International Journal of Offshore and Coastal Engineering
Vol.10 | No. 1 | pp. 1 – 6 | May 2026
e-ISSN: 2580-0914

Submitted: Desember 8, 2025 | Revised: January 2, 2026 | Accepted: February 14, 2026

Wave Simulation Based Recurrent Neural Network Long-Short Term Memory (RNN-LSTM)

Wimala L. Dhanistha [a*], Fajar Sebastian S [a]., Haryo D. Armono [a],

[a)] Department of Ocean Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
Corresponding author: wimala.danistha@its.ac.id

ABSTRACT
The Recurrent Neural Network Long-Short Term Memory (RNN-LSTM) algorithm was used to perform 6-hour time series forecasting for the variables significant wave height (Hs), maximum wave height (Hmax), zero-crossing period (Tz), and peak period (Tp) in a sea wave simulation using five datasets, each with a duration of ten years and from various locations. The ability of the RNN-LSTM algorithm to simulate sequential data, including time series, led to its selection. According to the research findings, the five RNN-LSTM simulations produced fairly good forecasts for Tz (MAPE below 11%) and good forecasts for Hs and Hmax (MAPE below 9%). However, they had trouble simulating Tp (MAPE up to 34.06%). The tendency for Tp data to have a higher standard deviation (up to 2) than other variables and Tp's lower correlation with the other variables are factors that make Tp forecasting challenging. We find that RNN-LSTM can forecast Tp with moderate accuracy, but it can produce dependable forecasts for Hs, Hmax, and Tz
Keywords: *LSTM, RNN , Wave Simulation*

1. INTRODUCTION

Accurately understanding wave characteristics (such as height and period) at all times is key to ensuring safety in navigation, fishing activities, weather forecasting, and various other marine engineering considerations. Therefore, wave height forecasting is necessary. Various methods are used to predict wave height, ranging from hydrodynamic models such as Simulating Waves Nearshore (SWAN) and WAVEWATCH III, wave energy models based on the principle of energy conservation to calculate wave height based on wind and current conditions, statistical methods that use historical wave height data for forecasting, and machine learning methods [1, 2].
With the availability of accurate databases and the increasing volume of historical data, there is an argument that data-driven models using Machine Learning (ML) and Artificial Intelligence (AI) can act as replacements for physical hydrodynamic models, offering shorter computation time and superior accuracy [3, 4]. Technological advances in Machine Learning have proven to be a solution for wave forecasting, even capable of rivaling and improving earlier hydrodynamic models such as SWAN [5].

This research explores the use of ML with the Recurrent Neural Network (RNN) Long Short-Term Memory (LSTM) algorithm for wave forecasting [6]. In this research, input data was selected from five different marine locations using real buoy data. A simulation will be conducted to examine the performance of this algorithm.
This research simulation will assess the accuracy and practicality of the algorithm. Real data will be compared with the simulation results to measure accuracy. The data variables used as input are historical data with a 6-hour timestep including significant wave height (Hs), maximum wave height (Hmax), peak period (Tp), and zero-crossing period (Tz). Accuracy will be measured using the Mean Absolute Percentage Error (MAPE) metric. The results of the study will demonstrate the performance of RNN-LSTM in different marine conditions and data scenarios.

Coulomb Friction theory explains the interaction between pipelines and soil to prevent lateral displacement. Until the 1970s, Coulomb Friction theory was the only method to estimate soil resistance against pipeline displacement due to hydrodynamic loads. Wagner et al [8] developed Coulomb Friction theory into an empirical model for pipelines, where total lateral resistance is assumed as the sum of Coulomb Friction and soil passive resistance components. Research indicates that design methods based on Coulomb Friction theory are often overly conservative.
In the analysis of subsea pipeline stability as referenced in DNVGL RP F109, there are three methods for lateral stability analysis: General Lateral Stability (GLS), Absolute Static Lateral Stability (ALSS), and Dynamic Lateral Stability (DLS). Dynamic analysis imposes more complex requirements, such as considering the pipeline's response to combinations of hydrodynamic loads from random waves, steady currents, and soil resistance in the time domain to

simulate factual outcomes.

2. LITERATURE REVIEW

The method used in analyzing wave simulation. The steps taken in this research can be explained as follows:

2.1 Wave Records as Time Series

Wave records are typically represented as time series, as illustrated in Figure 1, where the vertical axis is expressed in meters and the horizontal axis in seconds. Wave crests are indicated by dashed lines, and all zero down-crossing points are marked with circles. The wave period (T) refers to the time interval between two consecutive zero down-crossings (or up-crossings, depending on the plotting convention), while the wave height (H) is the vertical distance from a trough to the subsequent crest as observed in the record. A more widely used metric is the zero-crossing wave height (Hz), defined as the vertical distance between the maximum and minimum values within the interval between two zero down-crossings (or up-crossings).

When the record contains waves of varying periods, the number of crests may exceed the number of zero down-crossings, resulting in differences between crest-to-trough wave heights and Hz. Due to the inherently random nature of the sea surface, measured wave records never repeat exactly. However, if the sea state is considered stationary, the statistical properties of the wave period and height distributions will remain consistent across records. Consequently, statistical parameters are the most suitable descriptors for characterizing sea states based on measured wave data. Several commonly used parameters are presented in the formulations below.

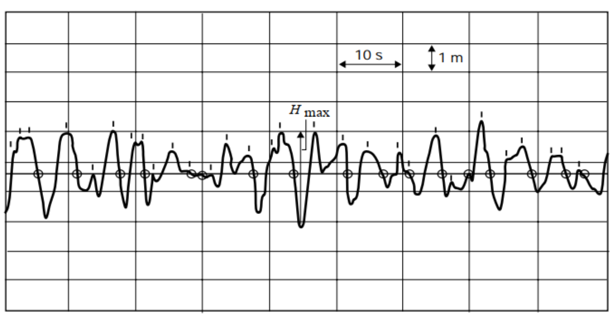

Figure 1. Wave records represented as time series [7].

2.2 Recurrent Neural Network

The Recurrent Neural Network (RNN) overcomes this challenge by unrolling the network across multiple time steps, adding successive layers, and recalculating prediction errors, thereby forming a deep neural network. In this framework, connections between nodes in the hidden layer create a directed graph along the temporal sequence, enabling information to persist over time. This temporal dependency introduces a form of memory within the RNN, allowing the architecture to retain and utilize contextual information. Consequently, each layer can receive input not only from the current state but also from several preceding layers in the sequence.

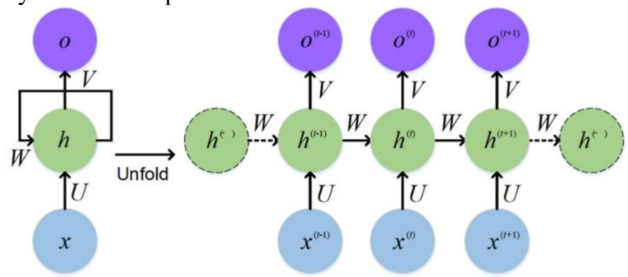

Figure 2. RNN with hyperbolic tangent as the activation function [6].

The Long Short-Term Memory (LSTM) network is an advanced variant of the RNN designed to overcome the limitations of standard RNNs in capturing long-term dependencies due to the vanishing or exploding gradient problem. LSTM introduces a memory cell that can preserve information over extended time periods and employs three primary gating mechanisms—input gate, forget gate, and output gate to regulate the flow of information.

The input gate controls the extent to which new information is stored in the cell state, the forget gate determines which information from the previous cell state should be discarded, and the output gate regulates the information that is passed on to the next hidden state. Through this gating structure, LSTMs are capable of learning both short-term and long-term temporal dependencies more effectively than conventional RNNs. As a result, LSTMs are widely applied in sequence modeling tasks such as time-series prediction, natural language processing, and speech recognition, where capturing contextual patterns over varying time scales is essential.

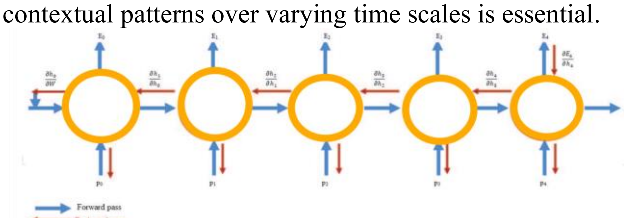

Figure 3. An unrolled RNN illustrating the flow of information from one input element in a sequence to the next [6].

3. ANALYSIS AND DISCUSSION

3.1 Collection of 5 Datasets

The collected data consists of time series wave data from real ocean wave measurements at 5 buoys in Australia, each with a minimum data duration of 10 years. The location details are provided in Table 1 and mapped in Figure 4. The data characteristics can be seen in Table 2. The real data obtained has a timestep of either 15 minutes, 30 minutes, or 1 hour. In addition to Hs, Hmax, Tz, and Tp, the dataset also includes wave direction angle and sea surface temperature;

however, these two variables were not used in this research.

Table 1. The Buoy Locations That Serve as the Sources of Real Data in this Study

No	Location	Coordinate
1	Abbot Point	-19,8605670; 148.0943330
2	Moreton Bay	-26,8999000; 153.28151670
3	Mooloolaba	-26,5660500; 153.18248300
4	Townsvile	-19,1761000; 147.07486700
5	Cairns	-16,7305; 145.7125951

Table 2. The Characteristics of Each Dataset

No	*Hs* (m)	*Hmax* (m)	*Tz Avg** (m)	*Tp Avg** (m)
1	0.579	4.777	3.209	4.182
2	0.360	2.953	2.632	3.161
3	1.120	9.042	5.346	8.850
4	0.696	5.503	3.484	4.932
5	0.506	3.320	3.232	4.902

[a]*Avg* – Average.

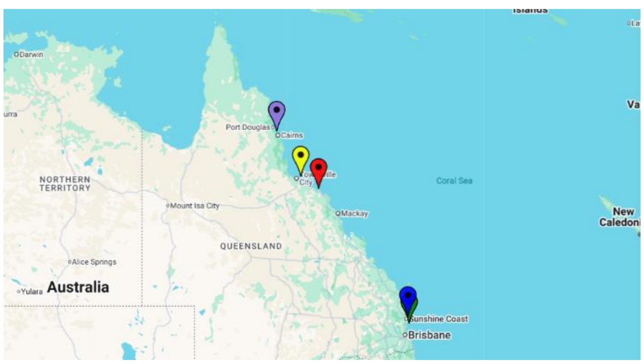

Figure 4. Map of buoy locations used as data sources, as listed in Table 1

3.2 Design and Simulation of 5 Machine Learning

The simulation in this study uses the Machine Learning algorithm RNN LSTM with 6 layers, along with the number of neurons, which are specifically detailed in the list below:

- Input (30)
- LSTM (32)
- LSTM (16)
- *Dropout* (16)
- LSTM (10)
- *Dense* (4)

In terms of input and output, the Machine Learning model used follows the structure shown in Figure 5, where the notation 0 (e.g., Hs0) indicates the current timestep, notation 1 refers to a future timestep, and negative notation refers to past timesteps. The ML model receives input in the form of 4 variables (often called features) over the past 30 timesteps. The hidden layers are trained to produce the output. The output consists of the values of the 4 variables at 1 timestep in the future. The list below describes the datasets based on the quantity and structure of the input and output data [8,9] :

- X_train set : (14430, 4, 30)
- Y_train set : (14430, 4, 1)
- X_test set : (120, 4, 30)
- Y_test set : (120, 4, 1)

X is the input, while Y is the output. The train set is used for training and loss validation, whereas the test set is used for simulation. In the first dimension, the 14,430 training data points are obtained from (10 years * 365 days/year * 4 timesteps/day) – 120 timesteps. The last 120 timesteps are used as test data. This study applies a training/validation split with a ratio of 75% / 25%. The training data and training loss validation data are randomly selected by the machine learning model using a 75% / 25% ratio from the X_train set..

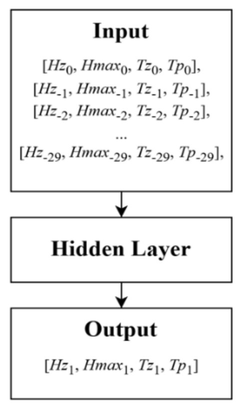

Figure 5. Input and Output Representation of the RNN-LSTM Used.

The data from the source was cleaned and resampled into a 6-hour timestep. Resampling was performed using the average, except for Hmax, which used the maximum value. After all the above steps were completed, the resulting data is shown in Figure 6. Min-Max Scaler was applied to the dataset before training the Machine Learning model. The Min-Max Scaler transforms the data scale into a range of 0–1 based on the minimum and maximum values, so that the 4 input variables fall within the 0–1 range. This facilitates the training process in Machine Learning. After training is completed, the Machine Learning model can be used to generate simulations on the test data. The simulated results are then converted back to their original scale using the Inverse Scaler. The simulation is performed using the trained model and the test input data. The simulation type is single step or T+1, where T is the current timestep, and T+1 is one timestep in the future. The simulation results are plotted on a graph, where the simulation output is compared to the actual test data, as shown in Figure 10.

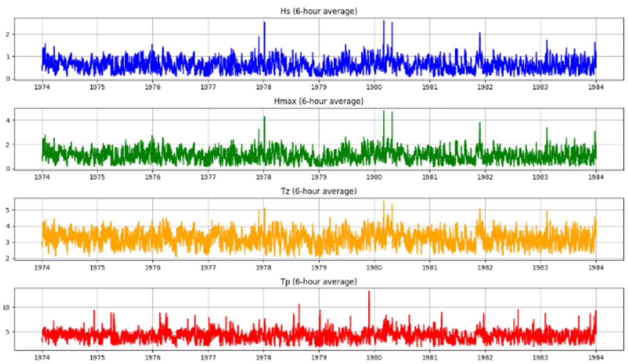

Figure 6. Dataset number 1 after being cleaned and resampled into a 6-hour timestep.

3.3 Result Analysis

All The parameter used in this study is MAPE. MAPE indicates the error in percentage form, making it easier to compare errors across different simulations. MAPE is applied to the simulation results generated by the trained RNN LSTM model using input data from the test set.

After the simulation is completed, MAPE evaluation is carried out. The validation requirement in this study is that the MAPE for each simulation across all output variables must be below 40% [10]. Several trials were conducted to improve accuracy until the validation criteria were met.

Table 3. Simulation Results from the 5 Datasets.

No	MAPE *Hs*	MAPE *Hmax*	MAPE *Tz*	MAPE *Tp*
1	5.67%	5.58%	6.57%	12.25%
2	7.37%	7.33%	8.39%	18.96%
3	3.35%	3.58%	9.14%	12.73%
4	7.31%	7.11%	7.50%	17.28%
5	5.72%	6.20%	10.55%	34.06%

After training the 5 Machine Learning models, 5 simulations were carried out, resulting in error values shown in Table 3. In analyzing the results of this study, several aspects and correlations can be considered to better understand the simulation outcomes. These aspects include:

- Simulation input data graph
- Correlation matrix
- Train vs validation loss
- Characteristics of the simulation data
- Simulation result graphs
- Simulation MAPE results

The input data graph, as shown in Figure 6, provides an overview of the data distribution that can be correlated with the resulting MAPE. Seasonal patterns, long-term trends, and fluctuations can be observed in this graph. Additionally, the input data graph also indicates the presence and magnitude of outliers in the data. Outliers are difficult to simulate and can explain cases of overshooting or undershooting in the simulation results when compared to real data. If the input data appears stable, the model is more likely to produce accurate predictions.

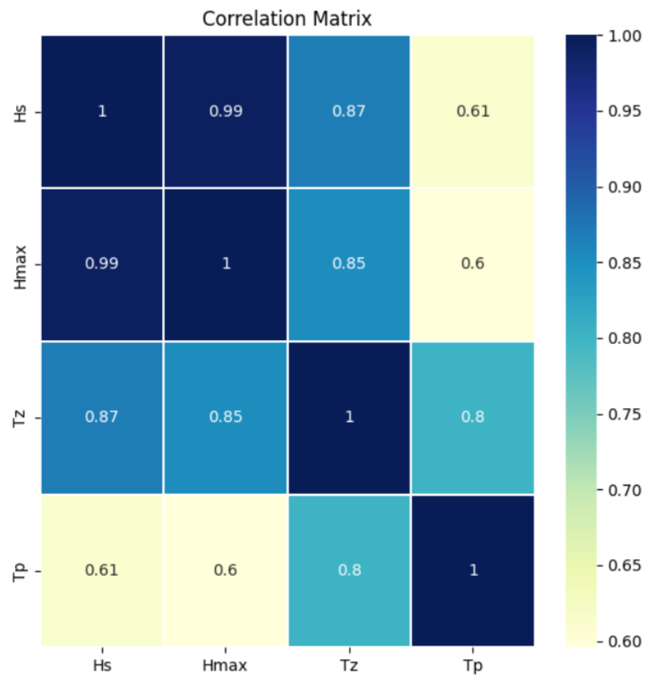

Figure 7. Correlation matrix of input data from dataset number 1

The correlation matrix (as shown in Figure 7) is important to analyze because it shows the correlation between each variable. The correlation of each input variable with other input variables affects the accuracy or error of each output result. Each cell in the correlation matrix can have a value ranging from 1.00 to -1.00. A value of 1.00 indicates a perfect positive correlation, where both variables increase or decrease together in perfect alignment according to their scale in the time series data. A value of -1.00 indicates a perfect negative correlation, where the variables move in completely opposite directions.

In Machine Learning, variables with correlations close to 1.00 or -1.00 are considered highly correlated and are very helpful in training the model to produce low-error outputs [11]. High correlation indicates that the model can use information from one variable to predict another. One of the reasons an output variable may have a high error is the absence of any input variables that are strongly correlated with it. The higher the correlation among input variables, the more effectively RNN-LSTM can learn internal representations. Correlation analysis also helps in selecting input and output variables by avoiding inclusion of irrelevant or unhelpful parameters.

The train vs validation loss graph indicates whether the model is experiencing overfitting, underfitting, or is well-trained. If the training loss decreases and the validation loss also decreases, it means the model is learning well. If the training loss decreases but the validation loss increases, overfitting is occurring. If both losses remain high and stagnant, it indicates underfitting, or that the data is too complex for the Machine Learning model to learn effectively. A model with balanced loss between training and validation is more likely to produce predictions with low MAPE and simulation graphs that closely match the actual data.

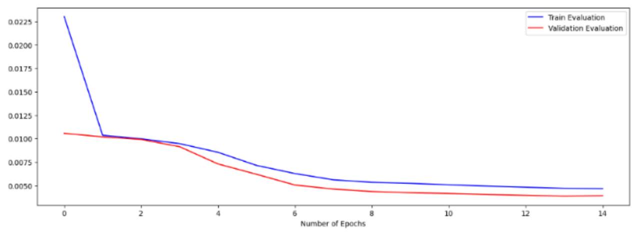

Figure 8. Train vs Validation loss from simulation 1

	Hs	Hmax	Tz	Tp
count	14610.000000	14610.000000	14610.000000	14610.000000
mean	0.578971	1.027856	3.208950	4.182462
std	0.271489	0.477295	0.436153	0.933770
min	0.070000	0.123333	2.046667	1.803333
25%	0.366667	0.656667	2.890000	3.546667
50%	0.553333	0.980000	3.196667	4.200000
75%	0.756667	1.336667	3.510000	4.763333
max	2.600000	4.776667	5.563333	13.290000

Figure 9. Data characteristics from simulation 1

Data characteristics (as shown in Figure 9), such as standard deviation (std), median (50%), maximum value (max), and mean, can provide insights into the nature of the data and its correlation with simulation results. A high standard deviation indicates large data fluctuations, which can make training more difficult. A low standard deviation suggests that the data is relatively stable and potentially easier to learn. The median can be used to assess data skewness; if the median is far from the mean, the distribution may not be normal. An excessively high maximum value may also indicate the presence of outliers or extreme events (such as ocean storms).

Simulation number 3 achieved the best MAPE. As shown in Table 3, the simulation results for Hs and Hmax consistently yielded low error, demonstrated by MAPE values below 10% and closely aligned MAPE across all simulations. The good MAPE performance for Hs and Hmax is partly influenced by the strong correlation between the two variables (as seen in Figure 7). In contrast, Tp consistently recorded MAPE values above 10% (as illustrated in Figure 10), which can be explained by the tendency of Tp data to have a higher standard deviation compared to other variables (especially in simulation number 5) and its weak correlation with the other variables (as shown in Figure 7).

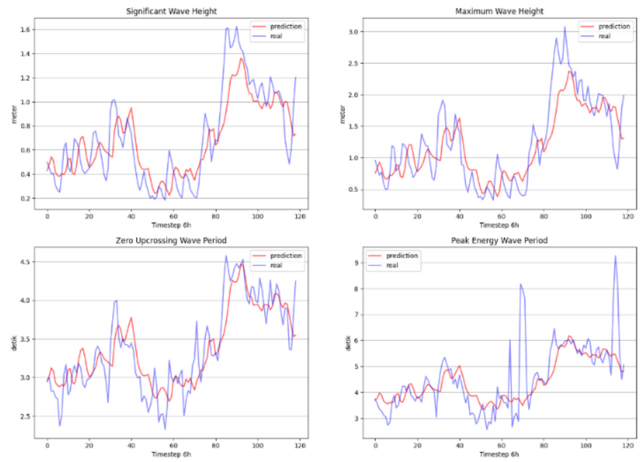

Figure 10. Comparison between real data and simulation results on the test set of dataset number 1.

4. CONCLUSIONS

From the five simulation results, the following conclusions were obtained:

1. RNN-LSTM is capable of producing good simulations for Hs and Hmax (MAPE below 10%), good results for Tz (MAPE below 11%), and fair results for Tp (MAPE up to 34.06%).
2. Within a 10-year dataset range at a 6-hour timestep (total of 14,550 data points), the four wave parameters can be simulated with good to fair accuracy (MAPE ranging from 3.6% to 34.06%).
3. The lower the standard deviation of a dataset and the stronger the correlation of a variable with other variables, the higher the simulation accuracy achieved for that variable.

REFERENCES

[1] Boukabara, S. A., Krasnopolsky, V., Stewart, J. Q., Maddy, E. S., Shahroudi, N., & Hoffman, R. N. (2019). Leveraging modern artificial intelligence for remote sensing and NWP benefits and challenges. *Bulletin of the American Meteorological Society*, *100*(12), ES473–ES491. https://doi.org/10.1175/BAMS-D-18-0324.1

[2] Boukabara, S. A., & Hoffman, R. N. (2022). Optimizing Observing Systems Using ASPEN: An Analysis Tool to Assess the Benefit and Cost Effectiveness of Observations to Earth System Applications. *Bulletin of the American Meteorological Society*, *103*(10), E2417–E2439. https://doi.org/10.1175/BAMS-D-22-0004.1

[3] Zhang, X., Li, Y., Gao, S., & Ren, P. (2021). Ocean wave height series prediction with numerical long short-term memory. *Journal of Marine Science and Engineering*, *9*(5). https://doi.org/10.3390/jmse9050514

[4] Minuzzi, F. C., & Farina, L. (2023). A deep learning approach to predict significant wave height using long short-term memory. *Ocean Modelling*, *181*. https://doi.org/10.1016/j.ocemod.2022.102151

[5] Khatimah, H., Jaya, I., & Atmadipoera, A. S. (n.d.). Significant Wave Height Forecasting using Long-Short Term Memory (LSTM) in Seribu Island Waters. *IJCCS*, *x, No.x*, 1–5.

[6] H. Okut, "Deep Learning for Subtyping and Prediction of Diseases: Long-Short Term Memory," in Artificial Neural Network and Deep Learning, IntechOpen, 2022.

[7] Guodong, Q., & Zhongxian, C. (2023). A Comparison of Ocean Wave Height Forecasting Methods for Ocean Wave Energy Conversion Systems. *Water (Switzerland)*, *15*(18). https://doi.org/10.3390/w15183256

[8] Makarynskyy, O. (2004). Improving wave predictions with artificial neural networks. *Ocean Engineering Journal*, *31*, 5–6.

[9] Dhanistha, W. L., Atmoko, R. A., Juniarko, P., & Akbar, R. (2017). Prediction of Significant Wave Height Using Neural Network in the Java Sea (North of Surabaya). *Applied Mechanics and Materials*, *862*, 72–77. https://doi.org/10.4028/www.scientific.net/amm.862.72

[10] Aisjah, A. S., & Arifin, S. (2013). Maritime weather prediction using fuzzy logic in Java Sea for shipping feasibility. *International Journal of Artificial Intelligence*, *10*(13 S), 112–122

[11] Dhanistha, W. L., Silvianita, S., & Wardhana, W. (2018). Wave height prediction based fuzzy logic. *International Journal of Civil Engineering and Technology*, *9*(9), 1611–1616