Tropical Cyclone Track Prediction in the TCWC Indonesia Monitoring Area Using Deep Recurrent Neural Networks

Fakhrul Alam
Computer Science Department, BINUS Graduate Program-Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia
fakhrul.alam@binus.ac.id

Gede Putra Kusuma
Computer Science Department, BINUS Graduate Program-Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia
inegara@binus.edu

Ida Pramuwardani
Agency of Indonesian Meteorology, Climatology and Geophysics (BMKG), Jakarta, Indonesia
ida.pramuwardani@bmkg.go.id (corresponding author)

Received: 3 June 2025 | Revised: 4 September 2025 | Accepted: 11 September 2025

 DOI: https://doi.org/10.48084/etasr.12531

ABSTRACT

Tropical Cyclones (TCs) are rapidly rotating large-scale storm systems and rank as the second most destructive natural hazards after earthquakes. Disaster mitigation in TC-prone regions is critical, particularly in view of the two-fold increase in population in these areas over the past five decades. This research focuses on the Area of Monitoring (AoM) in Indonesia for TC, which spans from 20°N to 20°S latitude and from 90°E to 141°E longitude. The dataset is sourced from the International Best Track Archive for Climate Stewardship (IBTrACS), filtered to include only TCs that occur in this AoM region. The preprocessing involved using a sliding window with a sequence length of three to generate input features. Four Recurrent Neural Network (RNN) models were evaluated: Long Short-Term Memory (LSTM), bidirectional LSTM (BiLSTM), Gated Recurrent Unit (GRU), and Bidirectional GRU (BiGRU). All models were trained using incremental learning, comprising 158 iterations for the northern AoM region and 69 iterations for the southern AoM region. The evaluation results indicate that the LSTM model has high performance, with an average Mean Absolute Error (MAE) of 0.24485 degrees in the northern region and 0.22330 degrees in the southern region.

Keywords-tropical cyclone; track prediction; deep learning; recurrent neural network; disaster mitigation

I. INTRODUCTION

TCs are rapidly rotating, large-scale storms that rank as the second most destructive natural disasters after earthquakes. Over the past 50 years, 79,324 people have been victims, and economic losses have reached 1.4 trillion USD. In the last five decades, there has been a two-fold increase in the population living in TC-prone areas [1]. The average radius of a TC reaches 150 to 200 km. TCs form over vast ocean areas with warm sea surface temperatures exceeding 26.5°C. Wind speeds near the center of a TC can exceed 63 km/h (34 knots) [2]. Therefore, accurate prediction of TC is essential for early warning, prevention, and mitigation efforts [3]. Intense precipitation and strong, persistent winds associated with these systems can lead to significant damage to infrastructure and impede both social and economic development in affected regions [4]. TC observations involve various weather parameters, such as cloud patterns, sea surface temperature, air pressure, vertical shear, upper-level divergence, vorticity, atmospheric wave disturbances, and other weather parameters.

When a TC is located within the Area of Responsibility (AoR) of TCWC Jakarta, defined as 00°–10°S and 90°E–118°E, 00°–11°S and 118°E–125°E, and 00°–9°S and 125°E–141°E, observations are conducted every six hours (four times per day). This region is indicated in red in Figure 1. In contrast, if the TC is within the Area of Monitoring (AoM) of TCWC Jakarta, covering 20°N–20°S and 90°E–141°E but outside the AoR, observations are carried out every 12 hours (twice per day). These areas are shown in blue in Figure 1.

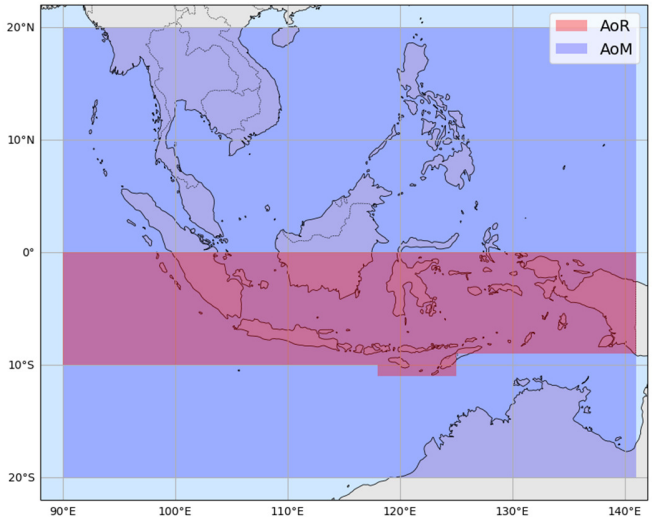

Fig. 1. AoR and AoM.

Several studies aimed to accurately predict TC tracks to enhance early warning systems and disaster mitigation. In [5], the RNN, LSTM, GRU, BiGRU, and BiGRU+ models were used for TCs in the Northwest Pacific Ocean from 1988 to 2017. BiGRU+ was the best model, with an MAE of 0.0199 in latitude and 0.0121 in longitude. This model has obvious advantages in mid- to long-term track forecasting, especially in the next 72 h. The study in [6] focused on the Northern Indian Ocean, using RNN, LSTM, and GRU models. The LSTM model demonstrated the best performance, achieving an MAE of 0.11 km, an RMSE of 0.14 km, and an MAPE of 0.61%. The dataset used in this study included features such as latitude, longitude, Sea Surface Temperature (SST), humidity, wind shear, and others. In addition, the research incorporated Chi-Square feature selection combined with a random forest classifier. In [7], the variational RNN provided a satisfactory approximation compared to conventional RNNs, while maintaining prediction accuracy. This study used RNN, Bayes-RNN, LSTM, and Bayes-LSTM models. The dataset regions included the North Indian Ocean, where Bayes-LSTM achieved a test ES of 0.006, the Northwest Pacific Ocean, with a Bayes-LSTM test ES of 0.011, the South Indian Ocean, with a Bayes-LSTM test ES of 0.007, and the South Pacific Ocean, with a Bayes-LSTM test ES of 0.014. The dataset included latitude, longitude, and maximum sustained wind speed.

Building on insights from prior research, this study utilizes latitude, longitude, storm speed (representing the translational velocity of tropical cyclones), and storm direction as input variables, with latitude and longitude as the predicted outputs. This study employs four RNN architectures for comparison: LSTM, BiLSTM, GRU, and BiGRU. In preprocessing, the sliding window method is implemented to capture localized temporal patterns within the sequential data. In addition, an incremental learning approach is adopted, which enables knowledge transfer from previously trained models to subsequent training stages, thereby enhancing model adaptability and efficiency.

Based on previous research, it is evident that RNN-based models, such as LSTM, BiLSTM, GRU, and BiGRU, can achieve satisfactory performance. However, none of these models has been applied to the AoM in Indonesia. Therefore, this research implements TC track prediction specifically for this region. In addition, TC track speed and cardinal direction data are incorporated into the learning dataset. This research formulates the problem of how RNN models can be applied to predict TC tracks, training them using incremental learning.

II. METHODOLOGY

This study aims to determine which model is most suitable for predicting TC tracks among RNN-based architectures, namely LSTM, GRU, BiLSTM, and BiGRU, using TC location data within the AoM of TCWC Indonesia.

A. Dataset

This step encompasses both the acquisition and the preparation of the data, including region-specific filtering, cleansing, and the application of the sliding window method. The dataset used in this investigation was obtained from the International Best Track Archive for Climate Stewardship (IBTrACS) [8]. The data were downloaded and filtered based on AoM. The TC dataset has been available since 1842 (downloaded on 22 May 2025 at 11:00 WIB) and contains 13,446 TCs comprising 718,315 data rows [9]. Of these, a total of 3,445 TCs occurred in the northern AoM (00°–10°N and 90°E–141°E), comprising 105,706 data rows, while 1,068 TCs occurred in the southern AoM (00°–10°S and 90°E–141°E), comprising 45,521 data rows. From the filtered AoM, the following parameters were extracted: SID (Storm Identification Number), SEASON (the year of the cyclone season), NUMBER (the sequence number of the cyclone in that season), BASIN (the climatological region in which the TC occurs), SUBBASIN (a smaller division of the basin), NAME (the TC's name), ISO_TIME (historical timestamp of TC), LAT (latitude), LON (longitude), WMO_WIND (maximum sustained wind speed), WMO_PRES (minimum central pressure), STORM_SPEED (translational speed of the cyclone), and STORM_DIR (translational direction of the cyclone). The TC parameters were then processed through a data cleansing stage to ensure data quality and consistency. The outcome of the preprocessing stage resulted in six main features: SID, ISO_TIME, LAT, LON, STORM_SPEED, and STORM_DIR. In previous state-of-the-art methods, latitude and longitude have been consistently included as essential features for TC track prediction.

B. Data Preparation

The next step involves applying a sliding window method, which is used across four RNN models. This method ensures that the input data remains sequentially connected and temporally consistent. This method is quite good at capturing the variation [10]. As illustrated in Figure 2, T1 represents the initial TC location, T2 the second step, and so on up to T6, which denotes the sixth step. T1, T2, and T3 are used as input to predict T4, and the process continues iteratively for subsequent predictions.

Following the data cleansing process, the datasets from both the northern and southern AoM were divided using a sliding window technique with a sequence length of 3. This resulted in 95,567 sliding windows for the northern AoM and 42,365 for the southern AoM. The method used in this study defines the input as consisting of four features: LAT, LON, STORM_SPEED, and STORM_DIR. The input has a shape of 3×4, corresponding to 3 time steps and 4 input features. The output includes only the LAT and LON coordinates, with a shape of (2), representing the predicted TC coordinate positions.

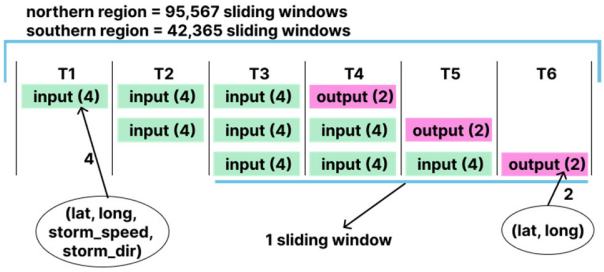

Fig. 2. Sliding windows method.

As shown in Figure 3, the dataset was divided into training, validation, and testing sets with a ratio of 60%, 20%, and 20%, respectively. The dataset was divided into 600 sliding windows for training data, 200 sliding windows for validation, and 200 sliding windows for testing data, to ensure that the model can learn well and be tested accurately. The first batch consists of sliding windows 1 to 1,000, the second batch spans from 601 to 1,600, the third batch from 1,201 to 2,200 sliding windows, and so on. This overlapping allows the model to be trained and evaluated iteratively on sequential segments of the dataset. Following the sliding window method, the datasets from both the northern and southern AoM were divided into batches; a total of 158 batches were generated from the northern AoM dataset and 69 batches from the southern AoM dataset.

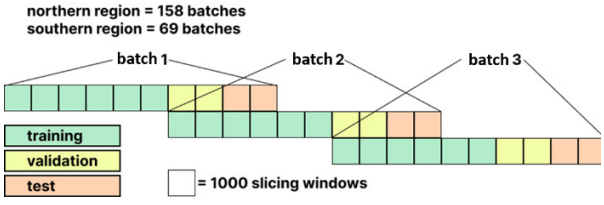

Fig. 3. Training, validation, and testing.

C. RNN Models

RNNs are deep learning algorithms designed for time series analysis, evolved to address the vanishing gradient problem commonly encountered in vanilla RNNs. LSTM and GRU models can retain information over longer sequences due to their more complex internal gating mechanisms compared to vanilla RNNs. This study employed bidirectional variants of LSTM and GRU, enabling the model to learn from both forward and backward temporal dependencies. Figure 4 illustrates the architecture of the proposed models.

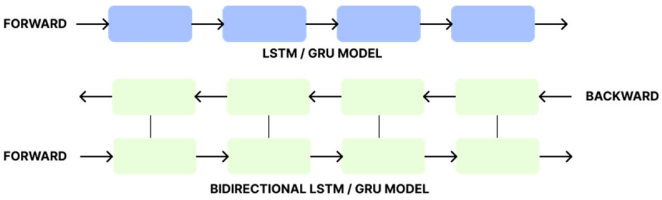

Fig. 4. LSTM/GRU vs Bidirectional LSTM/GRU.

1) Long Short-Term Memory (LSTM)

LSTM can solve complex time-series tasks that have never been solved by previous RNN algorithms [11]. LSTM can maintain values because it has more calculations compared to RNNs. In one LSTM cell, various calculations are performed using the sigmoid σ and tanh activation functions. The LSTM cell includes three primary gating mechanisms: a forget gate, an input gate, and an output gate. Following a series of experiments, an effective LSTM model architecture was established, presented in Table I. The LSTM layer employs 64 units with an input shape of 3×4. The number of trainable parameters in the LSTM layer is 17,664, while the dense layer contains 130 parameters. This LSTM configuration is considered optimal for capturing long-term dependencies in TC track prediction. Moreover, the input generated through the sliding window method has proven effective for learning short-term temporal patterns within the data.

TABLE I. ARCHITECTURE OF LSTM MODEL

Layer	Shape	Parameters
Input	3×4	
LSTM	64	17,664
Dense	2	130

2) Bidirectional Long Short-Term Memory (BiLSTM)

BiLSTM is an effective architecture compared to the vanilla RNNs in both speed and accuracy [12]. BiLSTM extends the LSTM architecture by utilizing two parallel LSTM layers: one processing the input sequence in the forward direction, while the other processing it in reverse. Although the core mechanism is the same as the standard LSTM, the final output is obtained by combining the outputs from both the forward and backward LSTM layers, allowing the model to capture contextual information from both past and future time steps. After experimentation, an effective BiLSTM model architecture was established, presented in Table II. The BiLSTM layer employs 128 units (2×64 units), comprising 64 units in each direction with an input shape of 3×4. The number of trainable parameters in the BiLSTM layer is 35,328, while the dense layer contains 258 parameters. This BiLSTM configuration is considered optimal for capturing long-term dependencies in TC track prediction from forward and backward.

TABLE II. ARCHITECTURE OF BILSTM MODEL

Layer	Shape	Parameters
Input	3×4	
BiLSTM	128	35,328
Dense	2	258

3) Gated Recurrent Unit (GRU)

GRU is a simplification of LSTM. GRU uses gates to control the flow of information without requiring a separate memory unit like LSTM. Architecturally, GRU is simpler than the LSTM architecture, but with the same goal, namely, to avoid vanishing gradients in time series [13]. In one GRU cell, various calculations are performed using the sigmoid σ and tanh activation functions. The GRU cell includes two primary gating mechanisms: the reset gate and the update gate [5]. Following experimentation, an effective GRU model architecture was established, presented in Table III. The GRU layer employs 64 units with an input shape of 3×4. The number of trainable parameters in the GRU layer is 13,440, while the dense layer contains 130 parameters. This GRU configuration is considered optimal for capturing long-term dependencies in TC track prediction. Moreover, the input generated through the sliding window method has proven effective for learning short-term temporal patterns within the data.

TABLE III. ARCHITECTURE OF GRU MODEL

Layer	Shape	Parameters
Input	3×4	
LSTM	64	13,440
Dense	2	130

4) Bidirectional Gated Recurrent Unit (BiGRU)

BiGRU extends the GRU architecture by utilizing two parallel GRU layers: one processing the input sequence in the forward direction, while the other processing it in reverse. Although the core mechanism is the same as the standard GRU, the final output is obtained by combining the outputs from both the forward and backward GRU layers, allowing the model to capture contextual information from both past and future time steps. Following a series of experiments, an effective BiGRU model architecture was established, as presented in Table IV. The BiGRU layer employs 128 units (2×64 units) with an input shape of 3×4. The number of trainable parameters in the BiGRU layer is 26,880, while the dense layer contains 258 parameters. This BiGRU configuration is considered optimal for capturing long-term dependencies in TC track prediction. Moreover, the input generated through the sliding window method has proven effective for learning short-term temporal patterns within the data.

TABLE IV. ARCHITECTURE OF BIGRU MODEL

Layer	Shape	Parameters
Input	3, 4	
BiGRU	128	26,880
Dense	2	258

D. Incremental Training

Incremental learning refers to learning from streaming data, with limited memory resources and, ideally, without sacrificing model accuracy [14]. The model generated at each incremental stage is subsequently used to initialize the training of the next stage. Pre-trained models are used for transfer learning or retrained with new data from the new batch.

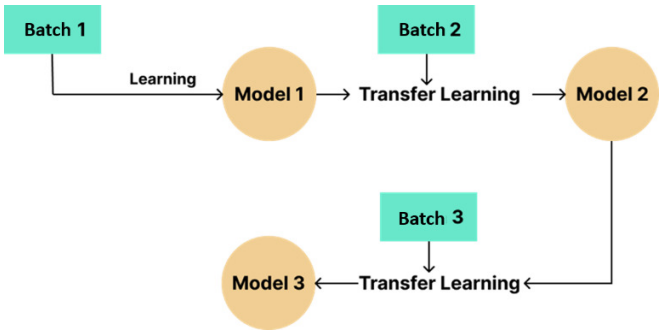

Fig. 5. Incremental learning.

Each model was trained using incremental learning, consisting of 158 batches for the northern region and 69 batches for the southern region. This resulted in a total of 908 incremental training iterations (transfer learning steps) for all models. Table V presents a summary of these training iterations.

TABLE V. INCREMENTAL TRAINING

Model	Northern	Southern
LSTM	158	69
BiLSTM	158	69
GRU	158	69
BiGRU	158	69

E. Hyperparameter Settings

Training parameters determine the success of training the model. Table VI summarizes the parameters for training the LSTM, BiLSTM, GRU, and BiGRU models.

TABLE VI. HYPERPARAMETERS SETTING

Parameter	Value
Learning rate scheduling	0.001 – 0.000001
Batch size	32
Epochs	500
Optimizer	Adam
Weight decay	0.0001
Gradient clipping	1.0
Early stop	20 epochs

Fine-tuning the learning rate is very important because the learning rate heuristic achieves remarkable success in stabilizing training, accelerating convergence, and improving generalization for adaptive stochastic optimization algorithms such as RMSProp and Adam [15]. The efficient use of vast batch sizes significantly reduces the number of parameter updates required to train a model [16]. Fine-tuning with Adaptive Moment estimation (Adam) performs optimization well compared to other optimizers such as SGD, RMSProp, Adagrad, and Adadelta. This optimizer requires little memory, being ideal for large-scale optimization, and exhibits fast and robust performance. Adam combines the benefits of SGD and RMSProp optimizers [17]. Numerical simulations have shown that weight decay can improve generalization in a feed-forward neural network [18].

III. RESULTS AND DISCUSSION

A. Training and Validation Result

The training and validation results based on the final loss value in each incremental training step are presented in the following figures. The x-axis in each graph represents the incremental training steps, with a maximum of 158 steps for the northern region (Figures 6, 8, 10, and 12) and 69 steps for the southern region (Figures 7, 9, 11, and 13). The light blue line represents the final training loss, the orange line represents the final validation loss, the dark blue dotted line indicates the average final training loss, and the red dotted line indicates the average final validation loss.

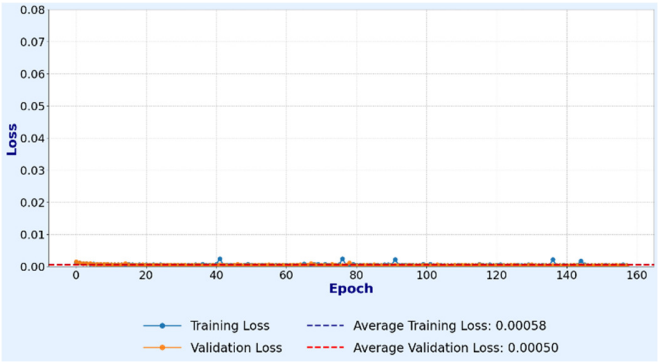

Fig. 6. Training and validation results of incremental learning for LSTM on Northern Region data.

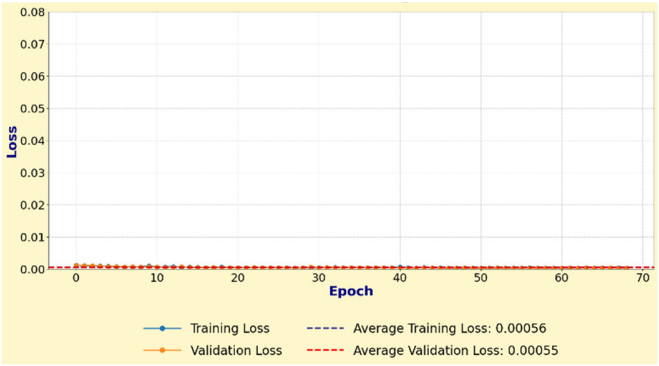

Fig. 7. Training and validation results of incremental learning for LSTM on Southern Region data.

LSTM exhibited an ideal incremental training pattern for both the northern and southern regions, as indicated by the consistently stable and nearly linear loss curves. All models achieved good results, with deviations within acceptable limits that did not significantly impact overall performance. Table VII shows the training and validation results.

The lowest average final training loss was achieved by the LSTM model in the southern region, with a value of 0.00056 on a normalized scale. Similarly, the lowest average final validation loss was also obtained by the LSTM model in the southern region, with a value of 0.00055 on a normalized scale.

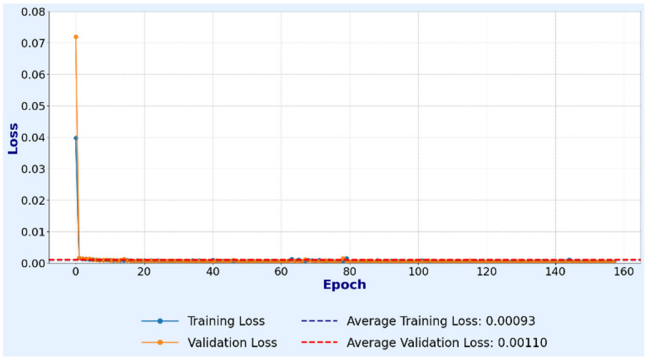

Fig. 8. Training and validation results of incremental learning for BiLSTM on Northern Region data.

Fig. 9. Training and validation results of incremental learning for BiLSTM on Southern Region data.

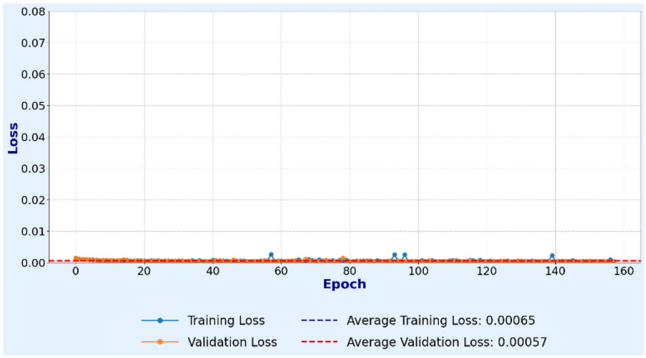

Fig. 10. Training and validation results of incremental learning for GRU on Northern Region data.

Fig. 11. Training and validation results of incremental learning for GRU Southern Region data.

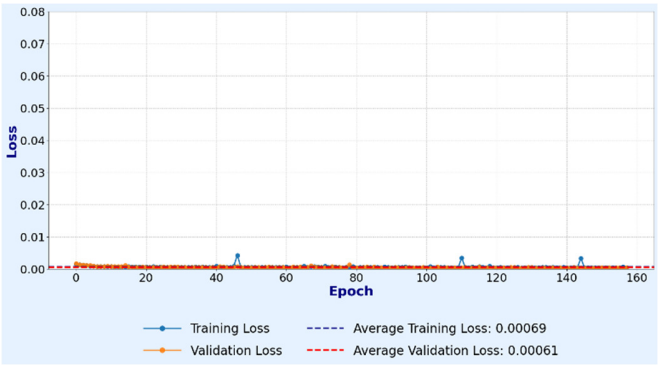

Fig. 12. Training and validation results of incremental learning for BiGRU on Northern Region data.

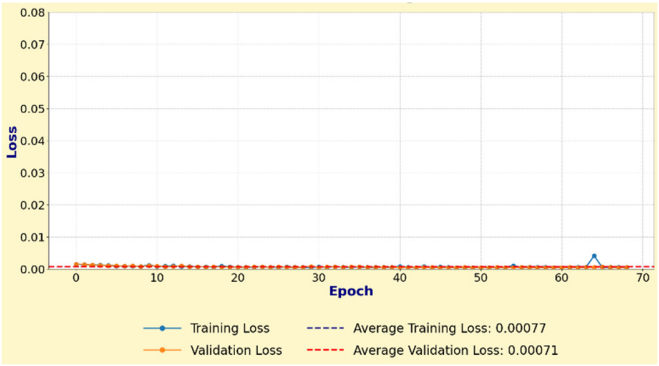

Fig. 13. Training and validation results of incremental learning for BiGRU on Southern Region data.

TABLE VII. TRAINING AND VALIDATION RESULTS

Model	Region	Avg. training loss	Avg validation loss
LSTM	Northern	0.00058	0.00050
LSTM	Southern	0.00056	0.00055
BiLSTM	Northern	0.00093	0.00110
BiLSTM	Southern	0.00075	0.00074
GRU	Northern	0.00065	0.00057
GRU	Southern	0.00059	0.00058
BiGRU	Northern	0.00069	0.00061
BiGRU	Southern	0.00077	0.00071

B. Testing Results

The results of the models were evaluated using Euclidean distance, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). Euclidean distance measures the geometric distance between two points in a multidimensional space as a straight-line distance [19]. RMSE and MAE are widely used to evaluate model performance [20].

Figure 14 shows that the lowest MAE was achieved by LSTM (Southern) at a value of 0.22330 degrees. Based on the MAE values obtained for both the Northern and Southern region models, the LSTM demonstrated the best overall performance in both. Figure 15 shows that the lowest RMSE was achieved by LSTM (Southern) with a value of 0.30815 degrees.

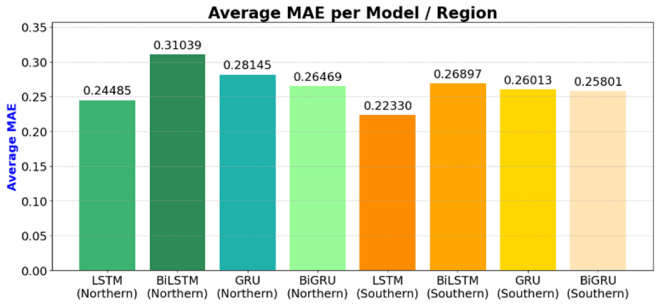

Fig. 14. Average MAE of models on testing data.

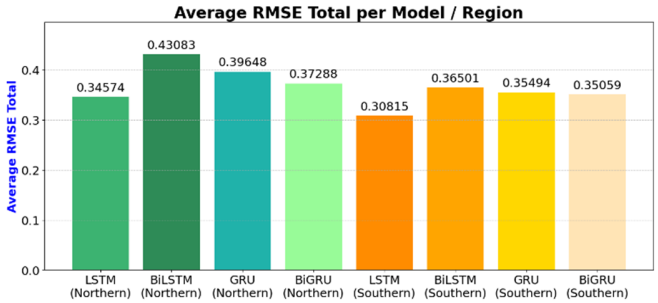

Fig. 15. Average RMSE of models on testing data.

Figure 16 shows that the lowest Euclidean distance was achieved by LSTM (Southern) with a value of 0.35397 degrees. Based on both the RMSE and Euclidean distance values obtained for both the Northern and Southern region models, the LSTM architecture demonstrated the best overall performance in both.

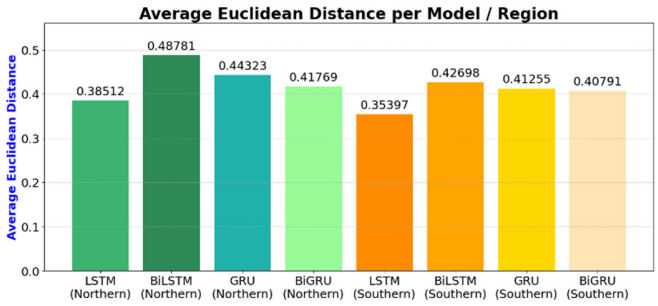

Fig. 16. Average Euclidean distance of models on testing data

C. Qualitative Analysis of TC Track Prediction

The evaluation results for all models indicate high predictive accuracy, with the predicted track lines closely following the observed cyclone tracks. In the following visualizations, the green dot represents the initial observation point, the red dot indicates the predicted location, and the blue line depicts the actual observed track. Figures 17-20 present the predicted track of TC Toraji within the AoM from 8 November 2024, 06 UTC, to 12 November 2024, 18 UTC. The predictions show relatively high accuracy compared to the observed (actual) TC track. Figures 21-24 illustrate the predicted track of TC Courtney with relatively high accuracy within the AoM from 24 March 2025, 12 UTC, to 29 March 2025, 12 UTC.

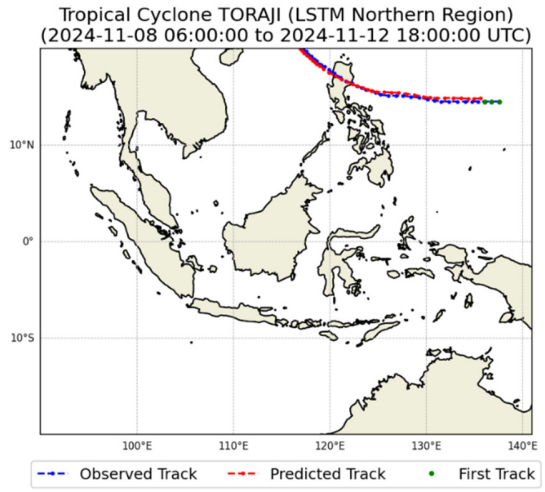

Fig. 17. Qualitative analysis of TC track prediction with LSTM in the Northern region.

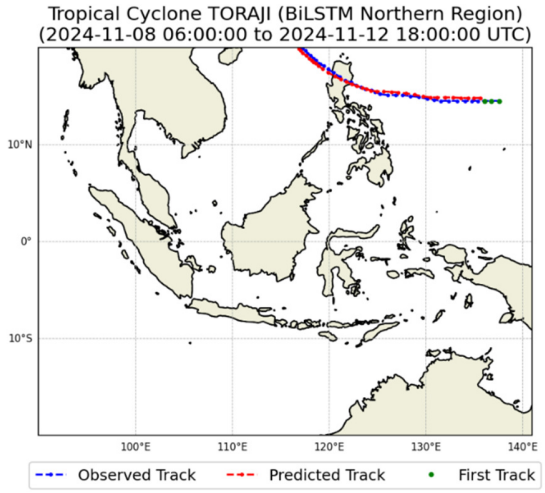

Fig. 18. Qualitative analysis of TC track prediction with BiLSTM in the Northern region.

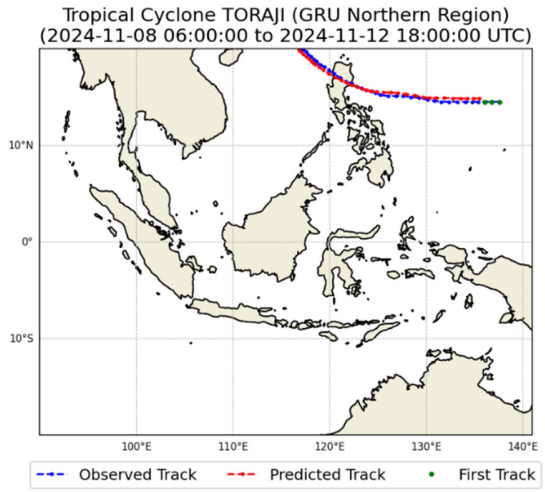

Fig. 19. Qualitative analysis of TC track prediction with GRU in the Northern region.

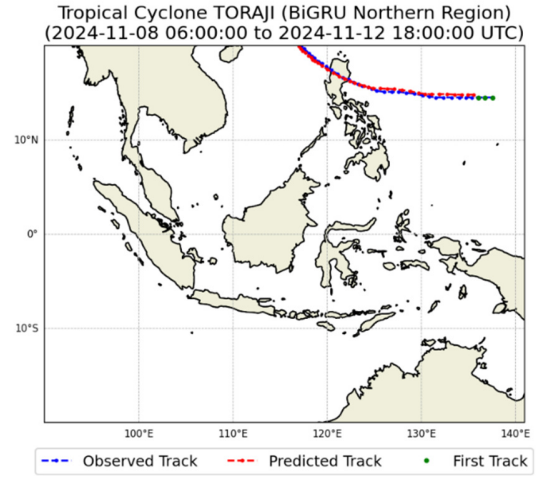

Fig. 20. Qualitative analysis of TC track prediction with BiGRU in the Northern region.

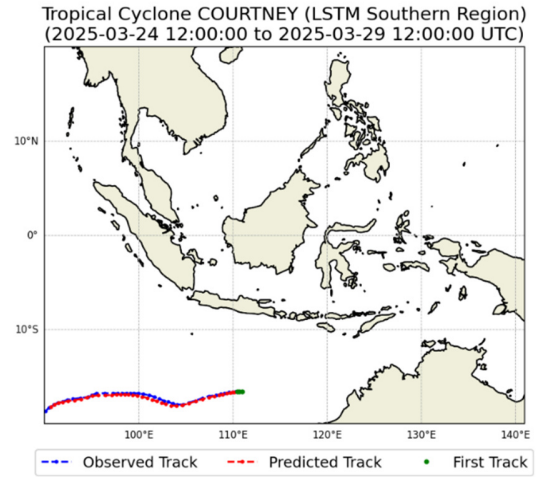

Fig. 21. Qualitative analysis of TC Track prediction with LSTM in the Southern region.

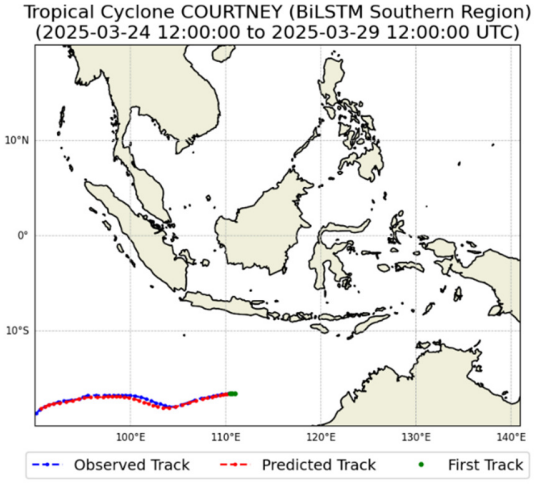

Fig. 22. Qualitative analysis of TC track prediction with BiLSTM in the Southern region.

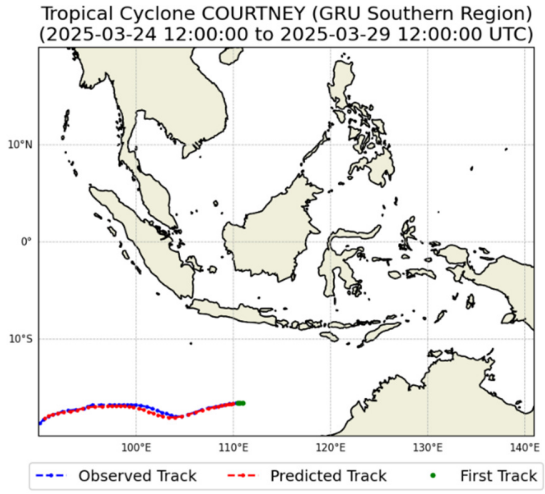

Fig. 23. Qualitative analysis of TC track prediction with GRU in the Southern region.

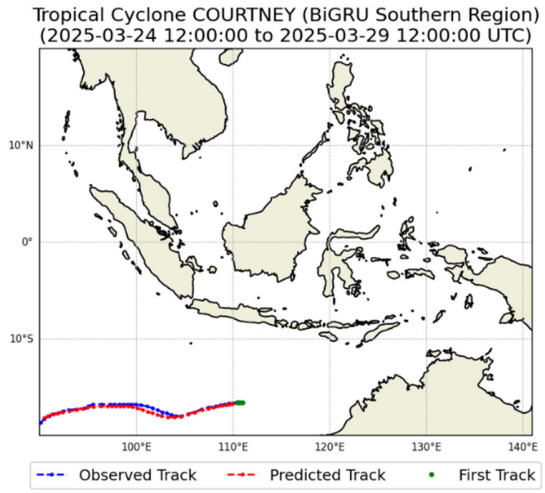

Fig. 24. Qualitative analysis of TC track prediction with BiGRU in the Southern region.

IV. CONCLUSION

This study performed preprocessing using the sliding window technique and employed incremental learning, which leverages transfer learning from previously trained models. For the Northern Indonesia AoM, transfer learning was applied across 157 iterations, resulting in a total of 158 learning processes. Similarly, for the Southern Indonesia AoM, transfer learning was applied across 68 iterations, leading to a total of 69 learning processes. Each RNN (LSTM, BiLSTM, GRU, and BiGRU) was trained separately for both the Northern and Southern regions, yielding a total of eight models (four for the North and four for the South). The performance of the models in predicting TC tracks showed that LSTM achieved the best performance. Specifically, LSTM achieved the lowest MAE, with values of 0.22330 degrees for the northern AoM region and 0.24485 degrees for the southern AoM region, the lowest RMSE, with values of 0.30815 degrees in the northern AoM and 0.34574 degrees in the southern AoM, and the lowest Euclidean distance, with values of 0.35397 degrees for the northern AoM and 0.38512 degrees for the southern AoM. However, all models had relatively high accuracy in approximating the center track of the observed TC. The results of this study are promising; however, the TC track is still derived from manually constructed time series data. Future work should consider developing this approach with images using semantic or instance segmentation methods.

The datasets used in this study are openly available from the International Best Track Archive for Climate Stewardship (IBTrACS) [9].

ACKNOWLEDGMENT

The authors gratefully acknowledge Bina Nusantara University, the Agency of Meteorology, Climatology, and Geophysics of Indonesia (BMKG), and the Ministry of Communication and Digital (Kementerian Komunikasi dan Digital) for awarding the Domestic Aspiration Scholarship (Beasiswa Aspirasi Dalam Negeri) through the KOMDIGI program. Their support and contributions have been instrumental in this research.

REFERENCES

[1] "Tropical cyclone," *World Meteorological Organization*, Dec. 16, 2022. https://wmo.int/topics/tropical-cyclone.

[2] K. Kiki, "Analisis musim siklon tropis 2021/2022 di Samudra Hindia Selatan Indonesia," *Megasains*, vol. 14, no. 1, Sep. 2023, https://doi.org/10.46824/megasains.v14i1.110.

[3] X. Wenwei *et al.*, "Deep Learning Experiments for Tropical Cyclone Intensity Forecasts," *Weather and Forecasting*, Jun. 2021, https://doi.org/10.1175/WAF-D-20-0104.1.

[4] S. Jayasree and K. R. Ananthapadmanaban, "Discrete Migratory Bird Optimizer with Deep Learning Driven Cyclone Intensity Prediction on Remote Sensing Images," *Engineering, Technology & Applied Science Research*, vol. 15, no. 2, pp. 21605–21610, Apr. 2025, https://doi.org/10.48084/etasr.8842.

[5] T. Song, Y. Li, F. Meng, P. Xie, and D. Xu, "A Novel Deep Learning Model by BiGRU with Attention Mechanism for Tropical Cyclone Track Prediction in the Northwest Pacific," *Journal of Applied Meteorology and Climatology*, vol. 61, no. 1, pp. 3–12, Jan. 2022, https://doi.org/10.1175/JAMC-D-20-0291.1.

[6] S. Rahman, M. Fahim Faisal, P. Mondal, and Md. Mahfuzur Rahman, "Tropical cyclone track prediction harnessing deep learning algorithms: A comparative study on the Northern Indian Ocean," *Results in Engineering*, vol. 26, Jun. 2025, Art. no. 105009, https://doi.org/10.1016/j.rineng.2025.105009.

[7] A. Kapoor, A. Negi, L. Marshall, and R. Chandra, "Cyclone trajectory and intensity prediction with uncertainty quantification using variational recurrent neural networks," *Environmental Modelling & Software*, vol. 162, Apr. 2023, Art. no. 105654, https://doi.org/10.1016/j.envsoft.2023.105654.

[8] K. R. Knapp, M. C. Kruk, D. H. Levinson, H. J. Diamond, and C. J. Neumann, "The International Best Track Archive for Climate Stewardship (IBTrACS): Unifying Tropical Cyclone Data," *Bulletin of the American Meteorological Society*, vol. 91, no. 3, pp. 363–376, Mar. 2010, https://doi.org/10.1175/2009BAMS2755.1.

[9] "International Best Track Archive for Climate Stewardship (IBTrACS)," *National Centers for Environmental Information (NCEI)*, Jul. 17, 2024. https://www.ncei.noaa.gov/products/international-best-track-archive.

[10] P. Kapoor and S. S. Bedi, "Weather Forecasting Using Sliding Window Algorithm," *ISRN Signal Processing*, vol. 2013, pp. 1–5, Dec. 2013, https://doi.org/10.1155/2013/156540.

[11] H. Cen, G. Han, X. Lin, Y. Liu, and H. Zhang, "Deep learning-based forecasting model for chlorophyll-a response to tropical cyclones in the

Western North Pacific," *Ocean Modelling*, vol. 189, Jun. 2024, Art. no. 102345, https://doi.org/10.1016/j.ocemod.2024.102345.

[12] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5–6, pp. 602–610, Jul. 2005, https://doi.org/10.1016/j.neunet.2005.06.042.

[13] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation." arXiv, Sep. 03, 2014, https://doi.org/10.48550/arXiv.1406.1078.

[14] A. Gepperth and B. Hammer, "Incremental learning algorithms and applications," in *European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, 2016.

[15] L. Liu *et al.*, "On the Variance of the Adaptive Learning Rate and Beyond." arXiv, Oct. 26, 2021, https://doi.org/10.48550/arXiv.1908.03265.

[16] S. L. Smith, P. J. Kindermans, C. Ying, and Q. V. Le, "Don't Decay the Learning Rate, Increase the Batch Size." arXiv, Feb. 24, 2018, https://doi.org/10.48550/arXiv.1711.00489.

[17] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization." arXiv, Jan. 30, 2017, https://doi.org/10.48550/arXiv.1412.6980.

[18] A. Krogh and J. Hertz, "A simple weight decay can improve generalization," *Advances in neural information processing systems*, vol. 4, 1991.

[19] P. E. Danielsson, "Euclidean distance mapping," *Computer Graphics and Image Processing*, vol. 14, no. 3, pp. 227–248, Nov. 1980, https://doi.org/10.1016/0146-664X(80)90054-4.

[20] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geoscientific Model Development*, vol. 15, no. 14, pp. 5481–5487, Jul. 2022, https://doi.org/10.5194/gmd-15-5481-2022.