

Pengembangan Model RNN LSTM pada Prediksi Produksi Daging Sapi dalam Perencanaan Pembangunan Nasional

Yulianingsih¹, Tri Yani Akhirina², Za'imatun Niswati³

¹²³ Department of Informatic Engineering, Universitas Indraprasta PGRI, Indonesia

Article Info

Article history:

Received Apr 10, 2022

Revised June 19, 2022

Accepted Oct 14, 2022

Keywords:

Data

Recurrent Neural Network

LSTM

RMSE

ABSTRACT (10 PT)

Data from one time can be used for another processing afterwards. This means that predictions in the future can be made by looking at additional information from previous data. RNN (Recurrent Neural Network) LSTM (Long Short-Term Memory) is a deep learning architecture that has such capability with a number of looping processes carried out backwards. LSTM is a variant of RNN to overcome the problem of vanishing gradient which requires a number of good time series data, this study attempt to observe how the predicted results of a data (x) on a certain time segment (t) in the time series data. The data used are statistical data on beef meat production and product consumption per capita per week from BPS by taking 12 records for each data source. The results were predictions of beef meat production and product consumption in the following few years with the best results using 3 years data with RMSE 32121,297 for beef meat production and 0.001 for product consumption. The purpose of this study is expected to provide information support for national development planning decision makers in meeting the supply needs of national food needs.

Copyright © 2022 Universitas Indraprasta PGRI.
All rights reserved.

Corresponding Author:

Yulianingsih,

Department of Informatic Engineering,

Universitas Indraprasta PGRI,

Jl. Nangka No. 58 C, Tanjung Barat, Jagakarsa, Jakarta Selatan.

Email: yuliaunindra@gmail.com

1. PENDAHULUAN

Berdasarkan amanat Undang-Undang Nomor 25 Tahun 2004 Tentang Sistem Perencanaan Pembangunan Nasional, perencanaan pembangunan harus didasarkan pada data dan informasi yang akurat dan dapat dipertanggungjawabkan. Dengan demikian ketersediaan data merupakan komponen utama dalam proses penyelenggaraan pembangunan, karena data akan mendukung dalam pengambilan kebijakan/keputusan, alat kontrol untuk mencegah terjadinya kesalahan serta mendukung penyelenggaraan pemerintahan yang transparan, akuntabel dan partisipatif [1]. Data harus dapat diolah menjadi sumber informasi yang dapat memberikan makna salah satunya dengan melakukan prediksi.

Kebutuhan daging sapi di Indonesia sangatlah besar, terutama menjelang hari-hari raya. Berdasarkan data yang diperoleh bahwa Indonesia masih belum dapat mencukupi swasembada daging sapi [2]. Upaya untuk mencapai swasembada daging akan dihadapkan pada banyak tantangan. Pergerakan yang cenderung menurun terjadi pada Industri sapi potong pada 40 tahun terakhir [3]. Oleh sebab itu agar dapat memenuhi kebutuhan

masyarakat, perlu dilakukan prediksi produksi daging sapi. Hasil prediksi ini dapat digunakan untuk pihak-pihak pemangku kepentingan untuk mengambil kebijakan dalam memenuhi kebutuhan nasional.

Dalam setiap pengambilan keputusan prediksi dapat dianggap sebagai komponen utama. Efektifitas suatu keputusan umumnya tergantung dari beberapa faktor yang tidak dapat kita lihat pada waktu keputusan diambil yang didasarkan pada data yang ada pada waktu sekarang dan waktu lampau [4]. *Time series forecasting* adalah sebuah area pada *machine learning* dimana waktu merupakan *attribute* yang diamati. Analisa berdasarkan urutan data yang sekuensial terhadap waktu, dimana prediksi pada masa yang akan datang diperoleh berdasarkan data pada waktu sebelumnya.

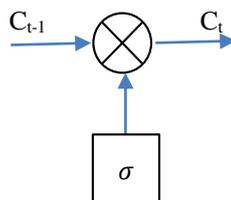
Recurrent Neural Network (RNN) adalah arsitektur mendalam yang banyak digunakan untuk pemodelan urutan, generasi, dan prediksi [5]. LSTM dan RNN konvensional telah berhasil diterapkan pada berbagai tugas prediksi dan pelabelan urutan. Mekanisme RNN memungkinkan untuk dapat mengeksplorasi jendela kontekstual yang berubah secara dinamis atas riwayat urutan input daripada yang statis seperti pada jendela berukuran tetap yang digunakan dengan jaringan umpan-maju [6]. Dalam pemodelan bahasa, RNN konvensional telah memperoleh pengurangan signifikan dalam kebingungan dibandingkan model n-gram standar [7] dan model LSTM RNN telah menunjukkan peningkatan dibanding RNN konvensional.

Long Short Term Memory (LSTM) merupakan bagian dari jaringan syaraf tiruan *Recurrent Neural Network* (RNN) yang bekerja berdasarkan urutan dan dilakukan secara berulang. Informasi yang ada pada waktu sebelumnya menjadi bagian masukan bagi informasi saat ini yang saling terhubung dan dapat digunakan sebagai prediksi pada masa yang akan datang [8]. Model LSTM telah terbukti memiliki kinerja yang lebih baik daripada RNN pada pembelajaran bahasa bebas konteks dan peka konteks [9]. LSTM berisi informasi di luar aliran normal dari *recurrent network* dalam *gate cell*. Keputusan tentang apa yang harus disimpan dan kapan mengizinkan pembacaan, penulisan dan penghapusan, melalui *gate* yang terbuka dan tertutup merupakan fungsi dari *cell*. *Gate* menyampaikan informasi berdasarkan kekuatan yang masuk ke dalamnya dan akan difilter menjadi bobot dari *gate* itu sendiri. Bobot tersebut sama seperti bobot input dan *hidden* unit yang disesuaikan melalui proses *learning* pada *recurrent network* [10].

Melalui riset ini penulis mencoba untuk melakukan prediksi produksi daging sapi dan konsumsi produk pada waktu tertentu yaitu 3 dan 6 tahun dengan pendekatan *time series forecasting* menggunakan RNN LSTM (*Recurrent Neural Network Long Short Term Memory*). Penulisan didukung oleh data statistik yang bersumber dari data statistik dan kesehatan hewan, Direktorat Jenderal Peternakan dan Kesehatan Hewan Kementerian Pertanian RI dari tahun 2007 sampai dengan 2018 dan data BPS dari situs BPS. Data yang diolah adalah produksi daging yang berasal dari ternak besar atau karkas hasil pemotongan ternak di wilayah nasional ditambah dengan *edible offal* (bagian yang dapat di makan) selama waktu tertentu dan data konsumsi produk peternakan perkapita perminggu. Data yang digunakan sebanyak 12 records untuk setiap sumber data (produksi daging dan konsumsi produk).

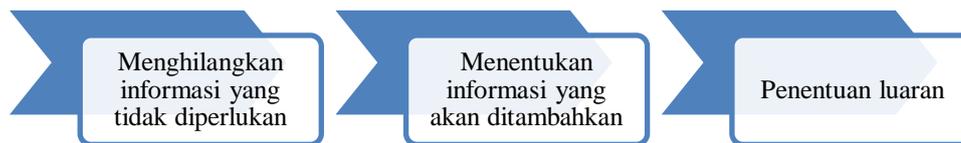
2. METODE

Model prediksi yang diusulkan dikembangkan menggunakan arsitektur *Recurrent Neural Network* (RNN) *Long Short Term Memory* (LSTM) yang dianggap lebih mampu menyelesaikan permasalahan dependensi jangka panjang atau permasalahan *vanishing gradient* pada RNN. Dengan memberikan gerbang jalur (*sigmoid*, σ) yang menghubungkan antara *cell state* sebelumnya (C_{t-1}) dengan *cell state* (C_t) untuk mengatur apakah sebuah informasi dapat diteruskan atau tidak dengan lebih baik [11] ilustrasi diberikan pada gambar 1.



Gambar 1. gerbang LSTM tunggal
Sumber: Suyanto *et al*

Untuk menyelesaikan arsitektur LSTM maka dilakukan sejumlah tahapan kegiatan yang dapat menyusun rangkaian gerbang tersebut :



Gambar 2. Tahapan pengembangan model

Tahapan 1, inialisasi bobot, *forget gate* menggunakan fungsi sigmoid utk menghilangkan informasi yang tidak diperlukan dengan persamaan 1.

$$f_t = \sigma(w_f \cdot [s_{t-1}, x_t] + b_f) \quad (1)$$

Tahapan 2, menentukan informasi yang akan ditambahkan ke *cell state* C_t yang diperoleh dari hasil penggabungan menggunakan dua fungsi sigmoid untuk mentransformasikan nilai antara -1 dan 1 menjadi nilai antara 0 dan 1 dan fungsi tanh yang mengatur nilai selalu berada antara -1 dan 1. Kemudian dilakukan perkalian pada kedua hasil fungsi tersebut yang akan menjadi nilai C_t , kemudian ditambahkan dengan *output* dari *forget gate* dari langkah pertama. Berikut persamaan 2a, 2b dan 2c yang menggambar langkah pada tahapan kedua.

$$i_t = \sigma(w_i \cdot [s_{t-1}, x_t] + b_i) \quad (2a)$$

$$C_t = \tanh(w_c \cdot [s_{t-1}, x_t] + b_c) \quad (2b)$$

$$C_t = f_t * C_{t-1} + i_t * C_t \quad (2c)$$

Tahapan 3, *Output LSTM* diperoleh dengan mengalikan nilai *output gate* dan tanh dari C_t dengan persamaan sebagaimana persamaan 3a dan 3b untuk mendapatkan hasil akhir dari arsitektur LSTM.

$$o_t = \sigma(w_o \cdot [s_{t-1}, X_2] + b_o) \quad (3a)$$

$$s_t = o_t * \tanh(C_t) \quad (3b)$$

Model prediksi dibangun dengan neuron sejumlah 4, *epoch* 100 dan *Batch size* 1. Format data yang digunakan CSV platform Jupyter Notebook dan Google Colab menggunakan modul keras. Masing-masing data dianalisis dengan algoritma menggunakan LSTM *forecast* model untuk one step univariate *time series forecasting* problem. Algoritma optimizer yang digunakan untuk menghitung RMSE adalah *Adaptive Moment Estimation* (Adam).

3. HASIL DAN PEMBAHASAN

RNN merupakan bagian dari *area machine learning* yang berbasis ekstraksi fitur dari sebuah data secara lebih rinci sekaligus merupakan sebuah neural network yang dikembangkan dari Jaringan Syaraf Tiruan (JST) yang memiliki kemiripan yang hampir sama dengan dengan arsitektur *Multi Layer Perception* (MLP).

Deep Learning merupakan bagian dari *machine learning* yang meniru sistem kerja dari otak manusia dimana informasi yang diperoleh dikarenakan adanya hubungan antar neuron. LSTM dikembangkan untuk menyelesaikan permasalahan memori jangka pendek yang ada ditemukan pada RNN [12].

Riset dimulai dengan langkah awal melakukan unggah data pada Github untuk masing-masing data produksi sapi dan konsumsi sapi dengan format CSV. Melakukan load dataset dari CSV file ke Github <https://raw.githubusercontent.com/yuliagniya/data/master/sapi-produksi.csv> sementara untuk data produksi daging sapi dan konsumsi produk peternakan <https://raw.githubusercontent.com/yuliagniya/data/master/sapi-konsumsi.csv> untuk data konsumsi produk peternakan.

Masing-masing sebanyak 12 rows data dari tahun 2007 hingga 2018 jumlah data yang digunakan terbatas untuk melihat terapan data time series pada arsitektur LSTM, terlampir pada tabel 1 data produksi daging dan konsumsi.

Tabel 1. Data Produksi Daging dan Konsumsi

Tahun	Produksi	Konsumsi
2007	339479	0.008
2008	392511	0.007
2009	409308	0.006
2010	436450	0.007

2011	485335	0.009
2012	508905	0.007
2013	504819	0.005
2014	497669	0.005
2015	506660.77	0.008
2016	518484.03	0.008
2017	486319.65	0.009
2018	497971.7	0.009

Tahapan selanjutnya melakukan perubahan data sesuai dengan model LSTM: Mengubah data menjadi *supervised learning problem*

```
# frame a sequence as a supervised learning problem
def timeseries_to_supervised(data, lag=1):
    df = DataFrame(data)
    columns = [df.shift(i) for i in range(1, lag+1)]
    columns.append(df)
    df = concat(columns, axis=1)
    df.fillna(0, inplace=True)
    return df
```

Kemudian mengubah data menjadi *stationary*

```
# create a differenced series
def difference(dataset, interval=1):
    diff = list()
    for i in range(interval, len(dataset)):
        value = dataset[i] - dataset[i - interval]
        diff.append(value)
    return Series(diff)

# invert differenced value
def inverse_difference(history, yhat, interval=1):
    return yhat + history[-interval]
```

Diikuti dengan proses mengubah data sehingga memiliki skala dari -1 sampai dengan 1

```
# scale train and test data to [-1, 1]
def scale(train, test):
    # fit scaler
    scaler = MinMaxScaler(feature_range=(-1, 1))
    scaler = scaler.fit(train)
    # transform train
    train = train.reshape(train.shape[0], train.shape[1])
    train_scaled = scaler.transform(train)
    # transform test
    test = test.reshape(test.shape[0], test.shape[1])
    test_scaled = scaler.transform(test)
    return scaler, train_scaled, test_scaled
```

Menyesuaikan model jaringan LSTM stateful ke dalam data pelatihan

```
# fit an LSTM network to training data
def fit_lstm(train, batch_size, nb_epoch, neurons):
    X, y = train[:, 0:-1], train[:, -1]
    X = X.reshape(X.shape[0], 1, X.shape[1])
    model = Sequential()
    model.add(LSTM(neurons, batch_input_shape=(batch_size, X.shape[1], X.shape[2]), stateful=True))
    model.add(Dense(1))
    model.compile(loss='mean_squared_error', optimizer='adam')
```

Mengevaluasi model LSTM pada data uji

```
for i in range(nb_epoch):
    model.fit(X, y, epochs=1, batch_size=batch_size, verbose=2, shuffle=False)
    model.reset_states()
```

Langkah akhir dari alur algoritma adalah menghasilkan laporan kinerja perkiraan.

```
# walk-forward validation on the test data
predictions = list()
for i in range(len(test_scaled)):
    # make one-step forecast
    X, y = test_scaled[i, 0:-1], test_scaled[i, -1]
    yhat = forecast_lstm(lstm_model, 1, X)
    # invert scaling
    yhat = invert_scale(scaler, X, yhat)
    # invert differencing
    yhat = inverse_difference(raw_values, yhat, len(test_scaled)+1-i)
    # store forecast
    predictions.append(yhat)
    expected = raw_values[len(train) + i + 1]
    print('Month=%d, Predicted=%0.3f, Expected=%0.3f' % (i+1, yhat, expected))
```

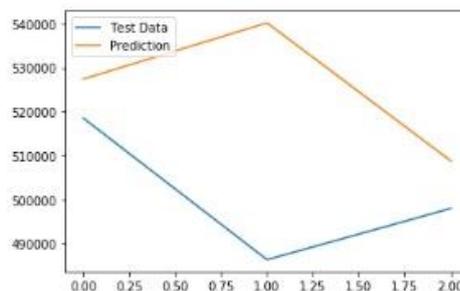
Kemudian dilakukan prediksi selama 3 tahun dan 6 tahun kedepan terhadap data produksi daging dengan memperlihatkan prediksi pada tiap tahunnya sebagaimana hasil yang diperlihatkan pada tabel 2 hasil prediksi dalam 3 tahun dan prediksi dalam 6 tahun tabel 2 berikut.

Tabel 2. Perbandingan hasil prediksi 3 dan 6 tahun

3 Tahun	6 Tahun
Y=1, Predicted=527391.882, Expected=518484.030	Y=1, Predicted=540946.210, Expected=504819.000
Y=2, Predicted=540159.995, Expected=486319.650	Y=2, Predicted=543151.691, Expected=497669.000
Y=3, Predicted=508797.618, Expected=497971.700	Y=3, Predicted=540720.410, Expected=506660.770
	Y=4, Predicted=550008.353, Expected=518484.030
	Y=5, Predicted=561117.103, Expected=486319.650
	Y=6, Predicted=533385.176, Expected=497971.700
Test RMSE: 32121.297	Test RMSE: 45417.445

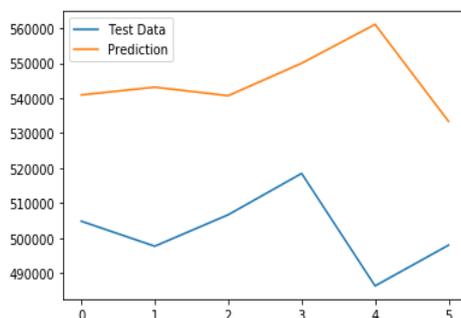
Semua perkiraan pada kumpulan data uji dikumpulkan dan skor kesalahan dihitung untuk merangkum ketrampilan model, dan melihat seberapa besar kesalahan antara prediksi dengan obeservasi (tingkat akurasi) dengan *Root Mean Squared Error* (RMSE). Dimana nilai RMSE terbaik adalah mendekati nol, maka berdasarkan hasil pengujian model dapat disimpulkan bahwa prediksi produksi daging sapi dengan data time serer 3 tahun lebih akurat jika dibandingkan dengan 6 tahun dengan capaian nilai RMSE 32121.297 hal juga dapat disimpulkan semakin banyak tahun prediksi maka semakin jauh tingkat akurasi yang diperoleh.

Berdasarkan hasil prediksi tersebut dilakukan plot data atau penggambaran hasil prediksi dalam bentuk grafik untuk mempermudah pembacaan hasil. Dapat terlihat pada grafik bahwa gap pada gambar 1 (data 3 tahun) lebih kecil dibandingkan pada gambar 2 (data 6 tahun).



Gambar 3. Plot line data untuk 3 tahun produksi daging

Berdasarkan hasil prediksi pada 6 tahun dilakukan plot data dengan hasil sebagai berikut



Gambar 4. Plot Produksi Daging Dalam 6 Tahun

4. PENUTUP

Hasil prediksi merupakan informasi yang dapat digunakan untuk membantu manajemen atau pemangku kebijakan dalam melakukan pengambilan keputusan. Pada sejumlah data yang berkelanjutan (time series) arsitektur LSTM menjadi bagian yang dapat dipertimbangkan. Harapan kedepannya dapat dilakukan pengujian pada beberapa varian LSTM lainnya.

DAFTAR PUSTAKA

- [1] Direktorat Jenderal Peternakan dan Kesehatan Hewan, *Statistik Peternakan dan Kesehatan Hewan 2018/ Livestock and Animal Health Statistics 2018*. 2018.
- [2] S. Rusdiana, "Fenomena Kebutuhan Pangan Asal Daging Dapat Dipenuhi Melalui Peningkatan Usaha Sapi Potong Di Petani," *SOCA J. Sos. Ekon. Pertan.*, vol. 13, no. 1, p. 61, 2019.
- [3] N. Ilham, "Kebijakan Pengendalian Harga Daging Sapi Nasional," *Kebijak. Pengendali. Harga Daging Sapi Nas.*, vol. 7, no. 3, pp. 211–221, 2016.
- [4] A. A. Rizal and S. Hartati, "PREDIKSI KUNJUNGAN WISATAWAN DENGAN RECURRENT NEURAL NETWORK EXTENDED KALMAN FILTER Program Studi Informatika , STMIK Bumigora Mataram Jurusan Ilmu Komputer dan Elektronika , FMIPA UGM , Yogyakarta," vol. X, no. 1, pp. 7–18, 2017.
- [5] A. Mali, A. Ororbia, D. Kifer, and C. L. Giles, "Recognizing Long Grammatical Sequences Using Recurrent Networks Augmented With An External Differentiable Stack," 2020.
- [6] D. Lee *et al.*, "Long short-term memory recurrent neural network-based acoustic model using connectionist temporal classification on a large-scale training corpus," *China Commun.*, vol. 14, no. 9, pp. 23–31, 2017.
- [7] T. Mikolov, M. Karafiát, L. Burget, C. Jan, and S. Khudanpur, "Recurrent neural network based language model," *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH 2010*, no. September, pp. 1045–1048, 2010.
- [8] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks," no. September, 2019.
- [9] F. A. Gers and J. Schmidhuber, "LSTM recurrent networks learn simple context-free and context-sensitive languages," *IEEE Trans. Neural Networks*, vol. 12, no. 6, pp. 1333–1340, 2001.
- [10] A. A. Rizal and S. Soraya, "Multi Time Steps Prediction Dengan Recurrent Neural," vol. 18, no. 1, pp. 115–124, 2018.
- [11] Suyanto, N. R. Kurniawan, and M. Satria, *Deep Learning Modernisasi Machine Learning Untuk Big Data*. Informatika, 2019.
- [12] E. C. Djamal, "Prediksi Harga Saham menggunakan Metode Recurrent Neural Network," *Semin. Nas. Apl. Teknol. Inf. 2019*, p. A-33-A-38, 2019.