Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 6, No. 3, August 2021, Pp. 231-238

Gold price prediction using convolutional neural network-long short-term memory (CNN-LSTM)

I Wayan Krisna Gita Santika*[1], Siti Saadah[2], Prasti Eko Yunanto[3]
School of Computing, Telkom University, Indonesia[1,2,3]

Article Info
Keywords:
Gold Price, CNN-LSTM, Deep Learning, Price Prediction, Hyperparameter

Article history:
Received: April 09, 2021
Accepted: August 05, 2021
Published: August 31, 2021

Cite:
Krisna Gita Santika, I. W., Sa'adah, S., & Yunanto, P. E. (2021). Gold price prediction using Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM). Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 6(3).
https://doi.org/10.22219/kinetik.v6i3.1253

*Corresponding author
I Wayan Krisna Gita Santika
E-mail address:
krisnagita@student.telkomuniversity.ac.id

Abstract
Gold has an important role in worldwide economics. Gold is not only used in jewelry but also can be a good deal for investment however several factors can affect the fluctuation in gold which can make the risk of investing in gold is bigger for many people. Therefore, it is very important to predict the gold price for people who invest in gold to help reduce the investment risk. This study implemented a hybrid method from Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) with tuned hyperparameters. CNN can extract useful knowledge and learn the internal representation of time-series data, and LSTM networks will identify short-term and long-term dependencies effectively. Those algorithms were combined in order to predict the gold price in daily time frame and weekly time frame with testing ratio 40% and testing ratio 20%. The best result obtained in the daily time of testing ratio 40% with RMSE 13.67953 and MAE 9,40998, while in testing ratio 20% has RMSE 15,53199 and MAE 10,32953. In weekly time has obtained the RMSE 37,59087 and MAE 28,01416 for testing ratio 40% while in testing ratio 20% the result was RMSE 32,68774 and MAE 22,65841. Those results have showed that CNN-LSTM model with tuned hyperparameters could predict the trend of daily time frame gold price.

1. Introduction

Gold is a global commodity in economic and currency markets. Gold is not only used in jewelry decoration and beautification but also has an important role in industries as a raw material [1]. The importance of gold makes gold a good deal for investment. Investing in gold has proven useful as a form of protection against economic instability [2]. History shows, when some countries had problems with the Balance of Payment they used gold as a protection against loans [3]**.** The price of gold is also having an important part of the world economy. The moment equities and bonds have a poor performance, gold becomes a good alternative to invest and get financial protection [4], but the fluctuation of gold price can increase the investment risk in gold [4]. Several factors can affect the fluctuation of the gold price, such as the exchange of the dollar, the cost of gold production, inflation, monetary policy, and geopolitics [5][6]. The fluctuation of gold price caused by the changes in the price of gold can produce large profits or significant losses for investors [7]. Therefore, the prediction of the gold price will be very important for investors in determining investment plan or policy to be chosen.

In the last two decades, gold price prediction which is a time series problem has been explored by many researchers. They used many methods, from statistical models to a modern way such as the machine learning method. Auto Regressive Moving Average (ARIMA) was applied by Nambier et al. to predict the gold price [8], but ARIMA has the pre-assumed linear from the model as the limitation. During the implementation of machine learning is getting popular, Dubey has developed and compared Support Vector Regression (SVR) and Adaptive Neural Fuzzy Inference System Learning to predict the gold price, the result was SVR method had a better ability to predict than ANFIS [9].

Deep Learning has extensive attention recently in the field of time-series prediction. In recent years, many studies also used deep learning to handled time-series prediction problems, a method called Back Propagation Bidirectional Extreme Learning Machine (BP-BELM) was applied by Zou and Xia in order to predict traffic flow, that method can obtain a 24,1797 RMSE score and more efficient than Bidirectional Extreme Leaning Machine (BELM) [10]. Zheng et al. has developed an algorithm called Deep Belief Network (DBN) to predict exchange rate in INR/USD and CNY/USD, the result was 0,0073 for INR/USD forecasting and 0,0017 for CNY/USD in MAPE score [11]. The study from Zahrah et al. using LSTM to predict Foreign Exchange Rate in COVID-19 pandemic, the results are 0,00624 RMSE in daily price and 0,00135 in 1-hour price [12]. Deep learning can deal with the nature problem and noise of time-series prediction to produce more accurate prediction [5], however selecting optimum hyperparameter is one of the main challenges in applying deep learning. Determine optimum hyperparameter can produce a model that performed better than untuned parameters [13]. Tuning hyperparameter is also often more important than choosing an algorithm [14].

The previous research by Liviries et al. used a hybrid method between Convolutional Neural Network and Long Short-Term Memory (CNN-LSTM) to predicting multivariate daily gold price [5]. The result shows that CNN-LSTM has a better performance than SVR, Feed Forward Neural Network (FFNN) and LSTM in predicting gold. The data from previous was obtained from the Finance Yahoo website. The advantage of CNN can extract useful knowledge and learn the internal representation of time-series data, and LSTM networks will identify short-term and long-term dependencies effectively. Those advantages can lead to a good prediction in gold price time-series prediction, but it is necessary to determine what are the optimum hyperparameters in model CNN-LSTM that can lead to increase performance. The problem discussed in this research is how to identify the optimal hyperparameter in CNN-LSTM to predict the fluctuating gold price.

This research focuses on implementing the CNN-LSTM algorithm with tuned hyperparameters. The CNN-LSTM model from the previous research if observed closely used the fixed number of hyperparameters [4]. The contribution of this research is finding the best hyperparameters in the CNN-LSTM model that can predict the fluctuating gold price with tuning in batch size, LSTM unit, and using early stopping function. The study aimed to make the CNN-LSTM model perform better than the previous research [4]. With the best hyperparameters, it is expected the optimal result to predict the gold price to help people who want to invest in gold avoid losses.

2. Research Method

2.1 Split Data

The data used in this research are 4175 for daily prices and 835 for weekly prices. Those data are from the years 2005 until 2021 in USD rate and obtained from the World Gold Council website, the data is in univariate form. World Gold Council using LBMA Gold Price, as well as several regional prices. The LBMA Gold Price is used as an important benchmark throughout the gold market and the other regional gold price is important for local markets. Some examples of the data are shown in Table 1. The data is split into train data and test data. There are two ratios data that are used in this research, the first is 40% testing ratio and the second is 20% ratio.

Table 1. Examples of Gold Price Data

Date	Daily Price	Date	Weekly Price
2005-03-28	425.15	2005-04-01	427.15
2005-03-29	426.10	2005-04-08	425.20
2005-03-29	426.45	2005-04-15	424.60
2005-03-31	427.50	2005-04-22	434.60
2005-04-01	427.15	2005-04-29	435.70

2.2 Pre-Processing Data

Pre-processing data is a process after split data was done. Data normalization is applied in train data and test data to improve the prediction accuracy [15]. Data normalization also avoids the differences range of the values that are too far away, because the values that are too far will not have much influence on the result from the model [16]. Data normalization in this research using Min-Max Scaling, Equation 1 shown the calculation behind Min-Max Scaling [17] where $\hat{x}$ is data result from normalization, x is data that will be normalized, $max(x)$ is the maximum value from all data, and $min(x)$ is the minimum value of all data. Data train and data test after normalization process are transformed into the shape input of CNN-LSTM, which are input data (feature) and output data (label). The 4 price rows data of gold price become input data, while one row afterward becomes output data.

$$\hat{x} = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

2.3 Convolutional Neural Netwok – Long Short-Term Memory (CNN-LSTM)

Convolutional Networks (CNN) is Neural Network that is created to produce pattern from an image [18]. Deep learning can manipulate features automatically, so it is not necessary to feature extraction models which are often complicated [19]. Deep Learning is divided into 4 parts, Deep Unsupervised Learning (DUL), Deep Supervised Learning (DSL), Deep Semi-Supervised Learning (DSSL), and Deep Reinforcement Learning (DRL) [20]. Based on those 4 parts of deep learning, this research using CNN as Deep Supervised Learning because we split data into train and test.

The parameters of the Convolutional Layer consist of the number of filters (kernels), filter length, and a small set of neurons. While the learning process of the neuron, each filter performs a dot product calculation between the filter itself and the dataset (1D or 2D), that process caused the networks to learn the input data while detecting some specific features such as the characteristics, spatial position, and weight distribution [21].

Max Pooling Layer is usually used in CNN to reduce the amounts of parameters (training weight and filters) and features. Max Pooling Layer will choose the maximum value from a field covered by Pooling Layer [22]. The output of the Max Pooling Layer will be calculated in Fully-Connected Layer using the equation shown in Equation 2. Where y_j is output from Fully-Connected Layer in j^{th} neuron, while n is the length from input data 1D (*x*); $w_{i,j}$ is weight neuron between i^{th} input value and j^{th} neuron and b_1 is bias [21].

$$y_j = \sum_{i=1}^{n} w_{i,j}x_i + b_1 \quad (2)$$

Long Short-Term Memory (LSTM) is one of the Recurrent Neural Networks (RNN) which can learn a long-term dependence through the utilization of the connection. RNN is trying to solve the memory shortage problem in Feed-Forward Networks which causes poor performance on time-series problems [5]. However, because RNN has a Vanishing Gradient, an LSTM was created. LSTM used memory cells and gate units in its architecture. The gate unit consists of several gates, such as Input Gate, Forget Gate, and Output Gate. The input that will be added to the Cell State and only stores the information needed, while the gate that determines the output is Output Gate [23]. LSTM structure is as shown in Figure 1.

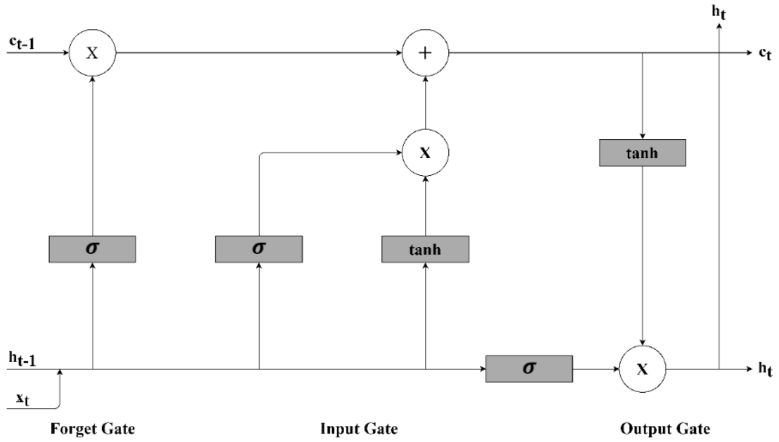

Figure 1. Long Short-Term Memory Structures

Equation 3 is a formula of Forget Gate. Forget Gate will forget information that is not needed and keep important and useful information. Equation 4 is an Input Gate formula. Input Gate will put in new information from the current input value. Equation 5 is a formula of hidden state candidates which is calculated based on previous hidden state and current input. Equation 6 is an Output Gate formula which is a gate that chooses what output is. Equation 7 and Equation 8 are formulas for hidden state and cell state, where * shows the elements multiplication. Equation 9 is a formula for the sigmoid function. Equation 10 is a formula for Hyperbolic activation. In Equation 2,3,4,5,6,7 *t* stand for time. The input to the LSTM unit which *x* in Equation 2,3,4,5,8,9. *W* and *b* are weights and bias vectors in Equations 2,3,4,5 [24].

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (3)$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (4)$$

$$g_t = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \quad (5)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (6)$$

$$c_t = g_t * i_t + c_{t-1} + f_t \quad (7)$$

$$h_t = o_t * \tanh(c_t) \quad (8)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (9)$$

$$\tanh(x) = 2\sigma(2x) - 1 \quad (10)$$

CNN-LSTM is a hybrid algorithm of the Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) methods. Convolutional later can extract knowledge that is useful in studying the internal representation of time-series data., while LSTM is effective for identifying short-term and long-term dependencies. The advantage of these two methods, a merger is carried out [5]. CNN-LSTM architecture is as shown in Figure 2.

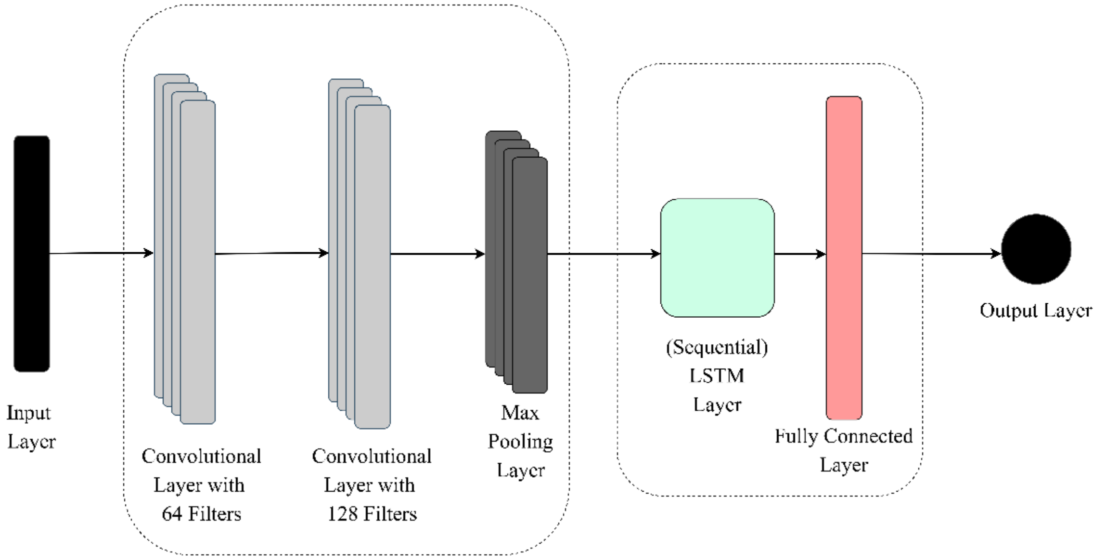

Figure 2. CNN-LSTM Model Architecture

The CNN-LSTM model consists of two main components, namely Convolutional Layer as the main component where mathematical operations are carried out to develop input data features, while the second main components make use of the features generated by LSTM and Dense Layer [5]. Adaptive Moment Estimation (Adam) is also applied as an optimizer. Adam algorithm is easy to implement and has good computational efficiency. Apart from that, Adam algorithm does not require a large amount of memory and is suitable for solving many data or parameter problems [25]. The effectiveness of Adam on CNN is proven to be able to provide a small cost to the training process [26]. In addition to CNN, Adam can also be applied to LSTM to optimize loss functions [23]. The CNN-LSTM model in this research also used Early Stopping function in order to avoid choosing epoch manually.

2.4 Training Process

The training process in this research using a Convolutional Layer with 64 filters and a Convolutional Layer with filter 128 those two layers using kernel size = 2, Max Pooling Layer with pool size = 2, a LSTM Layer and Fully Connected Layer. In CNN and LSTM using activation ReLu. ReLu is one of the most used activation functions since it learns data farther in a network with many layers [17]. The hyperparameter is then selected to become the hyperparameter set for the training process using CNN-LSTM. The hyperparameter candidates are shown in Table 2.

Table 2. Hyperparameter Candidates for CNN-LSTM Model

Hyperparameter	Value
Number of Batch Size	[128, 256, 512]
Number of LSTM Unit	[100, 200, 300]

The training process also used Adam as an optimizer and Early Stopping function. The training process has been done by two processes. The first process used 60% train data, while the second process used 80% train data for each time frame data, daily and weekly. The result of the training process is the CNN-LSTM model which will be used in the testing process.

2.5 Testing Process

The Testing process is carried out by using the test data from the split data as a dataset for the CNN-LSTM model obtained from the training process. The testing process has been done by two processes, the first process is using 40% data test, while the second process using 20% data test for each time frame data, daily and weekly. The result of testing process is a predictable gold price. Detailed system design for all processes can be found in Figure 3.

2.6 Performance Evaluation

Performance evaluation from the proposed model will be evaluated using Root Mean Square (Error) and Mean Absolute Error (MAE). The measurement of a model's performance can be measured using RMSE which is usually used as the standard metrics. RMSE is widely used in predictive modeling in verifying the results obtained from the modeling. The closer RMSE value to 0, the model being measured for performance can be said to be good [27], the same thing happened in MAE. Detailed formulations can be found in [5] for RMSE and MAE.

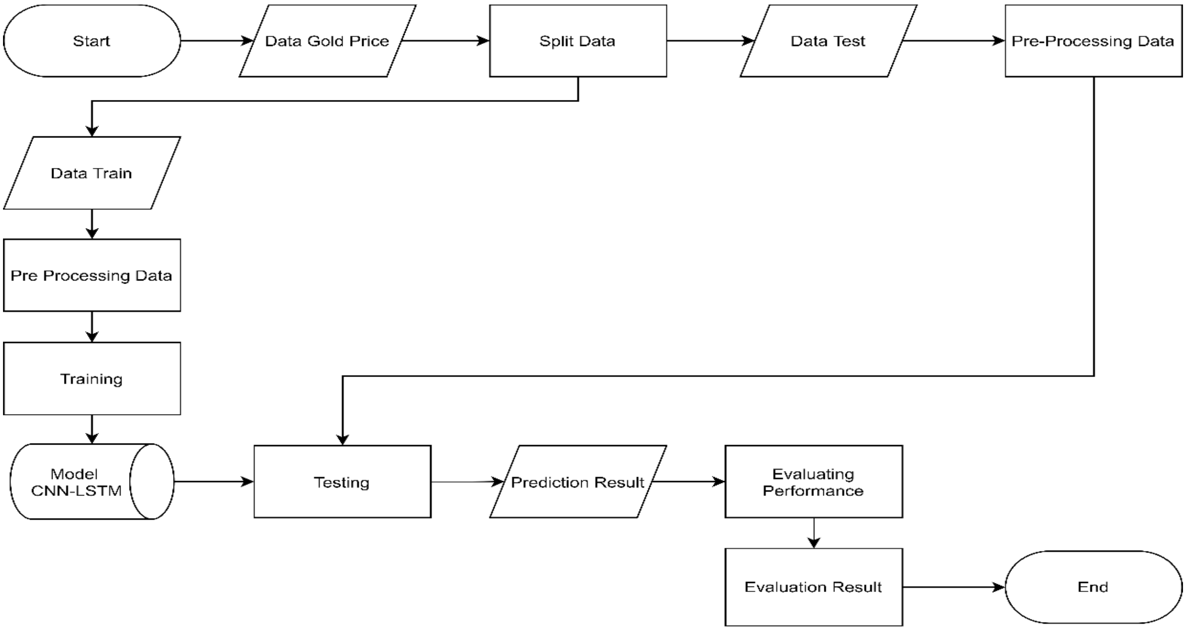

Figure 3. CNN-LSTM Gold Price Prediction System Design

3. Results and Discussion

The training process has been done to get the best hyperparameter in batch size and LSTM unit for each data ratio. In purpose to obtain the right data ratio for model CNN-LSTM shown in Figure 2, we conduct experiments in daily time frame and weekly time frame, those time frame also tested with different testing ratio which are 40% testing ratio and 20% testing ratio.

3.1 Best Hyperparameter Daily Time Frame

Table 3 shows the error performance result of CNN-LSTM model in daily time frame. The best hyperparameter in testing ratio 40% for daily time frame is 256 batch sizes, 100 LSTM units and 388 epochs with RMSE 13,67953 and MAE 9,40998, while in the testing ratio 20% the model with 512 batch sizes, 300 LSTM units and 458 epochs become the best hyperparameter with RMSE 15,53199 and MAE 10,32953. The use of 512 batch sizes has bigger result than the other batch size in testing ratio 40%, while in testing ratio 20% it has not happened. The number of epochs is decreasing when using the bigger batch size in testing ratio 40%, while in testing ratio 20% it has not happened. That also indicates a faster model CNN-LSTM when using the bigger batch size.

Table 3. Error Performance Results of CNN-LSTM Model in Daily Time Frame

Batch Size	LSTM Unit	Testing Ratio 40%			Testing Ratio 20%		
		Epoch	RMSE	MAE	Epoch	RMSE	MAE
128	100	328	14,83775	10,28374	243	15,78465	10,60598
128	200	205	14,80532	10,46320	166	18,02917	12,47424
128	300	204	14,80813	10,10875	147	16,04103	10,68298
256	100	388	13,67953	9,40998	390	15,89670	10,70721
256	200	380	15,24165	10,79648	298	16,04112	10,92605
256	300	377	14,21339	9,77594	309	17,11259	11,95636
512	100	54	18,04599	12,81577	673	16,01128	10,87941
512	200	51	17,95656	12,74110	47	20,72584	13,96864
512	300	49	18,01348	12,78120	458	15,53199	10,32953

3.2 Best Hyperparameter Weekly Time Frame

The error performance of the CNN-LSTM model in weekly time frame is shown in Table 4. The best hyperparameters in testing ratio 40% is 512 batch sizes, 300 LSTM units and 66 epochs with RMSE 37,59087 and MAE 28,01416. In testing 20%, the best hyperparameter is 128 batch sizes, 100 LSTM units and 479 epochs with RMSE 32,68774 and MAE 22,65841. Table 4 shows the result affected by LSTM unit, the bigger LSTM unit used the smaller result obtain in testing ratio 40%, while it has not happened in testing ratio 20%. The use of 512 batch size has a bigger number of epoch than the other number of batch size just in testing ratio 40%.

Table 4. Error Performance Results of CNN-LSTM Model in Weekly Time Frame

Batch Size	LSTM Unit	Testing Ratio 40%			Testing Ratio 20%		
		Epoch	RMSE	MAE	Epoch	RMSE	MAE
128	100	46	38,55176	28,71913	479	32,68774	22,65841
128	200	39	38,12133	28,43629	301	32,85492	22,75110
128	300	42	38,01949	28,32035	404	33,07555	23,24337
256	100	49	39,01046	29,03865	735	32,61283	22,74638
256	200	47	38,56566	28,72874	67	38,93458	26,86441
256	300	55	38,36413	28,61409	70	38,75133	26,75463
512	100	68	43,02718	31,60975	70	41,31657	29,10652
512	200	63	41,83229	30,85900	131	39,46210	27,59952
512	300	66	37,59087	28,01416	68	39,72639	27,76307

3.3 Model Comparison

When comparing the CNN-LSTM model in daily time frame, the performance of CNN-LSTM with testing ratio 40% is mostly better than the performance of CNN-LSTM with testing ratio 20% especially in batch size 128 and 256 no matter the number of epochs. The different results happened in weekly time frame, the performance of CNN-LSTM with testing ratio 20% mostly has a better performance than the testing ratio 40%, especially in batch size 128. The best score in daily 13.67953 for RMSE and 9,40998 for MAE, while the best score in weekly time frame 32,68774 for RMSE and 22,65841 for MAE. As a result, based on Table 3 and Table 4, the CNN-LSTM model in daily time frame has the better result than weekly time frame, this can be caused by the different amount of total data in daily time frame and weekly time frame, this reason supported by the result on Table 4 that describes the use of testing ratio 20% better than testing ratio 40%, it makes sense because the total data in weekly time frame not many as total data in daily time frame, so to predict the gold price in weekly time frame need more training ratio.

The state of the art is shown in Table 5 and Table 6. Table 5 shows the performance of our model compared to the other model for predicting gold price with different testing ratios. The model of CNN consists of 64 filters with kernel size 2 and fully connected layers 24 with batch size 256 while the model of LSTM consists of 2 LSTM layers with 100 LSTM units for each layer, the number of batch size is 128 for testing ratio 40% and 256 for testing ratio 20%. All the models we compared using early stopping function. Based on the result in Table 5, CNN-LSTM has a better performance than the other model in testing ratio 40% while in testing ratio 20% CNN-LSTM is better than CNN. This implies that CNN-LSTM has a better performance for predicting daily gold price than other prediction methods.

Table 5. Perfomance Comparison with Other Model Prediction

Model	Testing Ratio 40%		Testing Ratio 20%	
	RMSE	MAE	RMSE	MAE
CNN	21,20573	15,03605	21,94643	14,98170
LSTM	18,64879	13,40581	15,18284	10,05871
CNN-LSTM	13,67953	9,40998	15,53199	10,32953

Table 6 shows the comparison of our CNN-LSTM model with the model CNN-LSTM from previous study [4]. The model from the previous study was built without an early stopping function. The previous study model we called CNN-LSTM$_1$ and we called our model CNN-LSTM$_2$. The use of the early stopping function will give the advantage to not determine the epoch manually and stop the training process when there is no improvement in the training process, so it can lead us to get the best model. Table 6 shows our model CNN-LSTM$_1$ by applying the best hyperparameters has a better performance in predicting daily gold price with different testing ratios. In 40 % testing ratio our model has RMSE 13,67953 and 9,40998 MAE while in 20% testing ratio our model has RMSE 15,53199 and MAE 10,32953. In testing ratio 40% our model reduces the error by 25% in RMSE and 28% in MAE, while in testing ratio 20% our model reduces the error by 19% in RMSE and 20% in RMSE. This indicates that the model CNN-LSTM with tuned hyperparameter in LSTM unit and batchsize has a better performance than previous study. The use of the early stopping function also helps to avoid the overfitting that happened in the model and make the performance of CNN-LSTM better than previous study.

Table 6. Performance Comparison with Other Model CNN-LSTM

Model	Testing Ratio 40%		Testing Ratio 20%	
	RMSE	MAE	RMSE	MAE
$CNN\text{-}LSTM_1$	18,40281	13,11903	19,28486	12,94614
$CNN\text{-}LSTM_2$	13,67953	9,40998	15,53199	10,32953

4. Conclusion

Gold price prediction using the CNN-LSTM model was implemented in historical data to predict the trend of the gold price. The best hyperparameter was obtained in daily time frame, the parameters are 256 batch sizes, 100 LSTM units and 388 epochs with RMSE 13,67953 and MAE 9,40998 in testing ratio 40%, while in testing ratio 20% the parameters with 512 batch sizes, 300 LSTM units and 458 epochs become the best hyperparameter with RMSE 15,53199 and MAE 10,32953. In weekly time frame obtained parameters with 512 batch sizes, 300 LSTM units and 66 epochs with RMSE 37,59087 and MAE 28,01416 become the best hyperparameter in testing ratio 40%, while the best parameters of testing ratio 20% were the parameter with 128 batch sizes, 100 LSTM units and 479 epochs with RMSE 32,68774 and MAE 22,65841. The model CNN-LSTM in daily time frame gave a better result than the daily time frame. It can occur because the amount of train data in daily time frame is bigger than the amount of train data in weekly time frame.

The results show that CNN-LSTM that is tuned with the best hyperparameters and early stopping function has predicted the trend of gold price with improvement in reducing metrics error compared to model CNN-LSTM from previous study. In testing ratio 40% our model reduces the error by 25% in RMSE and 28% in MAE, while in testing ratio 20% our model reduces the error by 19% in RMSE and 20% in MAE. The results also show that tuning in batch size, LSTM unit and early stopping affected the model that led to improvement in metric error. Our research made our model is prospective to be used in predicting gold price, especially in daily time frame. In the future, building a more complex model or using the other tuning parameter technique could help to improve the performance of the model to predict the trend of gold price.

Acknowledgement

The researcher would like to thank the School of Computing, Telkom university that has given the opportunity and support to handle this research. This research code can be found in github.com/krisnagita/cnn-lstm-goldpriceprediction.

References

[1] E. Bouri, A. Jain, P. C. Biswal, and D. Roubaud, "Cointegration and nonlinear causality amongst gold, oil, and the Indian stock market: Evidence from implied volatility indices," *Resour. Policy*, vol. 52, no. March, pp. 201–206, 2017. https://doi.org/10.1016/j.resourpol.2017.03.003
[2] P. K. Mishra, J. R. Das, and S. K. Mishra, "Gold Price Volatility and Stock Market Returns in India," Am. J. Sci. Res. ISSN, vol. 1450, no. 9, pp. 47–55, 2010.
[3] D. C. North and R. P. Thomas, *The Rise of the Western World*. Cambridge University Press, 1973.
[4] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN–LSTM model for gold price time-series forecasting," *Neural Comput. Appl.*, vol. 32, 2020. https://doi.org/10.1007/s00521-020-04867-x
[5] N. A. Zainal and Z. Mustaffa, "A literature review on gold price predictive techniques," *2015 4th Int. Conf. Softw. Eng. Comput. Syst. ICSECS 2015 Virtuous Softw. Solut. Big Data*, pp. 39–44, 2015. https://doi.org/10.1109/ICSECS.2015.7333120
[6] X. Yang, "The Prediction of Gold Price Using ARIMA Model," *Adv. Soc. Sci. Educ. Humanit. Res.*, vol. 196, no. Ssphe 2018, pp. 273–276, 2019. https://dx.doi.org/10.2991/ssphe-18.2019.66
[7] Z. Xie, X. Lin, Y. Zhong, and Q. Chen, "Research on gold etf forecasting based on lstm," *Proc. - 2019 IEEE Intl Conf Parallel Distrib. Process. with Appl. Big Data Cloud Comput. Sustain. Comput. Commun. Soc. Comput. Networking, ISPA/BDCloud/SustainCom/SocialCom 2019*, pp. 1346–1351, 2019. https://doi.org/10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00193
[8] G. DeepikaM, G. Nambiar, M. Rajkumar, and A. Vishwavidhyapeetham, "Forecasting Price and Analysing Factors Influencing The Price of Gold Using ARIMA Model and Multiple Regression Analysis," vol. 2, no. 11, 2012.
[9] A. D. Dubey, "Gold price prediction using support vector regression and ANFIS models," *2016 Int. Conf. Comput. Commun. Informatics, ICCCI 2016*, 2016. https://doi.org/10.1109/ICCCI.2016.7479929
[10] W. Zou and Y. Xia, "Back propagation bidirectional extreme learning machine for traffic flow time series prediction," *Neural Comput. Appl.*, vol. 31, no. 11, pp. 7401–7414, 2019. https://doi.org/10.1007/s00521-018-3578-y
[11] J. Zheng, X. Fu, and G. Zhang, "Research on exchange rate forecasting based on deep belief network," *Neural Comput. Appl.*, vol. 31, pp. 573–582, 2019. https://doi.org/10.1007/s00521-017-3039-z
[12] H. H. Zahrah, S. Sa'adah, and R. Rismala, "Foreign Exchange Rate Prediction Using Long-Short Term Memory: A Case Study in COVID-19 Pandemic," vol. 6, no. 2, pp. 94–105, 2021.
[13] J. Wong, T. Manderson, M. Abrahamowicz, D. L. Buckeridge, and R. Tamblyn, "Can Hyperparameter Tuning Improve the Performance of a Super Learner?: A Case Study," *Epidemiology*, vol. 30, no. 4, pp. 521–531, 2019. https://doi.org/10.1097/EDE.0000000000001027
[14] N. Lavesson and P. Davidsson, "Quantifying the impact of learning algorithm parameter tuning," Proc. Natl. Conf. Artif. Intell., vol. 1, no. 1, pp. 395–400, 2006.

[15] K. Chen, Y. Zhou, and F. Dai, “A LSTM-based method for stock returns prediction: A case study of China stock market,” *Proc. - 2015 IEEE Int. Conf. Big Data, IEEE Big Data 2015*, pp. 2823–2824, 2015. https://doi.org/10.1109/BigData.2015.7364089
[16] D. A. Nasution, H. H. Khotimah, and N. Chamidah, “Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN,” *Comput. Eng. Sci. Syst. J.*, vol. 4, no. 1, p. 78, 2019. https://doi.org/10.24114/cess.v4i1.11458
[17] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. https://doi.org/10.1038/nature14539
[18] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” pp. 1–11, 2015.
[19] Suyanto, *Machine Learning Tingkat Dasar dan Lanjut*. Bandung: Informatika, 2018.
[20] Suyanto, *Modernisasi Machine Learning untuk Big Data*. Bandung: Informatika, 2019.
[21] Z. He, J. Zhou, H. N. Dai, and H. Wang, “Gold price forecast based on LSTM-CNN model,” *Proc. - IEEE 17th Int. Conf. Dependable, Auton. Secur. Comput. IEEE 17th Int. Conf. Pervasive Intell. Comput. IEEE 5th Int. Conf. Cloud Big Data Comput. 4th Cyber Sci.*, pp. 1046–1053, 2019. https://doi.org/10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00188
[22] L. Seymour, P. J. Brockwell, and R. A. Davis, *Introduction to Time Series and Forecasting.*, vol. 92, no. 440. 1997.
[23] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. https://doi.org/10.1162/neco.1997.9.8.1735
[24] R. Dey and F. M. Salemt, “Gate-variants of Gated Recurrent Unit (GRU) neural networks,” *Midwest Symp. Circuits Syst.*, vol. 2017-Augus, no. 2, pp. 1597–1600, 2017. https://doi.org/10.1109/MWSCAS.2017.8053243
[25] Z. Chang, Y. Zhang, and W. Chen, “Effective Adam-Optimized LSTM Neural Network for Electricity Price Forecasting,” *Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS*, vol. 2018-Novem, no. Figure 1, pp. 245–248, 2019. https://doi.org/10.1109/ICSESS.2018.8663710
[26] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–15, 2015.
[27] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature,” *Geosci. Model Dev.*, vol. 7, no. 3, pp. 1247–1250, 2014. https://doi.org/10.5194/gmd-7-1247-2014, 2014