INTI NUSA MANDIRI
DOI: https://doi.org/10.33480/inti.v17i1.3277

VOL. 17. NO. 1 AGUSTUS 2022
P-ISSN: 0216-6933 | E-ISSN: 2685-807X
Diterbitkan Oleh: LPPM Nusa Mandiri

PREDIKSI HARGA SAHAM TWITTER DENGAN LONG SHORT-TERM MEMORY RECURRENT NEURAL NETWORK

Ibnu Akil[1], Indra Chaidir[2]

Fakultas Teknologi Informatika
Universitas Bina Sarana Informatika
www.bsi.ac.id
ibnu.ial@bsi.ac.id[1], indra@bsi.ac.id[2]

Abstract— Today the trading business has become a trend to get money easily without having to work hard as long as you have capital. In every fields of infestation, the possibility of getting unwanted loss in decision making can be happen. To get maximum results and avoid losses, it is necessary to have expertise in predicting the ups and downs of the stock market value. The purpose of this research is to utilize machine learning technology to predict the fluctuation of stock value by using the Long Short-Term Memory RNN method. From the results of this study, it was found that LSTM+RNN is suitable for use in single-step models.

Keywords: stock price, machine learning, recurrent neural network, lstm

Abstrak—Dewasa ini bisnis trading menjadi suatu trend untuk mendapatkan uang dengan mudah tanpa harus bekerja keras asalkan memiliki modal. Dalam semua bidang investasi kemungkinan terjadinya resiko kerugian yang tidak diharapkan dalam pengambilan keputusan bisa saja terjadi. Untuk mendapatkan hasil yang maksimal dan menghindari kerugian maka diperlukan keahlian di dalam memprediksi naik turunya nilai bursa saham. Tujuan dari penelitian ini adalah memanfaatkan teknologi machine learning untuk memprediksi naik turunnya nilai saham dengan menggunakan metode Long Short-Term Memory RNN. Dari hasil penelitian ini didapatkan bahwa LSTM+RNN cocok untuk digunakan pada model single-step.

Kata kunci: harga saham, machine learning, recurrent neural network, lstm

PENDAHULUAN

Dewasa ini trading saham menjadi suatu trend di masyarakat millennial. Apa itu trading saham? “Trading saham adalah transaksi jual beli surat kepemilikan atas perusahaan maupun perseroan terbatas dalam jangka pendek” (Redaksi OCBC, 2022). Sebagai seorang trader perlu memiliki kemampuan untuk menganalisa fluktuasi harga pasar, agar dapat memprediksi kapan harus menjual atau membeli saham. “Dalam semua bidang investasi kemungkinan terjadinya resiko kerugian yang tidak diharapkan dalam pengambilan keputusan tersebut bisa saja terjadi. Karena pada dasarnya tidak ada satu investasi pun yang terbebas dari resiko” (Titin, 2015). Menurut Mandala dkk, Seringkali para trader mengalami kesulitan untuk memprediksi harga di masa mendatang, untuk itu perlu adanya sebuah analisis yang tepat untuk mengambil keputusan agar seorang trader dapat memperoleh keuntungan (Mandala, Wahyuni, & Atmadja, 2019).

Untuk dapat memprediksi dengan tepat kecenderungan harga saham baik itu naik ataupun turun maka perlu dilakukan analisis teknikal. Menurut Ong Adianto, analisis teknikal merupakan suatu metode pengevaluasian saham, komoditas, ataupun sekuritas lainnya dengan cara menganalisis statistic yang dihasilkan oleh aktivitas pasar di masa lampau guna memprediksi pergerakan harga di masa yang akan datang (Ong, 2008). Dalam dunia statistic metode yang umum digunakan untuk memprediksi adalah regresi. Menurut I Made Yuliara, analisis regresi merupakan suatu kajian dari hubungan antara satu variabel, yaitu variabel yang diterangkan (the explained variabel) dengan satu atau lebih variabel, yaitu variabel yang menerangkan (the explanatory). Apabila variabel bebasnya hanya satu, maka analisis regresinya disebut dengan regresi sederhana. Apabila

VOL. 17. NO. 1 AGUSTUS 2022
P-ISSN: 0216-6933 | E-ISSN: 2685-807X
Diterbitkan Oleh: LPPM Nusa Mandiri

INTI NUSA MANDIRI
DOI: https://doi.org/10.33480/inti.v17i1.3277

variabel bebasnya lebih dari satu, maka analisis regresinya dikenal dengan regresi linear berganda (Yuliara, 2016).

Permasalahan dari analisis teknikal yang menggunakan perhitungan statistic manual adalah sulit mendapatkan prediksi secara real time karena dibutuhkan waktu untuk melakukan perhitungan, sementara dalam dunia trading, fluktuasi harga saham terjadi sangat cepat. Selain analisis teknikal, dengan memanfaatkan Artificial Neural Network, memprediksi harga saham menjadi lebih efektif dari sisi kinerja (Janeski & Kalajdziski, 2010). Menurut Chen (Chen, 2020), metode yang paling popular di dalam pemodelan time series adalah Auto Regressive Integrated Moving Average (ARIMA) yang diajukan oleh Jenkins dkk. Namun ARIMA adalah model linear yang tidak mempertimbangkan fluktuasi dari variansi harga saham yang mendasarinya.

BAHAN DAN METODE

Tujuan dari penelitian ini adalah merancang model pembelajaran mesin yang dapat memprediksi turun naiknya harga saham dengan mengoptimalkan pemodelan dengan menggunakan metode RNN (Recurrent Neural Network). Tahapan rancangan penelitian ini adalah sebagaimana terlihat pada gambar 1 berikut:

Sumber: (akil, 2022)

Gambar 1. Rancangan Tahapan / Proses Penelitan

Data Penelitian

Sebagai bahan penelitan yang utama adalah data penelitian tentang stock price. Data ini didapatkan di web Kaggle, url: https://www.kaggle.com/code/ahmeddraz/twitter-stock-price-predictions-arima/data. Dataset ini adalah data stock price milik Twitter selama periode 2013-2022. Data ini sudah dalam format csv yang lebih memudahkan untuk diproses dan dianalisa. Perhatikan table 1 berikut:

Tabel 1. Dataset Stock Price Twitter

Date	Open	High	Low	Close	Adj Close	Volume
2013-11-07	45,09999 8	50,09	44	44,90000 2	44,9	11770160 0
2013-11-08	45,93	46,93999 9	40,68999 9	41,65000 2	41,65	27925300
2013-11-11	40,5	43	39,40000 2	42,90000 2	42,9	16113900
2013-11-12	43,66	43,77999 9	41,83000 2	41,90000 2	41,9	6316700
2013-11-13	41,02999 9	42,86999 9	40,75999 8	42,59999 8	42,6	8688300
2013-11-14	42,34	45,66999 8	42,24000 2	44,68999 9	44,69	11099400
2013-11-15	45,25	45,27	43,43	43,98	43,98	8010600
2013-11-18	43,5	43,95000 1	40,84999 8	41,13999 9	41,14	12810600
2013-11-19	41,38999 9	41,90000 2	40	41,75	41,75	7436600

Sumber: https://www.kaggle.com/code/ahmeddraz/twitter-stock-price-predictions-arima/data

Analisis Data

Dataset ini terdiri dari 2118 record yang terdiri dari column date—adalah tanggal transaksi, open—adalah nilai saham diawal pembukaan, high—adalah nilai tertinggi pada tanggal tersebut, low—adalah nilai terendah pada tanggal tersebut, close—adalah nilai pada saat penutupan di tanggal tersebut, adj close—adalah harga penutupan yang disesuaikan untuk mencerminkan nilai saham tersebut setelah memperhitungkan Tindakan korporasi apa pun. Dan terakhir adalah volume—yang merupakan jumlah saham yang diperjual-belikan pada tanggal tersebut.

Recurrent Neural Network + LSTM

INTI NUSA MANDIRI
DOI: https://doi.org/10.33480/inti.v17i1.3277
VOL. 17. NO. 1 AGUSTUS 2022
P-ISSN: 0216-6933 | E-ISSN: 2685-807X
Diterbitkan Oleh: LPPM Nusa Mandiri

Pada umumnya neural network dapat dibagi menjadi dua kelas-kelas besar. Satu kelas berisi feedforward neural network (FNN), dan yang lainya berisi recurrent neural network (RNN). Perbedaan mendasar antara FNN dan RNN adalah adanya mekanisme hubungan umpan balik(feedback) antara neuron berikutnya. ANN tidak ada hubungan umpan balik antara neuron, sedangkan RNN memiliki sekurangnya satu hubungan umpan balik antara neuron (Yi & Tan, 2004).

RNN secara prinsip dapat menggunakan hubungan-hubungan umpan balik-nya untuk menyimpan representasi-representasi dari kejadian-kejadian input dalam bentuk aktivasi "Short-Term Memory" (STM), sebagai algoritma yang paling banyak digunakan untuk pembelajaran mesin adalah apa yang di simpan di dalam STM, namun terlalu memakan waktu, atau tidak bekerja sama sekali, khususnya ketika waktu jeda minimal antara input dan sinyal terkait cukup panjang. Mekipun secara teoritis memukau, metode ini tidak menyediakan keuntungan secara praktis yang jelas, seperti metode backpropagation dalam jaringan feedforward dengan waktu yang terbatas. Sebagai oposisi dari STM adalah "Long Short-term Memory" (LSTM), yang didesain untuk mengatasi masalah dari error back-flow. LSTM dapat belajar untuk menjembatani jarak waktu dalam urutan input yang tidak dapat dipadatkan, tanpa harus kehilangan kemampuan jeda waktu pendek (Hochreiter & Schmidhubber, 1997). Arsitektur LSTM-RNN di desain untuk memodelkan urutan-urutan sementara dan ketergantungan jangka panjang mereka. LSTM-RNN telah sukses diterapkan dalam banyak aplikasi, diantaranya; speech-recognition, hand writing recognition, dan speech synthesis (Zen & Sak, 2015).

Proses Pengolahan Data

Untuk implementasi algoritma dan pengolahan dataset akan menggunakan machine learning dalam bahasa python, karena python telah menyediakan banyak library untuk pemrosesan algoritma-algoritma machine learning. Diantaranya adalah; tensorflow, matplotlib, pandas, dan numpy. Sehingga proses pemodelan machine learning jauh lebih mudah.

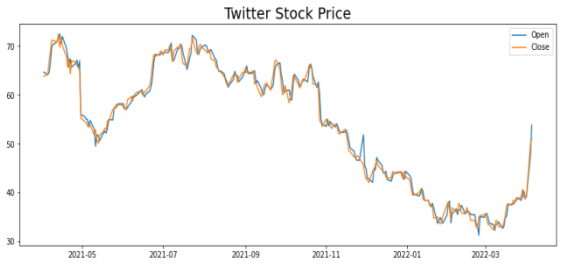
HASIL DAN PEMBAHASAN

Persiapan Data Awal

Sebagai sample data untuk analisa hanya digunakan data selama satu tahun mulai dari tanggal 01/05/2021-30/04/2022. Perhatikan cuplikan kode berikut:

```
# Read in the dataset ----------------
------------
data = pd.read_csv('/content/drive/MyD
rive/Colab Notebooks/Data/TWTR.csv')
# Konfigurasi data awal
df_tot = data.copy()
df_tot.Date = pd.to_datetime(df_tot.Da
te,dayfirst=True)
df_tot.set_index('Date',inplace=True)
df_tot = df_tot.asfreq('b','ffill')
df = df_tot[df_tot.index > '2021-04-
01'] ## sample data
```

sebagai sample data untuk diproses akan diambil dua field yaitu "Open" dan "Close". Open menunjukan nilai awal stock price pada hari tersebut, sedangkan Close menunjukkan nilai akhir setelah transaksi yang terjadi pada hari tersebut. Perhatikan gambar 2 berikut.



Sumber: (akil, 2022)

Gambar 2. Twitter Stock Price Periode 2021-2022

Pembagian Data

Data yang tampak pada gambar 1 akan di bagi menjadi tiga bagian 70% untuk training, 20% untuk validasi, dan 10% untuk test. Berikut adalah cuplikan kodenya:

```
column_indices = {name: i for i, name
in enumerate(df.columns)}
n = len(df)
train_df = df[0:int(n*0.7)]
val_df = df[int(n*0.7):int(n*0.9)]
test_df = df[int(n*0.9):]
```

Normalisasi Data

VOL. 17. NO. 1 AGUSTUS 2022
P-ISSN: 0216-6933 | E-ISSN: 2685-807X
Diterbitkan Oleh: LPPM Nusa Mandiri

INTI NUSA MANDIRI
DOI: https://doi.org/10.33480/inti.v17i1.3277

Selanjutnya sebelum diproses oleh model, data tersebut perlu melalui tahapan normalisasi. Normalisasi data adalah proses membuat beberapa variabel memiliki rentang nilai yang sama, tidak ada yang terlalu besar maupun terlalu kecil sehingga dapat membuat analisis statistik menjadi lebih mudah (Afifah, 2022). Selain itu untuk dapat membawa range nilai output ke dalam range input, maka data input harus dilakukan normalisasi data ke dalam range 0 sampai 1, sehingga outputnya dapat di denormalisasi ke dalam range nilai input (Chamidah, Wiharto, & Salamah, 2012). Ada banyak metode normalisasi diantaranya simple feature scaling, min-max, z-score, dan lain-lain. Disini akan dicoba menggunakan z-score untuk normalisasi. Perhatikan gambar 3 adalah data setelah dinormalisasi.

Date	Open	High	Low	Close	Adj Close
2021-04-02	0.532527	0.623618	0.530824	0.425731	0.425731
2021-04-05	0.460286	0.357092	0.287618	0.479667	0.479667
2021-04-06	0.531189	0.793825	0.618538	0.842753	0.842753
2021-04-07	0.794738	1.099933	0.855098	1.104543	1.104543
2021-04-08	1.232201	1.295209	1.280375	1.397906	1.397906
...	...	...	...	...	...
2021-12-09	-1.804623	-1.769832	-1.759029	-1.859342	-1.859342
2021-12-10	-1.886229	-1.983580	-1.950403	-1.977739	-1.977739
2021-12-13	-2.026699	-2.075940	-2.108554	-2.104030	-2.104030
2021-12-14	-2.228707	-2.235592	-2.119185	-2.136919	-2.136919
2021-12-15	-2.234059	-2.294966	-2.246768	-2.173753	-2.173753

Sumber: (akil, 2022)

Gambar 3. Data Setelah Proses Normalisasi

Konfigurasi Model

RNN adalah sebuah class Neural Network dari library keras yang merupakan bagian dari library tensorflow. RNN sangat bagus untuk memodelkan data berurut seperti time series. Skema dari RNN adalah; sebuah layer RNN menggunakan putaran untuk mengulang melalui suatu urutan waktu sambil mengelola keadaan informasi tentang urutan waktu yang sudah dilaluinya.

Model yang dikembangkan untuk studi kasus prediksi time series ini menggunakan dua macam models, yaitu single step model, dan multi step model.

Single step models adalah model paling sederhana yang dapat dibuat berdasarkan data semacam ini adalah model yang memprediksi nilai fitur tunggal—1 langkah waktu (satu hari) ke depan hanya berdasarkan kondisi saat ini. Sedangkan multi step models adalah prediksi multi-langkah, model perlu belajar memprediksi rentang nilai masa depan. Jadi, tidak seperti model single step, di mana hanya satu titik masa depan diprediksi, model multi step memprediksi urutan nilai masa depan. Disini model akan mengakumulasi status internal selama 24 hari, sebelum membuat satu prediksi untuk 12 hari ke depan. Kolom yang digunakan untuk memprediksi adalah kolom “Open” dan “Close”.

Konfigurasi Fungsi Compile dan Fitting Model

Cuplikan kode dibawah ini adalah konfigurasi untuk compile dan fitting model. Untuk fungsi loss menggunakan Mean Square Error (MSE), optimizer menggunakan fungsi Adam, dan metric (pengukuran error) menggunakan Mean Absolute Error (MAE). Dan untuk jumlah epochs (periode training) sebanyak 24 kali.

```
# fungsi compile dan fitting model ---
-----------------------------
MAX_EPOCHS = 24

def compile_and_fit(model, window, pat
ience=2):
  early_stopping = tf.keras.callbacks.
EarlyStopping(monitor='val_loss',

                patience=patience,

                mode='min')

  model.compile(loss=tf.keras.losses.M
eanSquaredError(),
                optimizer=tf.keras.opt
imizers.Adam(),
                metrics=[tf.keras.metr
ics.MeanAbsoluteError()])

  history = model.fit(window.train, ep
ochs=MAX_EPOCHS,
                      validation_data=
window.val,
                      callbacks=[early
_stopping])
  return history
```

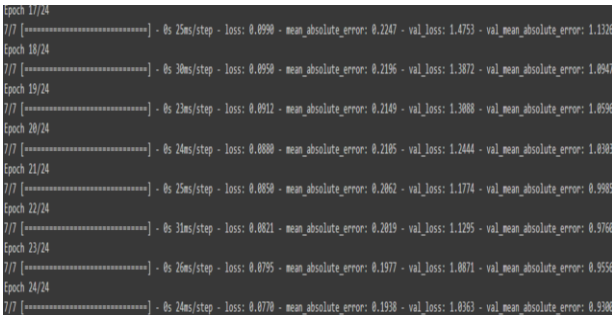
Konfigurasi Model Single Step

Untuk konfigurasi model single step dengan RNN adalah sebagai berikut; layer LSTM menggunakan 32 unit, dan layer dense menggunakan 1 unit. Perhatikan cuplikan kode berikut:

INTI NUSA MANDIRI
DOI: https://doi.org/10.33480/inti.v17i1.3277
VOL. 17. NO. 1 AGUSTUS 2022
P-ISSN: 0216-6933 | E-ISSN: 2685-807X
Diterbitkan Oleh: LPPM Nusa Mandiri

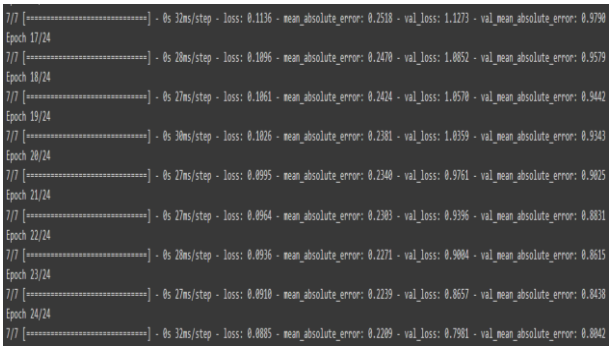
```
# RNN menggunakan single step --------
------------------------------
lstm_model = tf.keras.models.Sequentia
l([
    # Shape [batch, time, features] =>
 [batch, time, lstm_units]
    tf.keras.layers.LSTM(32, return_se
quences=True),
    # Shape => [batch, time, features]
    tf.keras.layers.Dense(units=1)

])
```



Sumber: (akil, 2022)

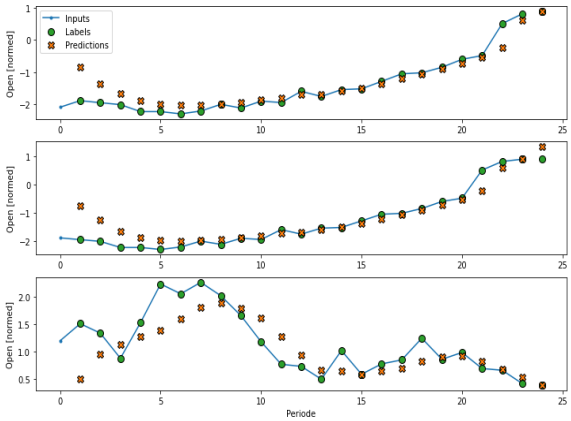
Gambar 4. Hasil Training RNN Single Step Model (Kolom Open)



Sumber: (akil, 2022)

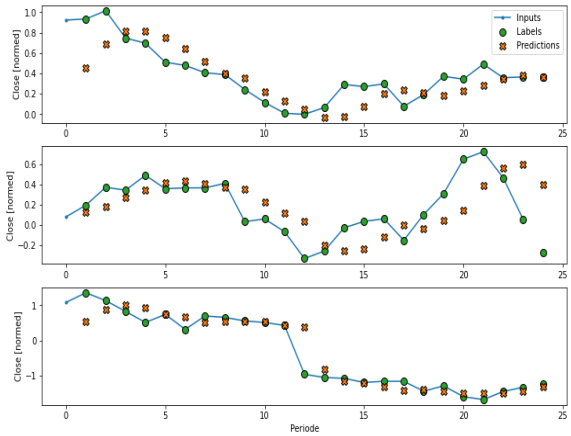
Gambar 5. Hasil Training RNN Single Step Model (Kolom Close)

Gambar 4 diatas adalah hasil training dari RNN dengan single step model untuk kolom Open, sedangkan gambar 5 untuk kolom Close. Nilai loss terakhir yang didapatkan dari 24 epochs adalah sebesar 0.0770 (Open), 0.0885 (Close). sedangkan nilai value loss sebesar 1.0363 (Open), 0.7981 (Close), dan nilai MAE sebesar 0.9300 (Open), 0.8042 (Close). Untuk hasil plot atau chart dapat dilihat pada gambar 6 dibawah ini. Dapat dilihat pada awal prediksi yang ditandai dengan titik (x) menunjukkan nilai yang jauh dari nilai target/label (titik berwarna hijau), namun nilai prediksi (titik x paling kanan) pada ke-tiga partisi tersebut, hampir mendekati nilai label yang sesungguhnya. Sedangkan untuk kolom Close yaitu gambar 7, hanya partisi ke-satu dan ke-tiga saja yang mendekati nilai target/label.



Sumber: (akil, 2022)

Gambar 6. Keluaran Plot RNN dengan Single Step Model (Kolom Open)



Sumber: (akil, 2022)

Gambar 7. Keluaran Plot RNN dengan Single Step Model (Kolom Close)

Konfigurasi Model Multi Step

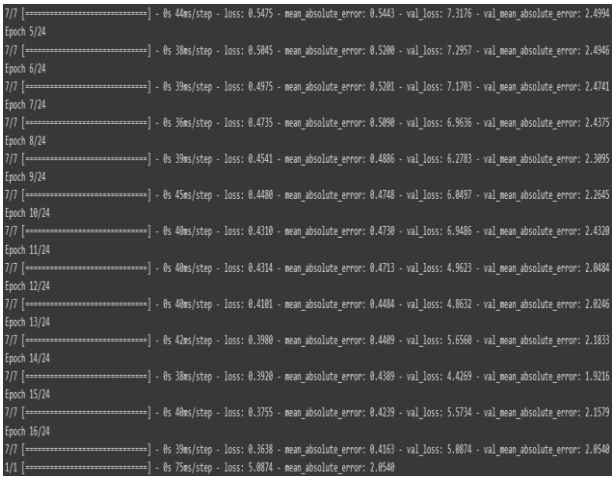
Untuk konfigurasi model multi step dengan RNN adalah sebagai berikut; untuk layer LSTM menggunakan 128 units. Dan jumlah epoch yang sama (24)

```
# RNN menggunakan multi step models --
------------------------------
```

VOL. 17. NO. 1 AGUSTUS 2022
P-ISSN: 0216-6933 | E-ISSN: 2685-807X
Diterbitkan Oleh: LPPM Nusa Mandiri

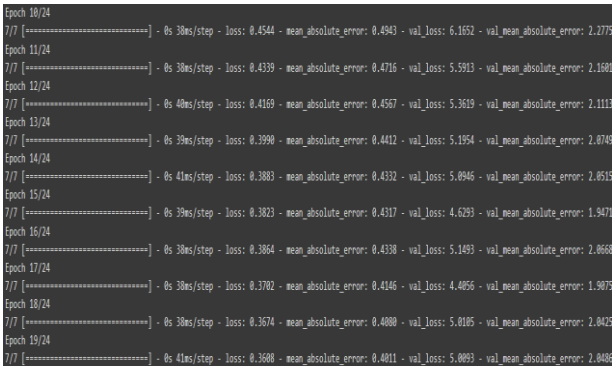
INTI NUSA MANDIRI
DOI: https://doi.org/10.33480/inti.v17i1.3277

```
multi_lstm_model = tf.keras.Sequential
([
    # Adding more `lstm_units` just ov
erfits more quickly.
    tf.keras.layers.LSTM(128, return_s
equences=False),
    # Shape => [batch, out_steps*featu
res].
    tf.keras.layers.Dense(OUT_STEPS*nu
m_features,
                          kernel_initi
alizer=tf.initializers.zeros()),
    # Shape => [batch, out_steps, feat
ures].
    tf.keras.layers.Reshape([OUT_STEPS
, num_features])
])
```



Sumber: (akil, 2022)

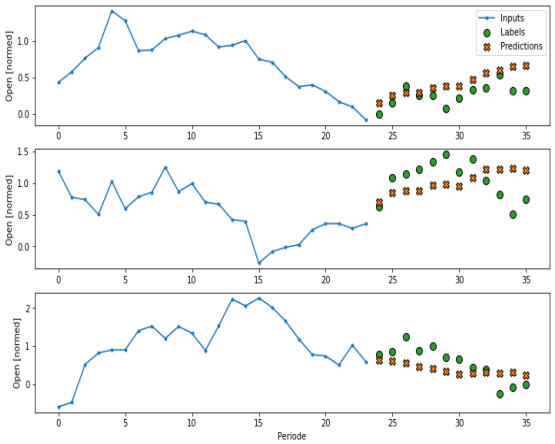
Gambar 8. Hasil Training RNN dengan Multi Step Model (kolom Open)



Sumber: (akil, 2022)

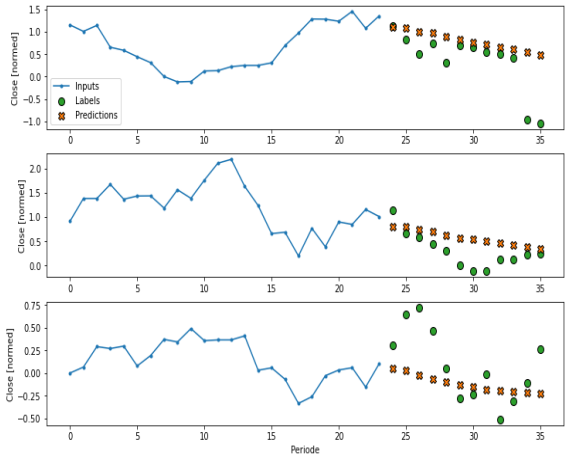
Gambar 9. Hasil Training RNN dengan Multi Step Model (kolom Close)

Setelah ditraining sebanyak 24 epoch, pada epoch ke 16 didapatkan nilai loss sebesar 0.3638 (Open) gambar 8, dan 0.3864 (Close) gambar 9, nilai value loss sebesar 0.4163 (Open), dan 5.1493 (Close), dan nilai MAE sebesar 2.0540 (Open), dan 2.0668 (Close). Untuk output plot-nya dapat dilihat pada gambar 10 untuk kolom Open, dan gambar 11 untuk kolom Close.



Sumber: (akil, 2022)

Gambar 10. Keluaran Plot RNN dengan Multi Step Model (Kolom Open)



Sumber: (akil, 2022)

Gambar 11. Keluaran Plot RNN dengan Multi Step Model (Kolom Close)

Evaluasi Model

Dari dua model yang dicoba yaitu single step model dan multi step model, RNN lebih akurat untuk single step model, sedangkan multi step model, notasi prediksi RNN terlihat kaku, perlu dilakukan optimasi konfigurasi model agar didapatkan hasil yang lebih maksimal.

KESIMPULAN

Penggunaan model RNN dengan menggunakan LSTM sebagai hidden layer di dalam neural network, dengan menggunakan dua model prediksi yaitu; single step dan multi step, di dapatkan bahwa untuk single step RNN cukup akurat dalam memprediksi data satu hari ke depan. Sedangkan untuk multi step, RNN tampaknya kehilangan dinamisme dalam memprediksi nilai yang bersifat turun naik. Sehingga untuk memprediksi data 12 hari berikutnya tampak kaku. Hal ini dapat dijadikan untuk penelitian lebih lanjut model RNN untuk multi step model.

REFERENSI

Afifah, L. (2022, Juli 12). *3 Metode Normalisasi Data (Feature Scaling) di Python*. Retrieved from ilmudatapy.com: https://ilmudatapy.com/metode-normalisasi-data/

Chamidah, N., Wiharto, & Salamah, U. (2012). Pengaruh Normalisasi Data pada Jaringan Syaraf Tiruan Backpropagation Gradient Descent (BPGDAG) untuk Klasifikasi. *Jurnal ITSmart*.

Chen, L.-P. (2020). Using Machine Learning Algorithms on Prediction of Stock Price. *Journal of Modeling and Optimization*, 84-99.

Hochreiter, S., & Schmidhubber, J. (1997). Long Short-Term Memory. *Neural Computation*.

Janeski, M., & Kalajdziski, S. (2010). Forecasting Stock Market Prices. *ICT Innovations 2010.*

Mandala, P. W., Wahyuni, M. A., & Atmadja, A. T. (2019). Determinasi Trader Dalam Pengambilan Keputusan Analisis Trading di Pasar Valas. *JIMAT*, 161-172.

Ong, E. (2008). *Technical Analysis for Mega Profit.* Jakarta: Gramedia.

Redaksi OCBC. (2022, 04 26). *Trading Saham: Definisi, Cara, dan Bedanya dengan Investasi*. Retrieved from OCBC NISP Web site: https://www.ocbcnisp.com/id/article/2021/11/04/trading-saham

Titin. (2015). Analisis Pengambilan Keputusan dalam Transaksi Trading Forex di Fxindo Regional Lamongan. *Jurnal Ekbis*, 689-695.

Yi, Z., & Tan, K. (2004). *Convergence Analysis of Recurrent Neural Network.* Springer Science + Business media.

Yuliara, I. M. (2016, Maret). Regresi Linear Sederhana. *Regresi Linear Sederhana*. Universitas Udayana.

Zen, H., & Sak, H. (2015). Unidirectional Long Short-Term Memory Recurrent Neural Network With Recurrent Output Layer for Low-Latency Speech Sythesis. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.