

Prediksi Data Time Series Saham Bank BRI Dengan Mesin Belajar LSTM (*Long ShortTerm Memory*)

Adhitio Satyo Bayangkari Karno^{1,*}

¹ Sistem Informasi; Universitas Gunadarma; Jl. KH. Noer Ali, Kalimalang Bekasi;
88860117; e-mail: adh1t10.2@gmail.com

* Korespondensi: e-mail: adh1t10.2@gmail.com

Diterima: 9 Mei 2020; Review: 10 Mei 2020; Disetujui: 29 Mei 2020; Diterbitkan: 17 Juni 2020

Abstract

This study aims to measure the accuracy in predicting time series data using the LSTM (Long Short-Term Memory) machine learning method, and determine the number of epochs needed to produce a small RMSE (Root Mean Square Error) value. The result of this research is a high level of variation in RMSE value to the number of epochs needed in the data processing. This variation is quite difficult to obtain the right epoch value. By doing an iteration of the LSTM process on the number of different epochs (visualized in the graph), then the number of epochs with a minimum RMSE value will be easier to obtain. From the research of BBRI's stock data prediction, a good RMSE value was obtained (RMSE = 227.470333244533).

Keywords: *long short-term memory, machine learning, epoch, root mean square error, mean square error.*

Abstrak

Penelitian ini bertujuan untuk mengukur ketelitian dalam memprediksi data *time series* menggunakan metode mesin belajar LSTM (*Long Short-Term Memory*), serta menentukan banyaknya *epoch* yang diperlukan untuk menghasilkan nilai RMSE (*Root Mean Square Error*) yang kecil. Hasil dari penelitian ini adalah tingkat variasi yang tinggi nilai rmse terhadap jumlah *epoch* yang diperlukan dalam proses pengolahan data. Variasi ini cukup menyulitkan untuk memperoleh nilai *epoch* yang tepat. Dengan melakukan iterasi dari proses LSTM terhadap jumlah *epoch* yang berbeda (di visualisasikan dalam grafik), maka jumlah *epoch* dengan nilai RMSE minimal akan lebih mudah diperoleh. Dari penelitian prediksi data saham BBRI diperoleh nilai RMSE yang cukup baik yaitu 227,470333244533.

Kata kunci: *long short-term memory, machine learning, epoch, root mean square error, mean square error.*

1. Pendahuluan

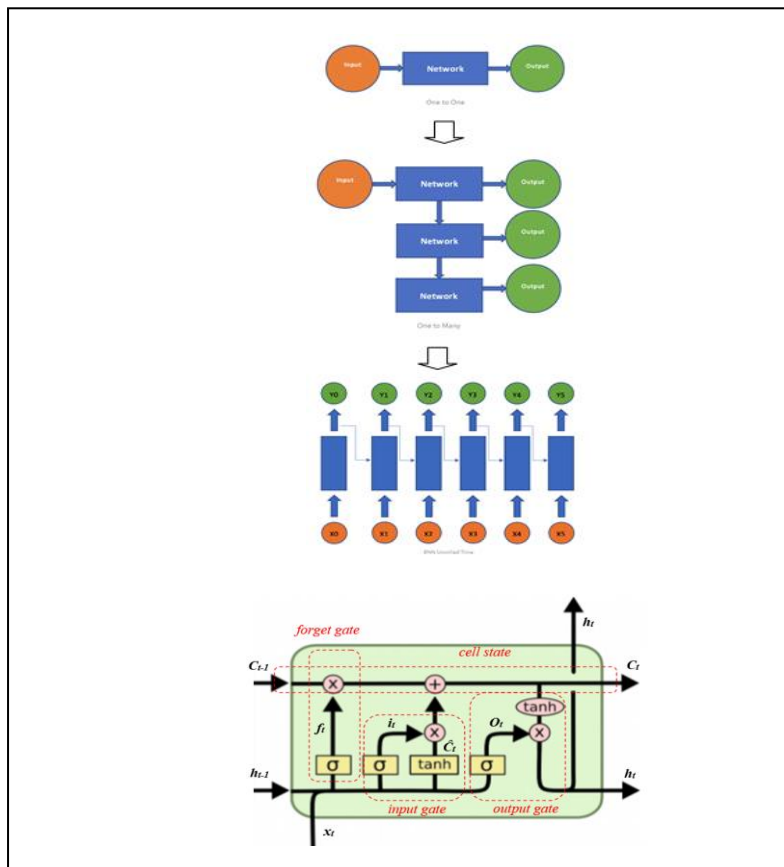
Tidak ada yang pernah mengetahui dengan tepat apa yang akan terjadi di masa yang akan datang, namun manusia dengan kemampuan berfikirnya berusaha untuk dapat memperkirakan kejadian dimasa depan dengan berbagai metode dan analisa untuk memperoleh tingkat kesalahan yang sekecil mungkin.

Sampai saat ini berbagai penelitian yang bertujuan untuk memprediksi suatu kejadian di masa mendatang dengan tingkat akurasi tinggi masih terus dilakukan. Metode tradisional

statistik masih menjadi hal yang umum dipergunakan, walaupun dalam era kecerdasan buatan (*Artificial Intelligence*) saat ini metode mesin belajar (*machine learning*) telah banyak berkembang dalam berbagai bidang kedokteran, pengenalan gambar (*image recognition*), game, robot, dan lain-lain, juga dalam hal prediksi data yang bersifat *time series*.

Mesin belajar merupakan suatu algoritma yang meniru cara kerja otak manusia, mesin ini mampu belajar dari kejadian-kejadian di masa lalu, semakin banyak data yang dipelajari maka mesin akan menjadi lebih pandai, layaknya otak manusia yang diberikan banyak informasi secara terus-menerus dan berulang-ulang menjadikan manusia mempunyai kemampuan akurasi berfikir lebih tinggi untuk menyelesaikan berbagai permasalahan.

Neural Network sederhana adalah model yang pertama kali dibentuk dari idea cara kerja otak manusia. Model ini terdiri dari tiga lapisan yaitu: lapisan masukan (*input layer*), lapisan tersembunyi (*hidden layer*) dan lapisan keluar (*output layer*). Tiap lapisan dapat terdiri dari beberapa node, setiap node lapisan tersembunyi merupakan suatu fungsi aktifasi dari akumulasi setiap masukan yang telah digandakan dengan suatu besaran bobot (*weight*) dan nilai bias. Setiap keluaran dari node ini akan menjadi masukan ke node di lapisan selanjutnya.



Sumber: Hasil Penelitian (2020)

Gambar 1. Perkembangan NN, RNN, LSTM

Model awal ini hanya mampu mengelola data yang bersifat regresi dan klasifikasi, kelemahan model ini tidak mampu mengkaitkan informasi baru dengan informasi sebelumnya.

(Hochreiter, S. & Schmidhuber, J., 1997). Agar mampu dipergunakan untuk data time series yang mempunyai keterkaitan dengan data sebelumnya. Maka hasil dari keluaran sebelumnya dipergunakan untuk memperbaharui nilai bobot baru. Bobot baru ini dipergunakan sebagai bobot untuk perhitungan data yang lebih baru dan seterusnya, sehingga ada keterkaitan antara data baru dengan data sebelumnya. Model pengembangan ini dinamakan dengan RNN (*Recurrent Neural Network*). RNN mampu mengolah data time series, namun ketidak mampuan untuk menampung memori yang panjang mengakibatkan RNN akan mengalami situasi dimana nilai untuk memperbaharui bobot akan menjadi hilang (*vanishing gradient*). Sehingga RNN hanya mampu mengolah data time series dengan nilai ketergantungan pendek. (Hochreiter, S. kombinasi gerbang yang agak rumit, yang kemudian sel ini dinamakan dengan LSTM.

Gerbang-gerbang sel LSTM yaitu: (Olah, Christopher, 2015).

Forget Gate, adalah gerbang yang memutuskan apakah masukan x_t dan keluaran h_{t-1} akan diteruskan ke *cell state*.

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

Input Gate, adalah gerbang masukan dengan dua fungsi aktivasi (sigmoid dan tanh), untuk memilih bagian yang akan di perbaharui.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

Cell State Gate untuk mengupdate nilai lama C_{t-1} menjadi nilai baru C_t .

$$C_t = f_t * C_{t-1} + i_t * \tilde{c}_t$$

Output Gate pertama adalah gerbang yang mengkombinasikan nilai lama dan nilai baru, yaitu

$$O_t = f_t * C_{t-1} + i_t * \tilde{c}_t$$

Output Gate kedua terdiri dari satu sel, yaitu

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

Sel LSTM mampu menghubungkan informasi sebelumnya dengan informasi selanjutnya, dan keefektifan untuk menyimpan informasi yang panjang ini sangat diperlukan dalam mengolah data time series. (Brownlee, Jason, 2018).

Pembuatan program dibuat dengan menggunakan bahasa Python, dikarenakan bahasa ini mempunyai banyak kelebihan dibanding dengan bahasa lain, terutama dalam hal program yang berhubungan dengan mesin belajar, Seperti banyak bahasa program yang lainnya yang bersifat umum (seperti interaktif, modular, dinamis), bahasa Python adalah open source dengan pustaka yang banyak dan terus bertambah, sehingga memudahkan pembuatan algoritma yang sulit menjadi lebih mudah, seperti *Numpy* (untuk operasi vektor dan matrik),

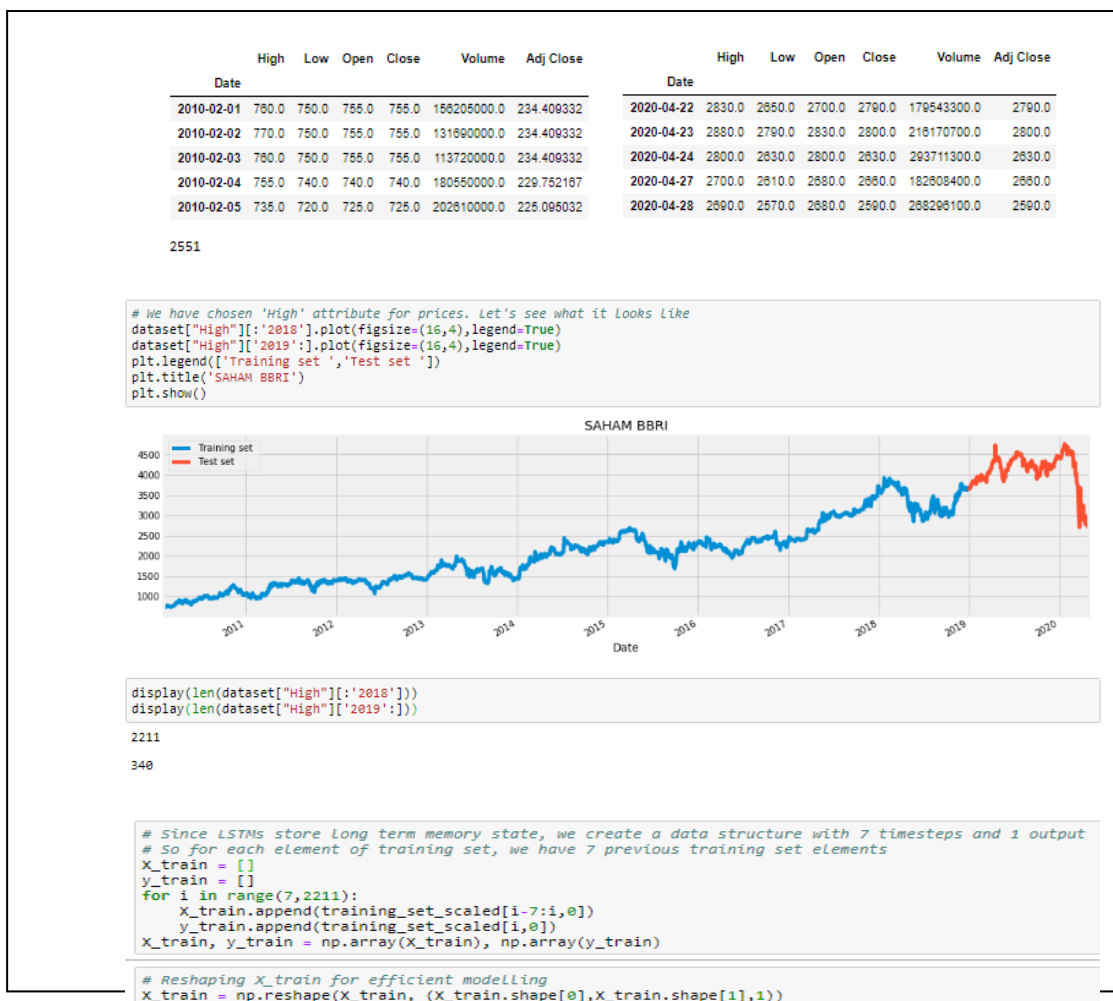
Scikit-learn (data analisis dan statistik), Pandas Data Frame (pengolahan data seperti layaknya Excell dan SQL), Matplotlib (visualisasi data grafik), dan Keras (API neural network yang bekerja di atas Tensor Flow dan Theano).

Dataset yang dipergunakan dalam penelitian ini adalah saham bank BUMN (Bank Usaha Milik Negara) yaitu BRI (Bank Rakyat Indonesia) dengan nama saham BBRI, saham yang populer dan banyak diminati oleh umum. Dalam penelitian ini juga diamati bagaimana perubahan epoch terhadap nilai rmse, sehingga program dapat dijalankan dengan mengambil nilai epoch yang tepat untuk menghasilkan prediksi yang tinggi.

2. Metode Penelitian

DataSet

Data yang dipergunakan adalah data saham Bank BRI bersumber dari <https://finance.yahoo.com>, dalam bentuk format CSV (Comma Separated Values) yang kemudian dirubah dalam format yang dapat dipergunakan Python dengan menggunakan modul Pandas.



Sumber: Hasil Penelitian (2020)

Gambar 2. Data set, grafik perjalanan dan training set saham BBRI

Data yang dipergunakan mulai dari 2010-02-01 sampai 2020-04-28, total data berjumlah 2551. Ada 6 kolom dari data set yang diperoleh, untuk penelitian ini diambil kolom "High" sebagai data time series. Perjalanan data dapat ditampilkan dalam bentuk grafik dibawah ini (Gambar 2). Untuk training data dipergunakan sampai dengan tahun 2018 (jumlah 2211 data), dan untuk test data dipergunakan mulai tahun 2019 (jumlah 340 data). Timestep diambil 7 dalam arti adalah data 1 sampai 7 dipergunakan untuk prediksi data di hari ke 8, data 2 sampai 8 dipergunakan untuk prediksi data di hari ke 9 dan seterusnya.

Arsitektur LSTM

Arsitektur LSTM yang dipergunakan dalam penelitian ini terdiri dari 4 hidden layer dengan tiap layer berisi 50 node, dropout 0,2 dan 1 output layer dengan 1 node.

```
# The LSTM architecture
regressor = Sequential()
# First LSTM Layer with Dropout regularisation
regressor.add(LSTM(units=50, return_sequences=True, input_shape=(X_train.shape[1],1)))
regressor.add(Dropout(0.2))
# Second LSTM Layer
regressor.add(LSTM(units=50, return_sequences=True))
regressor.add(Dropout(0.2))
# Third LSTM Layer
regressor.add(LSTM(units=50, return_sequences=True))
regressor.add(Dropout(0.2))
# Fourth LSTM Layer
regressor.add(LSTM(units=50))
regressor.add(Dropout(0.2))
# The output Layer
regressor.add(Dense(units=1))
```

S
umber: Hasil Penelitian (2020)

Gambar 3. Arsitektur LSTM

RMSE dan RMS

RMSE (*Root Mean Square Error*) adalah cara umum dipergunakan untuk mengukur kesalahan model dari prediksi data yang bersifat kuantitatif. RMSE dipergunakan untuk mengetahui ukuran sebaran simpangan titik data dari garis regresi linier atau untuk mengetahui konsentrasi data di sekitar garis regresi linier. MSE (*Mean Square Error*) adalah rata-rata dari kuadrat kesalahan (*error*) dipergunakan untuk menghitung kesalahan sample data dan tidak dipergunakan untuk estimasi model, sebagai berikut:

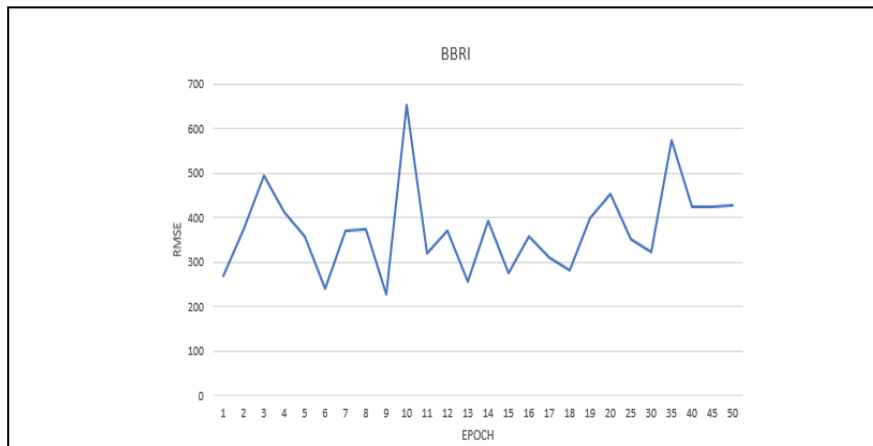
$$RMSE = \sqrt{\sum_{i=1}^n \frac{(y_i - \hat{y}_i)^2}{n}} \quad \text{dan} \quad MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n =$ nilai yang diprediksi
 $y_1, y_2, \dots, y_n =$ nilai yang diamati
 $n =$ jumlah data yang diamati

3. Hasil dan Pembahasan

Menentukan jumlah epoch

Untuk menentukan berapa jumlah epoch yang dipergunakan agar menghasilkan nilai rmse minimal masih agak sulit. Untuk itu dilakukan iterasi running program dengan jumlah epoch berbeda terhadap nilai rmse yang diperoleh, dinyatakan dalam grafik di bawah ini.



Sumber: Hasil Penelitian (2020)

Gambar 4. Grafik Epoch vs RMSE

Training data

Dari arsitekrut LSTM dan memilih jumlah epoch = 9 (berdasarkan gambar 6), maka training data dilakukan dengan mengukur nilai MSE (*Mean Square Error*) untuk tiap epoch yang dilakukan.

```
# Compiling the RNN
regressor.compile(optimizer='rmsprop',loss='mean_squared_error')
# Fitting to the training set
regressor.fit(X_train,y_train,epochs=9,batch_size=32)

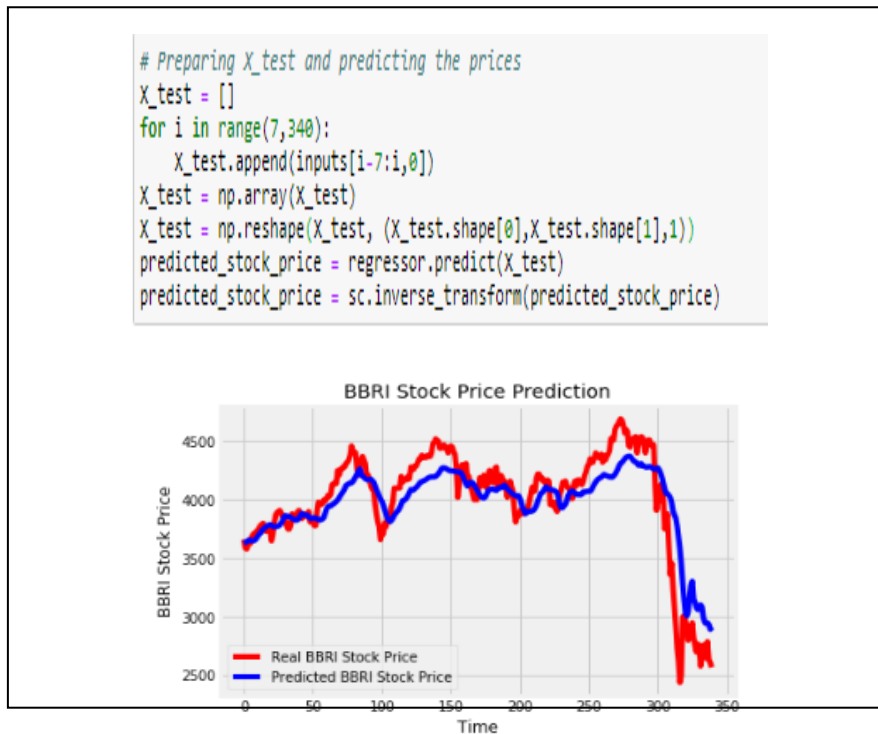
Epoch 1/9
2204/2204 [=====] - 59s 27ms/step - loss: 0.0208
Epoch 2/9
2204/2204 [=====] - 11s 5ms/step - loss: 0.0073
Epoch 3/9
2204/2204 [=====] - 11s 5ms/step - loss: 0.0063
Epoch 4/9
2204/2204 [=====] - 13s 6ms/step - loss: 0.0047
Epoch 5/9
2204/2204 [=====] - 10s 4ms/step - loss: 0.0047
Epoch 6/9
2204/2204 [=====] - 11s 5ms/step - loss: 0.0044
Epoch 7/9
2204/2204 [=====] - 9s 4ms/step - loss: 0.0041
Epoch 8/9
2204/2204 [=====] - 10s 5ms/step - loss: 0.0036
Epoch 9/9
2204/2204 [=====] - 11s 5ms/step - loss: 0.0035
```

Sumber: Hasil Penelitian (2020)

Gambar 5. Training Data

Testing data dan visualisasi hasil LSTM

Hasil pelatihan data, dipergunakan untuk memprediksi data selanjutnya dan mengukur tingkat akurasi dengan menggunakan data testing mulai tahun 2019 (jumlah 340 data).



Sumber: Hasil Penelitian (2020)

Gambar 6. Prediksi Data dan Grafik Visualisasi LSTM

Dari hasil testing data, diperoleh hasil yang cukup baik dengan nilai RMSE (*Root Mean Squared Error*) = 227,470333244533, Secara visual terlihat grafik prediksi hampir menyerupai grafik dari data asli (*data testing*).

4. Kesimpulan

Hasil prediksi data time series saham Bank BRI dengan menggunakan mesin belajar LSTM terlihat cukup berhasil untuk mendapatkan nilai akurasi tinggi (RMSE = 227,470333244533). Dan secara visual hasil plotting data prediksi memperlihatkan grafik yang mirip dengan data sebenarnya. Dari hasil penelitian ini dengan menggunakan data saham Bank BRI yang berjumlah 2551 (2211 untuk training dan 340 untuk testing), terlihat adanya variasi RMSE dan *Epoch* yang sangat bervariasi. Untuk itu pemilihan jumlah epoch terhadap penelitian ini harus dilakukan secara hati-hati agar mendapatkan nilai akurasi yang baik dan program dapat berjalan dalam waktu yang lebih cepat. Pengembangan penelitian selanjutnya masih banyak diperlukan dalam menggunakan arsitektur LSTM yang berbeda atau menggunakan jenis dan jumlah data *time series* yang berbeda (misal untuk jumlah data yang lebih minim)

untuk mengetahui serta menganalisa variasi RMSE dan *epoch* dari penelitian selanjutnya. Hal ini sangat diperlukan agar pemilihan jumlah epoch yang tepat dengan RMSE yang kecil, akan mempercepat proses jalannya program dari model LSTM. Dengan jumlah *epoch* yang lebih kecil tentunya akan mempercepat waktu proses jalannya program lebih cepat dibanding dengan jumlah *epoch* yang lebih besar. Perlu juga dilakukan penelitian selanjutnya dengan menggunakan jenis data yang sama atau berbeda untuk membandingkan LSTM dengan mesin belajar lainnya misal RNN (*Recurrent Neural Network*) atau GRU (*Gated Recurrent Unit*).

Daftar Pustaka

- Brownle, Jason on December 19, 2016 in [Time Series](#). "How To Backtest Machine Learning Models for Time Series Forecasting". <https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting>.
- Brownlee, Jason on November 14, 2018. "How to Develop LSTM Models for Time Series Forecasting", <https://machinelearningmastery.com/category/deep-learning-time-series>
- Hochreiter, S. & Schmidhuber, J., 1997, Long Short-Term Memory, *Neural Computation*, 8, 9, 1735–1780.
- Hastomo, Widi. Adhitio & Sudjiran (2019). Long Short Term Memory Machine Learning Untuk Memprediksi Akurasi Nilai Tukar IDR Terhadap USD. *Seminar Nasional Teknologi Informasi dan Komunikasi (SeNTIK STIK&K) STMIK Jakarta STIK&K, 22 Agustus 2019, Volume 3, ISSN : 2581-2327*
- L. J. Tashman, 2000. "Out-of-sample tests of forecasting accuracy: an analysis and review". *International Journal of Forecasting, Elsevier* 16(4), 437–450, 2000. <https://ideas.repec.org/a/eee/intfor/v16y2000i4p437-450.html>.
- Moody, James, Sep 6, 2019. "What does RMSE really mean?", <https://towardsdatascience.com/@james.moody>
- Neelabh Pant, Sept 7, 2017. "A Guide For Time Series Prediction Using Recurrent Neural Networks (LSTMs)", <https://blog.statsbot.co/time-series-prediction-using-recurrent-neural-networks-lstms-807fa6ca7f>
- Olah, Christopher, 2015. "Understanding LSTM Networks". <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> (2015)
- S. Varma and R. Simon. "Bias in error estimation when using cross-validation for model selection". *BMC Bioinformatics*, 7(1):91, Feb 2006. ISSN 1471–2105. doi: 10.1186/1471–2105-7–91.
- S. Akhil and R. Purva. 2019. "Appication of LSTM, GRU and ICA for Stock Price Prediction". *Proceeding of ICTIS 2018*, Volume 2. DOI:10.1007/978-981-13-1747-7 46