

Forecasting Monthly Inflation Rate in Denpasar Using Long Short-Term Memory

I Wayan Sumarjaya

Department of Mathematics, FMIPA, Universitas Udayana, Indonesia
e-mail: sumarjaya@unud.ac.id

Made Susilawati

Department of Mathematics, FMIPA, Universitas Udayana, Indonesia
e-mail: mdsusilawati@unud.ac.id

Abstract: *One of indicators of economic stability of a country is controlled inflation. In general, inflation provides information about the rise of goods and services in a region within a certain period which has strongly related to people's ability to purchase. The Covid-19 pandemic has affected almost any sectors especially the consumer price index. Bali, as a major tourist destination in Indonesia, has been severely affected by the pandemic. Information about future inflation rate plays an important role in determining the correct decision regarding economic policy. The aim of this research is to forecast the inflation rate in Denpasar using a deep learning method for time series. Deep learning, a part of machine learning, consists of layers of neurons that are designed to learn complex patterns and is able to make forecasting. In this research we deployed a special type of recurrent neural networks called long short-term memory (LSTM) that is suitable for use in time series analysis. We stacked the networks into two, three, and four layers to add capacity and to build deep networks for inflation rate series. A grid search for each layer is conducted to obtain optimal hyperparameters setting. We conclude that the optimum architecture for setting this deep network is stacked two LSTM layers. The monthly inflation rate forecasts suggest the inflation for 2022 fluctuates, but below one percent.*

Keywords: *inflation forecasting, time series deep learning, machine learning, long short-term memory, deep neural networks*

1. Introduction

One of indicators of economic stability of a country is controlled inflation. In general, inflation is the increase in goods and services within certain regions which in turn lowers the consumers purchase. The Covid-19 pandemic has affected all aspects of life, especially the economy such as the lower consumers price index. Bali, a major tourist destination in Indonesia, has been severely affected by the pandemic.

Bank Indonesia's August 2020 report (Bank Indonesia, 2020) for economic growth in the Province of Bali stated that the economic growth in the second quarter of 2020 was -

10,98% and inflation figures of 2,18%. Although the rate was lower, there was a fluctuation in the inflation rate see dynamic table of inflation of Badan Pusat Statistik Provinsi Bali (2021). The negative figures for inflation suggest that there was something wrong with the economy. Thus, the government needs information about how to control inflation.

Many researchers made forecast inflation rates in many cities in Indonesia. Stephani et al. (2015) predicted the national inflation rate using classical time series (ARIMA, ARIMAX, and transfer function models) and adaptive neuro fuzzy inference systems (ANFIS). Stephani et al. (2015) concluded that for general type inflation with money supply as input the transfer function models performs better than the other competing models and for food group type inflation ARIMAX performs better than the other models. As noted further in Stephani et al. (2015) finished food and transportation group type inflation are best modeled by time series with intervention, education inflation is best modeled with ARIMA, and health inflation is best modelled with ANFIS. The study of various methods of inflation forecasting method in Stephani et al. (2015) suggests that both methods have strength and weakness in terms of the type of inflation. Wulandari et al. (2016) predicted inflation rate in Surabaya using ARIMA, calendar variation, and intervention analyses. The study concluded that the general type inflation in Surabaya is best modelled with intervention models while for finished food, beverages, cigarettes, tobaccos, and clothing group type inflation are best modeled with calendar variation and ARIMA. Nawawi (2017) forecast monthly inflation rate in Bandung for year 2016 using ARIMA model and concluded that longer forecast periods tend to be less accurate. Similar study conducted by Santoso and Kharisma (2020) which forecast inflation rate in Bandung for year 2020 using autoregressive moving average (ARMA) and dummy seasonal concluded that the dummy seasonal significantly improved the forecast. Rukini (2015) forecast inflation rate in Denpasar using ARIMA exogenous. Rukini (2015) used foreign tourist visits, fuel prices, electricity basic tariffs, and Bali bombing as exogenous variables and concluded that the foreign tourist visits and fuel prices play significant role to the inflation rate. Finally, Purnama and Permana (2019) used fuzzy time series (FTS) and multilayer perceptron (MLP) to forecast inflation rate in Denpasar and concluded that the FTS performs better than MLP based on the mean square error value.

The above paragraph suggests that both classical time series method and machine learning methods which mainly based on neural networks may provide both weakness and strength. Another promising method that is of interest is deep learning for time series. The popularity of deep learning is mainly due to its successful applications in computer vision such as image classification. Deep learning is basically neural networks with many hidden layers. There many architectures of neural networks for deep learning. Among the popular architectures for deep learning are convolutional neural networks (CNN) and recurrent neural networks (RNN). RNN is specifically designed for sequence-to-sequence data

such as time series data. Furthermore, a specific type of RNN called long short-term memory (LSTM) capable of doing time series forecasting.

The article is organized as follows. In section one, we introduced the main motivation behind inflation forecasting. Section two discusses the methodology of our research. Results and discussion are presented in section three. Section four concludes the article and presents a few remarks.

2. Research Methodology

Deep learning, a subset of machine learning, gains its popularity after successful implementation in computer vision such as image classification. In addition, deep learning has been successfully applied to time series prediction mainly due to the wide availability of open-source packages or modules. A survey about application of deep learning in time series forecasting can be seen, for example, in Lim and Zohren (2021) and Torres et al. (2021).

In general, the architecture of deep learning for time series is recurrent neural networks (RNN). RNN is designed to handle sequential data such as text, audio, and video (Yu et al., 2019). RNN uses state variables to store past information and uses current input to determine current output. However, as it is further noted by Zhang et al. (2021), RNN suffers from numerical instability. Another problem of RNN is its inability to learn information when input gap is large (Yu et al., 2019).

Modern RNN such as long short-term memory (LSTM) introduces a memory cell or cell which is a kind of hidden state to record information (Zhang et al., 2021). LSTM has been successfully applied in fields such as unconstrained handwriting recognition, speech recognition, handwriting generation, machine translation, image captioning, and parsing (Goodfellow et al., 2016).

An LSTM cell consists of three gates: the input gate \mathbf{i}_t , the forget gate \mathbf{f}_t , and the output gate \mathbf{o}_t (see Figure 1).

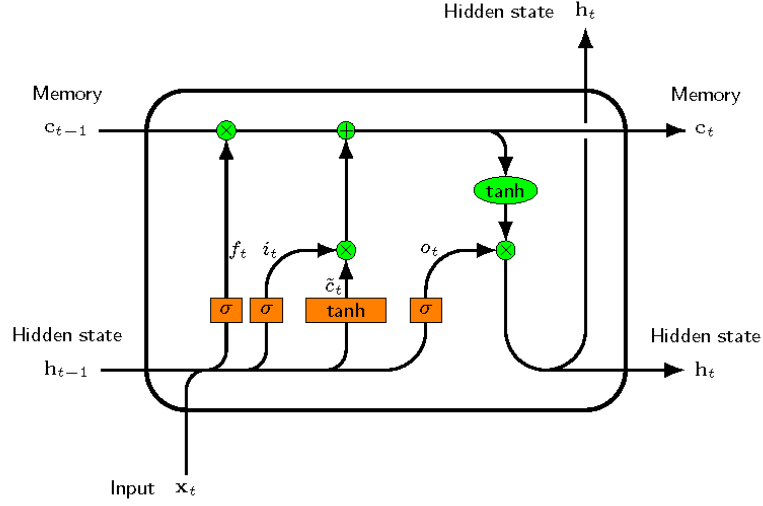


Figure 1. An architecture of a long short-term memory (LSTM) cell.

The role of each of the gates can be explained as follows. The forget gate decides what information to be removed from the cell state. If the value of the forget gate is 1, it keeps the data; otherwise, the data is discarded. A decision about when to read the data in the cell is done in the input gate. Finally, the output gate reads out the data from the cell. The relation among the gates can be seen in equations (1).

Suppose there are h hidden units, n batch sizes, and d inputs. Let $\mathbf{x}_t \in \mathbf{R}^{n \times d}$ be the input and $\mathbf{h}_{t-1} \in \mathbf{R}^{n \times h}$ be the hidden states of previous time step. Each of these gates is related to the activation function σ (usually sigmoid activation function) which has its input hidden states, weights, and biases. More specifically, these gates are calculated as follows (Yu et al., 2019) and (Zhang et al., 2021):

$$\begin{aligned}
 \mathbf{f}_t &= \sigma(\mathbf{x}_t \mathbf{w}_{xf} + \mathbf{h}_{t-1} \mathbf{w}_{hf} + \mathbf{b}_f), \\
 \mathbf{i}_t &= \sigma(\mathbf{x}_t \mathbf{w}_{xi} + \mathbf{h}_{t-1} \mathbf{w}_{hi} + \mathbf{b}_i), \\
 \tilde{\mathbf{c}}_t &= \tanh(\mathbf{x}_t \mathbf{w}_{x\tilde{c}} + \mathbf{h}_{t-1} \mathbf{w}_{h\tilde{c}} + \mathbf{b}_{\tilde{c}}), \\
 \mathbf{c}_t &= \mathbf{f}_t \cdot \mathbf{c}_{t-1} + \mathbf{i}_t \cdot \tilde{\mathbf{c}}_t, \\
 \mathbf{o}_t &= \sigma(\mathbf{x}_t \mathbf{w}_{xo} + \mathbf{h}_{t-1} \mathbf{w}_{ho} + \mathbf{b}_o), \\
 \mathbf{h}_t &= \mathbf{o}_t \cdot \tanh(\mathbf{c}_t).
 \end{aligned} \tag{1}$$

In equations (1) \mathbf{w}_{xf} , \mathbf{w}_{xi} , $\mathbf{w}_{x\tilde{c}}$, and \mathbf{w}_{xo} are weights with respect to the input x_t and are of dimension $\mathbf{R}^{d \times h}$. Similarly, \mathbf{w}_{hf} , \mathbf{w}_{hi} , $\mathbf{w}_{h\tilde{c}}$, and \mathbf{w}_{ho} are weights with respect to the hidden state \mathbf{h}_{t-1} and are of dimension $\mathbf{R}^{h \times h}$. Biases with respect to forget gate, input gate,

candidate memory, and output gate are denoted by \mathbf{b}_f , \mathbf{b}_i , \mathbf{b}_c , and \mathbf{b}_o and are of dimension $\mathbf{R}^{1 \times h}$.

The memory cell \mathbf{c}_t in (1) plays important role in addressing the problem of vanishing gradient and handling long dependencies in sequential data (Goodfellow et al., 2016). The next important component of an LSTM architecture is the hidden state which passes all data through the predictor when the output gate approximates one. However, when the gate closes to 0 the data received no further processing.

The LSTM layers can be stacked to add capacity and depth (see Figure 2). As can be seen from Figure 2 the output of l -th layer at time t is h_t^l and will be treated as input x_{t+1} of $(l+1)$ -th layer. Suppose that the sequence length is 5. We can unroll the stacked LSTM network in Figure 2 along the time dimension. This unrolled network can be seen in Figure 3.

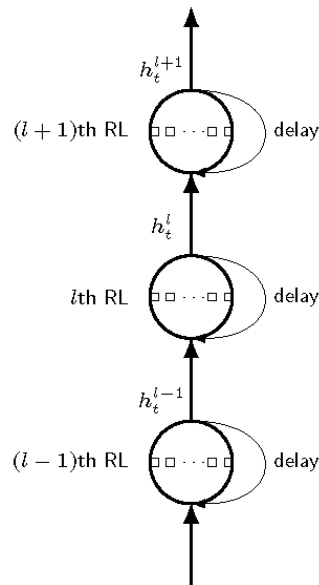


Figure 2. A stacked LSTM network with three recurrent layers (RL): $l-1$, l , and $l+1$. The small rectangular boxes are LSTM cells. This figure is adapted from Yu et al. (2019) with slight modification.

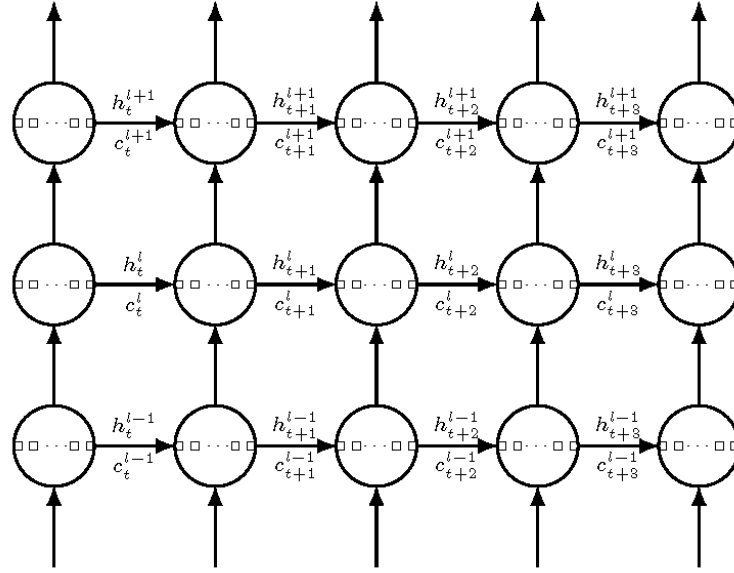


Figure 3. An unrolled stacked LSTM network with sequence length 5. This figure is adapted from Yu et al. (2019) with slight modification.

The l -th LSTM layer can be written as follows (Yu et al., 2019)

$$\begin{aligned}
 \mathbf{f}_t^l &= \sigma(\mathbf{w}_{hf}^l \mathbf{h}_{t-1}^l + \mathbf{w}_{xf}^l \mathbf{h}_t^{l-1} + \mathbf{b}_f^l), \\
 \mathbf{i}_t^l &= \sigma(\mathbf{w}_{hi}^l \mathbf{h}_{t-1}^l + \mathbf{w}_{xi}^l \mathbf{h}_t^{l-1} + \mathbf{b}_i^l), \\
 \tilde{\mathbf{c}}_t^l &= \tanh(\mathbf{w}_{hc}^l \mathbf{h}_{t-1}^l + \mathbf{w}_{xc}^l \mathbf{h}_t^{l-1} + \mathbf{b}_c^l), \\
 \mathbf{c}_t^l &= \mathbf{f}_t^l \cdot \mathbf{c}_{t-1}^l + \mathbf{i}_t^l \cdot \tilde{\mathbf{c}}_t^l, \\
 \mathbf{o}_t^l &= \sigma(\mathbf{w}_{ho}^l \mathbf{h}_{t-1}^l + \mathbf{w}_{wo}^l \mathbf{h}_t^{l-1} + \mathbf{b}_o^l), \\
 \mathbf{h}_t^l &= \mathbf{o}_t^l \cdot \tanh(\mathbf{c}_t^l).
 \end{aligned}$$

An excellent review of LSTM can be found in Schmidhuber (2015) and Yu et al. (2019) among others.

Due to its data-driven nature, the LSTM modeling relies heavily on the availability of open source or free software such as Python and various modules such as tensorflow, scikit learn, pandas, numpy, and keras among others. Steps to model and predict time series data, inflation rate in particular, are as follows.

1. Data partition

The data is split into training and testing dataset. For this particular purpose the proportions are as follows: training (70%) and testing (30%). Thus, we have 268 data for training and 111 data for testing.

2. Feature engineering

In this step the monthly inflation series are formulated as a supervised learning

problem. Here, the lag or window method is applied by creating twelve series which were lagged one period. This twelve-lagged series will serve as the features or co-variates and the original inflation series will serve as the label or the response in the supervised learning problem.

3. Data transformation

A suitable transformation is needed in order for the training and prediction process runs smoothly and match the activation functions. In this case, tangent hyperbolic (tanh) and sigmoid activation functions were used.

4. LSTM model specification

LSTM layers are stacked to build deep neural networks. At this stage, the tanh is used for activation function, hard sigmoid is used for recurrent activation, and linear is set for the dense layer (see Equation (1)). The root mean square error (RMSE) is used as a criterion to evaluate the fitness of the model and adaptive moment estimation (adam) is used as optimizer.

5. Setting hyperparameters

There is no hard and fast rule to find the best setting for the hyperparameters, especially for time series data. Thus, in order to get the optimal settings for hyperparameters a grid search can be done.

6. Compile and fit the models

Next, the model is compiled and fitted based on optimal settings for the hyperparameters obtained in the previous step.

7. Checking accuracy and tuning hyperparameters

In order to check for accuracy, the RMSE is used. Whenever the results of the training and prediction are unsatisfactory, the hyperparameters such as the batch sizes and epochs can be tuned. In addition, the decision of including look back or not can be done at this stage.

8. Prediction

In this step we the prediction that is obtained from the training and testing data is compared. Once this meets the criterion, the forecasting of monthly inflation rate can be proceeded.

Figure 4 shows the flowchart of this LSTM modeling.

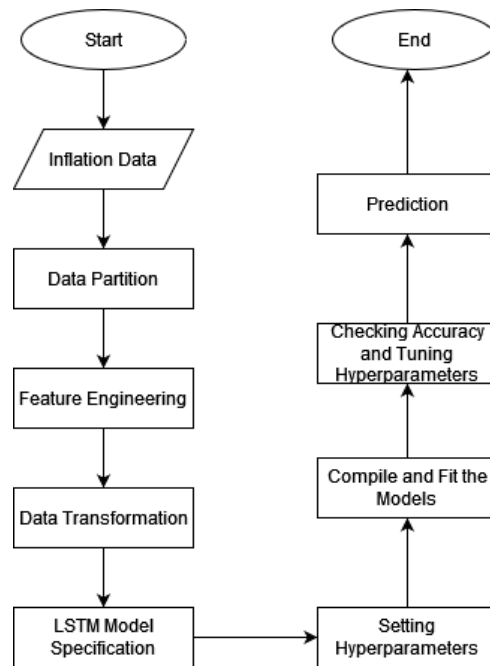


Figure 4. Flowchart of LSTM modeling

3. Results and Discussions

The data used in this research is the monthly Denpasar inflation rate from January 1990 to June 2021. This data can be accessed through Badan Pusat Statistik webpage (Badan Pusat Statistik Provinsi Bali, 2020) and (Badan Pusat Statistik Provinsi Bali, 2021). There are 378 months in this data and the plot of this series can be seen in Figure 5.

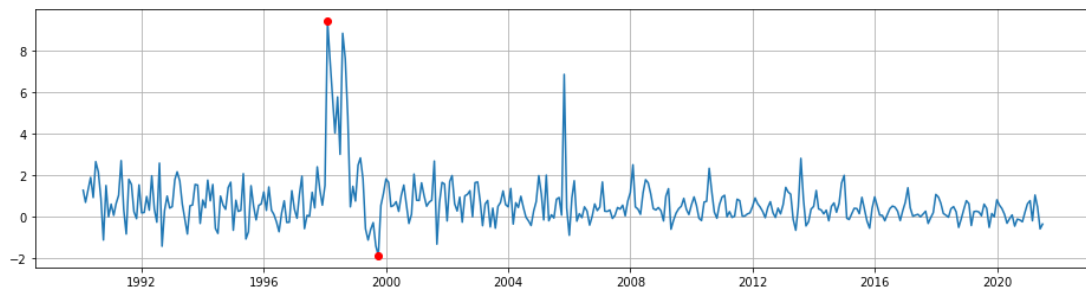


Figure 5. Denpasar inflation rate from January 1990 to June 2021.

As can be seen from Figure 5 the series look stationary, except that the rate soars 9.42 percent in early 1998 (see the first red dot after 1996) which is related to the monetary crisis in Indonesia. The average monthly inflation rate in Denpasar over the last three decades is 0.67. See Table 1 for summary of descriptive statistics of the inflation rate.

Table 1. Summary Statistics for Monthly Denpasar Inflation Rate

mean	sd	min	25%	50%	75%	max
0.67	1.25	-1.88	0.06	0.44	0.98	9.42

As can be seen from Table 1, the minimum inflation is -1.88 percent (see the second red dot closed to year 2000) which occurs several months after the 1997—1998 monetary crisis. This wild fluctuation suggests a period of economic instability during that time.

Prior to modeling the inflation using LSTM, the data is split into training and testing. Figure 6 shows training data (268 points connected by line in blue) and training data (111 points connected by line in orange).

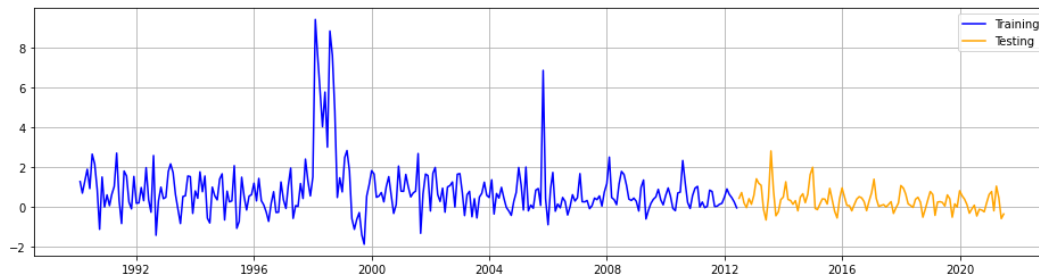


Figure 6. Training dan testing dataset for Denpasar inflation rate.

In section 2, it is mentioned that in order to obtain optimal settings for hyperparameters a grid search can be done. In our research the LSTM cells are stacked into two, three, and four layers and the combination of the following settings:

1. nodes: 25, 50, 75, 100;
2. epochs: 10, 25, 50, 100;
3. batches: 1, 10, 25, 50, 75, 100.

The total number of configurations from all the above settings is 96. The results of each stacked LSTM layers and their corresponding RMSE can be seen in Tables 2, 3, and 4. In each of these tables the settings for hyperparameters (HP) are determined by the number of nodes (N), epochs (E), and batch sizes (B).

Tabel 2. Hyperparameters Configuration and Corresponding RMSE for the Stacked Two LSTM Layers

HP N, E, B	RMSE	HP N, E, B	RMSE	HP N, E, B	RMSE	HP N, E, B	RMSE
25, 10, 1	0.609	25, 10, 10	0.588	25, 10, 25	0.576	25, 10, 50	0.574
25, 10, 75	0.573	25, 10, 100	0.620	25, 25, 1	0.584	25, 25, 10	0.578
25, 25, 25	0.582	25, 25, 50	0.572	25, 25, 75	0.581	25, 25, 100	0.572
25, 50, 1	0.570	25, 50, 10	0.564	25, 50, 25	0.592	25, 50, 50	0.587
25, 50, 75	0.579	25, 50, 100	0.586	25, 100, 1	0.624	25, 100, 10	0.553
25, 100, 25	0.586	25, 100, 50	0.587	25, 100, 75	0.585	25, 100, 100	0.586
50, 10, 1	0.577	50, 10, 10	0.602	50, 10, 25	0.593	50, 10, 50	0.588
50, 10, 75	0.567	50, 10, 100	0.553	50, 25, 1	0.635	50, 25, 10	0.586

50, 25, 25	0.587	50, 25, 50	0.580	50, 25, 75	0.590	50, 25, 100	0.594
50, 50, 1	0.604	50, 50, 10	0.575	50, 50, 25	0.587	50, 50, 50	0.599
50, 50, 75	0.580	50, 50, 100	0.579	50, 100, 1	0.676	50, 100, 10	0.550
50, 100, 25	0.566	50, 100, 50	0.594	50, 100, 75	0.597	50, 100, 100	0.586
75, 10, 1	0.600	75, 10, 10	0.583	75, 10, 25	0.578	75, 10, 50	0.585
75, 10, 75	0.583	75, 10, 100	0.569	75, 25, 1	0.611	75, 25, 10	0.607
75, 25, 25	0.573	75, 25, 50	0.591	75, 25, 75	0.580	75, 25, 100	0.585
75, 50, 1	0.642	75, 50, 10	0.566	75, 50, 25	0.584	75, 50, 50	0.574
75, 50, 75	0.580	75, 50, 100	0.574	75, 100, 1	0.663	75, 100, 10	0.600
75, 100, 25	0.562	75, 100, 50	0.611	75, 100, 75	0.586	75, 100, 100	0.587
100, 10, 1	0.575	100, 10, 10	0.578	100, 10, 25	0.588	100, 10, 50	0.591
100, 10, 75	0.587	100, 10, 100	0.592	100, 25, 1	0.646	100, 25, 10	0.579
100, 25, 25	0.596	100, 25, 50	0.576	100, 25, 75	0.578	100, 25, 100	0.587
100, 50, 1	0.620	100, 50, 10	0.562	100, 50, 25	0.581	100, 50, 50	0.583
100, 50, 75	0.595	100, 50, 100	0.581	100, 100, 1	0.640	100, 100, 10	0.654
100, 100, 25	0.554	100, 100, 50	0.586	100, 100, 75	0.558	100, 100, 100	0.579

Table 2 shows the hyperparameter settings for stacked two LSTM layers. The numbers 25, 10, 1 under HP means hyperparameter settings with 25 nodes in the first layer, 25 nodes in the second layer, 10 epochs, and 1 batch size. Similarly, the number 100, 100, 100 under HP means 100 nodes in the first layer, 100 nodes in the second layer, 100 epochs, and 100 batch sizes. For each of these combinations the RMSE is calculated. The optimal settings for the hyperparameters in the two LSTM layers are shown by the numbers 50, 100, 10 and RMSE of 0.550 in the table (highlighted in bold).

Table 3. Hyperparameters Configuration and Corresponding RMSE for the Stacked Three LSTM Layers

HP N, E, B	RMSE	HP N, E, B	RMSE	HP N, E, B	RMSE	HP N, E, B	RMSE
25, 10, 1	0.602	25, 10, 10	0.612	25, 10, 25	0.572	25, 10, 50	0.594
25, 10, 75	0.569	25, 10, 100	0.608	25, 25, 1	0.591	25, 25, 10	0.585
25, 25, 25	0.587	25, 25, 50	0.601	25, 25, 75	0.597	25, 25, 100	0.583
25, 50, 1	0.584	25, 50, 10	0.565	25, 50, 25	0.579	25, 50, 50	0.585
25, 50, 75	0.588	25, 50, 100	0.583	25, 100, 1	0.617	25, 100, 10	0.557
25, 100, 25	0.558	25, 100, 50	0.598	25, 100, 75	0.578	25, 100, 100	0.580
50, 10, 1	0.625	50, 10, 10	0.588	50, 10, 25	0.589	50, 10, 50	0.574
50, 10, 75	0.584	50, 10, 100	0.562	50, 25, 1	0.633	50, 25, 10	0.591
50, 25, 25	0.594	50, 25, 50	0.584	50, 25, 75	0.592	50, 25, 100	0.600
50, 50, 1	0.618	50, 50, 10	0.572	50, 50, 25	0.583	50, 50, 50	0.580
50, 50, 75	0.590	50, 50, 100	0.589	50, 100, 1	0.679	50, 100, 10	0.589
50, 100, 25	0.561	50, 100, 50	0.576	50, 100, 75	0.576	50, 100, 100	0.575
75, 10, 1	0.611	75, 10, 10	0.603	75, 10, 25	0.600	75, 10, 50	0.582
75, 10, 75	0.583	75, 10, 100	0.600	75, 25, 1	0.627	75, 25, 10	0.587
75, 25, 25	0.586	75, 25, 50	0.604	75, 25, 75	0.590	75, 25, 100	0.587
75, 50, 1	0.619	75, 50, 10	0.590	75, 50, 25	0.599	75, 50, 50	0.631
75, 50, 75	0.597	75, 50, 100	0.592	75, 100, 1	0.634	75, 100, 10	0.622
75, 100, 25	0.573	75, 100, 50	0.590	75, 100, 75	0.580	75, 100, 100	0.582

100, 10, 1	0.619	100, 10, 10	0.600	100, 10, 25	0.604	100, 10, 50	0.596
100, 10, 75	0.604	100, 10, 100	0.586	100, 25, 1	0.645	100, 25, 10	0.611
100, 25, 25	0.590	100, 25, 50	0.585	100, 25, 75	0.587	100, 25, 100	0.584
100, 50, 1	0.610	100, 50, 10	0.590	100, 50, 25	0.601	100, 50, 50	0.597
100, 50, 75	0.595	100, 50, 100	0.607	100, 100, 1	0.637	100, 100, 10	0.596
100, 100, 25	0.555	100, 100, 50	0.612	100, 100, 75	0.566	100, 100, 100	0.575

Table 3 shows the RMSE values for 96 hyperparameter settings for stacked three LSTM cells. The numbers 25, 10, 1 under HP means hyperparameters settings with 25 nodes in the first layer, 25 nodes in the second layer, 25 nodes in the third layer, 10 epochs, and 1 batch size. For this stack, the optimal setting is 100, 100, 25 with RMSE value 0.555.

Table 4. Hyperparameters Configuration and Corresponding RMSE for the Stacked Four LSTM Layers

HP N, E, B	RMSE	HP N, E, B	RMSE	HP N, E, B	RMSE	HP N, E, B	RMSE
25, 10, 1	0.602	25, 10, 10	0.612	25, 10, 25	0.572	25, 10, 50	0.621
25, 10, 75	0.569	25, 10, 100	0.608	25, 25, 1	0.591	25, 25, 10	0.581
25, 25, 25	0.587	25, 25, 50	0.601	25, 25, 75	0.597	25, 25, 100	0.600
25, 50, 1	0.584	25, 50, 10	0.565	25, 50, 25	0.579	25, 50, 50	0.614
25, 50, 75	0.588	25, 50, 100	0.583	25, 100, 1	0.617	25, 100, 10	0.554
25, 100, 25	0.558	25, 100, 50	0.598	25, 100, 75	0.578	25, 100, 100	0.581
50, 10, 1	0.625	50, 10, 10	0.588	50, 10, 25	0.589	50, 10, 50	0.611
50, 10, 75	0.584	50, 10, 100	0.562	50, 25, 1	0.633	50, 25, 10	0.570
50, 25, 25	0.594	50, 25, 50	0.584	50, 25, 75	0.592	50, 25, 100	0.631
50, 50, 1	0.618	50, 50, 10	0.572	50, 50, 25	0.583	50, 50, 50	0.582
50, 50, 75	0.590	50, 50, 100	0.589	50, 100, 1	0.679	50, 100, 10	0.582
50, 100, 25	0.561	50, 100, 50	0.576	50, 100, 75	0.576	50, 100, 100	0.569
75, 10, 1	0.611	75, 10, 10	0.603	75, 10, 25	0.600	75, 10, 50	0.609
75, 10, 75	0.583	75, 10, 100	0.600	75, 25, 1	0.627	75, 25, 10	0.595
75, 25, 25	0.586	75, 25, 50	0.604	75, 25, 75	0.590	75, 25, 100	0.576
75, 50, 1	0.619	75, 50, 10	0.590	75, 50, 25	0.599	75, 50, 50	0.580
75, 50, 75	0.597	75, 50, 100	0.592	75, 100, 1	0.634	75, 100, 10	0.590
75, 100, 25	0.573	75, 100, 50	0.590	75, 100, 75	0.580	75, 100, 100	0.572
100, 10, 1	0.619	100, 10, 10	0.600	100, 10, 25	0.604	100, 10, 50	0.615
100, 10, 75	0.604	100, 10, 100	0.586	100, 25, 1	0.645	100, 25, 10	0.579
100, 25, 25	0.590	100, 25, 50	0.585	100, 25, 75	0.587	100, 25, 100	0.580
100, 50, 1	0.610	100, 50, 10	0.590	100, 50, 25	0.601	100, 50, 50	0.568
100, 50, 75	0.595	100, 50, 100	0.607	100, 100, 1	0.637	100, 100, 10	0.587
100, 100, 25	0.555	100, 100, 50	0.612	100, 100, 75	0.566	100, 100, 100	0.568

The RMSE for four stacked LSTM layers can be seen in Table 4. The minimum RMSE in this stacked is 0.554 which is obtained by the values 25, 100, 10 which means 25 nodes

in the first layer, 25 nodes in the second layer, 25 nodes in the third layer, 25 nodes in the fourth layer, 100 epochs, and 10 batch sizes.

Based on the minimum of value of RMSE among the three models we choose the two stacked LSTM layer with 50 nodes in the first layer, 50 nodes in the second layer, 100 epochs, and 10 batch sizes. For this chosen model there are 10,400 parameters in the first LSTM stack; 20,200 parameters in the second LSTM stack; and 51 parameters in the dense layers.

The prediction of monthly inflation rate of Denpasar can be seen in Figure 7. As can be seen from the figure, the monthly inflation rate is fluctuated but still below 1%. This suggests that the inflation rate is quite stable.

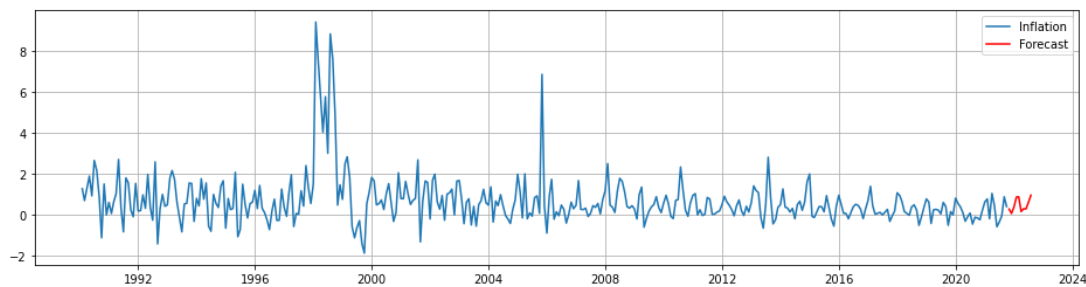


Figure 7. Forecast for Denpasar inflation rate (shown in orange).

4. Conclusions

The prediction of inflation rate of Denpasar suggests that inflation is fluctuated, but it is below 1%. In our experiment, various settings of hyperparameters also lead to almost similar results, for instance, fluctuation in the rate but still below 1%. This might suggest the stationarity of the data.

Our settings for both the stacked of LSTM layers and various hyperparameters settings are limited to our computational resources. This suggests that there might be other hyperparameter settings that can give better results. However, as can be seen in Tables (2)—(4) increasing the number of inputs (nodes), epochs, and batch sizes does not guarantee the smaller RMSE.

Acknowledgment

The authors would like to thank the Dean of FMIPA, Universitas Udayana for supporting this research under the scheme Penelitian Unggulan Program Studi (PUPS). This research is funded from DIPA PNPB Universitas Udayana year 2021 under contract grant

number B/798/UN14.2.28.II/PT.01.03 dated 31 May 2021. The authors also thank two anonymous reviewers for their helpful comments.

References

- Badan Pusat Statistik Provinsi Bali. (2020). *Inflasi Kota Denpasar dan Nasional, 1990-2013* (Vol. 2021, Issue July, 28).
- Badan Pusat Statistik Provinsi Bali. (2021). *Tabel Dinamis Subjek Inflasi* (Vol. 2021, Issue July, 28). <https://bali.bps.go.id/subject/3/inflasi.html#subjekViewTab5>
- Bank Indonesia. (2020). *Laporan Perekonomian Provinsi Bali Agustus 2020*. <https://www.bi.go.id/id/publikasi/kajian-ekonomi-regional/bali/Documents/Laporan%20Perekonomian%20Provinsi%20Bali%20Agustus%202020.pdf>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Lim, B., & Zohren, S. (2021). Time-series Forecasting with Deep Learning: A Survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194), 20200209. <https://doi.org/10.1098/rsta.2020.0209>
- Nawawi, M. (2017). Akurasi ARIMA dalam Peramalan Inflasi Kota Bandung. *Jurnal Manajemen Informatika*, 7(2). <https://doi.org/10.34010/jamika.v7i2.622>
- Purnama, I. N., & Permana, P. T. H. (2019). Perbandingan Peramalan Inflasi Kota Denpasar Menggunakan Metode Fuzzy Time Series dan Multilayer Perceptron. *Jurnal Teknologi Informasi Dan Komputer*, 5(3), 285–296. <http://dx.doi.org/10.36002/jutik.v5i3.801>
- Rukini. (2015). Model ARIMAX dan Deteksi GARCH untuk Peramalan Inflasi Kota Denpasar Tahun 2014. *Jurnal Ekonomi Kuantitatif Terapan*, 7(2), 168–182. <https://doi.org/10.24843/JEKT.2014.v07.i02.p09>
- Santoso, T., & Kharisma, B. (2020). Peramalan Inflasi Kota Bandung dengan Pendekatan Box-Jenkins. *Buletin Studi Ekonomi*, 178–189. <https://doi.org/10.24843/BSE.2020.v25.i02.p01>
- Stephani, C. A., Suharsono, A., & Suhartono. (2015). Peramalan Inflasi Nasional Berdasarkan Faktor Ekonomi Makro Menggunakan Pendekatan Time Series Klasik dan ANFIS. *Jurnal Sains Dan Seni ITS*, 4(1), D67–D72. <http://dx.doi.org/10.12962/j23373520.v4i1.8873>
- Torres, J. F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., & Troncoso, A. (2021). Deep Learning for Time Series Forecasting: A Survey. *Big Data*, 9(1), 3–21. <https://doi.org/10.1089/big.2020.0159>
- Wulandari, N., Setiawan, & Ahmad, I. S. (2016). Peramalan Inflasi Kota Surabaya dengan Pendekatan ARIMA, Variasi Kalender dan Intervensi. *Jurnal Sains Dan Seni ITS*, 5(1), D-90-D-95. <http://dx.doi.org/10.12962/j23373520.v5i1.14693>
- Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, 31(7), 1235–1270.
- Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). *Dive Into Deep Learning*. <https://d21.ai/d21-en.pdf>

