Cyclone Track Prediction with Matrix Neural Networks

Yanfei Zhang * ,Rohitash Chandra †‡, Junbin Gao *
* Discipline of Business Analytics, Business School, The University of Sydney, NSW 2006, Australia
† Centre for Translational Data Science, The University of Sydney, Australia
‡ School of Geosciences, The University of Sydney, NSW 2006, Australia

***Abstract*—Although machine learning and statistical methods have been extensively used to study cyclones, the prediction of cyclone trajectories remains a challenging problem. Matrix neural networks have the ability of handling spatial correlations in the data which made them suitable for image recognition tasks. Cyclone trajectories are defined by the latitude and the longitudes coordinates as a temporal sequence. Matrix neural networks are suitable for track prediction as the dataset can be conveniently given as input without vectorization that could result in loss of correlation of the spatial information. In this paper, matrix neural networks are used for cyclone track prediction for the South Indian Ocean. The results show that matrix neural networks have the ability to preserve spatial correlation that empower them to make a better prediction when compared to prominent recurrent neural network architectures.**

***Index Terms*—Cyclone trajectory prediction, meteorology, matrix neural networks, recurrent neural networks**

I. Introduction

Historically, the key methods for cyclone track (trajectory) prediction or forecasting were based on statistical methods [1], [2]. Initially, Neumann [3] utilised regression models with higher order polynomials cyclone track prediction [4]. Vickery *et.al* [5] applied multiple regression models to study different factors that would influence the probability of Hurricanes alongside the American western coastal line. The critical prediction target was the risk of reaching the landfall. Hall and Jewson [6] presented a model that considered non-parametric models for modelling the landfall risk and wind intensity. They featured all the cyclones instead of just selecting those that only reached landfall for training and gained improvement in performance over their counterparts [5].

Apart from statistical methods, deterministic models about the cyclone vortex structure [7] have been used for forecasting. DeMaria presented track forecasting with global, general circulation model governed by barotropic vorticity model [8]. Mohanty used deterministic models for track forecasting cyclone tracks for the bay of Bengal [9]. Goerss showed that a simple ensemble average or consensus forecast derived from a combination of different models are more accurate than the forecasts of the individual models [10]. A similar strategy was also shown by Weber for deterministic models that featured a statistical analysis of the annual performance of numerical track prediction models given the storm structure, location, and motion [11]. Furthermore, Roy and Kovordnyi presented a comprehensive review of the methods used for studying cyclones where they outlined that meteorological stations are shifting from traditional statistical methods to machine learning with neural networks [2].

Machine learning methods were gradually introduced as reliable methods for cyclone track and wind-intensity prediction. Lee and Liu [12] first proposed the track pattern classification by neural oscillatory elastic graph matching model. They also focused on track prediction with radial basis function neural networks for tropical cyclones in Oceania. Ali *et.al* [13] utilised the conventional feedforward neural network for cyclone track prediction in the Indian Ocean. The track distance was measured in degrees to fully capture the curvature of the global surface. Ali *et.al* [14] used conventional feedforward neural networks with tropical cyclone heat potential which is an important factor that influences the cyclone intensity. Zhang *et.al* [15] investigated whether the cyclone trajectories in West Pacific would have re-curvature (divert before landing) through reconstructing it as a classification problem with decision trees. Kordmahalleh *et.al* [16] used recurrent neural networks (RNNs) optimised by genetic algorithm (GA) to predict the American Hurricane path since 1851. Chandra *et.al* [17] [18] employed neuro-evolution of RNNs for cyclone path prediction for the South Pacific Ocean. Hong *et.al* [19] used convolutional neural networks (CNN) [20] to analysis the hyper-tensional satellite images to get the cyclone tracks in North Pacific. Cyclone path and wind-intensity as a joint problem can be considered as multi-variate time-series prediction. RNNs have shown to be well-suited for cyclone path prediction [17] [18]. Although their results were promising, the cyclones from different regions were stacked next to each other in the dataset. This gives biased prediction due to the spatial bias, i.e. the different starting locations (different magnitude of values in longitude and latitude) of various cyclones. Hence, we need to consider machine learning methods used for image recognition that preserve spatial correlations.

In image recognition tasks, it is important to preserve the spatial correlation among the pixels in an image. Therefore, one cannot simply transform or vectorize the input image simple as a vector for canonical feedforward neural networks. Therefore, convolutional neural networks (CNNs) [20] were developed that has achieved high level of success in a wide range of applications [21], [22]. However, CNNs preserve spatial correlation through its filtering and pooling layers that implicitly implements feature extraction. Alternately, more recently, matrix neural networks (MNNs) have been proposed for image recognition tasks [23]. MNNs take matrices directly as inputs, where each neuron processes information through bilinear mapping from lower layer units in exactly the same way as the canonical feedforward networks. Back-prorogation has been used to train MNNs which have shown to have very promising performance when compared to CNNs. Furthermore, MNNs can be more naturally extended for other domains of problems that have instances of data in matrix form.

In the case of prediction of cyclone trajectories, one relies on the trajectory information which is defined by the latitude and the longitude as a temporal sequence. Other features, such as the wind-intensity are also given. The features of matrix neural networks make them suitable for the case of cyclone trajectory prediction as the dataset for each of the cyclones can be conveniently given as input without vectorization that could result in loss of correlation of the spatial information which is similar to image recognition.

In this paper, MNNs are used for prediction of cyclone trajectories for the South Indian Ocean. We employ selected prominent RNN architectures as baseline models for comparison of the results with MNNs. This includes Elman RNNs [24], Long Short-Term Memory networks (LSTM) [25] and Gated Recurrent Unit networks (GRU) [26].

The remainder of this paper is organised as follows: Section II presents the data and methodology. Section III presents experiments and results that is followed by a discussion in Section IV. Section V provides the conclusion with a discussion of future research directions.

II. Methodology and Application

A. Data Pre-processing

The dataset features cyclones in the North Indian Ocean [27], where each cyclone features the latitude, longitude, wind-intensity recorded at 6 hour regular intervals for the duration of the cyclone. We only consider the track information given by the latitude and longitude and discard the wind-intensity, which can be considered for analysis in future work. Each cyclone generally occurs at a unique initial location ((x_0^j, y_0^j)) and duration given by the total time (time steps T_j). The unique initial location accounts spacial bias in the dataset. This is due to the way the latitude and longitude given as coordinates describe the cyclone track. For instance, if we consider two cyclones that have the same track pattern at different initial locations, due to the spatial bias, these cyclones would be interpreted as different cyclones by the model.

In order to remove the spacial bias, we reconstructed the trajectories by a phase-shift where all cyclones start from the same location (0,0). In other words, we are more concerned about the trajectory of the cyclone rather than the location at which it occurred. This is implemented by the following translation on the original data $(\mathbf{x}^j, \mathbf{y}^j)$ $(j \geq 0)$:

$$(x_t^j, y_t^j) := (x_t^j - x_0^j, y_t^j - y_0^j),$$

where $(\mathbf{x}^j, \mathbf{y}^j)$ indicates longitude and the latitude in the j-th cyclone respectively. These are column vectors that span $\mathbb{R}^{T_j}$. T_j indicates total number of time steps in the j-th cyclone, and t is the time step indicator in one cyclone, while the (x_0^j, y_0^j) is the initial point of the j-th cyclone.

The matrix neural network for track prediction will consider an autoregressive or sliding-window based prediction framework. This is also known as embedding of reconstruction of time series data using Taken's theorem [28]. Taken's theorem states that a time series reconstructed with overlapping windows of size α taken at regular intervals β (1 in our case) features the characteristics of the original time series. Given that cyclone track is given as bivariate sequences, each cyclone trak after embedding would be a matrix of size $(T_j - \alpha) \times (\alpha \times 2)$. Each data point contains two features (x_t, y_t), which correspond to the coordinates (latitude and longitude). Hence, for each cyclone, the input data $\mathbf{m}^j$ features $\mathbb{R}^{(T_j-\alpha)\times\alpha\times 2}$, and the corresponding output data $\mathbf{n}^j$ continues from the α-th data point in $\mathbb{R}^{(T_j-\alpha)\times 2}$.

The input data matrix $\mathbf{m}_t^j$ and output $\mathbf{n}_t^j$ are given as follows $(t \geq 0)$

$$\mathbf{m}_t^j = \begin{bmatrix} x_{t+1}^j & y_{t+1}^j \\ x_{t+2}^j & y_{t+2}^j \\ \vdots & \vdots \\ x_{t+\alpha}^j & y_{t+\alpha}^j \end{bmatrix} \quad \text{and} \quad \mathbf{n}_t^j = \begin{bmatrix} x_{t+(\alpha+1)}^j \\ y_{t+(\alpha+1)}^j \end{bmatrix}. \tag{1}$$

where, t represents the time (at regular intervals) for the data point.

B. Matrix Neural Networks

1) Feedforward Neural Networks: We first describe conventional feedforward neural networks (FNNs) that are known as universal function approximators which makes them appropriate for regression and time series prediction [29]. FNNs typically have one or more hidden layer(s) and an output layer. Each layer features neurons (represented as a vector) that are interconnected with the weights that are represented as a matrix. The mapping between the different layers of FNNs are defined by the following

$$\mathbf{x}^{l+1} = \sigma(\mathbf{W}^l \mathbf{x}^l + \mathbf{b}^l) \tag{2}$$

where, $\mathbf{x}^l$ is the input at layer l as a column vector, $\mathbf{x}^{l+1}$ is the output given by layer l (also a column vector), $\mathbf{W}^l$ is the fully connected weights and $\mathbf{b}^l$ is the bias at the given layer l. σ serves as an activation function which is normally a sigmoid. Equation (2) shows information processing from input to hidden layer(s) which involves the multiplication of a weight matrix $\mathbf{W}^l$ with column vector $\mathbf{x}$. Note that $\mathbf{x}$ represents neurons in input or hidden layer(s) depending on the number of hidden layers. The equation below shows an expanded representation of Equation (2).

$$\begin{bmatrix} x_{1j}^{l+1} \\ x_{2j}^{l+1} \\ \vdots \\ x_{pj}^{l+1} \end{bmatrix} = \sigma \left(\begin{bmatrix} w_{11}^{l} & w_{12}^{l} & \cdots & w_{1n}^{l} \\ w_{21}^{l} & w_{22}^{l} & \cdots & w_{2n}^{l} \\ \vdots & \vdots & \vdots & \vdots \\ w_{p1}^{l} & w_{p2}^{l} & \cdots & w_{pn}^{l} \end{bmatrix} \begin{bmatrix} x_{1j}^{l} \\ x_{2j}^{l} \\ \vdots \\ x_{nj}^{l} \end{bmatrix} + \begin{bmatrix} b_{1j}^{l+1} \\ b_{2j}^{l+1} \\ \vdots \\ b_{pj}^{l+1} \end{bmatrix} \right)$$

2) Matrix Neural Networks: Any problem that requires spatial correlation, such as those that involve coordinates in cyclone tracks, would lose information when the inputs are vectorized which is required by conventional FNNs. Therefore, matrix neural networks (MNNs) are more suitable in such types of problems [23].

MNNs have the ability to feature the input as a matrix instead of a vector. The update rule from one layer to another is also similar to FNNs

$$\mathbf{X}^{l+1} = \sigma(\mathbf{W}^l \mathbf{X}^l \mathbf{V}^{lT} + \mathbf{B}^l) \tag{3}$$

where, $\mathbf{X}^l$ is the input at layer l as a matrix, $\mathbf{X}^{l+1}$ is the output from layer l also in a matrix, $\mathbf{W}^l$ and $\mathbf{V}^l$ are the fully connected weights that updates both column and row of the original input unit (matrix), and $\mathbf{B}^l$ is the bias term, and σ also as an activation function, whose mapping can be intuitively explained by Figure 1.

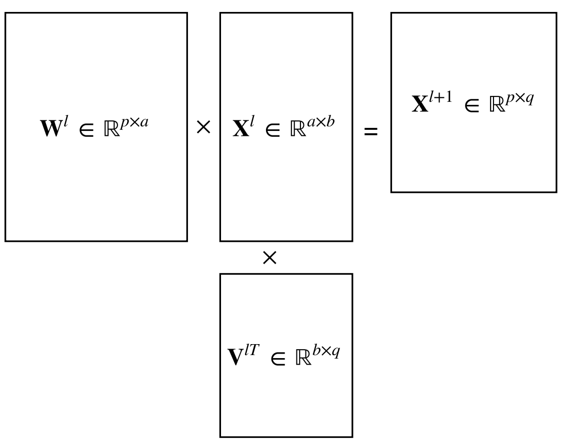

Fig. 1: Schematic of the internal representation of a matrix neural network showing information processing from input to output layer, which involves a matrix representation of input data as opposed to vector-based input in conventional FNNs. Further details were given in Equation (3)

Figure 1 shows a schematic representation of the input matrix $\mathbf{X}_i^l$ and the output matrix $\mathbf{X}_i^{l+1}$ for ith window (for time t to $t+\alpha$) that features latitude and longitude at time of a given cyclone. Note that dimensions changed from $a \times b$ to $p \times q$ for $\mathbf{X}_i^{l+1}$, which is achieved by *dot product* with respect to the dimension of the input matrix $\mathbf{X}_i^l$.

Given the definition of the cyclone tracks, the input of MNN $\mathbf{m}^j \in \mathbb{R}^{(T_j-\alpha)\times\alpha\times 2}$ is a 3-dimensional tensor, and the corresponding output is $\mathbf{n}^j \in \mathbb{R}^{(T_j-\alpha)\times 2}$, i.e., the number of sliding windows for both input and output data is $(T_j - \alpha)$, and each input unit $\mathbf{m}_t^j$ is of size $\alpha \times 2$, while the output unit $\mathbf{n}_t^j$ is of size 2. In other words, we shall find the following mapping: $f: \mathbb{R}^{\alpha\times 2} \to \mathbb{R}^2$.

The goal of the learning here is to minimize the error between network output ($\hat{\mathbf{n}}^j$) and the given data ($\mathbf{n}^j$) as shown in Equation 4. Therefore, the loss function is used to compute the gradients at each layer in order to update the weight matrices as done by error backpropagation in a conventional FNNs.

$$\min_{\mathbf{W}} \sum_{j=1}^{N} L(\hat{\mathbf{n}}^j, \mathbf{n}^j) = \sum_{j=1}^{N} \frac{1}{T_j - \alpha} \|\mathbf{n}^j - \hat{\mathbf{n}}^j\|_2^2 \tag{4}$$

Figure 2 shows how the training is implemented for prediction of cyclone tracks with MNNs. The original data is reconstructed by Taken's theorem into the input matrix m^j for each cyclone. The cyclones are stacked next to each other, as training and test dataset. The MNN is used to propagate the information from the input matrix to the output layer where the loss function is used to compute the gradients at each layer. The gradients are used to compute the weight update as in a conventional FNN. Further details for backpropagation in MNNs is given in [23].

III. Experiments and Results

In this section, we present experiments and results for cyclone track prediction using data from past few decades in the South Indian ocean. We use matrix neural networks and compared with prominent recurrent neural network architectures which are well suited for time series prediction.

A. Data Description

The dataset for cyclone trajectory prediction is obtained from Joint Typhoon Warning Centre (JTWC) [27] featuring the cyclones occurred in the South Indian ocean from 1985 to 2013. The dataset featured 286 cyclones in total that is separated into the training set and testing set. The training set features the first 70% of the cyclones and the testing set features the remaining 30%. The distribution of cyclone length or duration for both training set and testing set are given in Figure 3 and Figure 4. We notice that majority of the cyclones duration lie between 20 - 40 time points where each time point represents 6 hours. More precisely, the dataset shows that each data point was recorded at 600 hrs, 1200 hrs, 1800 hrs, and 24:00 hrs. The number of the data points in one cyclone ranged between 6 to 129. The maximum duration of cyclones in training set is about 90 time points while the testing set features about 120 time points. The cyclones featured from South Indian Ocean are displayed on the Figure 5.

B. Design of Experiments

The evaluation criteria is root mean square error (RMSE), where lower values indicate better performance.

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^{N} \left(\frac{1}{T_j - \alpha} \|\mathbf{n}_{test}^j - \hat{\mathbf{n}}_{test}^j\|_2^2\right)}$$

where the $\hat{\mathbf{n}}_{test}^j$ is the prediction of testing input $\mathbf{m}_{test}^j$.

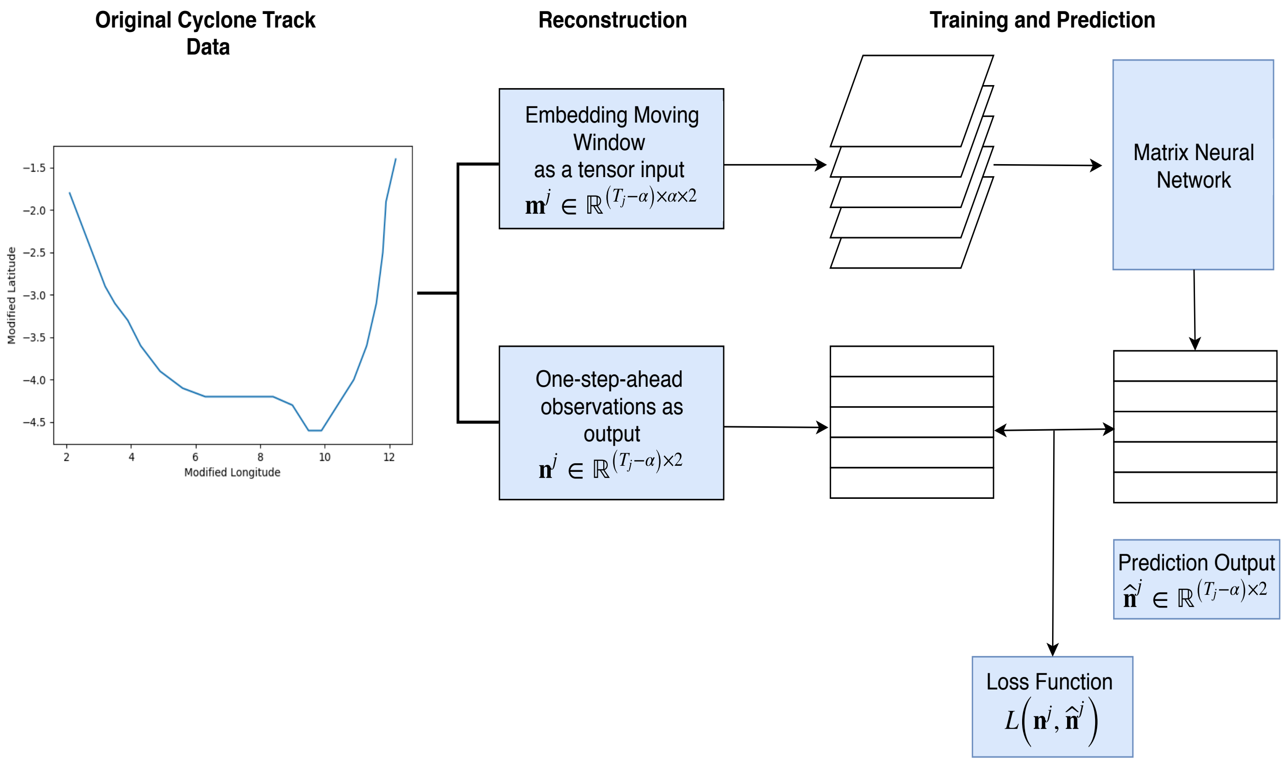

Fig. 2: The cyclone track data is reconstructed by overlapping windows $\mathbf{m}_t^j \in \mathbb{R}^{\alpha\times 2}$ that covers the observation from t to $t+\alpha$. We use the windows in a matrix form as input features to predict the subsequent 1-step ahead location (given by latitude and longitude) for the track $\mathbf{n}_t^j \in \mathbb{R}^2$

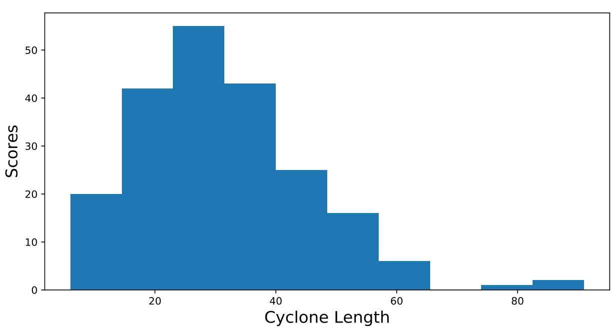

Fig. 3: Distribution of the cyclones duration (length) in training dataset. Note that each datapoint represents 6 hours.

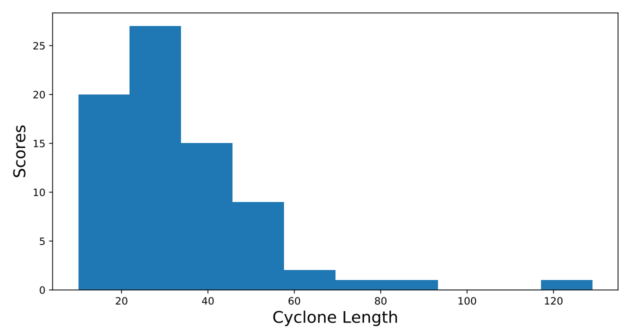

Fig. 4: Distribution of the cyclones duration (length) in testing dataset. Note that each datapoint represents 6 hours.

We used window size $\alpha = 4$ and time lag $\beta = 1$ for data reconstruction which we determined from trial experiments as most appropriate for the experiments. We used the 3 prominent recurrent neural architectures e (Elman RNN [24], LSTM [25], GRU [26]) for comparison of results given by MNN. Table I gives the details of the respective RNN topologies, with number of neurons in input, hidden and output layer. The same training and testing data used for MNN was used for the RNNs. The only difference was the presentation of the data which was vectorized for RNNs, as opposed to matrix input format in MNNs.

The matrix neural network is implemented directly with TensorFlow [30], while the RNN methods were implemented using Keras which is a Python based machine learning library [31] that also uses TensorFlow. The experiments were conducted on a laptop with Intel i7-6500U processor and memory of 8 gigabytes.

C. Results

The results of the prediction accuracy for the respective methods are shown in Table II. Note that the mean and 95 % confidence interval for 30 experimental runs with different

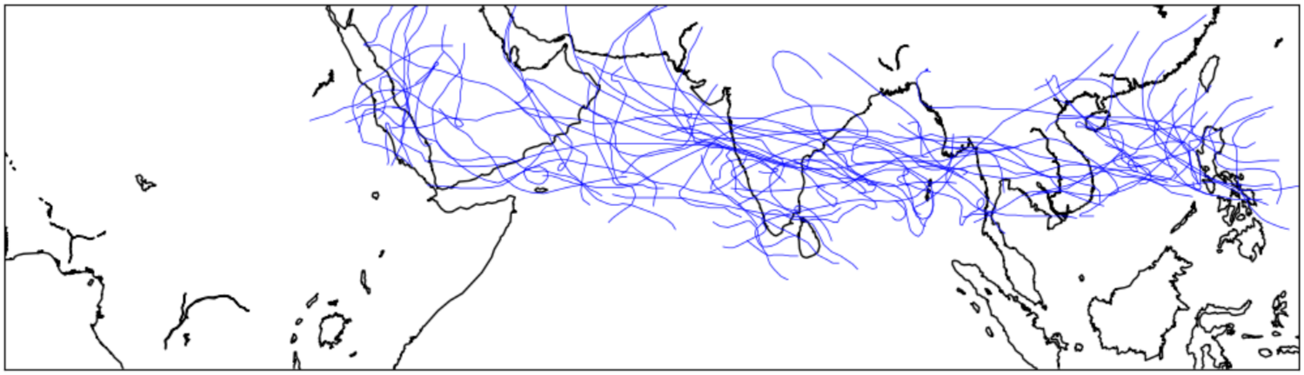

Fig. 5: Cyclone tracks in the South Indian ocean

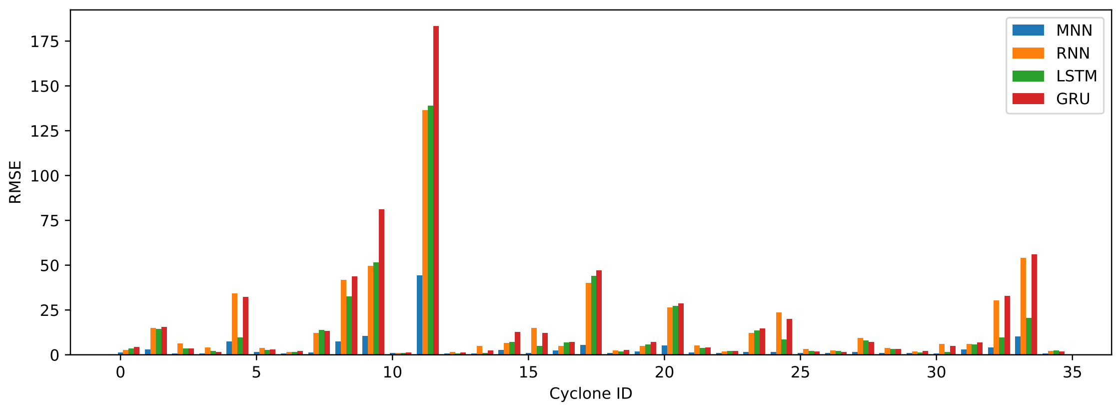

Fig. 6: RMSEs for the first to the 35-th cyclone predictions from MNN and other baseline models

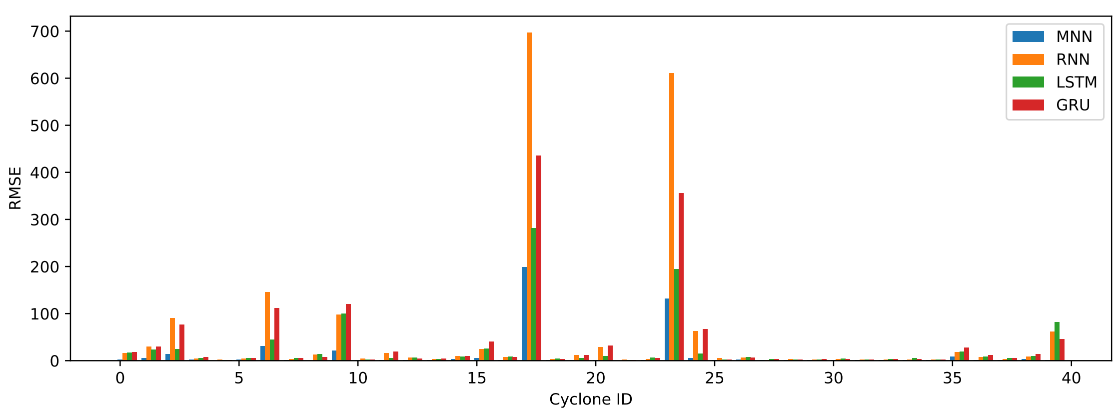

Fig. 7: RMSEs for the 36-th and onward cyclone predictions from MNN and other baseline models

Layer	Row	Column
Input Layer	α	2
Hidden Layer 1	25	8
Hidden Layer 2	50	10
Output Layer	1	2

TABLE I: Recurrent neural network experiment design

initial positions in weight space are shown. The results clearly show that MNN provides the best prediction accuracy (least RMSE) when compared with the other methods. Next, we select two of the cyclones and show the prediction performance of the respective methods.

Method	Testing RMSE
MNN	15.41±2.48
Elman RNN	39.91±3.12
LSTM	36.87±8.41
GRU	42.83±9.14

TABLE II: Performance of different methods on the test dataset

We select several representative cyclone predictions that have good performances, and compare them with respect to the baseline methods (RNN, LSTM, GRU) for further references (All cyclones mentioned are included in the appendices).

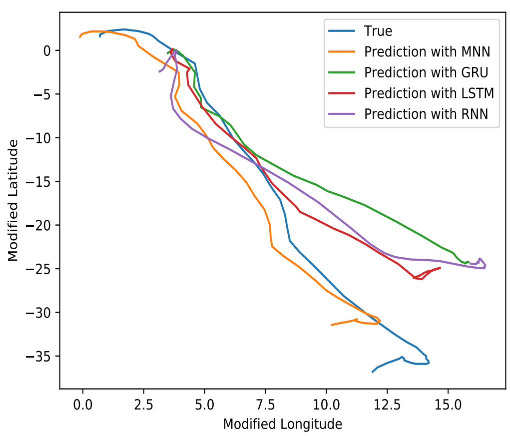

Fig. 8: 38-th Cyclone in Testing Set

Figure 8 shows the prediction performance of the respective method for the 38-th cyclone in the testing set. We can observe that the respective methods gave more accurate predictions in the first halt of the cyclone. In the second half, MNN was closest to the true cyclone trajectory.

Figure 9 shows the prediction performance for the 8-th cyclone in the testing set, where we observe that the MNN was able to give the most promising prediction from the beginning of the cyclone track when compared with other methods. GRU, LSTM and RNN could predict trajectories that were offset with a high value when compared to the initial point of the true trajectory.

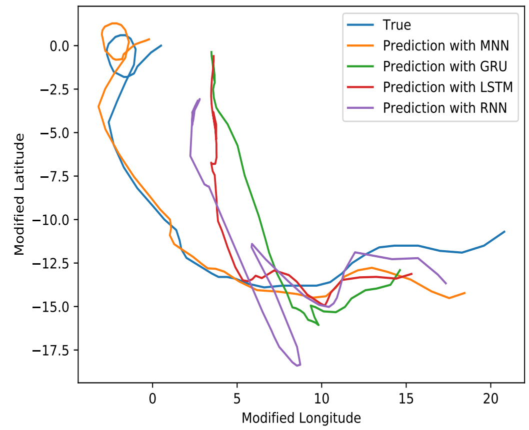

Fig. 9: 7-th Cyclone in Testing Set

For better understanding of the prediction performances given by MNN, the prediction accuracy across different methods were evaluated and visualised for each cyclone in the testing dataset as shown in Figure 6 and Figure 7. We observed high abnormally in prediction performance for 54-th and 60-th cyclones, therefore we provided further visualisation in Figure 10 and Figure 11.

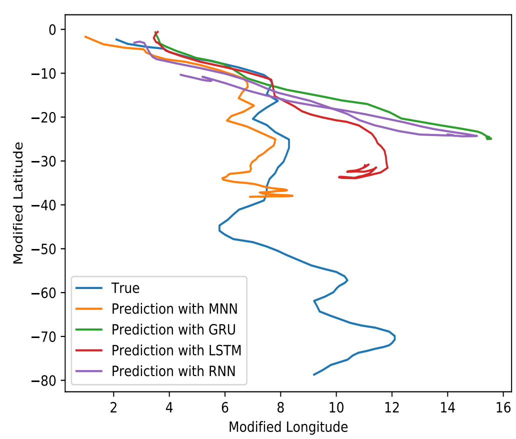

Fig. 10: 54-th Cyclone in Testing Set

Figure 10 shows the prediction performance for the 54-th cyclone in the testing set, where we observe that even no methods gave the ideal result on the cyclone prediction, at least MNN still reliably depicted this cyclone trajectory partially from the start to the middle, while other methods could not truly predict this cyclone trajectory at all.

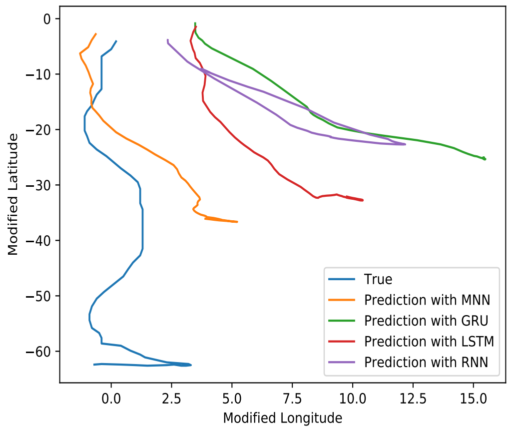

Fig. 11: 60-th Cyclone in Testing Set

Figure 11 shows the prediction performance for the 60-th cyclone in the testing set, where no method gave reliable result, but MNN did not deviate too much from the true cyclone track, while other 3 methods gave clearly outrageous results comparing with MNN.

The prediction accuracy for the cyclones shown in the respective figures are listed in Table III, which serves as further explanation mathematically.

Cyclone	MNN	RNN	LSTM	GRU
8 (Figure 9)	1.30	12.16	13.78	13.14
38 (Figure 8)	5.05	29.85	23.43	29.89
54 (Figure 10)	199.09	696.89	281.43	435.59
60 (Figure 11)	131.51	610.73	194.25	355.83

TABLE III: Prediction performance on selected cyclones

IV. Discussion

The results show that the proposed method (MNN) is most effective when compared to other methods (GRU, LSTM, FNN, RNN) given the chaotic nature of cyclone tracks as shown by the results. This is an important observation given that in the literature LSTMs and GRUs have shown to be powerful tools for a wide range of applications that includes time series problems. Howsoever, the prediction of cyclone trajectories is a special case of time series prediction that considers spatial relationships given by the latitude and longitude. Moreover, there are a number of other features, such as the distance of the cyclone to the landfall [18], the sea surface temperature [14], the pressure and wind-intensity [17]. which are not taken into account in this study. They could possibly help in achieving petter predictions.

Note that matrix neural networks are more flexible in its structure when compared with conventional FNNs. It enables us to adjust the input feature to any size and generate the output that is consistent or lower in dimension (D). In other words, the input can be of 2D for each input unit, and we can have 2D or 1D outputs. In conventional FNNs, we can only have 1D input unit and 1D output unit. 2D output units would be appropriate for multi-step ahead cyclone track prediction.

MNNs can more easily enable auxiliary features in the input while preserving the context in terms of spatial and temporal dependencies. It is believed that auxiliary features would improve the prediction [32]. For example, except for the 2 original input features (longitude, latitude), we can also introduce more input features like wind-intensity, precipitation, surface temperature, etc. All of those auxiliary features together for prediction would create the following mapping: $\mathbb{R}^{\alpha\times 6} \to \mathbb{R}^2$, which can more easily be implemented by MNNs.

The reliability of the data from the problem domain also plays an important part in quality of the predictions. Due to the fact that some of the cyclones occurred decades ago when the data capturing instruments were not sophisticated add to the limitations. Moreover, there are inconsistencies and noise in the dataset. There were a number of older cyclones with missing data. It would be reasonable to generate more data using approaches where synthetic datasets generated that resemble real cyclones [33], [34].

V. Conclusions and Future Work

We applied matrix neural network for cyclone trajectory prediction with matrix neural network. We also removed the spatial biases in the original dataset by normalising all the cyclones with their starting locations. The results show that the proposed method achieved better performance when compared to three established recurrent neural network methods from the literature. This result improved the related recurrent neural network methods method from the literature. It seems this was due to the feature of matrix neural network that preserve spatial information from the cyclone tracks, which is an advantage over the other methods.

In future work, uncertainty quantification through the Bayesian implementation of matrix neural networks can be done for the case of cyclone trajectory prediction. Furthermore, additional features from cyclones such as sea-surface temperature, distance to shore-line, humidity and pressure can be also used. Multi-step ahead cyclone track prediction can also be considered using matrix neural networks.

References

[1] C. J. McAdie and M. B. Lawrence, "Improvements in tropical cyclone track forecasting in the atlantic basin, 1970–98," *Bulletin of the American Meteorological Society*, vol. 81, no. 5, pp. 989–997, 2000.

[2] C. Roy and R. Kovordnyi, "Tropical cyclone track forecasting techniques a review," *Atmospheric Research*, vol. 104-105, pp. 40 – 69, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169809511002973

[3] C. J. Neumann, *An alternate to the HURRAN (hurricane analog) tropical cyclone forecast system*. National Weather Service, 1972.

[4] G. D. Atkinson and C. R. Holliday, "Tropical cyclone minimum sea level pressure/maximum sustained wind relationship for the western north pacific," *Monthly Weather Review*, vol. 105, no. 4, pp. 421–427, 1977.

[5] P. Vickery, P. Skerlj, and L. Twisdale, "Simulation of hurricane risk in the us using empirical track model," *Journal of structural engineering*, vol. 126, no. 10, pp. 1222–1237, 2000.

[6] T. M. Hall and S. Jewson, "Statistical modeling of north atlantic tropical cyclone tracks," *Tellus A*, vol. 59, no. 4, pp. 486–498, 2007.

[7] M. Fiorino and R. L. Elsberry, “Some aspects of vortex structure related to tropical cyclone motion,” *Journal of the atmospheric sciences*, vol. 46, no. 7, pp. 975–990, 1989.

[8] M. DeMaria, “Tropical cyclone track prediction with a barotropic spectral model,” *Monthly weather review*, vol. 115, no. 10, pp. 2346–2357, 1987.

[9] U. Mohanty, “Tropical cyclones in the bay of bengal and deterministic methods for prediction of their trajectories,” *Sadhana*, vol. 19, no. 4, pp. 567–582, 1994.

[10] J. S. Goerss, “Tropical cyclone track forecasts using an ensemble of dynamical models,” *Monthly Weather Review*, vol. 128, no. 4, pp. 1187–1193, 2000.

[11] H. C. Weber, “Hurricane track prediction using a statistical ensemble of numerical models,” *Monthly Weather Review*, vol. 131, no. 5, pp. 749–770, 2003.

[12] R. S. Lee and J. N. Liu, “Tropical cyclone identification and tracking system using integrated neural oscillatory elastic graph matching and hybrid rbf network track mining techniques,” *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 680–689, 2000.

[13] M. Ali, C. Kishtawal, and S. Jain, “Predicting cyclone tracks in the north indian ocean: An artificial neural network approach,” *Geophysical research letters*, vol. 34, no. 4, 2007.

[14] M. Ali, P. Jagadeesh, I.-I. Lin, and J.-Y. Hsu, “A neural network approach to estimate tropical cyclone heat potential in the indian ocean,” *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 6, pp. 1114–1117, 2012.

[15] W. Zhang, Y. Leung, and J. C. Chan, “The analysis of tropical cyclone tracks in the western north pacific through data mining. part i: Tropical cyclone recurvature,” *Journal of Applied Meteorology and Climatology*, vol. 52, no. 6, pp. 1394–1416, 2013.

[16] M. M. Kordmahalleh, M. G. Sefidmazgi, A. Homaifar, and S. Liess, “Hurricane trajectory prediction via a sparse recurrent neural network,” in *Proceedings of the 5th International Workshop on Climate Informatics*, 2015.

[17] 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 *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015, pp. 1–8.

[18] R. Chandra and K. Dayal, “Cooperative neuro-evolution of elman recurrent networks for tropical cyclone wind-intensity prediction in the south pacific region,” in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 1784–1791.

[19] S. Hong, S. Kim, M. Joh, and S.-k. Song, “Globenet: Convolutional neural networks for typhoon eye tracking from remote sensing imagery,” *arXiv preprint arXiv:1708.03417*, 2017.

[20] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, “Handwritten digit recognition with a back-propagation network,” in *Advances in neural information processing systems*, 1990, pp. 396–404.

[21] M. Egmont-Petersen, D. de Ridder, and H. Handels, “Image processing with neural networksa review,” *Pattern recognition*, vol. 35, no. 10, pp. 2279–2301, 2002.

[22] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural networks*, vol. 61, pp. 85–117, 2015.

[23] J. Gao, Y. Guo, and Z. Wang, “Matrix neural networks,” in *International Symposium on Neural Networks*. Springer, 2017, pp. 313–320.

[24] J. L. Elman, “Finding structure in time,” *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[25] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[27] “(2015) jtwc tropical cyclone best track data site,” http://www.usno.navy.mil/NOOC/nmfc-ph/RSS/jtwc/best tracks/.

[28] F. Takens, “Detecting strange attractors in turbulence,” in *Dynamical systems and turbulence, Warwick 1980*. Springer, 1981, pp. 366–381.

[29] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[30] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv preprint arXiv:1603.04467*, 2016.

[31] F. Chollet *et al.*, “Keras,” 2015.

[32] N. F. Rajani and R. J. Mooney, “Stacking with auxiliary features,” *arXiv preprint arXiv:1605.08764*, 2016.

[33] M. James and L. Mason, “Synthetic tropical cyclone database,” *Journal of waterway, port, coastal, and ocean engineering*, vol. 131, no. 4, pp. 181–192, 2005.

[34] M. James, L. Mason *et al.*, “Generation of a synthetic tropical cyclone database,” in *Coasts & Ports 1999: Challenges and Directions for the New Century; Proceedings of the 14th Australasian Coastal and Ocean Engineering Conference and the 7th Australasian Port and Harbour Conference*. National Committee on Coastal and Ocean Engineering, Institution of Engineers, Australia, 1999, p. 381.