Vol. 4, No.09, September 2023
E-ISSN: 2723-6692
P-ISSN: 2723-6595
http://jiss.publikasiindonesia.id/

Analysis of Time Series Water Level Data Prediction Using Deep Learning Method at the Water Gate of DKI Jakarta Water Resources Office

Supriyade[1], Gerry Firmansyah[2], Habibullah Akbar[3], Budi Tjahjono[4]
[1,2,3,4] Universitas Esa Unggul, Indonesia
Email : adejakarta770@student.esaunggul.ac.id, gerry@esaunggul.ac.id, habibullah.akbar@esaunggul.ac.id, budi.tjahjono@esaunggul.ac.id

KEYWORDS
Prediction of water level Deep Learning, Long Short Term Memory (LSTM), Recurrent Neural Network

ABSTRACT
Indonesia has 2 seasons, namely the dry season and the rainy season. During the rainy season, many points in the DKI Jakarta area experience flooding or inundation. The reason why Jakarta often experiences flooding is caused by several factors, including local rain floods, shipment floods and tidal floods. The DKI Jakarta Water Resources Agency currently does not have a system that can predict future water levels by referring to past and present water level data. Through this background, the author tries to conduct research in one of the floodgates in the northern area of DKI Jakarta in predicting water levels using deep learning methods, namely Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM). The purpose of this research is to analyze the best deep learning models and predict water level time series data. From the results of the analysis carried out, the best deep learning model is Long Short Term Memory (LSTM) using several tests such as n-input, split data with a composition of 90.33% train data and 9.67% test data , as well as testing of different parameters including epoch, batch size, learning rate, dropout , so the results obtained are the lowest error values with RMSE (17.65), MAPE (0.29), MAE (3.37) and the time needed in the process (runtime) is 39 minutes.

1. Introduction

The Jakarta Water Resources Agency currently does not have a system that can predict future water levels by referring to past and current water level data. In terms of monitoring, for now there is an application called JAKI in which there is a Jak Pantau feature that will make it easier to get all flood information. Flood events occur due to flood water runoff from rivers because flood discharge cannot be glimpsed by river channels or flood discharge is greater than the existing river drainage capacity (Fredrik et al., 2021). DKI Jakarta itself has 8 main water gates, namely Manggarai, Karet, Marina Ancol, Pulo Gadung, Istiqlal, Jembatan Merah, Flushing Ancol and Hek (Jakarta, 2020). The sluice gate serves to control water, so as to prevent flooding in fast and high flow. The process of opening or closing the floodgates is based on the level of rainwater level and water discharge rate. The flowing water will be directed by the sluice gate to the sea or river depending on the size and size of the flowing water discharge.

From this background, the problem in this study is related to flood control by predicting water level data in the form of *time series*. The purpose of this study is to analyze water level prediction with deep learning methods using *Python* programming language using *2* deep learning methods *, namely the RNN* (Recurrent Neural Network*)* and LSTM *(Long Short Term Memory*) methods. Where the data used is the water level dataset in January 2022.

The advantage of the LSTM method compared to the RNN method is that LSTM can remember data that is *time series* or data with *long-term dependency* information and LSTM can store previous information using cells contained in LSTM (Lattifia, Wira Buana, &; Rusjayanthi, 2022). The RNN method has a unique property, which is that it can store data in a network structure because it has at least one *feedback loop*. The advantage of the RNN model in forecasting algorithms is the ability to predict nonlinear time series data (Journal &; Mathematics, 2023). So, the two methods are compared with their respective prediction values by finding the lowest error value which is used to be a method in predicting water level values by the DKI Jakarta Water Resources Agency in the future.

2. Materials and Methods

2.1 Research Steps

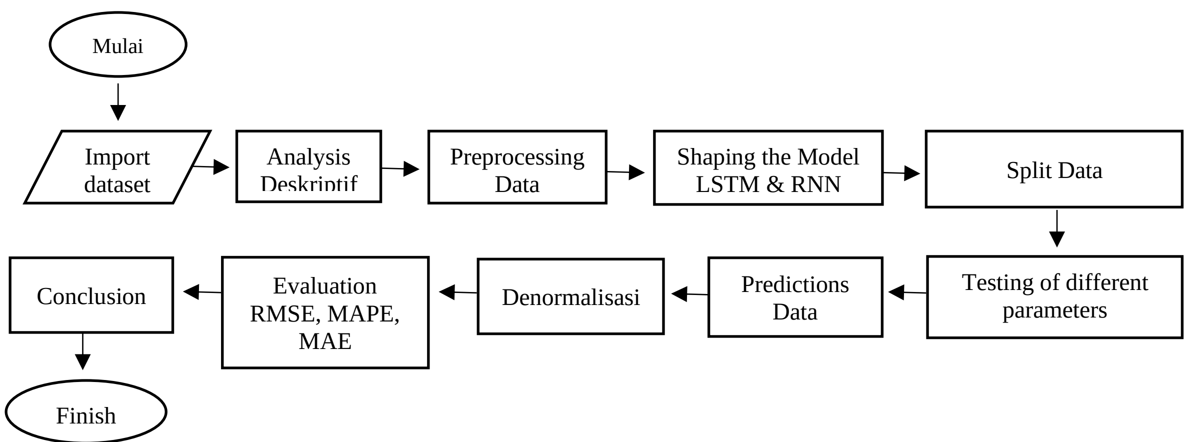

Figure 1. Deep Learning Flowchart Water Level Prediction

a. Start: This stage is the subject stage at the DKI Jakarta Water Resources Office that handles flood problems, namely in the Sub-Division of Flood Control and Drainage. The object taken was the prediction of water levels at the Marina Ancol Water Gate using LSTM and RNN algorithms.
b. Import : Input the water level dataset into the *google colaboratory*, the dataset used is the Water Level (TMA) data for January 2022.
c. Descriptive analysis: Perform LSTM and RNN model analysis
d. Data preprocessing: Normalize data with the min-max method, with data intervals between 0-1. Data normalization is done to accelerate the model in the learning process by scaling the data in the same range of values.
e. Forming RNN and LSTM Models: Perform random testing with different input values to get the best value.
f. Split Data: Determine the composition of data (split data), divide data into training and testing data. The data used water level data in January 2022, with 3 dataset division compositions
g. Testing Different Parameters: LSTM and RNN models are designed in advance and trained using training data to learn patterns of water level data. Determine the number of neurons on the *layer, epoch, n-input, batch size, dense.*
h. Making Predictions
i. Data denormalization: Denormalize test data using RMSE, MAPE and MAE
j. Model Evaluation: Calculate error values with RMSE, MAPE and MAE

k. Conclusion: Selection of the best method with the lowest error value

2.2 Literature Review

Some deep learning-based models that have high accuracy are when used for face detection, image processing, recommendation systems, *natural language processing*, and *time series prediction* (Sanjaya &; Budi, 2020). In *deep learning*, methods that are often used in previous research in processing time series data are *Recurrent Neural Network* (RNN) and *Long Short Term Memory* (LSTM).

2.2.1 Deep Learning

Deep Learning is one part of various *machine learning* methods that use *Artificial Neural Networks* (*ANN).* The advantages *of deep* learning compared to traditional *machine learning* methods are more complex feature extraction, less modeling and having more accurate predictions even when paid for with higher computation. *Deep learning* can be technically defined as *machine learning* that has more than one hidden layer. Deep Learning illustration can be seen in Figure 6 there are 4 layers and each layer has a different number of nodes (Rizki, Basuki, & Azhar, 2020)

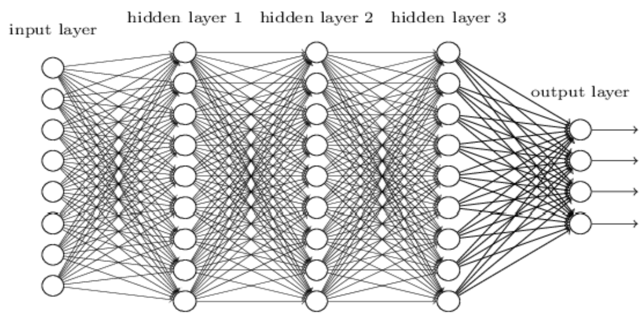

Figure 2. Deep learning illustration
(source: Journal of Repositors, 2020)

2.2.2 Recurrent Neural Network (RNN)

RNN is form of *Artificial Neural Network* (ANN) architecture specifically designed to process continuous or sequential data. RNN is usually used to solve historical data problems or *time series*, one example is weather forecasting. Here's the RNN architectural drawing:

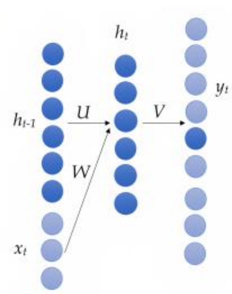

Figure 3. RNN architecture
(source: Journal and Mathematic, 2023)

2.2.3 Long Short Term Memory (LSTM)

LSTM is the architecture of RNN. LSTM can be used to process *sequential* data so that it can be used for *time* series data prediction. LSTM can detect data to be stored and data that is not used to be trimmed, because LSTM has 4 layers of neurons commonly called gates to organize memory on each neuron. The advantage of the LSTM method compared to the RNN method is that LSTM can remember data that is *time series* or data with *long-term dependency* information and LSTM can store previous information using cells contained in LSTM. There are 3 types of gates on LSTM, namely *forget* gates*, input gates, and* output gates. (Putri Sekti Ari &; Hanum, 2021) Here is a picture of the LSTM structure (Lstm, Kota, Sugiartawan, & Santoso, n.d.)*:*

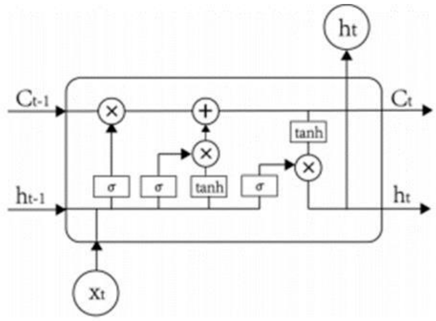

Figure 4. LSTM Structure
(Source: Corisindo National Seminar, 2022)

2.2.4 Python

Python is a programming language that can execute a number of multipurpose instructions directly with object-oriented and uses mass semantics to provide a level of readability of code or syntax. Most define Python as a language with a high level of capability, combining very clear capabilities and code syntax and complemented by the functionality of a very large and comprehensive base library. Although this python is classified as a high-level programming language, it is still designed in such a way that it is easy to understand and learn. Python can also run on many platforms such as Mac, Linux and Windows etc. Python is *open source* so there are still many people who contribute to developing (Pane &; Yogi Aditya Saputra, 2020). Python was chosen as a programming language in this study, because this language has many libraries that make it easy to create programs that involve a lot of vector and matrix manipulation, as well as visual displays of various attractive and easy-to-read graphics (*scikit learn matplotlib library*, and also *heatmaps (seaborn library)* to show correlations in the form of color and numerical maps (Hastomo, Karno, Kalbuana, Nisfiani, & ETP, 2021).

2.2.5 Collaboratory

Collaboratory or collab for short, is a product of Google research. *Colab* allows anyone to write and execute arbitrary python code through *a browser* and is perfect for machine learning data analysis and education. In addition to being easy to use, colab is quite flexible in its configuration and requires no setup. (Naik &; Girish, 2021) Some of the advantages of *Google Colaboratory* are : Support for python 2.7 and python 3.6, Free gpu acceleration, all major python libraries like TensorFlow, Scikit-Learn, Matplotlib, Keras among many others are pre-installed and ready to import, Support bash commands, Google colab notebooks are stored back on the drive.

2.3 Research Needs

In the process of data analysis, things are needed that support system testing or data processing, such as *hardware* and *software needs*.

2.3.1 *Hardware Requirements*

For hardware using 2 computers / laptops that have the following specifications:

Memory or RAM 8GB, Processor Intel Core™ i7-855OU CPU @ 1.80 GHz 1.99 Ghz and System Type 64 bit, Memory 16GB Intel®® Xeon® CPU E5-2609 @ 1.90 GHz

2.3.2 Software *Requirements*

For software include: Google Collaboratory, Python 3 programming language, Windows 10, Windows Server 2012 R2.

3. Result and Discussion

3.1 Data Collection

The water level data used for the study is water level data from the DKI Jakarta Water Resources Office in the northern region (marina water gate), which is data from January 2022 in the form of an Excel file totaling 744 records.

3.2 Preprocessing Data

Data preprocessing is prepared by going through a process to handle missing or empty data in various ways such as finding the average of an attribute for the same class. After that the data is normalized using *MinMaxScaler* with range (0, 1)

3.3 Split Data

In the data split process using experiment 3 split data. The following is the division process, among others:

a. First data split 1 day : 24 hours
- Testing data: 24 hours or **(3.25%)**
- Training data : Total data 744 – 24 = 720 hours or **(96.75%)**

b. Second data split 3 days :
- Testing data: 24 x 3 = 72 hours/3 days or **(9.67%)**
- Training data : Total data 744 – 72 = 672/3 days or **(90.33%)**

c. Third data split 7 days1(1 week) :
- Testing data: 24 x 7 = 168 hours/7 days or **(22.6%)**
- Training data: Total data 744 – 168 = 576 hours / 7 days or **(77.4%)**

3.4 Process Training

Training data is used to find the best parameters from the *Long Short Term Memory (*LSTM*)* and *Recurrent Neural Network (*RNN) *methods.* The results of the best parameters will be tested on testing data. The training process will be carried out, where the model will be trained using training data.

3.5 Deep Learning *Model Parameter Testing*

Deep learning model testing is carried out using several experiments with different parameters to get the best values, namely n-input, split data, batch size, learning rate, dropout and epoch.

3.5.1 Random Input Testing (n-input)

Table 1. Random Input Test Results

Input testing to-n	n-input	epoch	Batch size	Dense	Layer	Neuron	Rata-rata RMSE	Rata-rata MAE	Rata-rata MAPE
Testing 1st	2	10	1	1	1	100	11,98	3,1	0,25
Testing 2nd	3	10	1	1	1	100	11,88	3,14	0,24
Testing 3th	4	10	1	1	1	100	9,81	2,84	0,23
Testing 4th	5	10	1	1	1	100	18,98	4,07	0,33

Information:

From the table of random input test results above, it can be explained that the 3rd test with n-input 4 gives error values with the lowest average, including RMSE (9.81), MAE (2.84) and MAPE (0.23).

3.5.2 *Random Split Data Testing*

Table 2. Random *Split Data Test Results*

Split Data Testing to-n	Train Data Composition: Data Test	RMSE	MAE	MAPE
Testing 1st	96,75% : 3,25%	20,28	3.92	0.31
Testing 2nd	90,33% : 9,67%	16.23	3.76	0.30
Testing 3th	77,4% : 23,6%	26.74	4.55	0,36

Information:

From the table of random split data test results above, it can be explained that the composition of train data (90.33%) and test data (9.67%) gives low *error* values , including RMSE (16.23), MAE (3.76) and MAPE (0.30).

3.5.3 *Random Batch Size Testing*

Table 3. Result *of Random Batch size Testing*

Method	Average Runtime	Average RMSE	Average MAPE	Average MAE
LSTM	85 second	20.96	0.32	2.76
RNN	41 second	23.07	0.33	4.33

Information:

From the results of random *batch size* testing data above, it can be explained that the LSTM method has a low average error value compared to RNN, namely RMSE (20.96), MAPE (0.32) and MAE (2.76) and a process time (*runtime)* of 85 seconds, longer than the faster RNN of 42 seconds.

3.5.6 *Random Learning Rate Testing*

Table 4. Random *Learning Rate Test Results*

Method	Average Runtime	Average RMSE	Average MAPE	Average MAE
LSTM	160 second	25.55	0.35	4.55
RNN	88 second	38.28	0.45	5.65

Information:

From the random *learning rate* testing table above, it can be explained that the LSTM method has a low average error value compared to RNN, namely RMSE (25.55), MAPE (0.35) and MAE (4.55) and a process time (*runtime)* of 160 seconds, longer than the faster RNN of 88 seconds.

3.5.7 Pengujian *Random Dropout*

Table 5. *Random Dropout Test Results*

Method	Average Runtime	Average RMSE	Average MAPE	Average MAE
LSTM	170 second	39.8	0.40	5.52
RNN	89 second	46.67	0.44	6.04

Information:

From the random *dropout* testing table above, it can be explained that the LSTM method has a low average error value compared to RNN, namely RMSE (39.8), MAPE (0.40) and MAE (5.52) and a runtime time *of* 170 seconds, longer than the faster RNN of 89 seconds.

3.6 Results of LSTM Model and RNN Model Prediction Analysis

The results of previous test analysis using several hyperparameters with several different input values such as, n-input, split data, *batch_size,* learning_rate*,* dropout *can be drawn conclusions including* :

I. The 3rd parameter test (n-input = 4, epoch = 10, batch size = 1, dense = 1, layer = 1, neuron = 100) had the smallest average *error* value (RMSE, MAE, MAPE).

II. The test used a 3-day data split (3x24 hours = 72 hours) with a data train composition (90.33%), test data (9.67%) had the lowest *error* values , namely with respective values of RMSE (16.23), MAE (3.76), MAPE (0.3).

III. Parameter testing using *batch_size* can be seen that the *error* value in the LSTM model has an average value of RMSE (20.96), MAPE (0.32), MAE (2.76) with a runtime of 85 seconds, while for the RNN model has an average value of RMSE (23.07), MAPE (0.33), MAE (4.33) with a runtime of 41 seconds.

IV. In parameter testing using *learning_rate* it can be seen that the *error value in the LSTM model has an average value of RMSE (25.55), MAPE (0.35), MAE (4.55) with a runtime of* 160 seconds, while for the RNN model has an average value of RMSE (38.28), MAPE (0.45), MAE (5.65) with a runtime time of 88 seconds .

V. In parameter testing using *the dropout* table, it can be seen that the *error* value in the LSTM model has an average value of RMSE (39.8), MAPE (0.40), MAE (5.52) with a runtime of 170 seconds, while for the RNN model it has an average value of RMSE (46.67), MAPE (0.44), MAE (6.04) with a runtime *of 89* seconds.

From the conclusion of the analysis above, there are still differences in *error* values generated using learning_rate and dropouts, *the errors generated are still quite large compared to without using* learning_rate and *dropouts*. So, researchers will conduct further testing using the best model for predictive testing on *deep learning* methods, namely (RNN and LSTM), using (n-input = 4, epoch = 10, batch size = 1, dense = 1, layer = 1, neuron = 100), and split data (72 hours), and using different epoch values, among others, epoch 10, 50, 100 as comparison material. Here are the results:

Table 6. LSTM Method Prediction Analysis Results

Epoch	Average Runtime	Average RMSE	Average MAPE	Average MAE
10	185 Second	28.59	0.38	4.88
50	44 second	29.40	0.4	5
100	56 Minute	18.57	0.3	3.87

From the table of prediction analysis results using the LSTM method using different epoch counts, it can be explained that the greater the number of epochs, the lower the average error value produced and the runtime time *is* relatively longer.

Table 7. Results of RNN Method Prediction Analysis

Epoch	Average Runtime	Average RMSE	Average MAPE	Average MAE
10	44 second	29.40	0.4	5
50	4 minutes	27.73	0.38	4.8
100	9 minutes	32.52	0.4	5.15

From the table of prediction analysis results using the RNN method using different epochs, it can be explained that the greater the number of epochs, the higher the average error value produced and *the* runtime is relatively faster.

3.7 Best *Deep Learning* Model Evaluation Results

From the tests that have been done above, it shows that using the LSTM model provides better test results because it can produce low error values *from several parameters that have been tested before but have a* long runtime *in processing better prediction results, compared to using the RNN model provides test results with* error values which is higher, but in *a fairly fast* runtime.

3.8 LSTM Model Prediction Results

Table 8. Evaluation comparison of LSTM Method and RNN Method

Method	Average RMSE	Average MAPE	Average MAE	Average Runtime
LSTM	17.65	0.29	3.37	39 minutes
RNN	32.52	0.4	5.25	9 minutes

3.8.1 Operational Threshold of Marina Watergate North Jakarta

In the operation of the floodgates there are several categories or statuses where the water level as information that can be a reference in making decisions.

Table 9. Prediction results for the next 3 days of data (72 data)

e-ISSN: 2723-6692 p-ISSN: 2723-6595

Danger	standby	Alert	Normal
+ 250	201 - 250	171 - 200	0 - 170

3.8.2 Comparison of actual data with TMA prediction results

Table 10 below explains the results of the comparison between actual data in February 2022 and prediction data from January 2022 as many as 72 data *using the LSTM method* (n-input = 4, epoch = 100, batch size = 1, dense = 1, layer = 1, neurons = 100).

Table 10. Comparison results of actual and predicted TMA

Date Feb 2022	Tma Actual (cm)	Tma Prediction(cm)	Date Feb 2022	Tma Actual (cm)	Tma Prediction(cm)
1/2/2022 07.00	220	224	2/2/2022 21.00	137	148
1/2/2022 08.00	226	225	2/2/2022 22.00	141	153
1/2/2022 09.00	235	221	2/2/2022 23.00	147	159
1/2/2022 10.00	229	213	3/2/2022 00.00	154	164
1/2/2022 11.00	225	203	3/2/2022 01.00	158	170
1/2/2022 12.00	212	190	3/2/2022 02.00	165	177
1/2/2022 13.00	196	176	3/2/2022 03.00	171	186
1/2/2022 14.00	175	163	3/2/2022 04.00	176	197
1/2/2022 15.00	158	152	3/2/2022 05.00	173	209
1/2/2022 16.00	146	144	3/2/2022 06.00	185	221
1/2/2022 17.00	132	139	3/2/2022 07.00	196	231
1/2/2022 18.00	127	137	3/2/2022 08.00	201	238
1/2/2022 19.00	124	139	3/2/2022 09.00	213	239
1/2/2022 20.00	134	143	3/2/2022 10.00	219	233
1/2/2022 21.00	136	149	3/2/2022 11.00	221	224
1/2/2022 22.00	143	154	3/2/2022 12.00	213	212
1/2/2022 23.00	147	160	3/2/2022 13.00	202	197
2/2/2022 00.00	148	166	3/2/2022 14.00	186	182
2/2/2022 01.00	162	172	3/2/2022 15.00	172	167
2/2/2022 02.00	171	180	3/2/2022 16.00	153	155
2/2/2022 03.00	173	190	3/2/2022 17.00	147	146
2/2/2022 04.00	175	201	3/2/2022 18.00	136	141
2/2/2022 05.00	183	213	3/2/2022 19.00	133	139
2/2/2022 06.00	197	224	3/2/2022 20.00	132	141
2/2/2022 07.00	209	233	3/2/2022 21.00	146	145
2/2/2022 08.00	225	237	3/2/2022 22.00	153	151
2/2/2022 09.00	235	235	3/2/2022 23.00	156	156
2/2/2022 10.00	240	228	4/2/2022 00.00	161	162
2/2/2022 11.00	233	218	4/2/2022 01.00	161	167
2/2/2022 12.00	216	204	4/2/2022 02.00	170	174

2/2/2022 13.00	210	190	4/2/2022 03.00	173	182
2/2/2022 14.00	192	174	4/2/2022 04.00	178	192
2/2/2022 15.00	173	161	4/2/2022 05.00	185	203
2/2/2022 16.00	157	150	4/2/2022 06.00	187	215
2/2/2022 17.00	144	143	2/2/2022 21.00	137	148
2/2/2022 18.00	134	139	2/2/2022 22.00	141	153
2/2/2022 19.00	132	140	2/2/2022 23.00	147	159
2/2/2022 20.00	129	143	3/2/2022 00.00	154	164

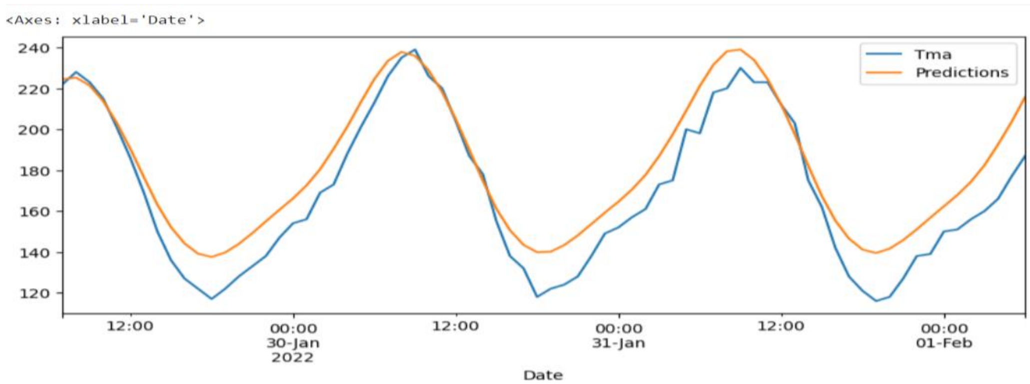

Figure 5. Water level prediction (tma) graph for February 2022

4. Conclusion

The following is a discussion related to water level prediction using the LSTM and RNN methods: 1. The application successfully processed the prediction of water level at the Marina sluice gate of DKI North Jakarta, using the water level dataset (Tma). 2. The composition of train data and test data with the most optimal results is with a train data composition of 90.33% and test data of 9.67%. This is because the composition of 90.33% train data and 9.67% test data has the lowest error rate, with an average value of RMSE (17.65), MAPE (0.29), MAE (3.37) and with an average runtime of 39 minutes. 3. The best parameters used in testing the LSTM method and RNN method are with data composition criteria (90.33%: 9.67%), n-input (4), dense (1), batch size (1), epoch (100), layer (1), neuron (100) provide the following evaluation:

Table 11. Conclusion Comparison of evaluation of LSTM model and RNN model

Method	Average RMSE	Average MAPE	Average MAE	Average Runtime
LSTM	17.65	0.29	3.37	39 minute
RNN	32.52	0.4	5.25	9 minute

4. After conducting analysis, implementation and test results of deep learning implementation using LSTM and RNN architecture for water level prediction at the DKI Jakata Marina sluice, it shows that the prediction results obtained using the LSTM method are quite good because they have the smallest error value. So it can be concluded that deep learning research with Long Short Term Memory (LSTM) architecture can work quite optimally. 5. Predictive testing using different epochs of 10, 50, and 100 can give different error values. It can be concluded that the larger the epoch used, the lower the error value produced and the longer the runtime

time. Vice versa, the smaller the epoch used, the higher the error value produced and the faster the runtime time.

5. References

Fredrik, Joan Arnold, Fredrik, Joan Arnold, Sudinda, Teddy W., Sejati, Wahyu, Fredrik, Joan Arnold, & Pendahuluan, A. (2021). *ANALISIS TINGGI MUKA AIR PADA KAWASAN SUNGAI CILIWUNG MT . HARYONO – PINTU AIR MANGGARAI DENGAN PROGRAM HEC-RAS 4 . 1 . 0 ANALYSIS OF WATER LEVEL IN THE CILIWUNG RIVER AREA MT . HARYONO – MANGGARAI WATER GATE WITH HEC-RAS 4 . 1 . 0 PROGRAM*. 100–106.

Hastomo, Widi, Karno, Adhitio Satyo Bayangkari, Kalbuana, Nawang, Nisfiani, Ervina, & ETP, Lussiana. (2021). Optimasi Deep Learning untuk Prediksi Saham di Masa Pandemi Covid-19. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, *7*(2), 133. https://doi.org/10.26418/jp.v7i2.47411

Jakarta, Posko Banjir DKI. (2020). Data Pintu Air DKI Jakarta.

Journal, Jambura, & Mathematics, O. F. (2023). *Peramalan Data Cuaca Ekstrim Indonesia Menggunakan Model ARIMA dan Recurrent Neural Network*. *5*(1), 230–242.

Lattifia, Tita, Wira Buana, Putu, & Rusjayanthi, Ni Kadek Dwi. (2022). Model Prediksi Cuaca Menggunakan Metode LSTM. *JITTER Jurnal Ilmiah Teknologi Dan Komputer*, *3*(1), 994–1000.

Lstm, Algoritma, Kota, Di, Sugiartawan, Putu, & Santoso, Servus Gusprio. (n.d.). *Multivariate Forecasting Curah Hujan Menggunakan*. 580–585.

Naik, Dr Poornima G., & Girish, Dr R. Nai. (2021). *Conseptualizing Python in Google Colab*. Sashwat Publication.

Pane, Syafrial Fachric, & Yogi Aditya Saputra. (2020). *Big Data classification behavior menggunakan python*. Kreatif Industri Nusantara.

Putri Sekti Ari, Dessanti, & Hanum, Latifah. (2021). Pengaruh Kualitas Pelayanan Website Djp Terhadap Kepuasan Pengguna Dengan Modifikasi E Govqual. *Profit*, *15*(01), 104–111. https://doi.org/10.21776/ub.profit.2021.015.01.11

Rizki, Muhammad, Basuki, Setio, & Azhar, Yufis. (2020). Implementasi Deep Learning Menggunakan Arsitektur Long Short Term Memory(LSTM) Untuk Prediksi Curah Hujan Kota Malang. *Jurnal Repositor*, *2*(3), 331–338. https://doi.org/10.22219/repositor.v2i3.470

Sanjaya, David, & Budi, Setia. (2020). Prediksi Pencapaian Target Kerja Menggunakan Metode Deep Learning dan Data Envelopment Analysis. *Jurnal Teknik Informatika Dan Sistem Informasi*, *6*(2), 288–300. https://doi.org/10.28932/jutisi.v6i2.2678