Forecasting of Sea Level Time Series using Deep Learning RNN, LSTM, and BiLSTM, Case Study in Jakarta Bay, Indonesia

Fikhri Masri[1], Deni Saepudin[2], Didit Adytia[3]

[1,2,3] Fakultas Informatika, Universitas Telkom, Bandung
[1]fikhrimasri@student.telkomuniversity.ac.id, [2]denisaepudin@telkomuniversity.ac.id,
[3]adytia@telkomuniversity.ac.id

Abstrak

Peramalan permukaan laut sangat penting dalam aktivitas di daerah pesisir, seperti pada bidang engineering dan juga dapat digunakan untuk menentukan navigasi pelayaran kapal. Selain itu, dapat digunakan untuk membuat rancangan pembangunan dan perencanaan pada daerah pesisir di masa depan, dan juga untuk mengurangi resiko yang ditimbulkan akibat pasang surut air laut. Peramalan permukaan laut dengan metode tradisional, seperti tidal harmonic analysis, tidak dapat mempertimbangkan kontribusi komponen non-tidal dalam peramalan permukaan laut. Dalam tugas akhir ini, penulis menggunakan metode deep learning untuk meramalkan permukaan laut. Penulis menggunakan tiga metode deep learning yaitu Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), dan Bidirectional Long Short Term Memory (BiLSTM). Ketiga metode deep learning ini dibandingkan untuk melihat performanya dalam meramalkan permukaan laut selama 48 jam, 72 jam dan 168 jam ke depan. Data yang digunakan pada tugas akhir ini yaitu data permukaan laut yang diperoleh dari observasi di Pelabuhan Jakarta, Indonesia, sebagai studi kasusnya. Hasil percobaan menunjukkan bahwa metode BiLSTM memberikan hasil yang lebih baik dibandingkan dengan RNN dan LSTM. Meningkatnya kinerja BiLSTM dari pada dua metode lainnya karena memungkinkan pelatihan tambahan dengan melintasi data input dua kali (yaitu, 1) dari kiri ke kanan dan 2) dari kanan ke kiri.

Kata kunci : Permukaan laut, Peramalan, RNN, LSTM, BiLSTM

Abstract

Sea level forecasting is vital in coastal activities, such as engineering and naval navigation. Moreover, it can be used for making strategies for future coastal development and planning, and also for mitigating its serious consequences. Traditional sea level forecasting, such as tidal harmonic analysis, do not consider a non-tidal component contribution in the sea level forecasting. In this study, we use a deep learning approach to forecast sea level. We use three deep learning methods: the Recurrent Neural Network (RNN), the Long Short Term Memory (LSTM), and the Bidirectional Long Short Term Memory (BiLSTM). These three methods of deep learning are compared to show their performances to forecast the sea level for 48 h, 72 h and 168 h ahead. We use the sea level data obtained from observation at Jakarta Port, Indonesia, as our study case. The results of the numerical experiment show that the BiLSTM method gives better performance than the RNN and the LSTM. The BiLSTM improve performance than the other two methods due enables additional training by traversing the input data twice (i.e., 1) left to right and 2) right to left.

Keywords: Sea Level, Forecasting, RNN, LSTM, BiLSTM.

1. Introduction

Sea level rise occurs as a result of many factors, one of which is global warming. Global warming occurs due to rising air temperatures by concentrations of gases known as greenhouse gases (GHGs). As a result of global warming, sea level continues to rise. Based on the report of the Intergovernmental Panel on Climate Change (IPCC), sea-level will increase 30-100 cm in 2100. [11].

Sea level rise has an impact on areas around the coast, one of which is Indonesia's capital city, Jakarta. Jakarta is one of the largest coastal cities in the world[4], [5]. Sea level rise in Jakarta is a serious problem. Almost every year, Jakarta experiences flood due to the impact of sea-level rise and also land subsidence. Sea level rise in Jakarta Bay reaches 0.57 cm per year [5] while land subsidence annually is around 9.5 - 21.5 cm per year from 2007 to 2009 [3]. In paper [1], it is predicted that in 2030 Jakarta sea level will increase as high as 2.88 m. The impact of rising sea levels is not only flooding but can also cause some islands around Jakarta to sink.

Based on the paper [14], five small islands around Jakarta had sunk as a result of rising sea levels. Sea level forecasting is important in coastal activities, such as engineering, and naval navigation. Moreover, it can be used for making strategies for future coastal development and planning, and also for mitigating its serious consequences [6]. Forecasting methods commonly used are quantitative forecasting methods by using historical data about the variables to be predicted and assuming that historical data patterns will continue into the future. Many algorithms and methods can be used to forecast sea levels, such as statistical methods and neural networks. Srivastava [16] proposed sea-level forecasting using the Exponential Smoothing Models and Autoregressive integrated moving average (ARIMA) methods. Forecasting uses Arabian sea level data with 17 years of history data (1994-2010), the results of forecasting Exponential smoothing state-space models method are better than ARIMA. Meanwhile, Sepideh Karimi et al. [9] proposed sea-level forecasting in Darwin Harbor, Australia, using Adaptive Neuro-Fuzzy Inference (ANFIS) method, Artificial Neural Network (ANN) and ARMA. The ANN and ANFIS methods yield similar accuracy and better than ARMA.

Recently deep learning is increasingly popular for time series forecasting problems especially Recurrent Neural Network (RNN) based model. There are several variations of RNN based models. Most of these RNN based models differ mainly because of their capabilities in remembering input data. The common issues of vanilla RNN are the vanishing and exploding gradient problems. This issues make it hard for vanilla RNN to capture the long term dependencies. Since the issues of vanishing gradient and exploding gradient, then seem special type of RNN is called Long Short Term Memory (LSTM). The LSTM was made to enhance network memory to remember previous states and preserve long term dependencies. The LSTM has shown a outperform ability to learn long term dependencies by preserving a memory cell to determine which unnecessary features should be forgotten and which necessary features should be remembered during the learning process [12],[17]. An interesting question is whether or not its performance may be more improved by incorporating additional layers of training data into the LSTM.

To investigate whether incorporating additional layers of training into the LSTM improves its prediction, this paper analyze the performance of Bidirectional LSTM (BiLSTM). In the BiLSTM enables additional training data by traversing the input data twice(i.e., 1) left to right and 2) right to left). In specific, we would compare the performance of the three methods to prediction sea level.

2. Methodology

2.1 Study area and data

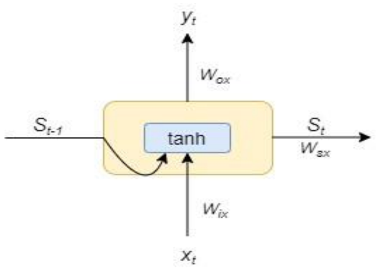
Kolinamil the port of Jakarta, Indonesia, located at -6.10 Latitude and 106.89 Longitude. Jakarta is the capital of Indonesia, the area of Jakarta is 661.52 km^2, consists of 5 regions, the area in Jakarta has a topographic slope between 0-2% in the north and center while in the south 5% [4].

Jakarta is located in the lowlands and is a coastal area that causes frequent flooding. The data source in this paper is sea level data in Kolinamil, Jakarta port, with one-year historical data (January-December 2019). Figure 1 shows the geographical location of the port of Jakarta.

Gambar 1. Location Jakarta Port

2.2 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) is one of the architectures of neural networks that are processed repeatedly to process inputs that are usually sequential data such as time-series data. RNN differs from traditional Neural Network because in RNN each processing produced is not only influenced by current input but is influenced by an internal state which is the result of previous input processing, which means that when RNN makes a decision the time step t-1 can influence the decision to be taken at time step t [10]. RNN has been widely used for forecasting. In [2], RNN is used for forecasting wind speed, and also in [13], RNN is used for forecasting electricity consumption. The simple RNN architecture proposed in this study is shown in Figure 2.



Gambar 2. RNN Structure

For each time step t, first calculate the state S_t from the input X_t and the previous state S_{t-1}, each state multiplied by the W_{ix} and W_{sx} and then add the bias b_x after it is processed with the hyperbolic tangent activation function (tanh) as given by the equation (1)

$$S_t = \tanh(W_{ix}X_t + W_{sx}S_{t-1} + b_x). \tag{1}$$

Function $\tanh(x)$ is an activation function that has the following equation

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{2}$$

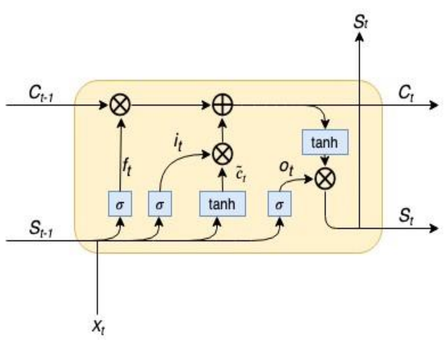
The range of the $\tanh(x)$ function is from (-1 to 1), and then the output value is given by (3)

$$y_t = W_{ox}S_t + b_y, \tag{3}$$

where y_t is output, W_{ox} is weight, S_t is states, and b_y is bias.

2.3 Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) is a type of modified Recurrent Neural Network (RNN), introduced by Hochreiter and Schmidhuber [8] by adding a memory cell that can store information for a long time. This vanishing gradient problem results in RNN failing to capture long term dependencies [15], thereby reducing the accuracy of a prediction on RNN [18]. In LSTM, data could be stored or discarded, since each neuron LSTM has several gates that regulate the memory of each neuron. Figure 3 shows the gates structure of LSTM.



Gambar 3. LSTM Structure

In LSTM there are three gates, namely f_t, i_t, and o_t as shown in Fig. 3. Gate f_t is the forget gate, i_t is the input gate, and o_t is the output gate. The equation of each gate are given by equation (4)-(9).

$$f_t = \sigma(W_{fx}X_t + W_{fs}S_{t-1} + b_f) \tag{4}$$

$$i_t = \sigma(W_{ix}X_t + W_{is}S_{t-1} + b_i) \tag{5}$$

$$\tilde{c}_t = \tanh(W_{cx}X_t + W_{cs}S_{t-1} + b_c) \tag{6}$$

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t \tag{7}$$

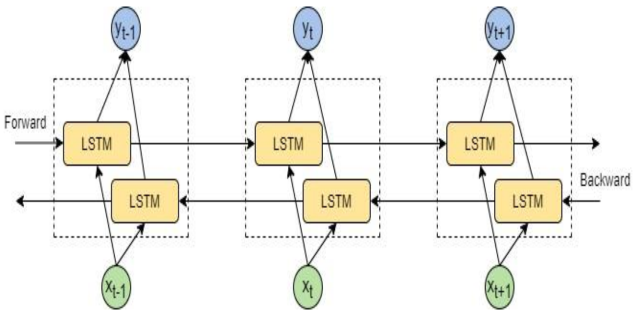
$$o_t = \sigma(W_{ox}X_t + W_{os}S_{t-1} + b_o) \tag{8}$$

$$s_t = o_t \times \tanh(c_t). \tag{9}$$

Symbol W_{fx}, W_{fs}, W_{ix}, W_{is}, W_{cx}, W_{cs}, W_{ox}, W_{os} is the weight, b_f, b_i, b_c, b_o is the bias, X_t is the input, S_{t-1} is the previous state, c_t is the cell state or memory cell and σ is the activation function sigmoid. The forget-gate determines the information to be stored or discarded in the previous state. The input gate regulates how many states for the current input to pass. The output gate decides the internal state to forward and the cell state or memory cell to forward old information with additional new information to the next cell state.

2.4 Bidirectional Long Short Term Memory (BiLSTM)

In LSTM, the output is only influenced by the previous time step, which means that it only gets information from the previous time step. Alex Graves and Jurgen Schmidhuber [7] introduce Bidirectional LSTM (BiLSTM) which is a variation of LSTM. The basic idea of BiLSTM is that output at time step t, not only influenced by the previous state $t-1$, but also by the next state t + 1, which means that in BiLSTM there are forward states and backward states. BiLSTM can improve the accuracy of the model. The structure of the BiLSTM is described in Figure 4.



Gambar 4. BiLSTM Structure

2.5 Metrics Evaluation

To evaluate the performance of the method in this paper, we use the Root Mean Square Error (RMSE) and the square of the correlation coefficient (R). RMSE represents the error between predictions and observations. The value of R measures the strength of a linear relationship between two variables. Expression of R and RMSE can be seen in equation (10) and (11).

$$R = \frac{\sum_{i=1}^{N}(x_i - x)(y_i - y)}{\sqrt{\sum_{i=1}^{N}(x_i - x)^2 (y_i - \bar{y})^2}} \tag{10}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - x_i)^2} \tag{11}$$

where y_i is the observed value, x_i is the predicted value, N is the amount of data, $\overline{x}$ is the average of predicted value, and $\overline{y}$ is the average value of the observation.

3. Result

Here we implement three different deep learning methods (RNN, LSTM, BiLSTM) for sea-level forecasting in Jakarta Port, Indonesia. The forecasting models are applied to forecast sea level for 72h, 120h and 168h ahead, and then their performance is measured. The All network hyperparameter used in this paper you can seen in Table 1.

Tabel 1. Hyperparameter Network

Hyperparameter	Value
Neuron	64
Optimization	Adam
Learning rate	0.01
Activation function	Tanh
Max Epoch	50

After setting hyperparameter, we also investigate the sensitiveness number of loopback that is used for the prediction in three models. We perform scenarios with various number of loopback, i.e. 36h, 45h, and 60h previous data to predict the next hour's. The forecasting result from three methods will form the future sea level of Jakarta for 48h , 72h and 168h ahead, such as shown in Table 2 - 4.

Tabel 2. Comparison Result RNN, LSTM, BiLSTM Using Loopback 36

	RNN		LSTM		BiLSTM	
	RMSE	R	RMSE	R	RMSE	R
48 h						
Training-1	0.0406	0.9566	0.0362	0.9654	0.0362	0.9654
Training-2	0.0392	0.9594	0.0384	0.9610	0.0362	0.9654
Training-3	0.0389	0.9599	0.0377	0.9624	0.0355	0.9667
Training-4	0.0426	0.9519	0.0362	0.9652	0.0345	0.9685
Training-5	0.0377	0.9624	0.0372	0.9634	0.0355	0.9667
Average	0.0398	0.9580	0.0373	0.9634	0.0355	0.9665
72 h						
Training-1	0.0378	0.9592	0.0325	0.9696	0.0325	0.9697
Training-2	0.0349	0.9650	0.0351	0.9647	0.0327	0.9693
Training-3	0.0352	0.9644	0.0340	0.9667	0.0322	0.9703
Training-4	0.0391	0.9562	0.0324	0.9699	0.0313	0.9719
Training-5	0.0339	0.9670	0.0340	0.9669	0.0324	0.9698
Average	0.0361	0.9623	0.0336	0.9675	0.0322	0.9702
168 h						
Training-1	0.0351	0.9742	0.0323	0.9780	0.0315	0.9790
Training-2	0.0336	0.9762	0.0348	0.9744	0.0321	0.9782
Training-3	0.0329	0.9772	0.0325	0.9776	0.0313	0.9792
Training-4	0.0367	0.9716	0.0316	0.9789	0.0320	0.9784
Training-5	0.0335	0.9763	0.0325	0.9777	0.0316	0.9788
Average	0.0335	0.9751	0.0327	0.9773	0.0317	0.9787

Tabel 3. Comparison Result RNN, LSTM, BiLSTM Using Loopback 45

	RNN		LSTM		BiLSTM	
	RMSE	R	RMSE	R	RMSE	R
48 h						
Training-1	0.0377	0.9624	0.0375	0.9628	0.0365	0.9648
Training-2	0.0377	0.9624	0.0359	0.9658	0.0368	0.9641
Training-3	0.0379	0.9619	0.0367	0.9643	0.0360	0.9656
Training-4	0.0373	0.9632	0.0359	0.9660	0.0371	0.9636
Training-5	0.0383	0.9611	0.0374	0.9629	0.0364	0.9649
Average	0.0377	0.9622	0.0366	0.9643	0.0365	0.9646
72 h						
Training-1	0.0337	0.9674	0.0340	0.9668	0.0326	0.9695
Training-2	0.0342	0.9665	0.0323	0.9701	0.0331	0.9686
Training-3	0.0341	0.9666	0.0327	0.9692	0.0322	0.9702
Training-4	0.0335	0.9678	0.0322	0.9703	0.0332	0.9683
Training-5	0.0345	0.9658	0.0337	0.9674	0.0329	0.9690
Average	0.0340	0.9668	0.0329	0.9687	0.0328	0.9691
168 h						
Training-1	0.0337	0.9760	0.0324	0.9778	0.0317	0.9787
Training-2	0.0321	0.9782	0.0316	0.9789	0.0316	0.9789
Training-3	0.0332	0.9768	0.0319	0.9784	0.0319	0.9785
Training-4	0.0326	0.9775	0.0313	0.9793	0.0317	0.9788
Training-5	0.0330	0.9769	0.0321	0.9782	0.0318	0.9787
Average	0.0329	0.9770	0.0318	0.9785	0.0317	0.9787

Tabel 4. Comparison Result RNN, LSTM, BiLSTM Using Loopback 60

	RNN		LSTM		BiLSTM	
	RMSE	R	RMSE	R	RMSE	R
48 h						
Training-1	0.0405	0.9565	0.0401	0.9574	0.0364	0.9650
Training-2	0.0398	0.9581	0.0357	0.9663	0.0383	0.9613
Training-3	0.0420	0.9634	0.0358	0.9660	0.0361	0.9654
Training-4	0.0381	0.9615	0.0355	0.9666	0.0367	0.9644
Training-5	0.0399	0.9578	0.0376	0.9625	0.0352	0.9672
Average	0.0400	0.9574	0.0369	0.9637	0.0365	0.9646
72 h						
Training-1	0.0365	0.9618	0.0367	0.9614	0.0327	0.9694
Training-2	0.0361	0.9626	0.0322	0.9702	0.0342	0.9663
Training-3	0.0383	0.9579	0.0323	0.9701	0.0323	0.9700
Training-4	0.0344	0.9661	0.0321	0.9705	0.0329	0.9690
Training-5	0.0362	0.9623	0.0340	0.9668	0.0318	0.9710
Average	0.0363	0.9621	0.0334	0.9678	0.0327	0.9691
168 h						
Training-1	0.0349	0.9743	0.0351	0.9740	0.0320	0.9784
Training-2	0.0350	0.9742	0.0322	0.9780	0.0321	0.9783
Training-3	0.0362	0.9723	0.0317	0.9787	0.0317	0.9788
Training-4	0.0331	0.9769	0.0318	0.9786	0.0321	0.9782
Training-5	0.0339	0.9757	0.0327	0.9774	0.0318	0.9786
Average	0.0346	0.9746	0.0327	0.9773	0.0319	0.9784

Based on Table 2 - 4 represent the result of the applied three models. To check the stability of the three models, we conducted a training scenario five times then average the five training values to result predict the future sea level of Jakarta. From the table, it is clear for RNN and LSTM loopback 45h perform better than loopback 36h and 60h, for BiLSTM loopback 36h better than 45h and 60h in term R and RMSE.

The overall evaluation of three models reveals that the BiLSTM models outperform than unidirectional LSTM and RNN. It seems that BiLSTM can capture the underlying context better by traversing the input data twice (i.e., 1) left to right (forward layer) and 2) right to left (backward layer)). The test results using optimum loopback for each models indicate that increasing prediction interval from 48 h to 168 h leads to a increase in the model accuracy. The R increases from 0.9622 to 0.9770 for RNN, from 0.9643 to 0.9785 for LSTM and from 0.9665 to 0.9787 for BiLSTM models. The RMSE decreases from 0.0377 to 0.0329 for RNN, from 0.0366 to 0.0318 for LSTM and from 0.0355 to 0.0317 for BiLSTM models.

4. Conclusion

Based on the experiments using RNN, LSTM, and BiLSTM models in forecasting sea-level was tested. Hourly sea-level observations at Kolinamil, Jakarta Port, Indonesia were used for training and testing of each model using optimal loopback. For each models has a different optimal loopback. In the RNN and LSTM have an optimal loopback 45h, and for BiLSTM have an optimal loopback 36h. The optimal RNN, LSTM and BiLSTM models were compared against each other to estimate sea levels 48 h, 72 h, and 168 h ahead. The results of the experiment represent that BiLSTM model gives better performance than RNN and LSTM.

References

[1] N. Anggraini, B. Trisakti, and T. E. B. Soesilo. Application of sattelite data to analyze inundation potential and the impact of sea level rise. Jurnal Penginderaan Jauh dan Pengolahan Data Citra Digital, 9(2), 2012.

[2] Q. Cao, B. T. Ewing, and M. A. Thompson. Forecasting wind speed with recurrent neural networks. European Journal of Operational Research, 221(1):148–154, 2012.

[3] E. Chaussard, F. Amelung, H. Abidin, and S.-H. Hong. Sinking cities in indonesia: Alos palsar detects rapid subsidence due to groundwater and gas extraction. Remote sensing of environment, 128:150–161, 2013.

[4] R. Djaja, J. Rais, H. Z. Abidin, and K. Wedyanto. Land subsidence of jakarta metropolitan area. In Proceedings of the 3rd FIG Regional Conference for Asia and the Pacific, pages 3–7, 2004.

[5] T. Firman, I. M. Surbakti, I. C. Idroes, and H. A. Simarmata. Potential climate-change related vulnerabilities in jakarta: Challenges and current status. Habitat International, 35(2):372–378, 2011.

[6] M. A. Ghorbani, R. Khatibi, A. Aytek, O. Makarynskyy, and J. Shiri. Sea water level forecasting using genetic programming and comparing the performance with artificial neural networks. Computers & Geosciences, 36(5):620–627, 2010.

[7] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural networks, 18(5-6):602–610, 2005.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997. [9]

S. Karimi, O. Kisi, J. Shiri, and O. Makarynskyy. Neuro-fuzzy and neural network techniques for forecasting sea level in darwin harbor, australia. Computers & Geosciences, 52:50–59, 2013.

[10] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang. Short-term residential load forecasting based on lstm recurrent neural network. IEEE Transactions on Smart Grid, 10(1):841–851, 2017.

[11] G. A. Meehl, W. M. Washington, W. D. Collins, J. M. Arblaster, A. Hu, L. E. Buja, W. G. Strand, and H. Teng. How much more global warming and sea level rise? Science, 307(5716):1769–1772, 2005.

[12] X. Qing and Y. Niu. Hourly day-ahead solar irradiance prediction using weather forecasts by lstm. Energy, 148:461–468, 2018.

[13] A. Rahman, V. Srikumar, and A. D. Smith. Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. Applied energy, 212:372–385, 2018.

[14] M. Ramdhan, S. N. Amri, and D. G. Priyambodo. Identification survey of sinking island in jakarta bay. Jurnal Riset Jakarta, 12(1):1–6, 2019.

[15] A. Saxena and T. Sukumar. Predicting bitcoin price using lstm and compare its predictability with arima model. Int. J. Pure Appl. Math, 119(17):2591–2600, 2018.

[16] P. K. Srivastava, T. Islam, S. K. Singh, G. P. Petropoulos, M. Gupta, and Q. Dai. Forecasting arabian sea level rise using exponential smoothing state space models and arima from topex and jason satellite radar altimeter data. Meteorological Applications, 23(4):633–639, 2016.

[17] Y.-T. Tsai, Y.-R. Zeng, and Y.-S. Chang. Air pollution forecasting using rnn with lstm. In 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), pages 1074–1079. IEEE, 2018.

[18] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu. Lstm network: a deep learning approach for short-term traffic forecast. IET Intelligent Transport Systems, 11(2):68–75, 2017.