# Raw Paper Material Stock Forecasting with Long Short-Term Memory

1st Febryo Kurniawan
*Faculty of Information Technology*
*Tarumanagara University*
Jakarta, Indonesia
k.febryo@gmail.com

2nd Dyah Erny Herwindiati
*Faculty of Information Technology*
*Tarumanagara University*
Jakarta, Indonesia
dyahh@fti.untar.ac.id

3rd Manatap Dolok Lauro
*Faculty of Information Technology*
*Tarumanagara University*
Jakarta, Indonesia
manataps@fti.untar.ac.id

2021 9th International Conference on Information and Communication Technology (ICoICT) | 978-1-6654-0447-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICoICT52021.2021.9527528

***Abstract*— The manufacturing business is one of the businesses in Indonesia that continues to show its development from year to year. Like a manufacturing business in general, one of the important efforts made in the printing business is the supply of raw paper materials to produce finished goods. The purpose of this research is making a forecasting of the raw paper material for printing company on 7 different types of 269 historical data with weekly intervals from January 2015 to February 2020 before the Covid19 pandemic season. Forecasting is done using the Long Short Term Memory method with Python language. The model architecture for training and testing is carried out using vanilla LSTM with single input, hidden and output layer with the configuration of 64 neurons in the hidden layer, 150 epoch, 12 batch size and Adam Optimizer (lr = 0.0001) which was repeated 10 times for best result. The test results show the best window size length in the model for each paper raw material differently from 4 to 16. All models was successfully forecasting the test data with an average MAPE of the overall forecast of 21.48%.**

***Keywords*— *Forecasting, Long Short Term Memory, Machine Learning, Mean Absolute Percentage Error, Printing***

## I. INTRODUCTION

The manufacturing business is one of the businesses in Indonesia that continues to show its development from year to year. [1] According to Badan Pusat Statistik, the manufacturing industry in Indonesia, especially the printing industry, has shown an increase of almost 20% in 2019. Like a manufacturing business in general, one of the important efforts that must be made in a printing business is the supply stocks needed to produce finished goods. This was also conveyed by Ziukov [2] regarding the importance of inventory and stocks for companies.

In the printing industry, one of the main components in production is the raw material for paper. In Indonesia there has been a 28% increase in paper prices since the beginning of 2019 [3]. In addition to this price increase, paper mill owners prioritized export contracts in providing supplies. This inflict new challenges for printing manufacturing businesses in terms of the availability of paper raw materials that are difficult to obtain and the high price of raw materials. Most small to medium size printing company like Bintang Sakti Printing Company must make pre-order far in advance to get raw paper materials. In certain circumstances, customers need goods to be sent faster than usual. This cause major problem such as hampered in production due to unpredictable availability of raw paper materials (out of stock).

The main purpose of the research is to forecast the the stock supply of paper raw materials needed for future period to optimize the operation and production of Bintang Sakti Printing Company. In this forecasting, of course based on relevant time series data in the previous period, which is the historical time series data on the use of paper raw materials in weekly intervals. The data collection as input data is 7 most used paper in Bintang Sakti Printing Company; Art Carton 210(AC210), Art Paper 150 (AP150), HVS80, HVS100, NCR, Chromo Sticker, HVS Sticker. These dataset range for 5 years, from January 2015 to February 2020, which right before the COVID19 pandemic season due to changes in the national economy and the consumption of printing products.

## II. Methods

### *A. Machine Learning*

This research used machine learning to do the forecast. The term machine learning can be interpreted literally as machine learning. Creating a program with machine learning means training computers itself to acquire rules by entering data and answers given by humans [4].

The learning method is divided into two parts, namely supervised learning, and unsupervised learning. This research used supervised learning, learning will be carried out by providing input data and output data. Examples of methods that use supervised learning are Linear and Logistic Regression, SVM, and Artificial Neural Networks (artificial neural networks). Whereas in unsupervised learning, learning is carried out by providing input data without any output data[4].

### *B. Sliding Window*

Machine Learning approach with supervised learning can be used for time series forecasting. Before modeling, time series data must be processed into supervised learning. The initial data is in the form of sequential data with columns for the amount of usage and time as the data index. The data will be converted into pair of input and output. The technique used is the Sliding window technique[5]. This technique will create a data sequence that is a pair of input (feature) which is the lagged value of a timestep, and output (label) which is the value in the current timestep. This technique can be seen in Figure 1.

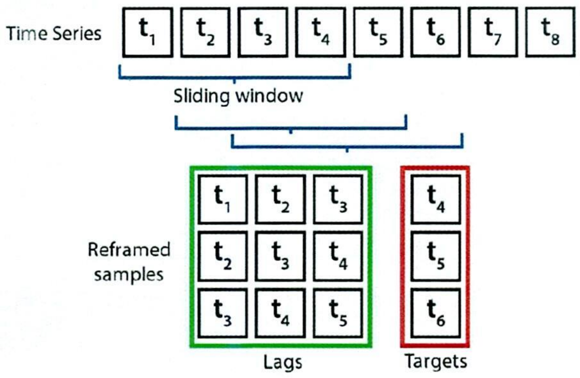

Fig. 6. Transform Time Series To Supervised Learning With Sliding Window

### C. Long-Short Term Memory

Long-Short Term Memory is an optimization of the common form of RNN which aims to avoid the long-term dependency problems that are often encountered in RNN. This LSTM model has a unique set of memory cells to replace neurons in the Hidden-Layer of the RNN [6]. LSTM will filter information through various gate structures that determine whether to update or maintain the state of the memory cells. This optimization was researched and proposed by Hochreiter & Schmidhuber in 1997[7].

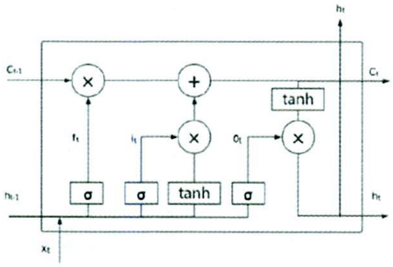

Fig. 7. Cell structure of Vanilla LSTM

However, the most commonly LSTM used nowadays is the LSTM architecture proposed by Graves & Schmidhuber in 2005 [8], which have the improvement of the added "forget gate". This architecture is so called as Vanilla LSTM[9]. Although in the recent year there is modification and variants in LSTM such as GRU[10], the study by Greff [11] compared popular LSTM variants and finding that the vanilla LSTM show good performance on mixed datasets across eight possible modifications of LSTM, this conclude no significantly improvement of other vanilla LSTM variant, including GRU.

In LSTM there is more than one activation function, in contrast to RNN which only has one activation function, namely the tanh function in each cell [6]. The processes contained in LSTM cells are described as follows.

The first step in LSTM is to determine whether the information from the previous cell state will be stored or removed from the cell. The formula for calculating the forget gate ($f_t$) is as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \qquad (1)$$

Where:

$\sigma$ = Sigmoid function

$W_f$ = Weight for forget gate layer

$h_{t-1}$ = Hidden state value at previous timestep

$x_t$ = Input value at current timestep

$b_f$ = Bias value of *forget gate layer*

Then at the input gate, this will determine how much information will be stored and entered into the cell state ($C_t$). In this layer there are two gates, the first one is the input gate ($i_t$) which will determine which value to update and the candidate gate ($\hat{C}_t$) contains the candidates for the cell state. The formula for calculating the input and candidate gate is as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \qquad (2)$$

$$\hat{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \qquad (3)$$

Where :

$W_i$ and $W_c$ are the weight for input and candidate gate

$b_i$ and $b_c$ are = bias value of input and candidate gate

The cell state or memory cell is the key to the LSTM itself. This information will go through two calculation phases, namely the forget gate and input gate stages. The cell state will then be forwarded for further cell calculations. The formula for cell state ($C_t$)is as follows:

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \qquad (4)$$

Where :

$C_{t-1}$=Cell state value at previous timestep

The last step in LSTM is to determine the output of the LSTM cells. The output value in a cell is the new hidden state. This value is obtained from the calculation of the output gate with the tanh function in the cell state. The formula for calculating the output gate ($O_t$) and hidden state ($h_t$) is as follows :

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \qquad (5)$$

$$h_t = O_t * \tanh(C_t) \qquad (6)$$

Where:

$W_o$ = the weight for output gate

$h_{t-1}$ = hidden state value at previous timestep

$b_o$ = bias value of output gate

### D. Mean Absolute Percentage Error

The Mean Absolute Percentage Error (MAPE) method is an evaluation calculation which is used to measure how accurate or precise a prediction is used. By using the MAPE method, we will get the difference between the actual value and the predicted value. This technique is used because it is easy to understand and interpret how big the forecast error is. The smaller the MAPE percent value indicates the more accurate a forecast is. The MAPE calculation formula is as follows[12].

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}i - yi}{yi}\right| \qquad (7)$$

Where :

ŷi = Predicted Value.

yi = Actual Value

$n$ = Number Of Data

## III. Experiment

### A. Experimtenal Settings

In this research, the LSTM used is from Keras library from Google Tensorflow backend. Other several libraries such as NumPy for numerical and Pandas for tabular data. The implementation is done in Python 3.7 environment. The thing to note is the requirement of Google Tensorflow version must match with the Python version in order to use GPU engine for accelerating the computation process. This experiment will be carried out on one of the dataset; HVS100 Paper with window size of 8 timesteps. There is no particular reason to consider using HVS100 Paper from paper other than those commonly used in everyday life.

### B. Pre-Processing

The first step is to collect historical data on the use of paper raw materials. The historical data on the use of paper raw materials obtained from the Bintang Sakti Printing. These dataset is a sequential time series of 269 weekly record (January 2015 - February 2020 before the COVID-19 pandemic) consisting of 2 columns; the date and the usage column in ream units (500 sheets). Each record is the amount of raw material usage for one week in that period. This data is saved in SQLite database, an example of an .xlsx file data for the input can be seen in Table 1**Error! Reference source not found.**. The selection of this .xlsx file rather than .csv is done because it is more common to use.

TABLE I. Weekly Use of Raw Paper Material Data Example

| Date | Paper Used in Ream (1 Ream = 500 Sheet) |
|---|---|
| 05/01/2015 | 8 |
| 12/01/2015 | 9 |
| 19/01/2015 | 9 |
| 26/01/2015 | 8 |
| 02/02/2015 | 8 |
| 09/02/2015 | 9 |

The data will be processed with the Pandas library as DataFrame structure for processing at a later step. In this machine learning model training, a supervised learning approach is used. The sliding window process is using the timeseriesgenerator function is used in the Keras library. This function will convert time series data into feature and label pairs as sliding window technique.

One of the common steps taken when pre-processing data is normalization, normalization is required if there is more than one feature that has a different value range, for example an age range of 0-100 years and an income range of 1-100 million. Income range is about 100,000 times bigger than age, these two features have different ranges. This will affect the learning process and the results of a model. Therefore, normalization of these two features should be carried out to equalize the range of the two features [13]. The data used in this research is univariate, which means there is only one value range. Therefore, the data normalization is not required in the pre-processing step.

For training and testing the model, the data is split into training and testing data. Training data are from the first 4 years, from January 2015 to December 2018. Test data are from January 2019 to February 2020.

### C. Model Architecture

Before forecasting the raw material for paper, an experiment was carried out to determine the architectural model to be used. To determine the model architecture, there are several initial parameters defined. These parameters are commonly used in LSTM forecasting with single input. Conceptually, the model architecture consists of 3 layers; the input layer, hidden layer, and output layer. These 3 layers can be seen in Figure 3.

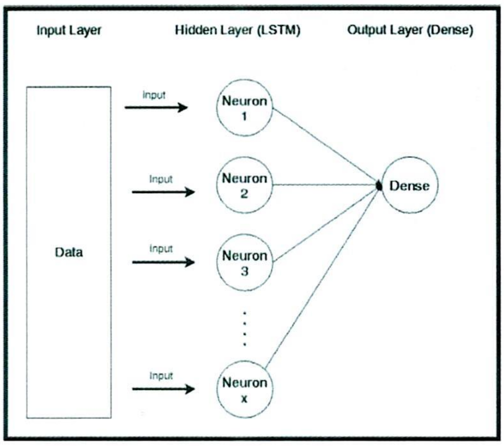

Fig. 8. Model Architecture

However, in this model design of made with the Tensorflow library, there are 2 layers arranged sequentially. Layer 1 is act as the input layer and the hidden layer. This layer is an LSTM layer which has x number of neurons which will be determined according to the experimental configuration. In this model architecture, the input layer is not declared explicitly because Tensorflow provides compatibility to add an input layer before the first layer by default[14]. The initial parameter for the layers can be seen in Table 2.

TABLE II. Initial Parameter

| Layer | Type | Parameter | |
|---|---|---|---|
| Layer 1 | LSTM | Activation Function | Sigmoid(Gate), Relu |
| | | Optimizer | Adam |
| | | Loss | MAPE |
| | | Metrics | MAPE |
| | | Batch Size | 12 |
| Layer 2 | Dense | Activation Function | Default(Linear) |

### D. Evaluation Measure

After specifying the configuration for the experiment, training and testing are carried out on the model architecture configuration. There are hundreds of hyperparameter can be used for configuration, in this experiment, the configuration used are 3 hyperparameters that more affect than other in forecasting performance. All experimental scenarios will be repeated 10 times, this is because the initial weight of the randomly generated model greatly affects the resulting performance. Therefore it is not enough to make one observation once.

In making a paper raw material forecasting model, several experiments will be carried out with several different model configurations. To evaluate the model, we will look at the model with the lowest MAPE forecasting accuracy value. If a model configuration is made more than one observation, the average MAPE accuracy value will be seen from a model configuration or take the best model from a model with the same configuration.

## IV. Result And Discussion

### A. Experiment Result

#### 1) Learning Rate Experiment

In machine learning, determining the learning rate is one of the steps that needs to be done. Learning rate will affect the rate of learning, it is how fast or aggressively the learning model of each learning epoch. The experiments will be carried out with learning rates of **0.001** and **0.0001**, The *window size* is 8 *timestep*, 32 neuron, dan 100 epoch.

After the training process is carried out for 10 repetitions of 0.0001 and 0.001 learning rates, the graph of the learning rate can be seen in FIGURE 4 and FIGURE 5.

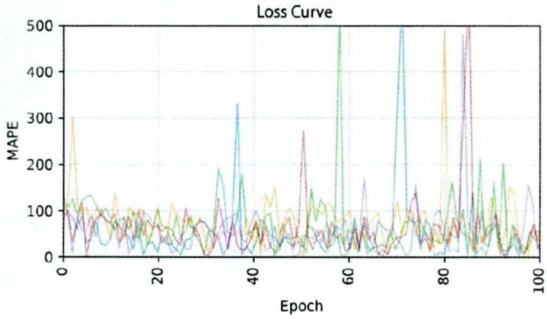

Fig. 9. Figure 1 Loss Curve with 0.001 Learning Rate

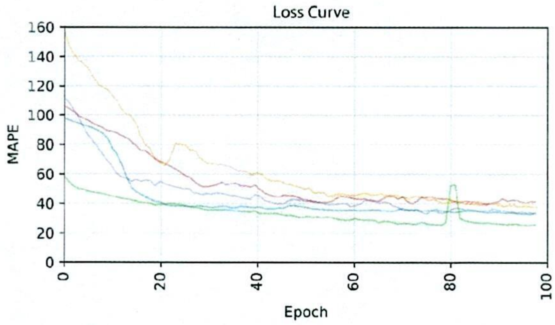

Fig. 10. Loss Curve with 0.0001 Learning Rate

In this experiment, the change in loss curve at the learning rate of 0.001 seems to jumping up and down, this is because the learning rate is too large resulting in the learning model being too fast so that the updated weight is too large, as a result the model cannot achieve optimal performance for forecasting. Therefore, the further experiment and model will use a learning rate of 0.0001.

#### 2) Neuron Experiment

In this experiment, we will find the optimal number of neuron units to be used in modeling. The training and testing process was carried out on 3 configurations and produced 30 models after 10 repetitions. Then the evaluation will be seen the average value for each of the 3 configurations. The result can be seen in Table 3 .

Based on the average results of training and testing in Table 3, it shows that configuration 3 with 64 neurons has the best performance with the lowest average MAPE test value of 36.53%.

TABLE III. NEURON EXPERIMENT RESULT

| **Config** | **Neuron** | **Epochs** | **Average MAPE Training** | **Average MAPE Testing** |
|---|---|---|---|---|
| Config 1 | 16 | 100 | 44.19 | 46.41 |
| Config 2 | 32 | 100 | 38.95 | 42.14 |
| **Config 3** | **64** | **100** | **32.20** | **36.53** |

In this experiment, it can also be seen that the addition of neurons in the hidden layer from 16 to 32 and to 64 shows an increase in performance. The insufficient number of neurons make the model unable to capture the complex relationship between the input and target variables, whereas if the number of neuron units is too many it can make for poor performance, this is because there are data that are not visible due to overparameterization.

TABLE IV. EPOCH EXPERIMENT RESULT

| **Config** | **Neuron** | **Epochs** | **Average MAPE Training** | **Average MAPE Testing** |
|---|---|---|---|---|
| Config 1 | 64 | 50 | 38.84 | 43.04 |
| Config 2 | 64 | 100 | 33.06 | 37.91 |
| **Config 3** | **64** | **150** | **29.93** | **35.49** |

#### 3) Epoch Experiment

In the second experiment, we will look for the optimal number of epochs to be used in modeling. The training and testing process was carried out on 3 configurations and produced 30 models after 10 repetitions. Then the evaluation will be seen the average value for each of the 3 configurations. The result can be seen in Table 4.

Based on the average results of training and testing in Table 4, it shows that configuration 3 with the number of epochs of 150 iterations has the best performance with the lowest average MAPE test value of 35.49%.

In this experiment, it can also be seen that the addition of epoch in the training model from 50 to 100 and to 150 indicates an increase in performance. The insufficient number of epochs affect the learning process of the model to be less, resulting in underfitting, on the other hand, if the number of epochs is too much it can result in overfitting.

Based on the three experiments conducted to determine the model architecture, it was found that the model architecture with 64 units of neurons and 150 iterations of epochs was the configuration with the best results. Therefore, to forecast paper raw materials, a model configuration will be set with 64 neurons and 150 iterations of epoch.

### B. Forecasting Result

After obtaining the best model architecture, a model will be made for each of the 7 types of paper data. The model architecture used is a model with the number of units of 64 and the number of epochs of 150. The model made will vary in the length of the window size; 4, 8, 12, and 16 timesteps to determine the optimal window size for each type of paper.

After determining the configuration scenario for the model, training and testing were carried out on the 7 dataset of weekly data; each will divided to 209 training data (January 2015 - December 2018) and 59 test data (January 2019-February 2020). After 10 repetitions, there are 40 models for each dataset. Of the 40 models, we will see the model with the best performance from each of the 4 Window

Size configurations. The 4 best models for each dataset can be seen in the following Table.

TABLE V. BEST MODEL ON AC210 DATASET

| Model | Window Size | Training MAPE | Testing MAPE |
|---|---|---|---|
| Model 25 | 4 | 20.84 | 55.03 |
| Model 14 | 8 | 20.58 | 48.56 |
| Model 27 | 12 | 20.12 | 48.06 |
| **Model 40** | **16** | **20.35** | **36.23** |

TABLE VI. BEST MODEL ON AP150 DATASET

| Model | Window Size | Training MAPE | Testing MAPE |
|---|---|---|---|
| Model 17 | 4 | 16.62 | 16.36 |
| Model 14 | 8 | 14.94 | 15.64 |
| Model 11 | 12 | 15.62 | 15.70 |
| Model 16 | 16 | 25.21 | 17.83 |

TABLE VII. BEST MODEL ON HVS80 DATASET

| Model | Window Size | Training MAPE | Testing MAPE |
|---|---|---|---|
| Model 33 | 4 | 11.78 | 21.91 |
| Model 14 | 8 | 8.76 | 22.38 |
| Model 11 | 12 | 9.21 | 23.29 |
| **Model 8** | **16** | **10.43** | **20.32** |

TABLE VIII. BEST MODEL ON HVS100 DATASET

| Model | Window Size | Training MAPE | Testing MAPE |
|---|---|---|---|
| Model 17 | 4 | 23.15 | 25.74 |
| Model 34 | 8 | 21.39 | 25.88 |
| Model 27 | 12 | 29.91 | 33.01 |
| Model 16 | 16 | 25.50 | 29.05 |

TABLE IX. BEST MODEL ON NCR DATASET

| Model | Window Size | Training MAPE | Testing MAPE |
|---|---|---|---|
| Model 37 | 4 | 16.58 | 18.61 |
| Model 10 | 8 | 12.57 | 15.49 |
| **Model 11** | **12** | **10.77** | **14.04** |
| Model 24 | 16 | 11.41 | 14.92 |

TABLE X. TABLE 10 BEST MODEL ON CHROMO STICKER DATASET

| Model | Window Size | Training MAPE | Testing MAPE |
|---|---|---|---|
| Model 37 | 4 | 16.71 | 25.42 |
| Model 34 | 8 | 16.92 | 26.33 |
| Model 23 | 12 | 16.46 | 26.61 |
| Model 24 | 16 | 15.63 | 26.57 |

TABLE XI. BEST MODEL ON HVS STICKER DATASET

| Model | Window Size | Training MAPE | Testing MAPE |
|---|---|---|---|
| Model 37 | 4 | 15.44 | 15.30 |
| **Model 2** | **8** | **14.39** | **12.95** |
| Model 27 | 12 | 13.75 | 13.74 |
| Model 4 | 16 | 13.45 | 14.08 |

If we look at each of the best models for each dataset, there are differences in the length of the timestep for the window size for each paper; Window size 16 for Art Carton 210 paper and HVS 80 paper; Window size 12 for NCR paper; Window size 8 for Art Paper 150 and HVS Sticker; Window size 4 for HVS 100 paper and Chromo Sticker. The difference in window size is due to differences in the time series patterns of each different paper raw material, ranging from trends, cyclic variants, seasonal and irregular fluctuations.

The results of forecasting training and testing data for the best model of each dataset can be seen on these following figures.

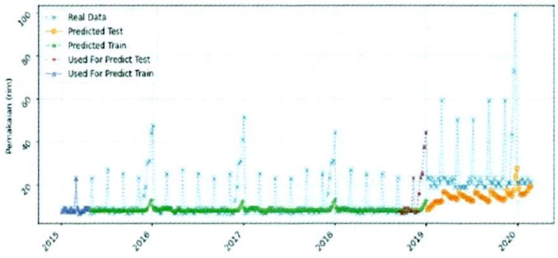

Fig. 6. Forecasting Result on AC210 Dataset Using Model 40

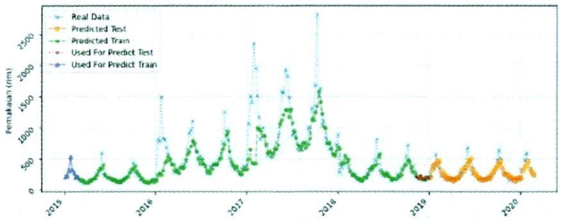

Fig. 7. Forecasting Result on AP150 Dataset Using Model 14

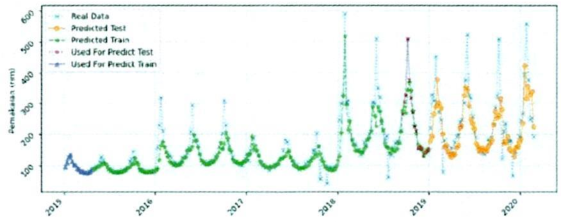

Fig. 8. Forecasting Result on HVS80 Dataset Using Model 8

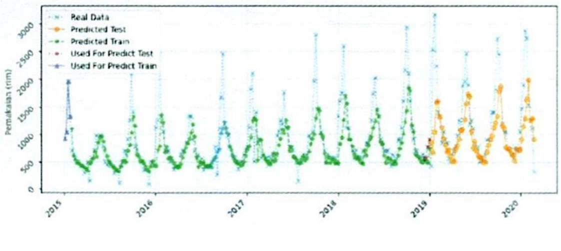

Fig. 9. Forecasting Result on HVS100 Dataset Using Model 17

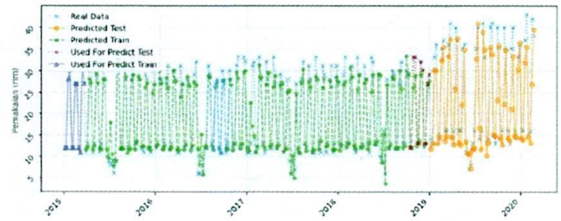

Fig. 10. Forecasting Result on NCR Dataset Using Model 11

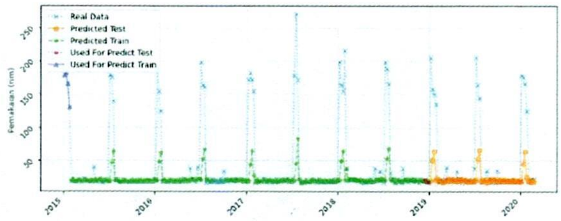

Fig. 11. Forecasting Result on Chromo Sticker Dataset Using Model 37

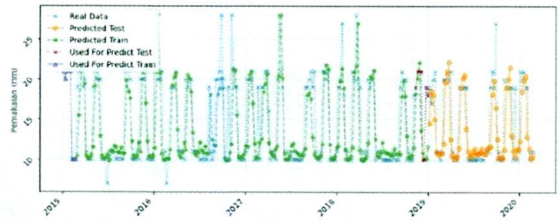

Fig. 12. Forecasting Result on HVS Sticker Dataset Using Model 2

The forecasting result for Art Carton 210 Dataset is a fairly high MAPE value of 36.23%. If we look at the graph of the forecasting results in_Figure 6, the model is unable to predict the Art Carton 210 paper test data. This is because there is a fairly large pattern change between the training data and the test data. The test data for 2019-2020 has a different pattern from the previous year so that the results of the forecasting model are not optimal for forecasting in that period.

## V. Conclusion

Based on the experiment, the learning rate of 0.0001 was chosen because the resulting learning curve graph was much better than the learning rate of 0.001. For the architecture of the forecasting model, the model with 64 neurons and 150 epochs had the best performance.

Based on the forecasting result, the LSTM model will result in poor forecasting performance if there were significant differences in the time series data pattern for the training data and test data.

All models are successful in forecasting the test data with an average MAPE of the overall forecast of 21.48%.

## References

[1] Badan Pusat Statistik, *Perkembangan Indeks Produksi Industri Manufaktur 2017-2019*. Jakarta: Badan Pusat Statistik, 2020.

[2] S. Ziukov, “A literature review on models of inventory management under uncertainty,” *Bus. Syst. Econ.*, vol. 5, no. 1, p. 26, Jun. 2015.

[3] A. Pernando, “Ini Masalah yang Masih Dihadapi Pengusaha Percetakan.” https://ekonomi.bisnis.com/read/20180902/257/834231/ini-masalah-yang-masih-dihadapi-pengusaha-percetakan (accessed Sep. 03, 2020).

[4] F. Chollet, *Deep Learning with Python*. New York: Manning Publications, 2017.

[5] A. M. Kotriwala, P. Hernandez-Leal, and M. Kaisers, “Load Classification and Forecasting for Temporary Power Installations,” in *Proceedings - 2018 IEEE PES Innovative Smart Grid Technologies Conference Europe, ISGT-Europe 2018*, Oct. 2018, pp. 1–6.

[6] C. Tian, J. Ma, C. Zhang, and P. Zhan, “A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network,” *Energies*, vol. 11, no. 12, 2018.

[7] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[8] 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.

[9] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, “Remaining useful life estimation of engineered systems using vanilla LSTM neural networks,” *Neurocomputing*, vol. 275, pp. 167–179, Jan. 2018, doi: 10.1016/j.neucom.2017.05.063.

[10] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, “On the Properties of Neural Machine Translation: Encoder–Decoder Approaches,” pp. 103–111, Sep. 2015, doi: 10.3115/v1/w14-4012.

[11] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, “LSTM: A Search Space Odyssey,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[12] S. Kim and H. Kim, “A new metric of absolute percentage error for intermittent demand forecasts,” *Int. J. Forecast.*, vol. 32, no. 3, pp. 669–679, Jul. 2016.

[13] S. Lakshmanan, “How, When, and Why Should You Normalize / Standardize / Rescale Your Data?” https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data (accessed Dec. 10, 2020).

[14] Tensorflow Team, “Tensorflow Core Documentation.” https://www.tensorflow.org/api_docs/python/tf/keras/layers/InputLayer (accessed Dec. 11, 2020).