14th International Conference on Structural Safety and Reliability - ICOSSAR'25
1-6 June, 2025 - Los Angeles California, USA
R. Ghanem, E. Johnson, S. Masri and A. Olivier (Eds)

DEEP LEARNING MOTIVATED DATA IMPUTATION OF TROPICAL CYCLONE RADIUS OF MAXIMUM WINDS

Swastik Agrawal[1], **Nishkal Hundia**[1], **Ziyue Liu**[2] **and Michelle Bensi**[1]

[1] University of Maryland

[2] Purdue University

Key words: Machine learning, Deep learning, Radius of Maximum winds, Historical storm data, Data Imputation

Abstract. Probabilistic coastal hazard assessments support risk-informed design approaches for critical infrastructure. These assessments typically are based on the Joint Probability Method, which requires accurate characterization of the joint distribution of tropical cyclone (TC) parameters. However, historical TC datasets used to derive the joint distribution often contain incomplete records for the Radius of Maximum Winds (R_{max}), a key variable in these analyses. While prior research has employed machine learning (ML) methods like Gaussian Process Regression and shallow Artificial Neural Networks to estimate missing R_{max} values, the potential of deep learning techniques for this task remains largely unexplored. Building on previous work, this study investigates the use of historical storm data to evaluate and compare various deep-learning strategies for R_{max} imputation. Specifically, we examine the performance of one-dimensional Convolutional Neural Networks (1DCNNs) and Long Short-Term Memory networks (LSTMs). This study also introduces physically informed model refinement and assessment approaches. We also compare these deep learning methods with previously unexplored classical ML methods, such as decision trees, particularly XGBoost. The study focuses on simplifying input features and leveraging the time series characteristics of storms to improve the imputation of missing R_{max} values in historical datasets. By exploring deep learning methodologies, this research aims to enhance the accuracy and reliability of imputing missing data in historical storm records.

1 INTRODUCTION

Storm-induced coastal hazards can impose a variety of adverse loads and conditions on structures. The probabilistic analysis of those hazards often relies on historical storm data [13][18]. However, historical storm records are often incomplete temporally or

spatially, particularly in the earlier years of record [16]. One particularly important yet often missing storm parameter is the Radius of Maximum wind speed (R_{max}) which is a representation of the size of a storm. Due to its physical characteristics, R_{max} plays a critical role in wind field estimation [5][4]. Thus, it is important to accurately estimate the value of R_{max} in datasets where it is missing. The goal is to predict missing R_{max} values. Figure 1 shows an example of the imputation process.

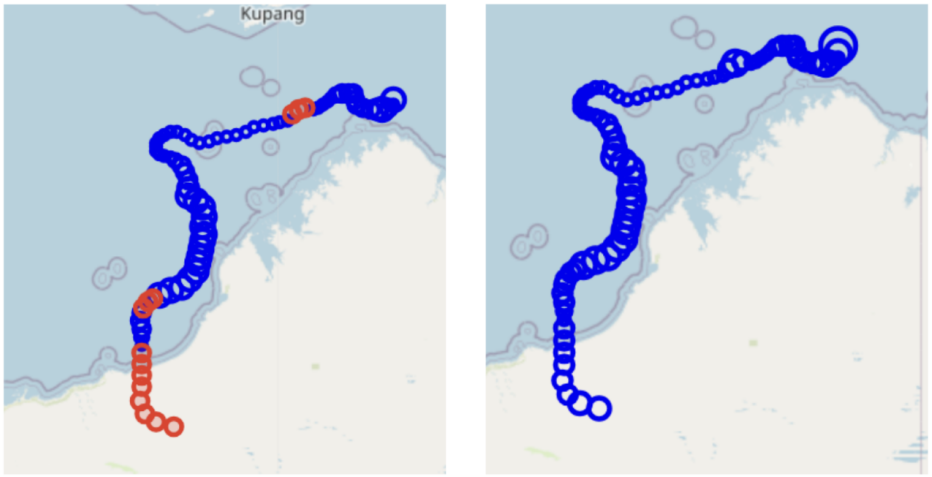

Figure 1: Left: A storm track with available R_{max} values indicated with blue circles, with the radius being proportional to the actual R_{max} values. Missing R_{max} values are indicated with red circles. Right: The same storm track with predicted R_{max} values.

Previous studies have proposed physical and statistical models to estimate the R_{max} [5][3][4][20]. However, these models may be limited in applicability across different regions and storm intensities. ML methods provide a data-driven alternative with advantages in efficiently modeling complex relationships and have been applied to storm-induced hazard analysis over the past decade [10][1][11][12]. Liu et al. 2024c explored the implementation of ML methods, including Gaussian process regression and shallow Artificial Neural Network (ANN), for historical storm data imputation and demonstrated the robustness of ML-based R_{max} imputation compared to existing approaches.

While shallow ML methods have shown promise, they are limited in their ability to capture complex temporal dynamics and dependencies inherent in storm evolution. Deep learning (DL), particularly architectures such as Convolutional Neural Networks (CNNs), and recurrent neural networks like Long Short-Term Memory networks (LSTMs) present an opportunity to capture temporal effects. LSTMs, for example, are well-suited to learning long-term dependencies in time series and have been successfully applied to TC intensity forecasting and track prediction tasks [19]. Similarly, one-dimensional Convolutional Neural Networks (1DCNNs) have been used to predict

peak storm surge from TC time series data [11].

However, despite these advances, the use of deep learning to impute missing R_{max} values in historical storm datasets remains largely unexplored. This study aims to address this gap by investigating deep learning-based imputation of R_{max} in TC track data. Specifically, we evaluate the performance of 1DCNN and LSTM in modeling the temporal nature of storm evolution and estimating missing R_{max} values. This approach, unlike previous studies that typically treat individual storm points independently, preserves the temporal dynamics of storms.

2 METHODOLOGY

2.1 Data Sources

We use historical TC reanalysis data sourced from the International Best Track Archive for Climate Stewardship (IBTrACS) for most of the testing done on various models [9]. The IBTrACS sources initial observations during the storm, followed by reanalysis based on other available data such as reports from ships, land stations, radars, satellites, etc. All of this is used to produce a best estimate of the storm's track and intensity. For our tests, we consider the USA_AGENCY data series in IBTrACS because it has consistent spatial and temporal coverage as well as wide availability for each variable. The temporal resolution of the data is generally 3-hourly, and the spatial resolution is 0.1 degrees, which is nearly 10 km [9].

2.2 Data Preprocessing

We begin by filtering the dataset to include only entries reported by the USA_AGENCY, ensuring consistency in the extracted data and measurement standards. We retain only the variables relevant to our testing methods from this filtered dataset, as given in Table 1.

We introduce a categorical flag (IS_LAND) in our feature set indicating whether the storm data point is over land or water. To meaningfully normalize the storm direction, we take the sine and cosine of the angle and add them to our feature set. To better capture spatial information for neural network-based models, we convert the latitude and longitude into 3D Cartesian coordinates (X, Y, Z) to better encode geospatial information. The transformation is performed using the following equations:

$$X = \cos(lat) \cdot \cos(lon) \tag{1}$$

$$Y = \cos(lat) \cdot \sin(lon) \tag{2}$$

$$Z = \sin(lat) \tag{3}$$

Once all relevant variables are extracted and transformed, we structure the time series data for input into sequential models such as CNNs and LSTMs. We divide each storm's track into fixed-length segments, referred to herein as blocks. Each block represents a

Table 1: Extracted Variables from IBTrACS for data preprocessing. Latitude is expressed in degrees due north [-90, 90], and longitude is expressed in degrees due east [-180, 180]. Storm direction θ, can range from 0-360 degrees.

Variable	**Symbol**	**Unit**
Minimum Pressure at sea level	P_c	millibars
Maximum sustained wind speed	V_{max}	knots
Latitude	lat	degrees
Longitude	lon	degrees
Distance from the center of the storm to the point at which the wind speed is 34 knots (mean of non-zero quadrants)	R_{34}	nmile (nautical miles)
Distance of the storm point to land	d	nmile (nautical miles)
Storm Speed	V_f	knots
Storm Direction	θ	degrees

contiguous subsequence of the storm's time series, capturing the evolution of relevant variables over a fixed window of time.

Blocks are generated independently for each storm and are temporally homogeneous (there typically, though not exclusively, exists a consistent 3-hour time interval between two consecutive data points). To prevent data leakage and ensure rigorous evaluation, we store metadata about each block's origin (i.e., storm ID as well as start and end indices of each block in the original dataframe). This allows us to enforce that all blocks from a given storm are assigned exclusively to one of the training, validation, or testing sets, and are never split across them.

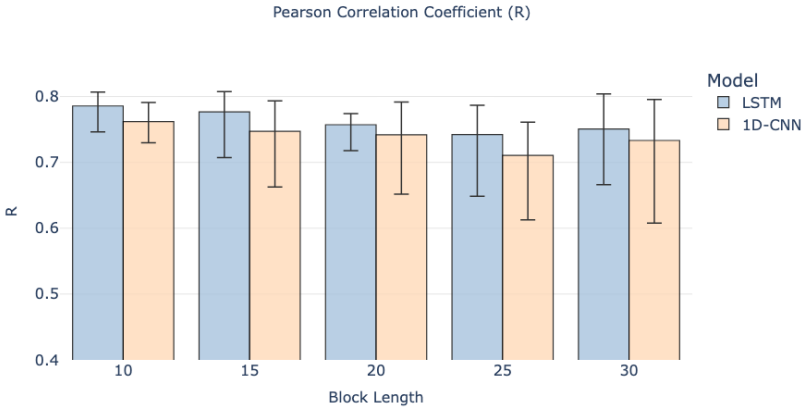

Figure 2: R Coefficient comparison of various block lengths. Error bars represent range (min, max)

After generating blocks, we perform a sensitivity study in which we train 1DCNN and LSTM models using a k-fold cross-validation split across the entire dataset. For each fold, blocks are partitioned into training, validation, and testing sets, maintaining strict isolation between storms across these sets. Multiple fixed lengths for the blocks were tested (10, 15, 20, 25, and 30 time steps), and we judged that 10 was the most effective block size.

As is evident from the results in Figure 2, a block length of 10 yields the highest correlation coefficient with the lowest variance across both models. This might be because a length of 10 results in a higher number of training samples for these models. Other evaluation metrics, including Mean Absolute Error (MAE) and Mean Squared Error (MSE), corroborate this finding.

Finally, we implement min-max scaling on the following variables: P_c, V_{max}, R_{34}, d, and V_f. To ensure that the training data is normalized for the training process. The scaling is applied to the training, validation, and testing sets for each fold to ensure no information leakage occurs between the training, validation, and test datasets.

2.3 Tested models (ANN, XGBoost, 1DCNN, LSTM)

We explored a range of ML techniques, focusing heavily on deep learning approaches, specifically 1DCNNs and LSTMs, due to their ability to leverage temporal dependencies in the data. Figure 3 provides a graphical summary of the trade-offs between temporal and non-temporal methods.

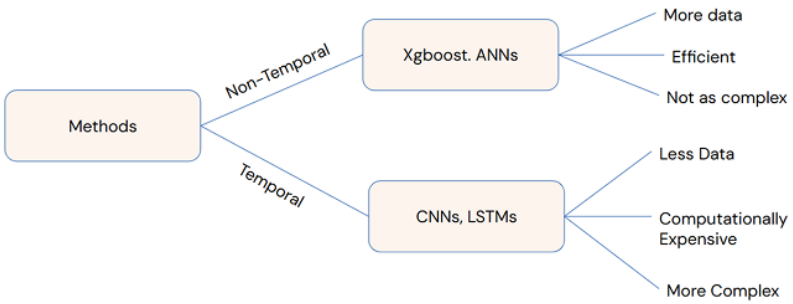

Figure 3: Tree of methods used and their benefits and drawbacks

1DCNNs operate by sliding convolutional filters over the input sequence to detect local temporal patterns. These models have been shown to be effective at identifying short-term dependencies and trends in multivariate TC storm surge time series data [11]. This model architecture includes sequential layers of convolutional layers, followed by pooling layers that reduce the dimensionality of data as it flows through the model.

LSTMs, on the other hand, are designed to model long-term dependencies in sequential inputs using gated memory cells [8]. This makes them good candidates for capturing

the progression of storm characteristics over time [7]. By maintaining both long-term and short-term memory, LSTMs are capable of retaining localized information about storm points near the current timestep, while also preserving broader contextual information about the overall storm track.

To identify the best architecture, we perform a random search (of 100 trials for each temporal model) over predefined architecture-defining parameters (such as number of layers, neurons, and activation functions) to optimize the hyperparameters for our models in a more efficient manner [2]. These identified architectures are presented in Figure 4. We then use the resulting model architecture to select the best block length for the 1DCNN and LSTM models.

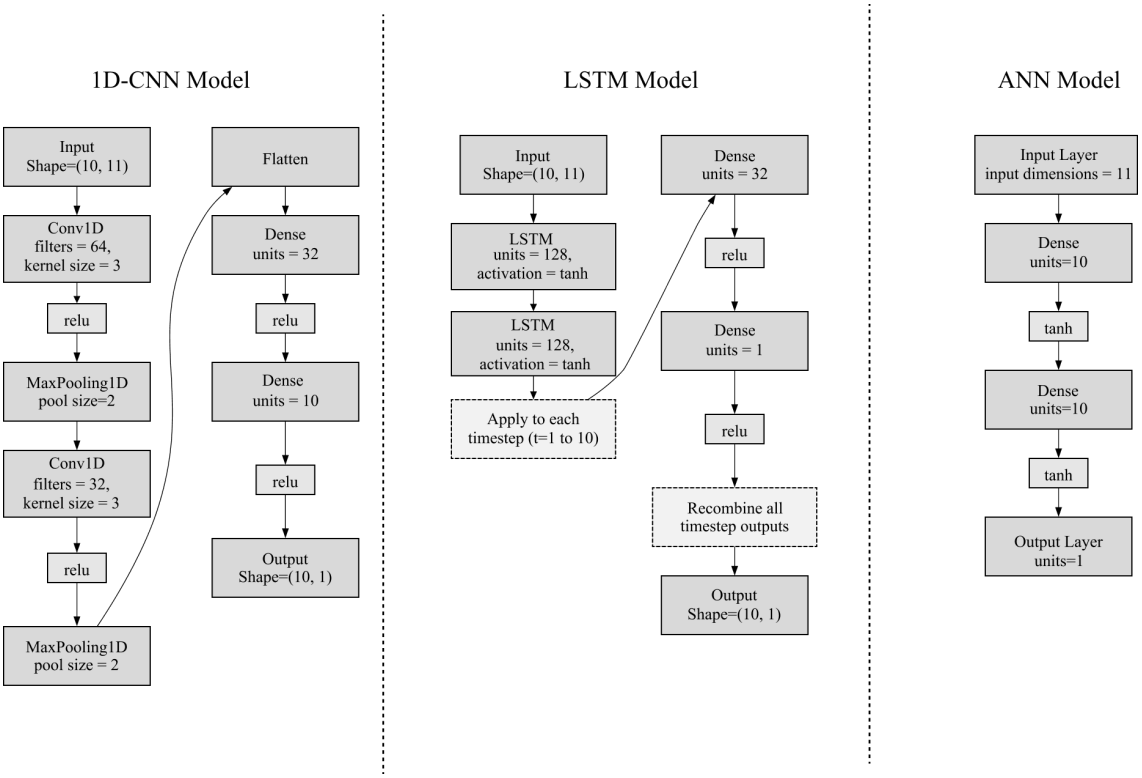

Figure 4: Flowchart depicting model architecture for the tested neural networks

To evaluate the effectiveness of our temporal deep learning implementations against standard non-temporal approaches, we compare the performance of the DL models against two baseline models XGBoost and ANN. These models are limited in their ability to capture the sequential dependencies and evolving dynamics within a storm's lifecycle. XGBoost helps capture complex nonlinear relationships and interactions within the data by sequentially constructing an ensemble of decision trees, where each tree corrects residuals via regularized gradient descent [6]. The specific ANN architecture

in this study has been derived from Liu et al. 2024c.

2.4 Alternative Variable Sets

For the final model evaluation, we experiment with three different input configurations: one including the R_{34} variable, one excluding it, and one where the data is scaled. The inclusion and exclusion of R_{34} is done because R_{34} is known to be highly correlated to R_{max} but it is also not very widely available in historical datasets [12]. The scaled configuration is based on the methodology outlined in Chavas and Knaff 2022, herein referred to as CK22 [4], and introduces additional parameters derived from empirical relationships into the input variable set, including R_{34}. The underlying hypothesis is that scaling the data using these known relationships can reduce the number of training steps required for the model compared to when the model must learn them independently. This effectively embeds established TC physics into the dataset, potentially enhancing model performance.

The equations used to get the scaled data based on CK22 are as given in the equations below

$$f = 2\sin(lat) \tag{4}$$

$$M_{34} = R_{34} \times 34 \text{ m s}^{-1} + \frac{1}{2} f R_{34}^2 \tag{5}$$

$$\frac{M_{\max}}{M_{34}} = 0.699 \exp\left[-0.00618(V_{\max} - 34 \text{ knots}) - 0.00210(V_{\max} - 34 \text{ knots})\left(\frac{1}{2} f R_{34}\right)\right] \tag{6}$$

$$M_{\max} = \left(\frac{M_{\max}}{M_{34}}\right) M_{34} \tag{7}$$

$$R_{\max} = \frac{V_{\max}}{f}\left(\sqrt{1 + \frac{2 f M_{\max}}{V_{\max}^2}} - 1\right) \tag{8}$$

where,

f is the Coriolis parameter

$R_{\max}, R_{34}, V_{\max}$ have the same definition as this paper,

$M_{\max}$ is The absolute angular momentum at $R_{\max}$,

M_{34} is The absolute angular momentum at R_{34}

Based on these equations the following variables are added to the variable set: $(R_{34})^2$, f, $\sin(lat)$, and the ratio $M_{\max}/M_{34}$.

3 RESULTS AND DISCUSSION

To evaluate model performance, we use two primary metrics: the correlation coefficient (R) and MAE between predicted and true R_{max} values. These metrics provide insights regarding each model's ability to accurately and consistently impute missing R_{max} values from multivariate storm data sequences. We compare the deep learning models (1DCNN and LSTM) against the non-temporal baselines (XGBoost and ANN). Our experiments were conducted using the IBTrACS reanalysis dataset, with 4-fold cross-validation.

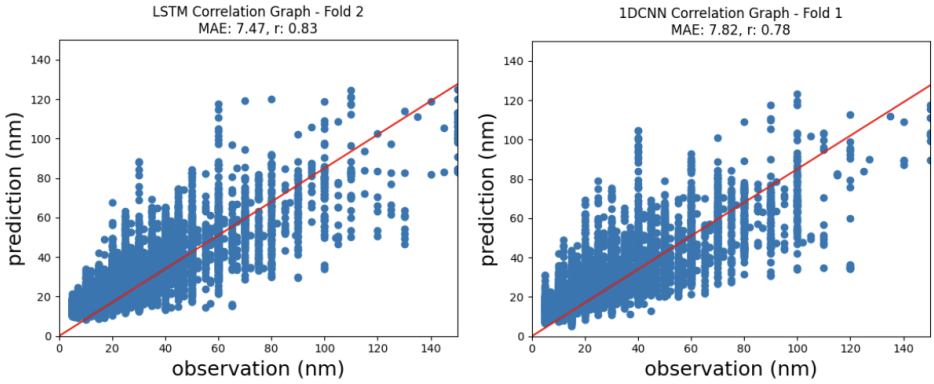

Figure 5: Scatter plot comparing the predicted and observed, and true R_max values for 1DCNN and LSTM models using scaled input data

Table 2 presents the LSTM model with scaled data achieves the best overall performance across all metrics, with the highest correlation coefficient (R = 0.78), the lowest MAE (= 7.83 nm). and the lowest MSE (= 158.2 nm). This supports our hypothesis that embedding empirically derived physical relationships, as done in the scaled data configuration, can significantly enhance model performance by reducing the burden on the model to learn these relationships from scratch. Figure 5 presents the correlation graphs for LSTMs and 1DCNN using scaled input data.

Across all models, those with access to the R_{34} variable tend to perform similarly well. This is expected, as R_{34} is strongly correlated with the target variable R_{max}. This strong correlation likely dominates any advantage of the temporal modelling capabilities of 1DCNNs and LSTMs, resulting in similar errors. For instance, 1DCNN, LSTM, and XGBoost models all achieve R values around 0.76–0.77 when R_{34} it is included.

For all model types, performance with scaled data remains similar to or better than the version with R_{34} alone. Notably, the LSTM with scaled data achieves better performance than both the LSTM with only R_{34} and the one without R_{34}. This suggests that integrating physical model-based variables derived from CK22 not only complements

Table 2: Model Performance Comparison (All mentioned metrics are mean metrics across four folds of cross-validation. The Sample size mentioned is of one of the four folds, and marginally changes across folds.)

	Model	R value	MAE	MSE	Train Samples	Test Samples
Temporal	1DCNN with R34	0.77	8.2	170.1	3,287	1,130
	1DCNN without R34	0.60	11.2	306.4	5,516	1,830
	1DCNN with scaled data	0.76	8.4	177.8	3,317	1,100
	LSTM with R34	0.77	7.9	160.9	3,342	1,075
	LSTM without R34	0.62	11.0	292.8	5,530	1,816
	LSTM with scaled data	**0.78**	**7.8**	**158.2**	3,333	1,084
Non-Temporal	XGBoost with R34	0.76	8.6	180.9	45,043	14,390
	XGBoost without R34	0.56	11.8	341.8	70,877	23,513
	XGBoost with scaled data	0.76	8.6	178.4	43,992	15,441
	ANN with R34	0.67	10.4	235.0	44,639	14,794
	ANN without R34	0.54	12.4	347.1	71,211	23,179
	ANN with scaled data	0.60	11.1	267.2	44,678	14,755

the presence of R_{34} but may also improve generalization and learning efficiency.

However, once R_{34} it is excluded, differences between temporal and non-temporal models become more apparent. In the absence of R_{34}, LSTM models outperform non-temporal approaches like XGBoost and ANN. This suggests that the LSTM is able to leverage temporal dependencies within the storm data to partially compensate for the missing R_{34} variable. Similarly, the 1DCNN model also shows a performance drop without R_{34} but still outperforms ANN and is closer in performance to LSTM, indicating some benefit from its convolutional structure in handling sequential patterns.

It is important to note that the temporal deep learning models (LSTM and 1DCNN) are trained on significantly fewer samples than the non-temporal models like XGBoost and ANN. This is due to the need for constructing temporally coherent input sequences (e.g., 10 time steps per sample), which limits the number of usable blocks from the dataset. Despite this limitation in sample size, the performance of temporal models is competitive or superior, indicating their effectiveness in capturing sequential dynamics crucial for R_{max} prediction.

To further assess model behavior, we visualize predicted and true R_{max} values across critical storm variables: P_c , V_{max} and V_f (results for P_c and V_{max} shown in Figure 6). We also plot prediction errors against these variables to identify patterns. Results show that storms characterized by high maximum sustained wind speeds and low minimum pressures tend to exhibit smaller prediction errors, suggesting that the model imputes R_{max} more reliably for stronger, more structurally consistent storms compared to weaker, more variable systems. But, this might also be due to the true R_{max} values themselves being relatively lower for stronger storms as compared to cases when the minimum pressure is higher and the wind speeds are lower, resulting in lower errors. The plots for 1DCNN also look similar to Figure 6.

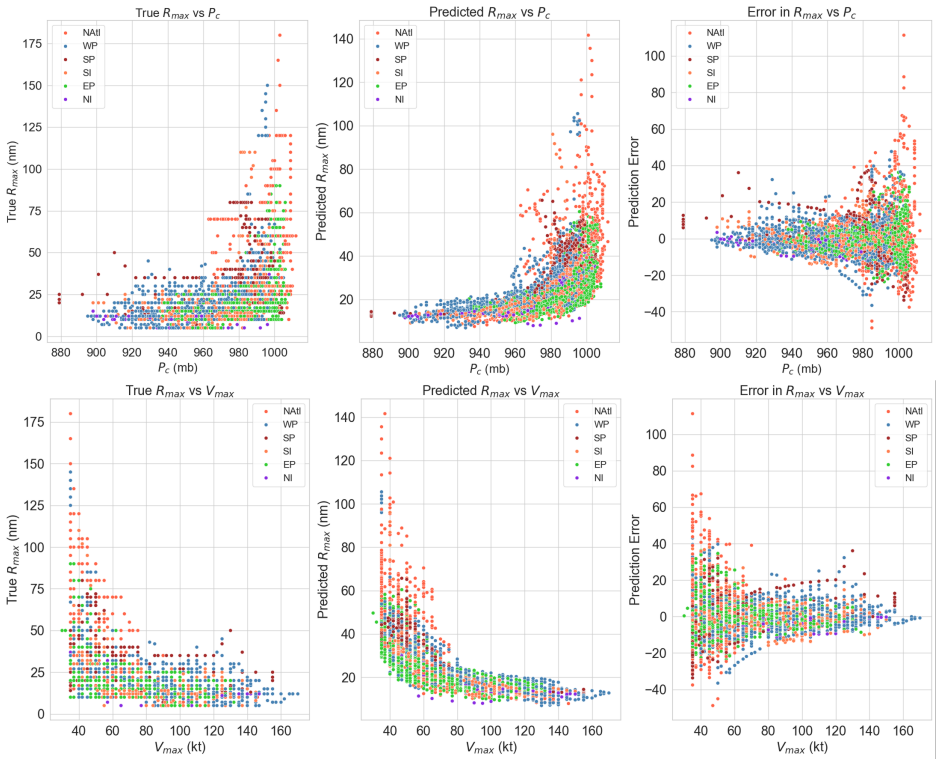

Figure 6: Relationship between true R_{max}, predicted R_{max} along with prediction error with P_c and V_{max} for the LSTM model using scaled input data. Points are colored by basin. WP: West North Pacific, EP: East North Pacific, NAtl: North Atlantic, SI: South Indian, SP: Southern Pacific, NI: North Indian.

4 CONCLUSION

This study evaluates the effectiveness of temporal DL models, such as LSTM networks and 1DCNN, for multivariate time series imputation of R_{max} against non temporal deep DL models. With reanalysis data from IBTrACS serving as training data for the models, we note that temporal models perform better than a shallow ANN while performing on par with XGBoost when R_{34} is excluded from the data. Performance is comparable between these models when R_{34} is included in the data, potentially because of the high correlation between R_{34} and R_{max}.

5 LIMITATIONS AND FUTURE STEPS

The temporal models in this study are sequence-to-sequence models (predicting R_{max} for entire 10-step blocks) that disregard existing observed R_{max} values that could improve subsequent predictions. Thus, the model ignores any existing R_{max} values that might be available for that block. Additionally, performance metrics show stochasticity across training instances—while temporal models consistently outperform ANN, XG-

Boost occasionally surpasses temporal models, likely due to our dataset's small sample size.

As an extension to this study, we are testing methods to handle data scarcity and improve generalization by exploring multiple data augmentation techniques, including pretraining temporal models on two synthetically generated tropical storm datasets and later fine-tuning on the IBTrACS reanalysis data, and implementing a rolling window approach, which increases the number of samples by generating overlapping blocks.

REFERENCES

[1] Al Kajbaf, A., and M. Bensi. Application of surrogate models in estimation of storm surge: A comparative assessment. *Applied Soft Computing*, 91:106184, 2020. https://doi.org/10.1016/j.asoc.2020.106184.

[2] Bergstra, J., and Y. Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13:281–305, 2012. https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf.

[3] Chavas, D. R., and K. A. Emanuel. A QuikSCAT climatology of tropical cyclone size. *Geophysical Research Letters*, 37(18), 2010. https://doi.org/10.1029/2010GL044558.

[4] Chavas, D. R., and J. A. Knaff. A simple model for predicting the hurricane radius of maximum wind from outer size. *Weather and Forecasting*, 37(5):563–579, 2022. https://doi.org/10.1175/WAF-D-21-0103.1.

[5] Chavas, D. R., N. Lin, and K. Emanuel. A Model for the Complete Radial Structure of the Tropical Cyclone Wind Field. Part I: Comparison with Observed Structure. *Journal of the Atmospheric Sciences*, 72(9):3647–3662, 2015. https://doi.org/10.1175/JAS-D-15-0014.1.

[6] Chen, T., and C. Guestrin. XGBoost: A Scalable Tree Boosting System. 2016. https://arxiv.org/pdf/1603.02754.

[7] Gao, S., et al. A Nowcasting Model for the Prediction of Typhoon Tracks Based on a Long Short Term Memory Neural Network. *Acta Oceanologica Sinica*, 37(5):8–12, 2018. https://doi.org/10.1007/s13131-018-1219-z.

[8] Hochreiter, S., and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. https://doi.org/10.1162/neco.1997.9.8.1735.

[9] National Centers for Environmental Information. International Best Track Archive for Climate Stewardship (IBTrACS), 2021. https://www.ncei.noaa.gov/products/international-best-track-archive.

[10] Jia, G., and A. A. Taflanidis. Kriging metamodeling for approximation of high-dimensional wave and surge responses in real-time storm/hurricane risk assessment. *Computer Methods in Applied Mechanics and Engineering*, 261:24–38, 2013.

[11] Lee, J.-W., et al. Rapid prediction of peak storm surge from tropical cyclone track time series using machine learning. *Coastal Engineering*, 170:104024, 2021. https://doi.org/10.1016/j.coastaleng.2021.104024.

[12] Liu, Z., A. Al Kajbaf, and M. Bensi. Applications of Statistical Learning Methods in Natural Hazard Assessment. PSAM17 & ASRAM24, Japan, 2024.

[13] Liu, Z., et al. Comparative Analysis of Joint Distribution Models for Tropical Cyclone Atmospheric Parameters in Probabilistic Coastal Hazard Analysis. *Stochastic Environmental Research and Risk Assessment*, 38:1741–1767, 2024. https://doi.org/10.1007/s00477-023-02652-5.

[14] Liu, Z., et al. A Multi-Tiered Bayesian Network Coastal Compound Flood Analysis Framework. Under review, 2025.

[15] Liu, Z., et al. A Bayesian Network Method for Deaggregation: Identification of Tropical Cyclones Driving Coastal Hazards. Under review, 2025.

[16] Liu, Z., et al. Machine Learning Motivated Data Imputation of Storm Data Used in Coastal Hazard Assessments. *Coastal Engineering*, 190, 2024. https://doi.org/10.1016/j.coastaleng.2024.104505.

[17] Nadal-Caraballo, N. C., et al. North Atlantic Coast Comprehensive Study–Coastal Storm Hazards from Virginia to Maine. US Army Engineer Research and Development Center, Vicksburg, MS, 2015.

[18] Nadal-Caraballo, N. C., et al. Coastal Hazards System–Louisiana (CHS-LA). US Army Engineer Research and Development Center, Vicksburg, MS, 2022.

[19] Tong, B., et al. Short-Term Prediction of the Intensity and Track of Tropical Cyclone via ConvLSTM Model. *Journal of Wind Engineering and Industrial Aerodynamics*, 226:105026, 2022. https://doi.org/10.1016/j.jweia.2022.105026.

[20] Vickery, P. J., and D. Wadhera. Statistical Models of Holland Pressure Profile Parameter and Radius to Maximum Winds of Hurricanes from Flight-Level Pressure and H*Wind Data. *Journal of Applied Meteorology and Climatology*, 47(10):2497–2517, 2008. https://doi.org/10.1175/2008JAMC1837.1.