Enhancing Exchange Rate Forecasting: Leveraging Adaptive Moment Estimation in Deep Long Short Term Memory Models Against Foreign Currencies

Sugeng D. Cahyono, Oliver S. Simanjuntak*, and Heru C. Rustamaji

Department of Informatics, Faculty of Industrial Engineering, UPN Veteran Yogyakarta, Sleman Yogyakarta 55281, Indonesia
oliver.simanjuntak@upnyk.ac.id

Abstract. This study focuses on constructing a forecasting model for the Indonesian Rupiah exchange rate against the USD and JPY using DLSTM-ADAM. Exchange rate data from October 2014 to October 2022, sourced from the API website https://ofx.com, is utilized. The preprocessing stages involved normalization and feature sliding window applications. Subsequently, various hyperparameter combinations were employed to train and test the LSTM model. The outcomes emphasize the significance of adding hidden layers to the LSTM model, substantially reducing the RMSE and MAPE values, and enhancing the model's overall performance. Furthermore, the study reveals that the Adam optimizer surpasses SGD regarding training performance. Specifically, for forecasting the IDR/USD exchange rate, the stacked layers and Adam optimizer yielded a remarkable 6,217% reduction in MAPE. Similarly, for the JPY/IDR exchange rate prediction, the MAPE reduction reaches 6,811%. These findings underscore the potential of this architecture for implementing effective time-series data-forecasting models.

Keywords: LSTM, forecasting, time series.

1 Introduction

The currency exchange rate significantly influences the Indonesian economy, particularly in relation to international debt and industrial production costs [1]. The exchange rate significantly impacts Indonesia's economy, affecting international debt and industrial production costs. A low rupiah exchange rate raises the burden of external debt, increasing payments for interest and principal, thereby pressuring the overall economy. Additionally, fluctuating exchange rates elevate import costs for raw materials, raising production expenses for domestic industries [2]. Accurate exchange rate predictions enable proactive measures, assisting policymakers and businesses in managing risks, adjusting production costs, and fostering economic stability and growth in Indonesia. Therefore, currency exchange rate analysis and forecasting are critical in this context [3].

A. Putro Suryotomo and H. Cahya Rustamaji (eds.), *Proceedings of the 2023 1st International Conference on Advanced Informatics and Intelligent Information Systems (ICAI3S 2023)*,
Advances in Intelligent Systems Research 181,
https://doi.org/10.2991/978-94-6463-366-5_22

Previous research has identified that Artificial Neural Networks are an effective method for predicting time-series data, including currency exchange rates [4]. ANNs have dynamic adaptability and learning capabilities, making them the preferred prediction method in this study. Furthermore, LSTM is selected as a variant of ANN because of its ability to address the issue of vanishing gradients and other weaknesses associated with RNNs [5, 6].

Exchange rate data have two significant features: the exchange rate data feature and the time variable, thus classifying the data as a time-series type. The Indonesian Rupiah exchange rate data from the source https://ofx.com shows a pattern of high fluctuation, categorizing the data as non-stationary or having very random patterns. Time-series data are suitably modelled in Recurrent Neural Networks (RNN) because they can produce better learning models than multilayer perceptrons [1]. An RNN has a drawback in multiple weightings, which leads to the Vanishing or Exploding Gradient effect. This phenomenon results in an unstable training processes [3]. The memory cell gate architecture in long short-term memory (LSTM) can prevent or reduce vanishing or exploding gradient phenomena. LSTM is highly suitable for time-series cases owing to the support of Memory Cell gates in each neuron block [7].

The capabilities of LSTM can be enhanced in several ways, such as modifying the LSTM architecture, adding hyperparameters, and modelling the optimizer function. Architectural modifications with a Stacked LSTM or Deep LSTM (DLSTM) model have been used for research on Speech Recognition [8] and Photovoltaic Power Forecasting [9]. In these studies, DLSTM has been proven to reduce the Error Rate and is superior to the conventional LSTM.

In general, updating LSTM weights uses the Stochastic Gradient Descent (SGD) optimizer learning algorithm. This algorithm has the advantage of a simple computational process [10]. However, SGD has several disadvantages: it requires hyperparameters, can become trapped in local optima, and requires many epochs to achieve convergent value [11]. The Adaptive Moment Estimation (Adam) optimizer has been applied to address the problems of the SGD algorithm in the Deep Belief Network (DBN) for time-series forecasting cases [12, 13]. The Adam Optimizer has proven superior to the SGD Optimizer, with lower error values and faster error evaluation processes. The Adam Optimizer reduces error values more rapidly because it uses dynamic learning rates [14].

The data used in this study are the IDR/USD and IDR/JPY exchange rate values. The reason for using this data is that the US Dollar is the global benchmark currency. Additionally, the United States is one of the countries with the most robust currency value and dominates global finance [5]. As cited by CNBC Indonesia, the Japanese Yen occupies the first position as the dominant currency in Asia [15].

In conclusion, seeking an optimal LSTM architecture model remains an interesting research topic because there is no standard rule for determining the architecture structure and hyperparameter values to produce an effective learning model. This research contributes to the implementation of the Adaptive Moment Estimation algorithm on the Deep LSTM architecture, which can be applied in forecasting the exchange rate of the

Indonesian Rupiah (IDR) against foreign currencies. The success of this study is measured based on evaluation metrics such as the Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and Directional Accuracy (DA).

2 Research Method

This study utilizes a quantitative research category, adopting an experimental method with a quasi-experimental type. A Quasi-experimental approach can be used, and the results after treatment are applied to the data [6]. This study aims to understand the performance of ADAM on DLSTM after conducting learning modelling and measuring the model's performance with several measurement metrics. The general flow of research can be described as in Fig. 1.

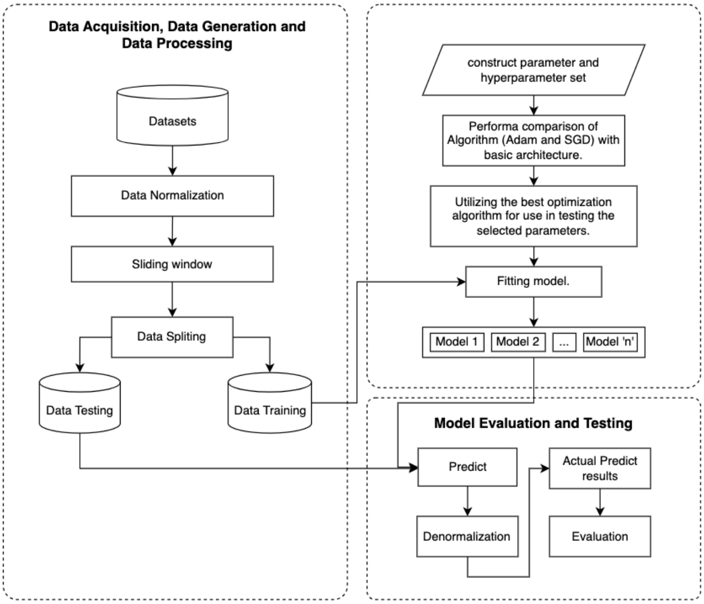


Fig. 1. General design of the prediction model fitting.

This diagram presents a workflow for a machine learning experiment as described in a research paper. Initially, data acquisition, generation, and processing are conducted, involving normalization and segmentation via a sliding window technique, followed by splitting the data into training and testing sets. Next, parameters and hyperparameters are constructed, algorithms (Adam and SGD) are compared, and the optimal algorithm is selected for parameter testing. The selected model is then fitted with the data. Multiple models are generated and subsequently evaluated, where predictions made by the models are denormalized for comparison against actual results, concluding with an evaluation phase to assess the model's performance.

2.1 Data Collection

The data for this study was obtained from an open API source at https://ofx.com, covering the period from October 10, 2014, to October 10, 2022. The acquired data were JSON documents; subsequently, close-price data with daily intervals were extracted. The next step involved processing the data according to testing requirements.

2.2 Preprocessing

The preprocessing stage in this study was carried out progressively as follows: Initially, data normalization was performed using the MinMax Normalization method with a scale of (-1, 1) to achieve convergence of the tanh function. Snippet the dataset series used for model training are shown in Table 1.

Table 1. Sample of datasets

Date	20	21	22	23	24	25
Rates	12112	11973	11985	12018	12058	12045
Norm	-0.941	-1	-0.994	-0.981	-0.964	-0.969

The next step involves applying feature engineering techniques using the sliding window method to create the input data features X and target y. This sliding window technique offers the advantage of a robust understanding of series data. Subsequently, the data were divided into training and test sets at ratios of 80% and 20%.

2.3 Hyperparameters and Model Architecture.

The best model was identified by first defining the variations in the test parameters and determining the LSTM hyperparameters. This study used the test indicators outlined in **Table 2**.

Table 2. Hyperparameter

No	Example	Parameter
1	Input	[7, 14, 30]
2	Neuron	[8, 16, 32, 64]
3	Hidden Size	2
4	Output	1
5	Epochs	100
6	Optimizer	[Adam*, SGD]
7	Learning rate	[0.1, 0.01, 0.001]
8	Dropout	[0.2]

In the case of predicting foreign exchange rates, the hyperparameters listed in the table including the sliding window inputs of 7, 14, and 30 days are utilized to capture the temporal dynamics of currency movements. The choice of these specific windows allows the model to incorporate short-term weekly trends, more extended biweekly patterns, and monthly data trends, which is a common approach in the volatile realm of currency exchange. The output value set to 1 signifies that the model's aim is to forecast the exchange rate for the subsequent day, using the historical data from these defined periods. This is particularly pertinent in currency trading where precision in daily forecasts can significantly impact trading strategies and financial outcomes.

In the model's testing phase, neurons define the network's complexity, with the optimizers—Adam and SGD—guiding how the model adjusts during learning. Adam is often more efficient due to its adaptive learning rate, while SGD is simpler but may require careful tuning. The learning rate controls the optimization step size, critical for effective learning without overshooting or stalling. Dropout at 20% helps prevent overfitting by randomly omitting neurons during training, encouraging a more generalized model. Finally, setting epochs to 100 allows the model to iteratively learn from the data, aiming to optimize performance without overfitting. These hyperparameters are calibrated to improve the model's predictive accuracy on future currency exchange rates.

The stage of constructing the best model architecture involves testing various data with the RMSE, MAPE, and DA metrics. Initially, the optimal learning rate (lr) was determined. Once the optimal lr is identified, testing continues for the best input and the first hidden layer. These tests could be performed concurrently, resulting in 12 model variations, all of which were tested using the above metrics.

The next test was to examine the second-stacked neuron. This test uses the input size and first neuron size from the best test results in the previous step; however, at this stage, four variations of the neuron were tested, resulting in four deep LSTM models from this experiment. The four models were then compared by testing the same test matrix. The steps mentioned above can be illustrated in Fig. 2.

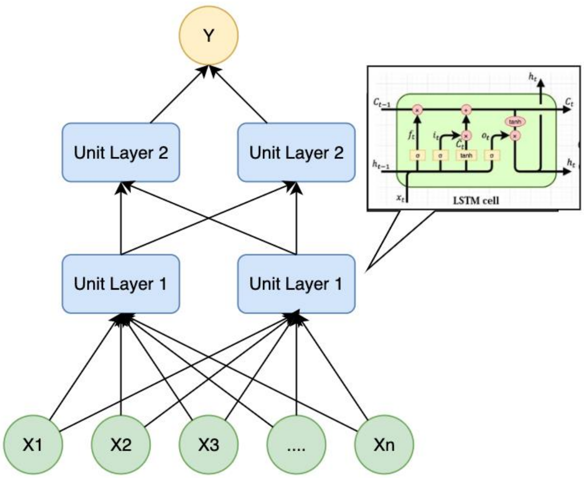


Fig. 2. Model architecture

At each node, the perceptron was replaced by an LSTM equation. The LSTM calculation involves one input signal (X_t) of the original value and two input signals, the hidden (h_{t-1}) input and context input (C_{t-1}), from the residual values of the calculations in the previous session. These input values were processed using LSTM calculations using the following equation:

$$i_t = \sigma(x_t U^i + h_{t-1} W^i) \tag{1}$$

$$f_t = \sigma(x_t U^f + h_{t-1} W^f) \tag{2}$$

$$o_t = \sigma(x_t U^o + h_{t-1} W^o) \tag{3}$$

$$\widetilde{C_t} = \tanh(x_t U^g + h_{t-1} W^g) \tag{4}$$

$$C_t = \sigma\left(f_t * C_{t-1} + i_t * \widetilde{C_t}\right) \tag{5}$$

$$h_t = \tanh(C_t) * o_t \tag{6}$$

The result of h_t is used as the signal *output* and the residual hidden output is $h_{t(n+1)}$ while C_t is used as the signal *context layer*.

2.4 Model Training

The LSTM training process generally uses a Deep Neural Network process that starts with initiating weights and biases, followed by a forward propagation process to obtain the initial error. After obtaining the first loss function, the backpropagation technique was performed while continuously updating the weights (w) and biases (b). This research proposes an Adaptive Moment Estimation algorithm for parameter updating.

The Adam optimization process updates the weight (w) and bias (b) parameters. The aim of Adam's estimation is to optimize parameter updates. In this study, the Adam process was used within the LSTM network. Thus, Adam's input parameters are the LSTM network values. This process can be illustrated in the flow of Fig. 3.

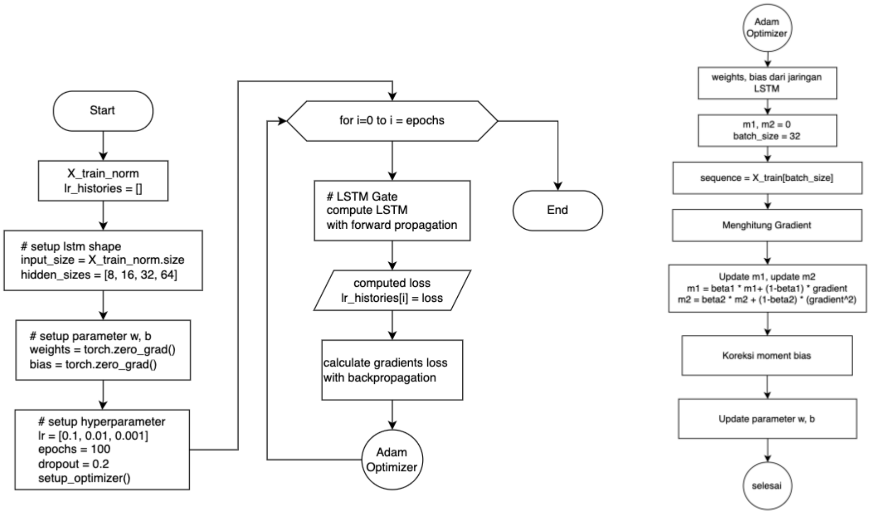


Fig. 3. Training flow of LSTM model and weight updating with Adam optimiser.

2.5 Model Testing

Testing and evaluation were performed using the back-testing method. Backtesting divides the data into two parts: training and testing. This method is often used to test predictive models using historical data [16]. Testing was first conducted using the training data to search for the hyperparameters of the model. The dataset was divided into 80% training data and 20% testing data, with the number of training and testing data determined by the sliding window size.

The model was evaluated using the parameters listed in **Table 2**. This is done to ensure a fair comparison between the methods, allowing for the consideration of the results of each method and making the validation parameters applicable to both methods. The resulting model was evaluated using the RMSE, MAPE, and DA metrics. Testing and evaluation aim to develop a model that provides the best values, achieves accurate forecasting, and fits well. The model building process will be evaluated using the illustration in Fig. 4.

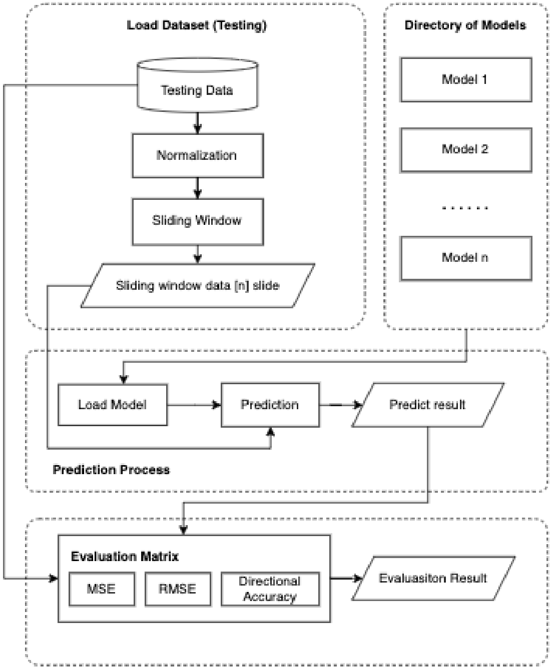
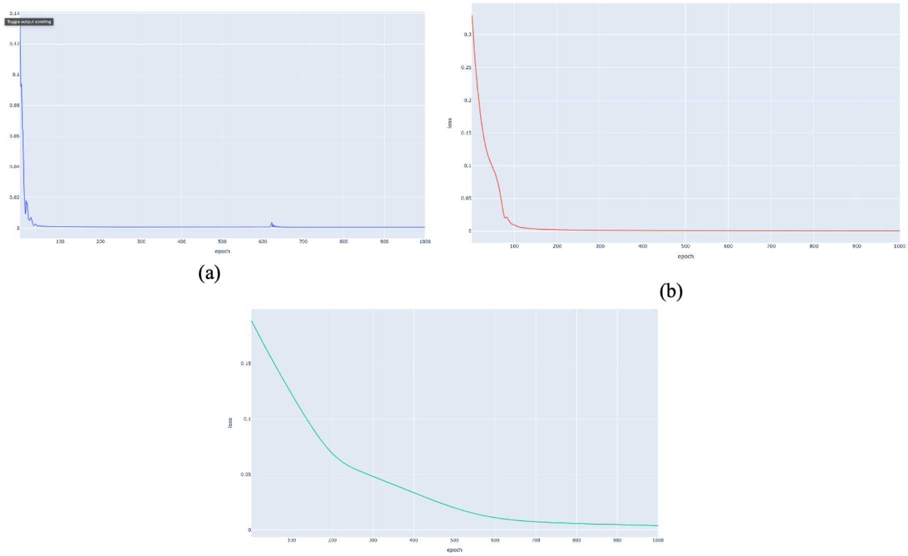


Fig. 4. Model prediction and evaluation

3 Result

3.1 Learning Rate

The test results will produce a record value for decreasing learning rate which is attached in Fig. 5. The results from the learning rate testing indicate that a value of 0.001 (c) has good learning performance, marked by a stable decrease in error and no mid-process error volatility. This contrasts with lr 0.1 (a) and 0.01 (b), which exhibited unstable increases during learning. Hence, this study uses a value of 0.001 as the learning rate (*lr*).



(a)

(b)

(c)

Fig. 5. Learning rate result 0.1(a), 0.01(b) and 0.001(c)

3.2 Evaluation of The Best Input and Stacked Layers.

Testing was conducted by comparing the variations in the models generated from the combination of the input parameters and test neurons. It was performed on all USD/IDR and JPY/IDR datasets. The test results are listed in Table 3. Observations at this stage indicate that the USD/IDR exchange rate performs best with the 14-64-1 architecture, while the JPY/IDR performs best with the 7-64-1 architecture.

Table 3. Evaluation of test results for the first input and stacked layers.

Input	Stacked 1	USD/IDR RMSE (IDR)	USD/IDR MAPE (%)	USD/IDR DA (%)	JPY/IDR RMSE (IDR)	JPY/IDR MAPE (%)	JPY/IDR DA (%)
7	8	633,318	3,805	39,137	2,894	2,164	43,808
7	16	215,352	0,989	39,908	1,522	1,033	44,272
7	32	338,661	1,597	39,753	1,524	1,059	44,118
7	64	120,397	0,602	40,832	**1,520**	**1,032**	**44,372**
14	8	150,558	0,787	39,567	3,081	2,259	43,721
14	16	270,946	1,321	39,258	1,725	1,183	43,876
14	32	159,625	0,764	40,958	1,457	1,016	45,581
14	64	**103,086**	**0,509**	**40,495**	1,264	0,878	44,341
30	8	737,557	4,597	40,683	8,525	6,277	45,016
30	16	164,294	0,784	39,441	1,896	1,310	43,925
30	32	125,585	0,622	40,528	1,482	0,984	44,548
30	64	113,397	0,578	40,528	1,266	0,860	44,081

The next step is to find the best value for the second stacked neuron layer; the process is the same as that for finding the best input and first neuron. This test yielded the best architecture for the USD/IDR currency as 14-64-64-1, while the exchange rate for JPY/IDR is 7-64-64-1. The results of observations at this stage are attached in Table 4.

Table 4. Evaluation of the second stacked layers.

Input	S1	S2	USD/IDR RMSE (IDR)	USD/IDR MAPE (%)	USD/IDR DA (%)	JPY/IDR RMSE (IDR)	JPY/IDR MAPE (%)	JPY/IDR DA (%)
7	64	8				1,607	1,129	47,214
7	64	16				1,333	0,903	46,594
7	64	32				1,425	0,996	45,975
7	64	64				**1,303**	**0,891**	**46,130**
14	64	8	239,455	1,115	41,422			
14	64	16	187,071	0,899	40,958			
14	64	32	106,185	0,527	40,958			
14	64	64	**93,585**	**0,463**	**41,267**			

In Table 3, we see the performance metrics of a single-layered model with various inputs and stacked layers, where the lowest RMSE (Root Mean Square Error) and MAPE (Mean Absolute Percentage Error) for USD/IDR are 103,086 and 0.509% respectively, and the highest DA (Directional Accuracy) for JPY/IDR is 44,372.

Table 4 presents a two-layered stacked model, where the best performance metrics for the same currency pairs improve, with RMSE and MAPE for USD/IDR at 93,585 and 0.463% respectively, and DA for JPY/IDR at 46,130.

The consistent improvement in RMSE, MAPE, and DA across the tables as the number of layers increase provides evidence that adding layers to a model can enhance its predictive accuracy and directional correctness in currency exchange rate forecasting. The DLSTM-Adam model with increased layer complexity shows the best performance, confirming the hypothesis that more complex architectures can capture patterns in data more effectively, leading to improved predictions.

3.3 Evaluation of Optimization Algorithms

After conducting a series of tests that included varying the learning rate, input size, number of neurons in the first LSTM layer (stacked 1), the number of neurons in the second LSTM layer (stacked 2), and the choice of the optimizer, a comparison of the best results from each condition was made. These conditions were the best LSTM, DLSTM-SGD, and DLSTM-Adam. The test results indicated that DLSTM-Adam provided the best performance for all currency exchange values, marked by the smallest Root Mean Square Error (RMSE) and largest Decision Accuracy (DA) as shown in Table 5.

Table 5. Evaluation of the second stacked layers.

Model	USD/IDR RMSE (IDR)	USD/IDR MAPE (%)	USD/IDR DA (%)	JPY/IDR RMSE (IDR)	JPY/IDR MAPE (%)	JPY/IDR DA (%)
LSTM	103,086	0,509	40,495	1,234	0,835	44,118
DLSTM-SGD	1.097,52	6,68	40,65	10,292	7,702	45,046
DLSTM-Adam	93,585	0,463	41,267	1,303	0,891	46,130

Testing was conducted using 20% of the total dataset, equivalent to approximately 600 days. The test results showed that there are two best models, namely DLSTM-Adam and two best models with DLSTM-SGD. The test results indicated that using the Adam optimizer in the DLSTM model reduced the MAPE value by 6.217% for the USD/IDR exchange rate and decreased the MAPE value by 6.811% for the JPY/IDR exchange rate, attached in Fig. 6.

Fig. 6. Test results from the best model *Adam* and *SGD* USD/IDR (a) JPY/IDR (b)

The graph comparison suggests that the DLSTM-Adam model closely follows the actual data trends, indicating higher accuracy and potentially better prediction performance than the DLSTM-SGD model, which shows greater deviation from the actual values. If empirical evidence supports these observations, it would confirm that the Adam optimizer outperforms SGD in this case, likely due to its adaptive learning rate mechanism that can navigate the data and parameter space more effectively.

3.4 Conclusion

The research framework, with the flow of (i) data collection, (ii) normalization, (iii) sliding window, and (iv) data splitting and considering the testing parameters, was able to produce a reasonably good prediction model. This research resulted in the best learning rate (LR) of 0.001 for both exchange rates, as evidenced by the measurement of the best prediction model for the USD/IDR exchange rate with RMSE, MAPE, and DA values of Rp93.585, 0.463%, and 41.267%, respectively. For the JPY/IDR exchange rate, the model produced RMSE, MAPE, and DA values of Rp1.303, 0.819%, and 46.130%, respectively.

References

1. Hazazi, M.A., Sihabuddin, A.: Extended Kalman Filter In Recurrent Neural Network: USDIDR Forecasting Case Study. IJCCS Indones. J. Comput. Cybern. Syst. 13, 293 (2019). https://doi.org/10.22146/ijccs.47802.
2. Nurjanah, R., Mustika, C.: The influence of imports, foreign exchange reserves, external debt, and interest rates on the currency exchange rates against the United States Dollar in Southeast Asia Countries. J. Perspekt. Pembiayaan Dan Pembang. Drh. 9,. https://doi.org/10.22437/ppd.v9i4.12706.
3. Prabowo, A.S., Sihabuddin, A., Sn, A.: Adaptive Moment Estimation On Deep Belief Network For Rupiah Currency Forecasting. IJCCS Indones. J. Comput. Cybern. Syst. 13, 31 (2019). https://doi.org/10.22146/ijccs.39071.
4. Saparudin, A., Maulidina, T.: PREDIKSI NILAI TUKAR DOLLAR (USD) KE RUPIAH (IDR) MENGGUNAKAN ARTIFICIAL NEURAL NETWORK. JUSS J. Sains Dan Sist. Inf. 2, 7 (2019).
5. Semaan, D., Harb, A., Kassem, A.: Forecasting exchange rates: Artificial neural networks vs regression. In: The Third International Conference on e-Technologies and Networks for Development (ICeND2014). pp. 156–161. IEEE, Beirut, Lebanon (2014). https://doi.org/10.1109/ICeND.2014.6991371.
6. Pangayom, G.R., A.S.: Model Peramalan Long-Short Memory Recurrent Neural Network Untuk Peramalan Nilai Tukar Mata Uang Dolar Amerika-Rupiah Indonesia, http://etd.repository.ugm.ac.id/penelitian/detail/114191, (2017).
7. ElSaid, A., Wild, B., Higgins, J., Desell, T.: Using LSTM recurrent neural networks to predict excess vibration events in aircraft engines. In: 2016 IEEE 12th International Conference on e-Science (e-Science). pp. 260–269. IEEE, Baltimore, MD, USA (2016). https://doi.org/10.1109/eScience.2016.7870907.
8. Graves, A., Mohamed, A., Hinton, G.: Speech Recognition with Deep Recurrent Neural Networks. ArXiv13035778 Cs. (2013).
9. Abdel-Nasser, M., Mahmoud, K.: Accurate photovoltaic power forecasting models using deep LSTM-RNN. Neural Comput. Appl. 31, 2727–2740 (2019). https://doi.org/10.1007/s00521-017-3225-z.
10. Hirata, T., Kuremoto, T., Obayashi, M., Mabu, S., Kobayashi, K.: Forecasting Real Time Series Data using Deep Belief Net and Reinforcement Learning. J. Robot. 4, 5 (2018).
11. Ginsburg, B., Castonguay, P., Hrinchuk, O., Kuchaiev, O., Lavrukhin, V., Leary, R., Li, J., Nguyen, H., Zhang, Y., Cohen, J.M.: Stochastic Gradient Methods with Layer-wise Adaptive Moments for Training of Deep Networks. ArXiv190511286 Cs Stat. (2020).
12. Ghazali, G., Si, M.: PREDIKSI SAHAM MENGGUNAKAN DBN (DEEP BELIEF NETWORK) STOCK PREDICTION USING DBN (DEEP BELIEF NETWORK). 16.
13. Sagheer, A., Kotb, M.: Time series forecasting of petroleum production using deep LSTM recurrent networks. Neurocomputing. 323, 203–213 (2019). https://doi.org/10.1016/j.neucom.2018.09.082.
14. Chang, Z., Zhang, Y., Chen, W.: Effective Adam-Optimized LSTM Neural Network for Electricity Price Forecasting. In: 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS). pp. 245–248. IEEE, Beijing, China (2018). https://doi.org/10.1109/ICSESS.2018.8663710.

15. Setiaji, H.: Hari Ini, Rupiah Jadi Mata Uang Terkuat Ketiga di Asia, https://www.cnbcindonesia.com/market/20180426161640-17-12643/hari-ini-rupiah-jadi-mata-uang-terkuat-ketiga-di-asia.
16. Brownlee, J.: Gentle Introduction to Models for Sequence Prediction with RNNs, https://machinelearningmastery.com/models-sequence-prediction-recurrent-neural-networks/, last accessed 2020/04/02.