An Investigative Comparison between State-of-the-art Methods and Long Short-Term Memory Recurrent Neural Networks for Cyclone Path Forecasting

ARSH CHAWLA*, Australian National University, Australia
ISAAC LEONG*, Australian National University, Australia
AARON NGUYEN*, Australian National University, Australia

Accurately predicting extreme weather events like cyclones is crucial for safeguarding lives and infrastructure. One standard modern approach to this problem deploys large weather models such as general circulation models. However, these are expensive to train and use even though they provide highly accurate inferences. Recent work has achieved a similar level of precision with machine learning techniques. However, these models are still considerably large-scale. In this paper, we examine the possibility of achieving comparable results using smaller-scale machine learning models. We have employed long short-term memory neural networks (LSTM) to predict tropical cyclone tracks by regarding them as time-series input data.

When compared to state-of-the-art computationally expensive forecasting models such as the Hurricane Analysis and Forecast System (HAFS), the accuracy of our model is respectable, with the absolute error being within 104% for 12-hour-lead forecasts. Meanwhile, when compared to other modern deep learning approaches, our model appears to outperform more complicated deep learning-based models by an average of 147.5% RMSE error. Our findings demonstrate that smaller-scale machine learning models, such as LSTMs, can provide competitive accuracy while significantly reducing computational costs. This suggests a promising avenue for more efficient and accessible cyclone forecasting, opening the door to broader applications in real-time weather prediction.

ACM Reference Format:
Arsh Chawla, Isaac Leong, and Aaron Nguyen. 2024. An Investigative Comparison between State-of-the-art Methods and Long Short-Term Memory Recurrent Neural Networks for Cyclone Path Forecasting. 1, 1 (May 2024), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Weather forecasting is an important technology, providing precautionary alerts to regions that will be affected by impending disasters. Like many other natural disasters, tropical cyclones can cause severe harm to vehicles, properties, and lives. Hence, the capability to predict tracks and intensity of cyclones is important in protecting potentially-affected residents.

There are two main methods to predict tropical cyclones tracks — by numerical calculation of the dynamic weather system [8], and with machine learning models to predict the next most likely location given the historical tracks of cyclones. General circulation models (GCMs), a popular numerical weather forecasting method, are powerful tools to

*These authors contributed equally to this research.

Authors' Contact Information: Arsh Chawla, u7252428@anu.edu.au, Australian National University, Canberra, ACT, Australia; Isaac Leong, u7643339@anu.edu.au, Australian National University, Canberra, ACT, Australia; Aaron Nguyen, Aaron.Nguyen@anu.edu.au, Australian National University, Canberra, ACT, Australia.

Manuscript submitted to ACM

output the "weather" by splitting the atmosphere into grids [22]. On the other hand, machine learning technologies, such as recurrent neural networks (RNN) and matrix neural networks allow for time-series prediction which is highly applicable for cyclone track detection [4, 18, 28]. Recently, a new approach that aims to combine the two different methods above was developed by Kochkov et al. [17]. This approach combines the explain-ability of physical models and the efficiency of neural networks.

The fundamental issue with current cyclone forecasting technologies is the use of Physical Climate Models for explain-ability, which are computationally expensive [9]. These traditional climate models perform simulations based on the physical interactions between four components: atmosphere, land, ocean, and sea ice [9]. These interactions are mathematically modelled based on air temperature, pressure, density, and wind magnitude among countless other physical factors [9, 22]. The immense detail present in climate models means there must exist a trade-off between accuracy/precision and computational cost, often dictated by model resolutions [9].

RNNs are a class of deep learning models derived from artificial neural networks, trained to perform data predictions based on sequential data inputs [3]. They consist of four key components — the input layer, output layer, hidden layer, and the loss function. The characteristic layer of an RNN is its hidden layer, located between the input and output layer. The hidden layer uses memory from previous inputs to make future predictions [12]. It is this memory based prediction which gives RNNs a unique edge in time forecasting problems, such as cyclone path prediction [5].

The methodology in this research involves processing a dataset of historical cyclone tracks, and using it to train and create different RNN models. Many factors that could affect the model performance are investigated, which include but are not limited to feature selection and extraction, and hyper-parameter tuning. The trained models are evaluated numerically and graphically by comparison to cyclone tracks in the dataset. Importantly, the models are compared to state-of-the-art models to assess the feasibility of predicting cyclone tracks with an RNN model.

First, related works regarding tropical cyclone track predictions are discussed in Section 2. Then, inside Section 3, the methodology section, a detailed description of the database in provided in Section 3.1 with supplementary information about feature engineering in Section 3.1.1. Then, the model architecture is outlined in Section 3.2, followed by model training and hyper-parameter tuning in Section 3.3. The evaluation and results of the model stated in Section 4 discussing single step and multi-step predictions and comparing with state-of-the-art performance. Finally, the limitations of our work and final discussion is stated in Section 5 and 6.

2 Related Work

State-of-the-art tropical cyclone track predictions are separated into two main approaches — numerical large-scale weather models including GCMs, and machine learning prediction models [17]. Recent advances in machine learning, particularly in applications of RNNs, have enabled more accurate machine learning predictions than previous convolution-based neural networks [15] for cyclone track predictions.

HAFS, developed by the National Oceanic and Atmospheric Administration (NOAA), is a state-of-the-art multi-scale numerical model that employs components such as a high-resolution physics engine and three-dimensional ocean coupling to make predictions [20]. Dong et al. [7] quantified HAFS' SAR (stand-alone regional) performance for the 2019 Atlantic Hurricane Season. They found HAFS-SAR demonstrated track forecast improvements over GFS (Global Forecast System), HWRF (Hurricane Weather Research Forecast Model), and HMON (Hurricane Multi-scale Ocean-coupled Non-hydrostatic) across forecast time lead times [7]. A recent major advance was by Kochkov et al. [17] in which they have used a machine learning model in substitution for the previously time-consuming higher-order differential equation numerical solver, resulting in 3 to 5 orders of magnitude of reduction in the computational resources.

Pure machine learning models are separated into three groups based on the input to the model — latitude and longitude-based, satellite images, and a fusion of the two [26]. Machine learning models, RNNs in particular, perform well when predicting non-linear atmospheric systems such as tropical cyclones [26]. Recent advances in the precision of machine learning predictions are first stated in the work by Moradi Kordmahalleh et al. [19] with a sparse recurrent neural network. Alemany et al. [2] later explored this idea by using long short-term memory neural networks [13] and incorporated the first grid approach where the output space is mapped to a $1^\circ \times 1^\circ$ grid cells.

Other machine learning models have aimed to extend the feature space by providing satellite images of the storm at each time step [7, 15]. To incorporate 2-dimensional imageries, these models are based on ConvLSTM, which is an extension to the existing LSTM by replacing the fully connected neural network layer with a convolutional neural network. By using a convolutional layer, the model can retain spatial features [7].

3 Methodology

3.1 Database: IBTrACS

The International Best Track Archive for Climate Stewardship (IBTrACS) project is currently the most complete dataset of global tropical cyclone tracks available [10, 16]. This dataset includes data from 12 weather agencies, such as RSMC-Tokyo and New Delhi, structured in a table. The index of each row is the identifier for the corresponding storm in byte string format and the time of the record, with a variety of numerical variables, such as latitude and longitude of the storm, which are averaged from across all the agencies [7]. Other variables, such as intensity, landfall, wind speed and pressures, are collected into variables categorised by each weather agency.

Given that there is a rich background to the dataset, the measurements of the tropical cyclone tracks are not unified; for example, cyclone intensity is measured differently in Japan than in the United States. This discrepancy is also observed for wind speed and air pressure. Therefore, necessary cleaning is required for the data before we proceed to model training. There are missing values in many of the columns of the dataset due to having multiple sources of the data and IBTrACS does not artificially interpolate the data [7]. Therefore, only latitude, longitude, and landfall are retained for further analysis.

Furthermore, even though the data is mainly collected in 3-hourly intervals, there are abnormal data points that differ from the 3-hourly intervals. Hence, these data points are removed from the dataset. There are also tropical cyclones with short sequences of observation that are not useful for either the training or the inference phrase. For this research, we have picked the window size for the predictions to be either five or ten. Therefore, cyclone track data of a sequence of less than five cannot be used for this research. After data cleaning, 3793 cyclones were left in the feature space.

3.1.1 Feature Engineering. The first important variable to be considered is the representation of the location of the storm. It is possible to simply keep the latitude and longitude feature sequence as input into the model [2, 24, 25], or a transformed set of latitude and longitude features [4]. However, as the latitude and longitude values are of a fairly wide range, normalisation is required for these values to avoid inefficiency in the model training and inference phase. [23]. It is also important to note that the latitude and longitude coordinates introduce an artificial discontinuity where the longitude changes from one end to the other. Other recent work has demonstrated that further expanding the latitude and longitude values into a fine-grained grid may improve model performance and reduce the possible output space of the model [2, 4].

On the other hand, we can apply unit sphere mapping to avoid the discontinuity problem introduced by latitude and longitude coordinates [21]. This transformation is also beneficial because it does not require normalisation since the scale of the data is the same [23]. However, this coordinate system has an underlying problem as it allows the storm to move up or down in the atmosphere.

$$x = \cos(\text{Latitude}) \cdot \cos(\text{Longitude})$$
$$y = \cos(\text{Latitude}) \cdot \sin(\text{Longitude})$$
$$z = \sin(\text{Latitude})$$

Given that these two different cyclone location representations have their own advantages and disadvantages, we have established models using both representations, allowing for comparison between the two types.

Other common features used in existing literature are wind speed and pressure [4, 24, 25]. Although results in these papers have shown that this may have an improvement in the performance of the model, it is unlikely to be useful in auto-regression. Neither the wind speed nor the pressure can be easily and accurately predicted given they involve more complicated weather process [25] and rather rely on pre-existing knowledge of the wind speed [4].

Finally, for model-building purposes, the dataset is separated into two subsets — training and validation, in a 80/20 ratio. Although it is common to have another test subset for pure evaluation purposes, we have chosen to ignore it since the predictions are made auto-regressively in which only the input for the first step prediction would be exactly identical to the dataset.

3.2 Model Architecture

The basis of the model is applying LSTMs to the input time-series data then further extracting the highly non-linear relationship by applying a NLPs to compute the final output of the model. The architecture of the LSTM is defined as the following. We are using LSTMs with forget gate as described by Gers et al. [11] and the compact equations are defined as the following with the architecture shown above in Figure 1.

In Equation 1, at time t, x_t is the input data, c_t is the cell state, and h_t is the hidden state. There are three gates included in the architecture as shown in Figure 1. f_t is the value of the forget gate. When it is 1, all previous cell state information is retained; when it is 0 all information is forgotten. On the other hand, i_t is the input gate which determines

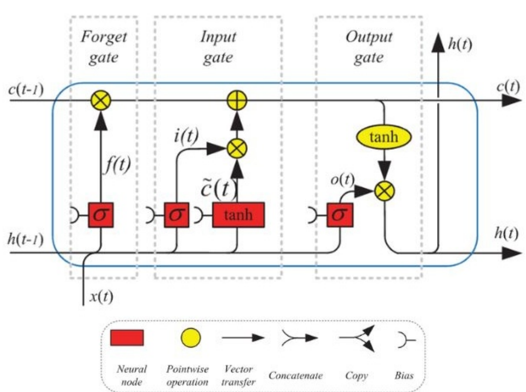

Fig. 1. Architecture of LSTM cell, by Yu et al. [27].

$$\begin{aligned}
f_t &= \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f),\\
i_t &= \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i),\\
\tilde{c}_t &= \tanh(W_{\tilde{c}h}h_{t-1} + W_{\tilde{c}x}x_t + b_c),\\
c_t &= f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t,\\
o_t &= \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o),\\
h_t &= o_t \cdot \tanh(c_t).
\end{aligned}$$

Eq 1: Underlying equations for LSTM cells.

Batch Size	Validation Loss	Number of Epochs
32	4.283e-06	15
64	6.558e-06	14
128	7.497e-06	20
256	1.178e-05	20
512	1.905e-05	16
1024	1.994e-05	23
2048	2.135e-05	39
4096	5.176e-05	42
8192	5.435e-05	68

Table 1. Effect of batch size on validation loss

Learning Rate	Validation Loss	Number of Epochs
1e-05	1.276e-05	812
0.0001	1.687e-05	181
0.001	3.013e-05	32
0.01	4.383e-05	17
0.1	2.25e-05	17

Table 2. Effect of learning rate on validation loss

Number of LSTM Units	Validation Loss	Number of Epochs
8	1.402e-05	140
16	2.337e-05	70
32	2.074e-05	43
64	2.427e-05	39
128	2.881e-05	26
256	3.164e-05	30
512	4.637e-05	19

Table 3. Effect of number of LSTM units on validation loss

what input information is maintained for the current inference. Finally, the output gate decides what information can be output based on the cell status by affecting the hidden state as the output for a RNN unit is the hidden state [27]. Finally, all the W matrices are different trainable parameter matrices at used for different gates.

3.2.1 Batch Normalisation. Batch normalisation is, as its name suggests, a normalisation technique that is applied to each training mini-batch, introduced by Ioffe and Szegedy [14]. This technique can also enhance model training speed by allowing for a larger learning rate. At the same time, batch normalisation is particularly useful for LSTM as it can allow faster convergence and better generalisation as shown by Cooijmans et al. [6]. Therefore, we have applied batch normalisation to latitude and longitude models as the scales after normalisation were still not unified enough.

3.3 Model Training and Hyper-parameter Tuning

For hyper-parameter tuning, models were fit with the `EarlyStopping` callback method from TensorFlow [1], which stops the fitting when the validation loss is no longer decreasing. We set the `patience` argument to 3, which means it will stop after 3 epochs with no improvement. The validation loss reported is the minimum validation loss achieved across all training epochs. This is to prevent possible fluctuations near the end from affecting the 'overall' validation loss of the model. Whenever not varied, the batch size is constant at 2048, the learning rate is 0.001, and one LSTM layer with 64 units. We also have a second layer, which is a dense layer with 3 units.

In Table 1, the validation loss is lower and therefore better for lower batch sizes. The validation loss appears to increase with the number of batch sizes. However, training was more time-consuming for lower batch sizes. Also, in Table 2, the validation loss generally decreases as the learning rate decreases, except for when the learning rate is 0.1. The training was more time-consuming for lower learning rates. In Table 3, the validation loss generally increases as the number of LSTM units increases, except for when there are 16 LSTM units, whose validation loss is greater than the loss for 32 units. Based on the discussed model architecture and hyper-parameter tuning, we have created a list of models as summarised in Table 4.

4 Model Evaluation and Results

This section aims to evaluate the relative performance of the proposed models, comparing them to state of the art methods, both in terms of performance, but also computational cost.

Model	Layers: Layer	Layers: Type	Layers: Units	Number of Epochs Trained	Batch Size	Learning Rate	Window Size	Input Type
A	1	LSTM	64	10	1024	0.001	5	Unit Sphere
	2	Dense	3					
B	1	LSTM	16	50	64	0.0001	10	Unit Sphere
	2	Dense	16					
	3	Dense	3					
C	1	Batch Norm.	n/a	20	256	Cosine Decay	5	Latitude & Longitude
	2	LSTM	64					
	3	Dense	2					
D	1	Batch Norm.	n/a	20	256	Cosine Decay	10	Latitude & Longitude
	2	LSTM	128					
	3	Dense	2					

Table 4. Summary of proposed models

4.1 Evaluation

The IBTrACS West Pacific cyclone dataset was used for all evaluation calculations. Methods to quantify the performance of time-series forecasting models can be split into two categories; single step, and multi-step. This study will investigate the single-step performance of the proposed models through haversine distance error and Precision Intersection over Union (IOU) metrics. Meanwhile the multi-step performance is evaluated through root mean squared error (RMSE) and mean absolute error (MAE).

4.1.1 Single Step. The average single-step haversine distance error metric, described in Figure 2(a), provides insight into the error of the model when predicting the next point of a cyclone, given an input set of points (equal to the window size). It was calculated as the haversine distance between the actual and predicted latitude and longitude points as per the equation below (where $r = 6371$km, the Earth's radius), and then averaged across 500 randomly selected tropical cyclones from the evaluation dataset. In the equation below, φ is latitude and λ is longitude of the two points.

$$d = 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1) \cdot \cos(\varphi_2) \cdot \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

Another key evaluation metric is precision, calculated using TP (true positive) and FP (false positive) values as per below.

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

Due to the non-classification nature of time-series models, predicted points do not naturally fall into the category of true positive or false positive. Therefore, to determine these values, we must devise a classification strategy where if a predicted point is *close enough* to the actual, the classification is true positive. More formally, to achieve this we must assign each point an area varied by a θ value (where $0 < \theta < 1$), then calculate the IOU (Intersection over Union)

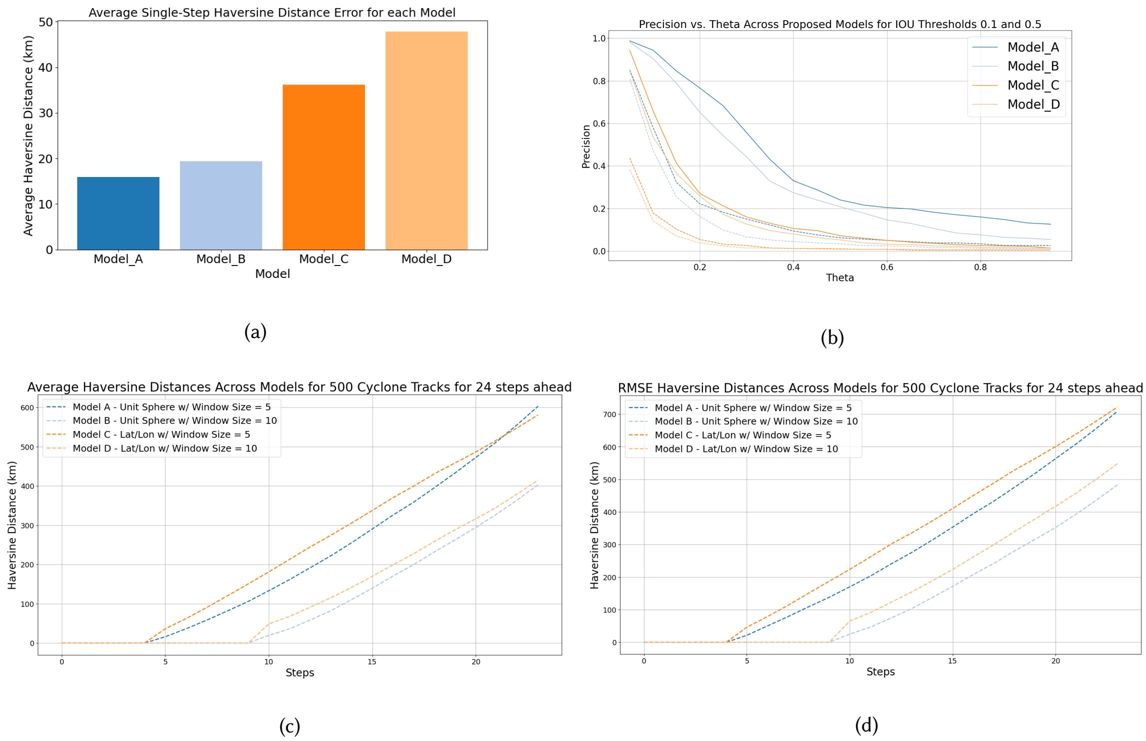

Fig. 2. Comparison of Proposed Models: (a) Single-Point Error (measured as haversine distance) for Proposed Models, (b) Precision vs. Theta of Proposed Models with IOU threshold 0.1 (dotted lines) and 0.5 (dotted lines). Same color represents same architecture results, (c) MAE over 24 time steps, and (d) RMSE over 24 time steps.

between the two areas. If the IOU value is greater than a defined threshold, the prediction is classified as true positive, otherwise false positive. Precision at different IOUs is given in Figure 2(b).

$$IOU_{\text{Intersection over Union}} = \frac{\text{Intersection of Area}}{\text{Union of Area}}$$

In order to determine IOU earth must be approximated as a 2D grid. The dx and dy in km between the predicted point and actual point are calculated based on the respective longitude and latitude values, with intersection and union values being calculated using point area thereafter. Precision curve is plotted for each model for IOU threshold values of 0.5 and 0.1, across θ values for $0 < \theta < 1$, where each predicted/actual point is represented as a square with area $\frac{25}{\theta^2} km^2$.

4.1.2 Multi Step. The performance of time-series models can also be quantified through auto-regressive multi-step analysis. These metrics differ from single-step as they measure the predictive performance of the model across an entire cyclone track, as opposed to just the next point. Approximately 500 cyclone tracks from the evaluation dataset were selected for analysis, through systematic sampling.

The averaged RMSE (root mean squared error) and MAE (mean absolute error) metrics for each model were plotted across all time steps from $t = 0$ to $t = 24$, described by Figure 2(c) and 2(d).

$$RMSE_{\text{root mean squared error}} = \frac{\sum (y_i - y_p)^2}{n} \quad MAE_{\text{mean absolute error}} = \frac{|(y_i - y_p)|}{n}$$

4.2 Results

4.2.1 Model Performance. In the single-step haversine distance evaluation, unit-sphere based Models A and B demonstrated the lowest errors, as shown in Figure 2(a), in the range of $15 - 20km$. The precision-theta curve in Figure 2(b) backed these findings, both at the 0.1 and 0.5 IOU thresholds. As expected, both sets of models performed relatively similarly across IOU thresholds. It appears the additional feature, in the form of the z coordinate significantly improved model performance for a single-step.

The autoregressive multi-step MAE in Figure 2(c) and RMSE in Figure 2(d) plots for Model A and Model B appeared to exhibit a lower gradient indicating less error, for shorter time frames (< 20 steps). As steps increase however, we notice the gradient of the latitude/longitude based Models decrease relative to their unit sphere counter parts, implying stronger performance for long-term path prediction. A key difference across models was the window size, ranging from 5 to 10. Across evaluation metrics, there was little evidence indicating window size had significant impact on model performance. Figure 2(c) and 2(d) back these findings, with corresponding models with the same the features (unit-sphere vs. lat/lon) displaying similar trends across window sizes.

Overall, we notice the unit sphere based implementations had stronger performance, most notably at the single-step. Specifically, from the precision-theta curve Figure 2(b) and the distance-error graphs 2(a), 2(c), 2(d), we notice Model A has strongest overall performance. Having identified our strongest performing models, the question is how these results compare to other state-of-the-art tropical cyclone prediction techniques.

4.2.2 Comparison to the State-of-the-Art. Kim et al. [15] built a set of ConvLSTM models for cyclone track prediction, quantifying error using RMSE, similar to Figure 2(d) [15]. From Figure 2(d), we notice the predictions of Model A for the first 3 time-steps (extracted from the raw data) have RMSE values of 21.34, 50.05, 78.74 respectively, whereas Kim et al. [15] found that their best model had 26.64, 140.97 and 148.74. While the first prediction (3 hours) is comparable, our proposed model appears to perform better across the next two predicted points, out-performing the ConvLSTM model by an average of 147.5% RMSE error. Dong et al. [7] set out to quantify the performance of state-of-the-art HAFS for hurricane prediction, based on MAE [7]. At the 12, 24, 36 and 48 hour mark Dong et al. [7] found HAFS's absolute error in km was 40, 55, 80 and 100, meanwhile the MAE of our proposed Model A was as per Figure 2(c), 81.60, 191.66, 325.82 and 472.42 (extracted from the raw data) at the same time steps. While across long periods of time, the performance of our model is magnitudes worse than HAFS, at the 12 hour time frame we notice results are within 102% of absolute error.

4.2.3 Comparison to NeuralGCM by Kochkov et al. [17]. In Figure 3, it is generally seen that Model D predictions follow the actual track for the first few time-steps, before deviating in direction. For example, there is a cyclone heading in a westward direction after making landfall in southern Philippines, but the prediction has the cyclone moving northwards towards southern mainland China. On the other hand, NeuralGCM predictions appear to be more random, where some predictions are visually further from the actual track, such as the track going along the coastline of eastern Russia. There is also a random prediction south of that track that does not correspond to any actual track. Despite the computational complexity of NeuralGCM, it is far from perfect in predicting cyclone tracks. Model D is much more light-weight in comparison, and can produce similar, if not more accurate predictions.

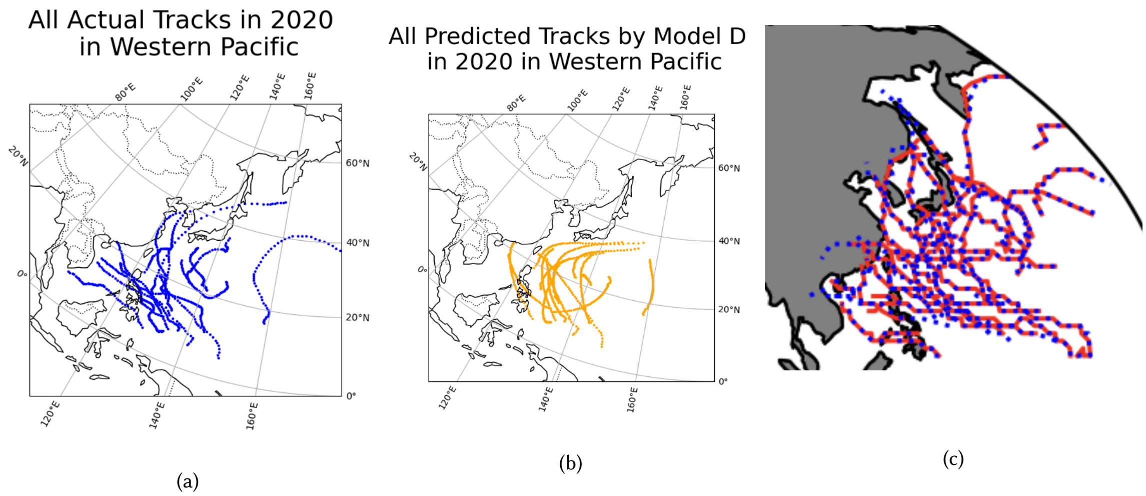

Fig. 3. Comparison of the predictions of model D versus NeuralGCM [17] on the same tracks. (a) is a graph of all the actual tracks in the western pacific basin in IBTrACS dataset. (b) is a graph of all the predicted tracks by our models limiting to 40 steps ahead. (c) is the NeuralGCM's predictions in the similar Western Pacific basin.

5 Limitations and Future Work

This paper investigated two types of location representations, latitude/longitude and unit sphere [2, 15], both of which had their own problems. The unit sphere approach has an inherent flaw, whereby the model may make predictions where the z coordinate is not coherent, i.e., is not on the surface of the Earth. While this problem is tackled by the latitude and longitude approach, new flaws are generated in the process. Specifically, the idea that latitude and longitude are bounded (from 0 to 360 or -180 to 180) has the potential to cause problems.

Future work could explore hybrid approaches that combine the strengths of both unit-sphere and latitude/longitude representations to mitigate their respective flaws. For example, incorporating post-processing techniques to ensure unit-sphere predictions remain coherent on the Earth's surface could improve accuracy. Additionally, advanced bounding strategies for latitude and longitude models may reduce issues related to coordinate wrapping, particularly when predicting across the poles or the 180-degree meridian. Further research could also investigate alternative geo-spatial representations or neural network architectures specifically designed for spherical data.

When it comes to future model design, it would be beneficial to perform more thorough hyper-parameter tuning as it is currently done exclusively by grid searching. Furthermore, this process can be aided with super-computers in order to search for a wider parameter space.

6 Conclusion

This investigation has provided deep insight into the performance of simple, lightweight machine learning models for cyclone forecasting. Results have shown that our simplified LSTM-based models perform at comparable levels to state-of-the-art methods, and even outperform certain more complex deep learning models. These findings highlight the potential of streamlined LSTM models to revolutionise cyclone forecasting, offering a powerful, cost-effective alternative to complex, resource-intensive systems.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 265–283.

[2] Sheila Alemany, Jonathan Beltran, Adrian Perez, and Sam Ganzfried. 2019. Predicting hurricane trajectories using a recurrent neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 468–475.

[3] AWS. [n. d.]. What is RNN? - Recurrent Neural Networks Explained - AWS. https://aws.amazon.com/what-is/recurrent-neural-network/

[4] Rikhi Bose, Adam L Pintar, and Emil Simiu. 2021. Forecasting the evolution of north atlantic hurricanes: A deep learning approach. *Technical Note (NIST TN), National Institute of Standards and Technology, Gaithersburg, MD.* (2021). https://nvlpubs.nist.gov/nistpubs/TechnicalNotes/NIST.TN.2167.pdf

[5] Nikolaj Buhl. 2023. Time Series Predictions with RNNs. https://encord.com/blog/time-series-predictions-with-recurrent-neural-networks/

[6] Tim Cooijmans, Nicolas Ballas, César Laurent, Çağlar Gülçehre, and Aaron Courville. 2016. Recurrent batch normalization. *arXiv preprint arXiv:1603.09025* (2016).

[7] Pingping Dong, Jie Lian, Hui Yu, Jianguo Pan, Yuping Zhang, and Guomin Chen. 2022. Tropical cyclone track prediction with an encoding-to-forecasting deep learning model. *Weather and Forecasting* 37, 6 (2022), 971–987.

[8] Kerry Emanuel. 2018. 100 years of progress in tropical cyclone research. *Meteorological Monographs* 59, 1 (2018), 15–1.

[9] Climate Extremes. 2023. A closer look at climate modelling. https://climateextremes.org.au/a-closer-look-at-climate-modelling/

[10] Jennifer Gahtan, Kenneth R. Knapp, Carl J. III Schreck, Howard J. Diamond, James P. Kossin, and Michael C. Kruk. 2024. (2024). *International Best Track Archive for Climate Stewardship (IBTrACS) Project, Version 4.01. [West Pacific]* (2024). https://doi.org/10.25921/82ty-9e16

[11] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to forget: Continual prediction with LSTM. *Neural computation* 12, 10 (2000), 2451–2471.

[12] Research Graph. 2024. An Introduction to Recurrent Neural Networks (RNNs). https://medium.com/@researchgraph/an-introduction-to-recurrent-neural-networks-rnns-802fcfee3098

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (11 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 arXiv:https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf

[14] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[15] Sookyung Kim, Hyojin Kim, Joonseok Lee, Sangwoong Yoon, Samira Ebrahimi Kahou, Karthik Kashinath, and Mr Prabhat. 2019. Deep-hurricane-tracker: Tracking and forecasting extreme climate events. In *2019 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 1761–1769.

[16] Kenneth R. Knapp, Michael C. Kruk, David H. Levinson, Howard J. Diamond, and Charles J. Neumann. 2010. The International Best Track Archive for Climate Stewardship (IBTrACS): Unifying Tropical Cyclone Data. *Bull. Am. Meteorol. Soc.* 91, 3 (March 2010), 363–376. https://doi.org/10.1175/2009BAMS2755.1

[17] Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, Sam Hatfield, Peter Battaglia, Alvaro Sanchez-Gonzalez, Matthew Willson, Michael P. Brenner, and Stephan Hoyer. 2024. Neural general circulation models for weather and climate. *Nature* 632, 8027 (Aug. 2024), 1060–1066. https://doi.org/10.1038/s41586-024-07744-y

[18] Jie Lian, Pingping Dong, Yuping Zhang, Jianguo Pan, and Kehao Liu. 2020. A Novel Data-Driven Tropical Cyclone Track Prediction Model Based on CNN and GRU With Multi-Dimensional Feature Selection. *IEEE Access* 8 (2020), 97114–97128. https://doi.org/10.1109/ACCESS.2020.2992083

[19] Mina Moradi Kordmahalleh, Mohammad Gorji Sefidmazgi, and Abdollah Homaifar. 2016. A sparse recurrent neural network for trajectory prediction of atlantic hurricanes. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. 957–964.

[20] NOAA. 2024. NOAA Hurricane Analysis and Forecast System (HAFS). https://registry.opendata.aws/noaa-nws-hafs

[21] Kamban Parasuraman. 2022. Hurricane Path Prediction using Deep Learning - Kamban Parasuraman - Medium. *Medium* (Oct. 2022). https://medium.com/@kap923/hurricane-path-prediction-using-deep-learning-2f9fbb390f18

[22] David A Randall. 2000. *General circulation model development: past, present, and future.* Elsevier.

[23] Khaled Mahmud Sujon, Rohayanti Binti Hassan, Zeba Tusnia Towshi, Manal A. Othman, Md Abdus Samad, and Kwonhue Choi. 2024. When to Use Standardization and Normalization: Empirical Evidence From Machine Learning Models and XAI. *IEEE Access* 12 (Sept. 2024), 135300–135314. https://doi.org/10.1109/ACCESS.2024.3462434

[24] Jinkai Tan, Sheng Chen, and Jun Wang. 2021. Western North Pacific tropical cyclone track forecasts by a machine learning model. *Stochastic Environ. Res. Risk Assess.* 35, 6 (June 2021), 1113–1126. https://doi.org/10.1007/s00477-020-01930-w

[25] B Tong, X Wang, JY Fu, PW Chan, and YC He. 2022. Short-term prediction of the intensity and track of tropical cyclone via ConvLSTM model. *Journal of Wind Engineering and Industrial Aerodynamics* 226 (2022), 105026.

[26] Zhen Wang, Jun Zhao, Hong Huang, and Xuezhong Wang. 2022. A review on the application of machine learning methods in tropical cyclone forecasting. *Frontiers in Earth Science* 10 (2022), 902596.

[27] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation* 31, 7 (2019), 1235–1270.

[28] Yanfei Zhang, Rohitash Chandra, and Junbin Gao. 2018. Cyclone Track Prediction with Matrix Neural Networks. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8. https://doi.org/10.1109/IJCNN.2018.8489077