000) Bureau of meteorology research centre. Accessed: January 21, 2015.

[20] S. Debsarma, "Cyclone and its warning system in bangladesh," National Disaster Reduction Day, 2001.

[21] *A Neural Network Regression Model for Tropical Cyclone Forecast*, 2005.

[22] R. Kovordányi and C. Roy, "Cyclone track forecasting based on satellite images using artificial neural networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 64, no. 6, pp. 513–521, 2009.

[23] G.-F. Lin and M.-C. Wu, "A hybrid neural network model for typhoon-rainfall forecasting," *Journal of Hydrology*, vol. 375, no. 3, pp. 450–458, 2009.

[24] S. Chaudhuri, S. Goswami, A. Middey, D. Das, and S. Chowdhury, "Predictability of landfall location and surge height of tropical cyclones over north indian ocean (nio)," *Natural Hazards*, vol. 75, no. 2, pp. 1369–1388, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11069-014-1376-0

[25] M. Samuelides, "Neural identification of controlled dynamical systems and recurrent networks," in *Neural Networks*. Springer, 2005, pp. 231–287.

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

[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] 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*, San Diego, CA, USA, 1996, pp. 491–496.

[29] 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.

[30] (2015) JTWC tropical cyclone best track data site. [Online]. Available: \url{http://www.usno.navy.mil/NOOC/nmfc-ph/RSS/jtwc/}

[31] "Smart bilo: Computational intelligence framework," accessed: 02-12-2015. [Online]. Available: http://smartbilo.aicrg.softwarefoundationfiji.org