Identification of Minimal Timespan Problem for Recurrent Neural Networks with application to Cyclone Wind-Intensity Prediction

Ratneel Deo* and Rohitash Chandra†
*†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

***Abstract*—Time series prediction relies on past data points to make robust predictions. The span of past data points is important for some applications since prediction will not be possible unless the minimal timespan of the data points is available. This is a problem for cyclone wind-intensity prediction, where prediction needs to be made as a cyclone is identified. This paper presents an empirical study on minimal timespan required for robust prediction using Elman recurrent neural networks. Two different training methods are evaluated for training Elman recurrent network that includes cooperative coevolution and backpropagation-though time. They are applied to the prediction of the wind intensity in cyclones that took place in the South Pacific over past few decades. The results show that a minimal timespan is an important factor that leads to the measure of robustness in prediction performance and strategies should be taken in cases when the minimal timespan is needed.**

I. Introduction

Computational intelligence methods have shown to be robust methods for time series problems [1]. Amongst popular computational intelligence methods, evolutionary neural networks have shown good potential for time series prediction [2], [3]. Recurrent neural networks (RNNs) due to their architecture are well-suited for modeling temporal sequences [4].

Tropical cyclones have aroused much attention due to their destructive nature [5]. Statistical models have been previously used to forecast the movement and intensity of the cyclones. Climatology and Persistence (CLIPER) is one of the computer-based forecast models which was able to give 5 days prediction, i.e., 72 hours of cyclone intensity [6].

Statistical hurricane intensity prediction scheme (SHIPS) has also been used for cyclone intensity forecasts although restricted to storms over the ocean only [7]. There has been a growing interest in computational intelligence techniques for cyclone prediction systems [8], [9]. Cyclone path and wind intensity prediction are seen as a time series problem. In the past, cyclone wind-intensity [10] and track prediction [11] have been tackled by cooperative neuro-evolution of recurrent neural networks. Track prediction was tackled as a two-dimensional time series problem where the latitude and longitude of the cyclone tracks were involved [11]. The results have been promising for cyclones in the South Pacific region.

The time span is a windowed snapshot of taken at regular intervals the observation period for a time series data [12]. The minimal timespan is an important factor when it comes to predicting the nature cyclones in terms of track and wind intensity. Cyclone path and track prediction need to be made as soon as possible when a cyclone is identified. It is important to identify if a prediction model can work with the shortest duration after which time series prediction can begin, i.e if the cyclone data is recorded every 6 hours, an important issue is the minimal timespan required to make a prediction. Robust predictions can be vital in reducing the impact of the calamity of cyclones through efficient planning and management.

This paper presents an empirical study on minimal timespan required for robust prediction using Elman RNNs. We train a prediction model and test it for robustness regarding minimal timespan. We perform this by training it with different size of timespan and using a different size for testing. For instance, in the case of cyclone wind-intensity prediction, this could be 36 hours (6 data points recorded every 6 hours) for training and then testing it with 12 hours (2 data points recorded every 6 hours). Therefore, in this case, there is a need to evaluate the quality of prediction within 12 hours when a cyclone is formed. We run several different types of experiments to test the robustness of Elman RNNs using two different training methods that include cooperative neuro-evolution and backpropagation-though-time. the training set in the test set set[13]. each of the

The rest of the paper is organized as follows. We give a brief background on neural networks for cyclone modeling and prediction in Section II, followed by a detailed presentation of the methods used for identifying the minimal timespan in Section III. Section IV presents the overview of minimal timespan problem. We discuss our experimental results in

Section V and conclude with directions for future research in Section VI.

II. Background and Related Work

A. Neural Networks for Cyclones

Neural network regression models have been used for the prediction of the maximum potential intensity of cyclones [8].The error back-propagation learning algorithm was used in a feedforward neural network 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 [8]. 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 [14]. The prediction of the direction of movement from previously unseen satellite images showed good performance.

A hybrid neural network model that clusters input data using self-organizing maps and feeds data from the different clusters to separate networks for training and prediction was proposed [15]. The method was used for forecasting actual typhoon-rainfall in Taiwan's Tanshui river basin. It showed improved performance over the conventional prediction methods.

An investigation was done on the impact of varying the number of layers and the number of neurons per layer for the prediction of the direction and intensity of cyclones over the North Indian Ocean [16]. 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 [17]. Chandra et al. [11] proposed a method for cyclone track prediction based coevolution of Elman RNNs for the South Pacific where the latitude and longitude were treated as separate dimensions. A similar approach was used for the prediction of wind intensities [10].

III. Problem Definition and Methodology

In this section, we identify the minimal timespan prediction problem and give details of the model that will be used to analyse the problem. We use Elman RNN as prediction model and two distinct training algorithms that include back-propagation through time and cooperative neuro-evolution.

A. Problem Definition: Minimal Timespan Prediction Problem

Weather prediction involves time series prediction for natural phenomenon such rainfall prediction, cyclones, tornadoes, wave surges and droughts[18], [19]. One needs to check how fast the prediction model can make a decision when the event occurs. If the model is training over specific months for rainy seasons for a decade, the system should be able to make a robust prediction from the beginning of the rainy season. We define the *event length* as the duration of an event which can

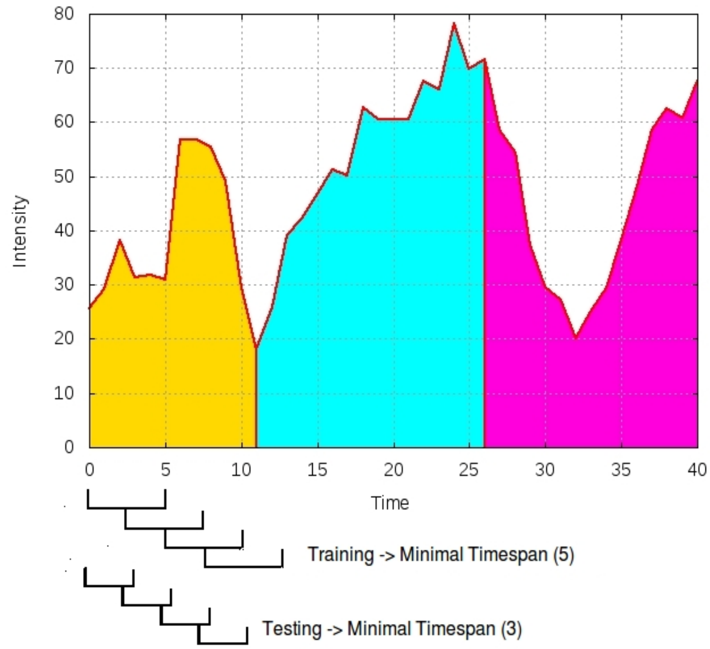


Fig. 1. Cyclone wind intensity time series showing relation between training and testing timespan.

be number of hours of a cyclone or number of days of drought or torrential rain.

In a conventional time series prediction problem, the large time series data set needs to be broken down into smaller sections or snapshots called windows, which is usually taken at regular intervals [20]. The size of the window is defined as the timespan. In the case of financial time series, there can be an issue if a prediction is made according to the division of stock market per month. When a month begins, one needs to evaluate an effective prediction model to check the number of days (data points) one needs in order for the model to make an efficient prediction.

The same problem lies when it comes to cyclones, one needs to measure how many hours after the cyclone is detected the model begin prediction regarding the track, wind or other characteristics of the cyclone. In the case of cyclones, predictions need to be made as quickly as possible in order to provide early warnings to the people so that they can get prepared. For example, data about a tropical cyclone in the South Pacific is recorded at six-hour intervals [13]. Therefore, if the timespan used is 6 data points then, the first prediction of any system used for predictions would come after 36 hours. It would take 36 hours to make the first prediction about the cyclone wind intensity. By that time, a lot of damage would have been already caused which may have been avoided if robust and accurate warnings been issued.

The problem with the existing models such as neural networks used for cyclones and related problems is that we do not know what would be the minimal time required to reach a

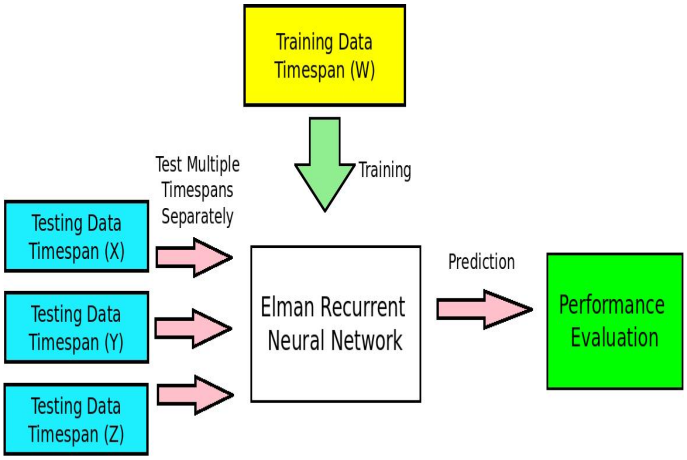


Fig. 2. Elman Recurrent neural network trained with timespan *W* and tested with *X,Y,Z* for identifying the minimal timespan.

decision about the first prediction. We introduce the problem of *minimal timespan* that defines the minimum duration needed for a model to effectively reach a prediction for a given time-series.

Figure 1 shows a portion of the wind intensity time series of tropical cyclones in the South Pacific. The event length is represented by the different color portion of the time series and it gives the duration of single cyclones. The timespan is of fixed length and moves through the time-series in a windowed motion. The movement at some points causes the timespan to overlap from one event to the other the at the point of transition of the events (cyclones). The figure shows how we extract two different timespan values from a single time-series.

Figure 2 shows the experimental method we used to identify the minimal timespan. The RNN was firstly trained to predict cyclone wind intensity using a timespan or embedding dimension *W*. Later we tested each of the fully trained network (trained with the separate timespan values) with multiple values of timespan (3,4,5,6,7,8). The predictions given by each of the timespan trained and tested with was analyzed to identify the overall minimal timespan.

B. Methodology: Recurrent Networks for Prediction

RNNs are dynamical systems that use states from previous time steps to compute current state; they are thus well-suited for modeling temporal sequences [4]. Elman RNNs use a context layer to compute the new state from the previous state and current inputs. The basic components of an observed dynamical system are represented in an Elman network using the input, context and the output layer [21].

The change of the hidden state neurons' activation in Elman RNNs [4] is given by Equation (1).

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

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.

Figure 3 shows the Elman recurrent neural used for cyclone wind intensity prediction where *D* represents the embedding dimension. Input data is preprocessed and it is fed to the RNN at single time-steps up till the size of the timespan being used is reached after which the wind intensity is predicted.

The performance of the two methods is measured using the root mean squared error (RMSE) and the mean absolute error (MAE) as given in Equation 2 and 3, respectively.

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

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|(y_i - \hat{y}_i)| \quad (3)$$

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

We employ two distinct algorithms for training the given RNN. These include, 1) cooperative neuro-evolution and 2)back-propagation through time which are described in detail in the sections to follow.

C. Training Algorithm: Cooperative Neuro-evolution

Cooperative coevolution (CC) is an evolutionary algorithm [22] that has been used to train neural networks and also known as cooperative neuro-evolution [3]. Cooperative neuro-evolution employs problem decomposition methods that divide a neural network into subcomponents and evolves them [23], [24].

Cooperative neuro-evolution (CNE) has given promising results for training recurrent neural networks for time series problems [3]. We employ neuron level problem decomposition [24] that has shown to be effective for time series problems whereby each node in the hidden, context and output layer of the recurrent neural network is regarded as a subcomponent as shown in Figure 3 [3]. Each subcomponent is implemented as a sub-population that in principle can employ any evolutionary algorithm. During evolution, all the sub-populations are evolved for a fixed number of generations in a round-robin fashion. Cooperation takes place for fitness evaluation when the best individual from the respective sub-populations are concatenated and then mapped into the recurrent neural network that provides the error (RMSE) which becomes the fitness. This process is repeated until a fixed number of evaluations is reached as given in Algorithm 1.

D. Training Algorithm: Backpropagation Through-Time

Backpropagation through time (BPTT) is a gradient descent based algorithm that is most widely used for training RNNs [25]. The algorithm unfolds a recurrent neural network in time into a deep multilayer feedforward network and employs the error back-propagation for weight update. When unfolded in time, the network has the same behavior as a recurrent neural network for a finite number of time steps. Algorithm 2

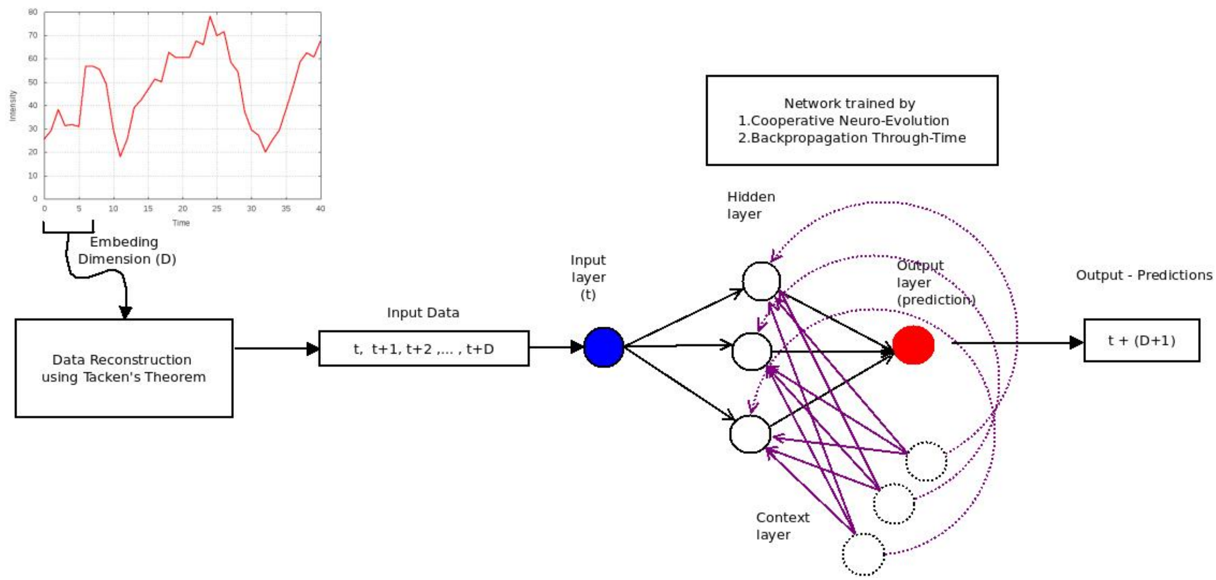


Fig. 3. Elman recurrent neural network used for tropical cyclone wind intensity prediction.

Algorithm 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

Algorithm 2: Backpropagation Through-Time for Training Elman RNNs

Step 1: Prepare Training and Testing dataset using Taken's theorem
Step 3: Initialize the RNN weights with small random numbers in range [-0.5, 0.5]
for each *Epoch* until termination **do**
  for each Sample **do**
    for n Time-Steps **do**
      Forward Propagate
    end for
    for n Time-Steps **do**
      i) Backpropagate Errors using Gradient Descent
      ii) Weight update
    end for
  end for
end for

shows the BPTT algorithm which was used to train the Elman recurrent neural network. An *epoch* is referred to a complete cycle through all the sets of input and output data.

IV. Experiments and Results

In this section, we provide the details of the experimental design and results where RNNs are training using cooperative neuro-evolution (CNE) and back-propagation through-time (BPTT) for the identified minimal span problem. We focus on minimal timespan for tropical cyclone wind intensity prediction as a case study.

In the testing stage, we pre-process the test dataset using different values of the timespan. In this way, we evaluate the effectiveness of the trained RNN for generalization performance on different values of timespan from which only value has been used during training.

A. Data Preprocessing and Reconstruction

We use Taken's theorem [26] to reconstruct the time series data into a 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. The

RNN unfolds k steps in time which is equal to the embedding dimension or timespan D [3], [27], [28].

We use tropical cyclone intensity data from the Southern Pacific region [13]. 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 the data set. All the cyclones in both the training and testing dataset were all concatenated into a single data stream to form the complete time series. Cyclones were placed consecutively in the data set based on their date of identification in ascending order.

B. Experimental Design

The sub-populations in cooperative neuro-evolution employ the generalized generation gap with parent-centric crossover (G3-PCX) evolutionary algorithm [29]. We use the population size of 200 with 2 parents and 2 offspring that has shown good results in previous work [3]. In the case of BPTT, we employ a learning rate of 0.2. In the case of cooperative neuro-evolution, we provide results for 3 hidden neurons as this showed optimal results in previous work [10]. We train with different numbers of hidden neurons in the case of BPTT and then present the case that gives best results to test different minimal timespan.

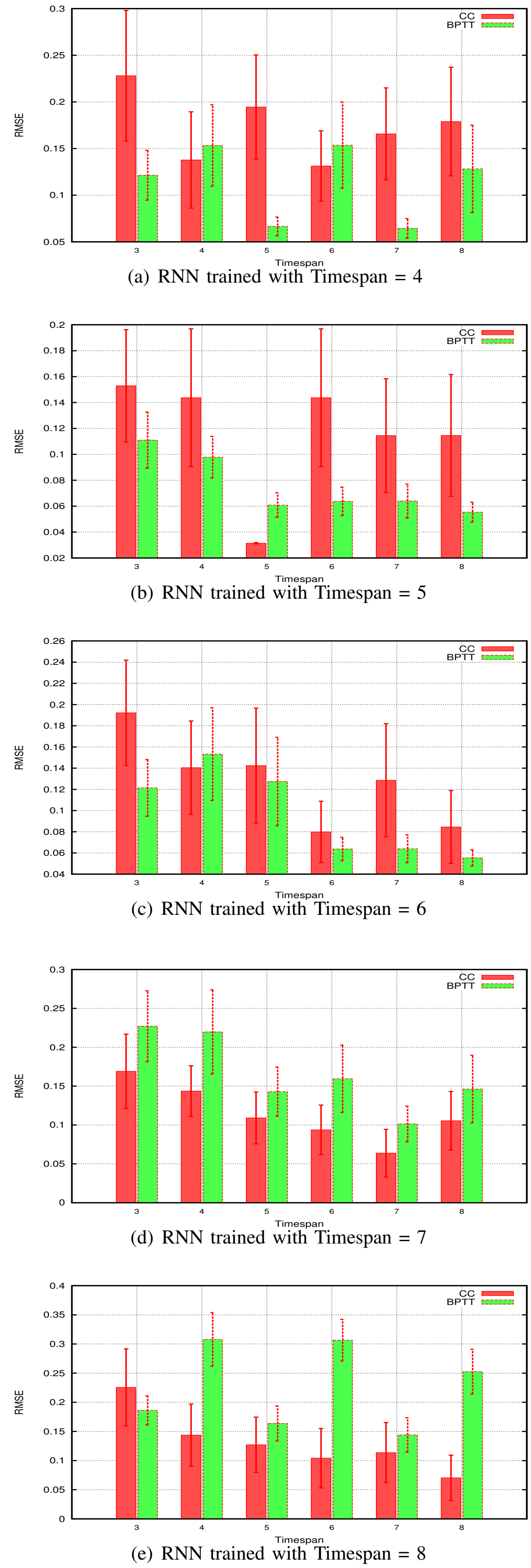
The original dataset comprised of cyclones from the past decades where each data-point was recorded at regular six-hour intervals. We reconstructed the data in order to test the effectiveness of the model for prediction within 18 hours (timespan of 3) and up to 48 hours (timespan of 8). Figure 2 gives more details of the experimental set up used. The neural network was trained with timespan *W* and tested with timespan *X,Y,Z*.

The implementation of both the designated training algorithms and the cyclone dataset are given *Smart Bilo: An Open Source Computational Intelligence Framework* [30].

C. Results

Figure 4 shows the performance CNE and BPTT on the testing data sets for the varying timespan values from the cyclone data. Each point in the bar-graph (CNE and BPTT) gives the performance of the RNN that had been tested with timespan ranging from 3 up to 8 with increments of 1. The 95% confidence interval reported by RMSE of 30 independent experimental runs is given by an error bar.

The sub-figure 4(a), 4(b) , 4(c), 4(d) and 4(e) are used to test the robustness of the CNE and BPTT training algorithms for evaluating the minimal timespan. We compare the performance of training algorithms with respect to the varied timespans ($TS \in \{4, 5, 6, 7, 8\}$) used in training. Figure 4 (b) has achieved the best performance which is given by the minimum error. The timespan of 5 has shown the best performance in testing dataset given the RNN was trained with timespan of 5. The best performance was given when the minimum timespan for testing dataset was the same for the training dataset. Similar trends were seen with all the other cases of the timespan that were used in training; except for *TS*4 as shown in Figure 4(a). We highlight the case for *TS*4, where there was only 0.006



(a) RNN trained with Timespan = 4

(b) RNN trained with Timespan = 5

(c) RNN trained with Timespan = 6

(d) RNN trained with Timespan = 7

(e) RNN trained with Timespan = 8

Fig. 4. Performance of CNE and BPTT in wind intensity prediction in the testing data set (2006 -2013) for tropical cyclones in the South Pacific.

TABLE I
BEST PERFORMANCE OF COOPERATIVE COEVOLUTION

TS(training)	TS (testing)	RMSE (Test)	MAE (Test)
4	6	0.1312 ± 0.0378	25.64 ± 7.749
5	**5**	**0.0314 ± 0.0005**	**4.962 ± 0.082**
6	6	0.0798 ± 0.0290	15.06 ± 6.153
7	7	0.0637 ± 0.0307	11.27 ± 6.088
8	8	0.0704 ± 0.0389	11.68 ± 6.739

difference between timespan 4 and 6. Therefore, we could still generalize that the same timespan used for training and testing provide the best performance.

The performance of CNE was able to beat BPTT for higher Timespans (TS [7 and 8]). CNE also showed good prediction accuracy for Figure 4.(b) and 4.(c) when the training timespan was same as the timespan tested that is 7 and 8 respectively. In lower timespans (TS [4, 5, 6]), BPTT had shown better performance.

Figure 6 gives the performance for a single run of the CNE together with the error in prediction. The initial 100 data points are shown for clear visualizations. The timespan of 5 is compared with timespan 6 and 7. We only used timespan 5, 6 and 7 for visualization purposes as timespan 5 showed most promising performance as seen in Figure 4(b).

Table I summarizes the best performance of CNE. As shown by the RMSE and MAE, the best possible value for both training and testing timespan is 5 as it has the least error.

D. Discussion

We defined minimal timespan as the least possible number of data points or the smallest window size necessary for time-series prediction. The results, in general, reveal that the minimal timespan is an important feature to test the robustness of the prediction model and the training algorithm. Cyclone wind-intensity prediction was used as it needs robust prediction model, however, other applications can also be explored to identify the minimal timespan problem.

In terms of the training algorithm, CNE was able to outperform BPTT for the higher timespan. CNE works towards dividing a larger problem into smaller components and solving them. The neural network gets larger in size with large timespan as the RNN unfolds longer in time to cater for the increased number of inputs. Figure 5 shows the comparison of the size of unfolded RNN. Timespan *TS(4) and TS(7)* are shown where it is evident that larger timespan unfolds into a larger network in time. Therefore, training the larger neural network is well suited for CNE as it is an evolutionary algorithm and the weight updates are done according to fitness of the entire network and not through gradients as in the case of BPTT. The results demonstrated that for *TS(7) and TS(8)*, CNE outperformed BPTT. This is due to the difficulty of BPTT in back-propagating errors as the size of the network that unfolds in time gets larger. As shown in the results, in the cases of smaller timespan, *(TS4, TS5, and TS6)*, BPTT performs better than CNE.

We found that training and testing timespan need to be same for best prediction performance. This shows that the RNN's we trained were unable to generalize well for the different timespan tested which implies that our training methodologies were not robust enough. This reaffirms that the choice of timespan as a good measure of robustness for the training algorithms and the prediction model. The challenge in future research is to develop a strategy that is able to give good prediction performance regardless of the size of the timespan in the testing dataset.

The results showed that the minimal timespan TS(5) gave the best performance. This implies that prediction of the model can take place within 30 hours from the identification of the cyclone. Since readings are taken every 6 hours, timespan of 5 is same as 30 hours from the beginning of the cyclone.

V. Conclusion

In this paper, we identified the minimal timespan problem for robust time series prediction with application to cyclone wind intensity. The minimal timespan has been defined as the least possible window size necessary to begin time-series prediction. Back-propagation through time and cooperative neuro-evolution algorithm were used to train Elman RNNs to find out the effect of minimal timespan.

According to the results, the minimal timespan is an important characteristic for robust time series prediction. The minimal timespan would be useful in training a RNN models that would be able to enhance predictions in future cases of cyclones.

According to the cyclone data, the data points were collected at six-hour intervals, we could predict cyclone wind intensity quite accurately after 30 hours from the start of the cyclone. This can enable better preparation for the cyclone, therefore, reducing the damages caused. The minimal timespan is of paramount importance when it comes to problems that require faster prediction as seen with cyclones. The problem of minimum timespan exists in a wide range of applications, especially in engineering problems that rely on intelligent decision making based on minimal data readings by sensors.

In future work, multi-objective and multi-tasking methods could be used for the minimal timespan problem. Further applications in other problems such as rainfall and those that require fast seasonal prediction at beginning of event such as earthquakes can be explored.

References

[1] F. E. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *Omega*, vol. 29, no. 4, pp. 309–317, 2001.

[2] 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.

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

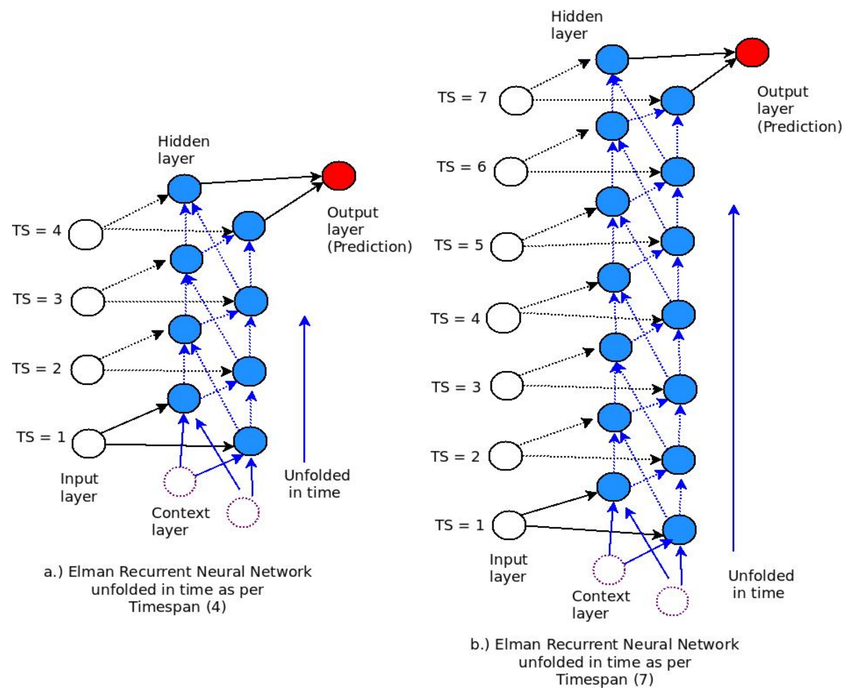
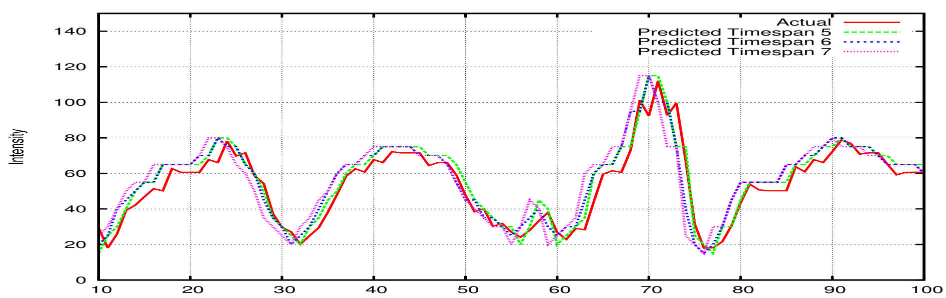
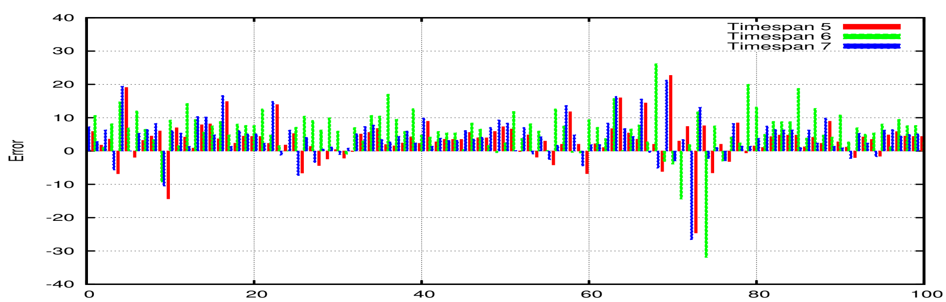


Fig. 5. Unfolded view of RNN.



(a) Performance of CNE on testing dataset



(b) Error in prediction by CNE

Fig. 6. Performance of CNE for a single experimental run

[4] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179–211, 1990.
[5] K. Emanuel, "Increasing destructiveness of tropical cyclones over the past 30 years," *Nature*, vol. 436, no. 7051, pp. 686–688, 2005.
[6] J. A. Knaff, M. DeMaria, C. R. Sampson, and J. M. Gross, "Statistical, five-day tropical cyclone intensity forecasts derived from climatology and persistence," *Weather Forecasting*, vol. 18, pp. 80–92, 2003.
[7] D. M. and J. Kaplan, "An updated statistical hurricane intensity prediction scheme (ships) for the atlantic and eastern north pacific basins," *Weather Forecasting*, vol. 14, pp. 236 – 337, 1999.
[8] *A Neural Network Regression Model for Tropical Cyclone Forecast*, 2005.
[9] 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.
[10] 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 Computtaion*, Sendai, Japan, May 2015, pp. 1784–1791.
[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] L. Sacchi, C. Larizza, C. Combi, and R. Bellazzi, "Data mining with temporal abstractions: learning rules from time series," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 217–247, 2007.
[13] (2015) JTWC tropical cyclone best track data site. [Online]. Available: http://www.usno.navy.mil/NOOC/nmfc-ph/RSS/jtwc/
[14] 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.
[15] 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.
[16] 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
[17] 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
[18] E. N. Lorenz, "Empirical orthogonal functions and statistical weather prediction," 1956.
[19] L. F. Richardson, *Weather prediction by numerical process*. Cambridge University Press, 2007.
[20] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Knowledge and Information Systems*, vol. 3, no. 3, pp. 263–286, 2001. [Online]. Available: http://dx.doi.org/10.1007/PL00011669
[21] M. Samuelides, "Neural identification of controlled dynamical systems and recurrent networks," in *Neural Networks*. Springer, 2005, pp. 231–287.
[22] 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.
[23] F. Gomez, J. Schmidhuber, and R. Miikkulainen, "Accelerated neural evolution through cooperatively coevolved synapses," *J. Mach. Learn. Res.*, vol. 9, pp. 937–965, 2008.
[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] 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.
[26] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980*, ser. Lecture Notes in Mathematics, 1981, pp. 366–381.
[27] 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.
[28] 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.
[29] 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.
[30] "Smart bilo: Computational intelligence framework," accessed: 02-12-2015. [Online]. Available: http://smartbilo.aicrg.softwarefoundationfiji.org