JIKO (Jurnal Informatika dan Komputer) Accredited KEMENDIKBUD RISTEK, No.105/E/KPT/2022
Vol. 7, No. 3, December 2024, pp. 232-241 p-ISSN: 2614-8897
DOI: 10.33387/jiko.v7i3.9247 e-ISSN: 2656-1948

TRANSFORMER WITH LAGGED FEATURES FOR HANDLING LONG-TERM DATA DEPENDENCY IN TIME SERIES FORECASTING

Eko Verianto[1*], Annisa Fikria Shimbun[2]

[1]Information Systems Study Program, Faculty of Science and Visual Communication, Universitas Cendekia Mitra Indonesia
[2]Informatics Study Program, Faculty of Science and Visual Communication, Universitas Cendekia Mitra Indonesia
Email: [*1]eko@unicimi.ac.id, [2]niesashimbun@unicimi.ac.id

(Received: 11 December 2024, Revised: 19 December 2024, Accepted: 27 December 2024)

Abstract

Data with long-term dependencies plays an important role in time series forecasting. However, studying data with long-term dependencies in time series data presents challenges for most algorithms. While some algorithms can forecast time series data, not all can model data with long-term dependencies effectively. The algorithm typically used for forecasting data with long-term dependencies is Long Short-Term Memory (LSTM), but LSTM can still face vanishing gradient issues, making it difficult to identify long-term dependencies in very long datasets. Another algorithm used for forecasting long-term time series data is the transformer. However, this algorithm has not yet shown better performance compared to simple linear models. The goal of this research is to develop an effective solution for forecasting time series data with long-term dependencies. The approach proposed in this research is the transformer with *lagged features* and also using *time series cross-validation* techniques. The results of this study show the performance of the transformer model in MAPE per fold on the BBCA stock dataset with a lag=5 and fold=5 configuration as follows: 0.0390, 0.0329, 0.0207, 0.0554, 0.0423. On the USD/IDR exchange rate dataset, the results are 0.0273, 0.0431, 0.0498, 0.0236, 0.237. The results of each fold are inconsistent and show unstable performance, indicating that the transformer with *lagged features* and using *time series cross-validation* techniques has not yet been able to provide its best performance in long-term time series forecasting.

Keywords: *Time Series Forecasting; Transformer; Self-Attention; Cross-Validation; Lagged Features*

**Corresponding Author: Eko Verianto*

1. INTRODUCTION

The technological developments in recent decades have triggered a digital revolution that affects every aspect of human life. Digital development essentially relies on internet technology [1]. Internet technology has currently transformed the economies in various parts of the world. This means that the internet allows products and services to reach a wider market share. This market expansion also leads to an increase in data volume known as *big data*. One of the causes of the increased data volume is the phenomenon of data being continuously altered, with changes occurring even in short intervals.

Data has become a primary commodity for business actors and stakeholders in decision-making. Currently, data is often referred to as the oil of the 21st century [2]. Data has a significant and beneficial impact on the development and progress of businesses. One of the advantages of data is the ability to make more accurate and effective decisions, including how data is analyzed and modeled for specific purposes such as forecasting.

Forecasting is defined as estimating future information using past data [3]. Forecasting is an important aspect to be implemented in business as it can accurately predict future trends and events, and is useful in many contexts, including business management [4].

Forecasting activities cannot be separated from past data, as this data serves as the primary raw material in the modeling process. The data available today tends to be more complex and has various characteristics. One example is data with long-term trends. Such data usually has long-term dependencies

that play an important role in time series forecasting [5].

Studying data with long-term dependencies present in time series data is a challenge for most algorithms [5]. While some algorithms can perform forecasting on time series data, not all algorithms can effectively model data with long-term dependencies. This issue can affect the quality of the model, which may not capture long-term patterns in time series data. As a result, making decisions based on such models can be difficult due to their inadequate performance in predicting long-term trends and changes in the data.

The algorithm typically used for forecasting data with long-term dependencies is *Long Short-Term Memory* (LSTM). This algorithm uses a recurrent structure that is built and designed to address the issues of *vanishing gradient* and *exploding gradient* [3]. However, LSTM can still experience vanishing gradient problems, making it difficult to identify long-term dependencies in very long data sequences [6].

Previous research proposed combining the estimation methods of f*ractional differentiation parameter* (and/or *Hurst parameter*) with recurrent neural networks to study and predict long-term dependencies in information. This research evaluated four different architectures: simple RNN, LSTM, BiLSTM, and GRU. The results of this study indicate that accurate predictions can be made one step ahead of the long-term memory parameter, particularly with the BiLSTM network, which achieved the best results using the proposed methodology. However, there are still challenges, such as recurrent neural networks failing to capture points that are very far apart [5].

Previous research on the inefficiency of LSTM in handling long-term dependencies due to the vanishing gradient problem in LSTM networks. This research provides an empirical analysis using a case study on NASA's turbofan engine degradation. The research shows that the longer the input sequence, the harder it is for the LSTM model to remember all the relevant information [6].

Previous research on the effectiveness of transformers for long-term time series forecasting. This research proposes a simple single-layer linear model to compare with the transformer model. The results of this study show that the single-layer linear model demonstrates better performance compared to the transformer model [7].

Referring to the issues from previous research, the main focus of this study is on developing effective solutions and building models for forecasting time series data with long-term dependencies. Based on this, an approach is proposed to address data with long-term dependencies. The approach proposed in this study is a transformer with *lagged features* and *time series cross-validation*. The transformer, with its *self-attention* mechanism, can analyze the internal characteristics of the data effectively and focus on important information both globally and locally [8]. Based on this, the *self-attention* mechanism is suitable for long-term trend time series forecasting problems. This study also proposes the use of *lagged features* to capture temporal dependencies in the data and the use of *time series cross-validation* to maintain the temporal order of the data.

2. RESEARCH METHOD

This research method uses a quantitative approach with an experimental design. Experimental research provides researchers with the opportunity to directly influence the research variables and is the only type of research that can test hypotheses about causal relationships [9]. The method in this research is structured, systematic, planned, and clear. The method in this research is presented in the form of a diagram as shown in Figure 1 below:

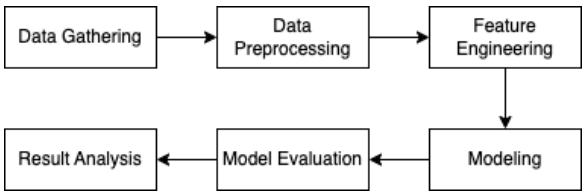

Figure 1. Research Stages

2.1 Data Gathering

The first step in this research is data gathering. Data gathering or data collection is a technique that researchers can use to collect data [10]. This research uses internet search techniques to collect data. The data used in this research consists of financial datasets, which include historical closing price data of BBCA stocks and historical closing price data of the exchange rate of the US Dollar to the Indonesian Rupiah. Each of these datasets has different trends. Figure 2 below shows the long-term trend of BBCA stock prices over a period of 10 years, which tend to increase.

Figure 2. "BBCA stock closing price chart

Unlike the long-term trend in BBCA stock prices, the closing price trend of the exchange rate between the US Dollar and the Indonesian Rupiah tends to be more volatile, although there are indications of an upward trend. Figure 3 below shows the closing price trend of the US Dollar against the Indonesian Rupiah over a period of 10 years.

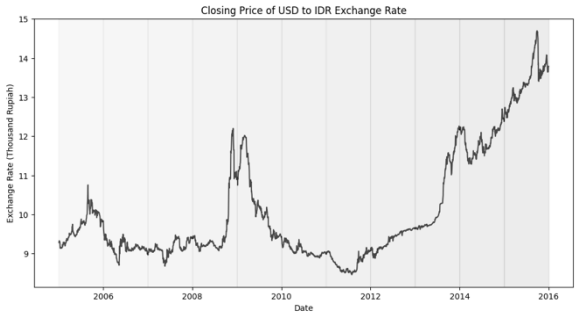

Figure 3. USD/IDR exchange rate closing price chart

2.2 Data Preprocessing

The second step in this research is data preprocessing. Data preprocessing is the initial data processing procedure used to transform raw data obtained from various sources into cleaner and more usable information for further analysis [11]. Data preprocessing in this research includes data cleaning and adjusting data according to the needs. In this stage, data is sorted according to the order of dates, duplicates are managed, and missing data is filled using the linear interpolation method. Linear interpolation is an interpolation that assumes values form a straight line [12]. The linear interpolation equation for time series data is shown in Equation 1 below.

$$y = y_0 + \frac{y_1 - y_0}{t_1 - t_0} \times (t - t_0) \quad (1)$$

The value to be determined is represented by y, y_0 is the known value at time t_0, y_1 is the known value at time t_1. t is the time of the interpolated value y, t_0 is the time of the interpolated value y_0 and t_1 is the time of the interpolated value y_1.

2.3 Feature Engineering

The third stage of this research is *a feature engineering. Feature engineering* is the process of extracting features from raw data and transforming them into a format suitable for *machine learning* models [13]. In this stage, temporal feature extraction is performed using *lagged features.* Lagged features are features that contain data from previous time steps [14]. Table 1 below is an example of the application of *lagged features* for univariate time series data on BBCA stock closing prices.

Table 1. Illustration of Lagged Features on BBCA Stock Data

Date	Closing Price	t-1	t-2	t-3
05/01/15	2640	NaN	NaN	NaN
06/01/15	2620	2640	NaN	NaN
07/01/15	2625	2620	2640	NaN
08/01/15	2595	2625	2620	2640
09/01/15	2585	2595	2625	2620
12/01/15	2560	2585	2595	2625

The illustration of *lagged features* in Table 1 uses a lag size of three, showing the stock prices from one day before (t-1), two days before (t-2), and three days before (t-3). These *lagged features* are used to help the model learn temporal patterns in the data, thereby improving prediction accuracy.

2.4 Modeling

The fourth stage of this research is *modeling.* This stage focuses on the development of a transformer model designed to address long-term dependencies in time series forecasting. In this stage, the model is designed using a transformer architecture with lagged features. The use of *lagged features* in the transformer architecture aims to enhance the model's ability to capture temporal patterns in the data. Figure 4 below shows the transformer architecture with *lagged features.*

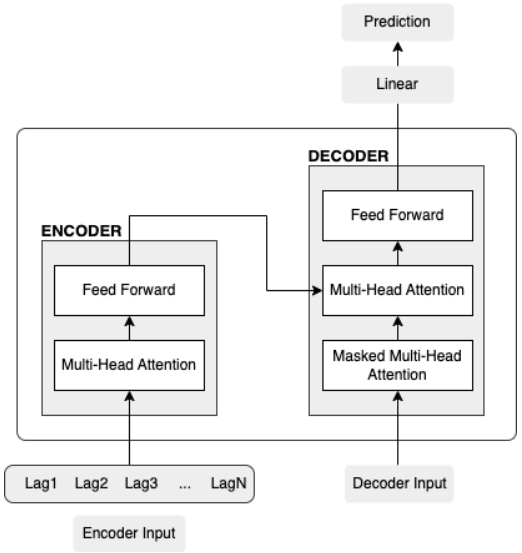

Figure 4. Transformer Architecture with *Lagged Features*

Figure 4 shows that the encoder part of the transformer architecture receives input in the form of past values from time series data expressed in lags. The encoder then uses a self-attention mechanism to capture the temporal relationships between lags in the input sequence. This is followed by a feed-forward process to strengthen feature representation. Masked multi-head attention is used to prevent future information leakage, where only previous values are used. The output from the encoder is then processed in the multi-head attention in the decoder (cross-attention). In this part, all the information from the encoder's output and the decoder is processed to capture the relationships from each piece of data. After the cross-attention stage, the next step in the decoder is the feed-forward process to learn the non-linear relationships between the data. The linear part functions to transform the decoder's output representation into the predicted value.

2.5 Model Evaluation

The fifth stage of this research is model evaluation. This research applies time series cross-validation to evaluate the performance of the forecasting model. Time series cross-validation focuses on data partitioning that maintains its temporal

order. The characteristic of time series cross-validation is that the validation samples consist of consecutive observations [15]. It works by ensuring that the training data only includes information available up to a certain point in time, while the testing data is taken from the future after the training data. Figure 5 below shows how time series cross-validation works.

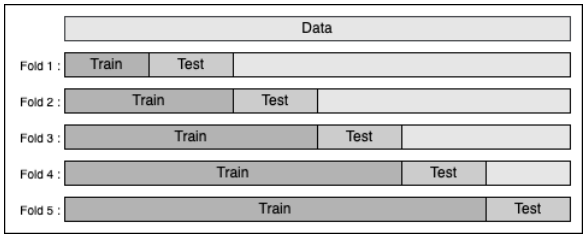

Figure 5. How time series cross-validation works

Figure 5 shows how *time series cross-validation* works. Each fold involves training data and testing data. The training data subset is gradually expanded by incorporating previous data. By applying *time series cross-validation* in this manner, each fold maintains the temporal order of the data.

At this stage, an evaluation of each model will be carried out. Model evaluation is done by measuring how well a model performs in making predictions. Model performance evaluation is carried out using *Mean Absolute Percentage Error* (MAPE). MAPE is used to measure the average prediction error proportionally to the actual value [16]. The *mean absolute percentage error* equation is shown in Equation 2 below.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{\hat{y}_i}\right| \quad (2)$$

Equation 2 consists of the summation of the absolute values obtained through the difference between the actual value y_i and the predicted value $\hat{y}$ divided by the predicted value $\hat{y}$. This value is then multiplied by one hundred percent.

2.5 Result Analysis

The final stage of this research is result analysis. This research focuses on analyzing the performance of the model in predicting long-term time series data. Observations focus on evaluation metrics to measure how well the model can predict values at each future time step. Comparing the prediction results of the transformer model with benchmark models provides deeper insights into the strengths and limitations of the model in handling long-term time series data. Visualization of prediction charts and error rate charts can also provide an understanding of the error patterns occurring in the model.

This analysis stage aims to provide a clearer picture of the capability and stability of the transformer model in predicting data movements. The results of this analysis are also used to identify potential improvements and further developments in the transformer model to handle long-term time series data.

3. RESULT AND DISCUSSION

3. 1 Dataset Processing

The dataset in this research consists of historical BBCA stock price data from January 2005 to December 2015 with daily frequency. The file format of the data used in this research is *comma-separated values* (csv). Figure 6 below shows a sample of the BBCA stock price data used in this research.

	Date	Open	High	Low	Close	Adj Close	Volume
2	3-Jan-05	295	295	292.5	295	198.73	126,180,000
3	4-Jan-05	295	302.5	295	300	202.1	291,730,000
4	5-Jan-05	300	315	300	307.5	207.15	564,730,000
5	6-Jan-05	310	310	302.5	307.5	207.15	161,150,000
6	7-Jan-05	305	305	292.5	295	198.73	592,910,000

Figure 6. Sample of BBCA stock data

In addition to the BBCA stock closing prices, this research also uses historical closing price data of the US Dollar against the Indonesian Rupiah from January 2005 to December 2015 with daily frequency. Figure 7 below shows a sample of the US Dollar to Indonesian Rupiah exchange rate data.

	Date	Close	Open	High	Low	Vol.	PercentageChange%
2	3-Jan-05	9.283,0	9.325,0	9.335,0	9.275,0		0,01%
3	4-Jan-05	9.300,0	9.300,0	9.310,0	9.265,0		0,18%
4	5-Jan-05	9.310,0	9.310,0	9.325,0	9.285,0		0,11%
5	6-Jan-05	9.327,5	9.292,5	9.327,5	9.264,5		0,19%
6	7-Jan-05	9.310,0	9.326,0	9.326,0	9.285,0		-0,19%

Figure 7. Sample of the US Dollar to Indonesian Rupiah exchange rate data

The two datasets used will then be processed to select the values to be used for modeling. The values used for modeling are the closing prices in each dataset. Figure 8 below shows the sample closing price vectors in the BBCA stock dataset and the US Dollar to Indonesian Rupiah exchange rate dataset.

```
Closing Stock Price [BBCA]:
0    295.0
1    300.0
2    307.5
3    307.5
4    295.0
Name: Close, dtype: float64
Currency Exchange Rate [USDIDR]:
0    9.2830
1    9.3000
2    9.3100
3    9.3275
4    9.3100
```

Figure 8. Sample closing price vectors for BBCA stock and the US Dollar to Indonesian Rupiah exchange rate

After feature selection on both datasets, the next step is *data preprocessing*. One of the activities in *data preprocessing* is data imputation. The technique used for data imputation is linear interpolation. Linear interpolation will fill in missing values by estimating

the value between two known data points, the calculation process of linear interpolation as shown in the equation 1.

The next step after *data preprocessing* is the *feature engineering* stage. In this stage, the data will be adjusted for modeling needs, such as determining *lag* and target as shown in Figure 9 below.

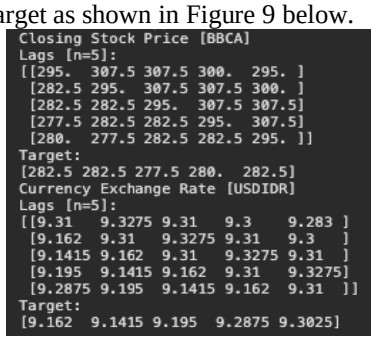

Figure 9. Sample lags and targets for BBCA stock closing prices and the US Dollar to Indonesian Rupiah exchange rate

Figure 9 shows the historical closing price data of BBCA stock and the historical closing price data of the US Dollar to Indonesian Rupiah exchange rate, already in the form of lag and target.

3. 2 Modeling and Evaluation

Model formation is carried out through the data training process. The data used in the training process is the training data with hyperparameter variations to find the best model. The first experiment conducted in this research is the experiment on determining the number of lags and folds. This experiment uses variations in the number of folds in time series cross-validation and variations in lags in lagged features.

Table 2 below shows the results of the lag experiment on the BBCA dataset and the US Dollar to Indonesian Rupiah exchange rate dataset. This experiment aims to observe the impact of the variation in the number of lags on the error rate and the execution time produced.

Table 2. Lag Experiment on Transformer Model

Dataset	Number of Folds	Number of Lags	MAPE	Execution Time (s)
BBCA Stock	5	5	0,0381	3312
		10	0,0400	3427
		15	0,0417	3815
		20	0,0574	3847
Currency Exchange Rate (USD/IDR)	5	5	0,0335	3506
		10	0,0536	3677
		15	0,0500	3836
		20	0,0546	3822

The lag experiment in Table 2 was evaluated using MAPE. The MAPE results show that increasing the lag does not always affect the error rate. The results also show that predominantly using lag=5 results in the minimum error rate. This indicates that the use of short-term information has good relevance in model formation. This experiment also shows that the number of lags affects the model's execution time. The larger the number of lags, the higher the potential for increased execution time.

This research also conducted experiments on the variation of the number of folds in both datasets. The aim of this experiment is to observe the impact of the variation in the number of folds on the error rate and the execution time produced. Table 3 below shows the results of the fold number experiment measured using MAPE.

Table 3. Fold Experiment on Transformer Model

Dataset	Number of Lag	Number of Folds	MAPE	Execution Time (s)
BBCA Stock	5	5	0,0381	3312
		10	0,0471	6781
		15	0,0518	10521
		20	0,0402	13581
Nilai Tukar Mata Uang (USD/IDR)	5	5	0,0335	3506
		10	0,0321	7019
		15	0,0285	10598
		20	0,0364	13202

Table 3 shows that increasing the number of folds does not always result in better error rates, indicating that the number of folds does not significantly affect the model's generalization ability, while execution time always increases with the number of folds.

To observe the influence of variations in the combination of lag and fold on the error rate and execution time, variations of both were made as shown in Table 4 below.

Table 4. Experiments on Fold and Lag in the Transformer Model

Dataset	Number of Folds	Number of Lags	Avg. MAPE Across all Folds	Execution Time (s)
Saham BBCA	5	5	0,0381	3312
		10	0,0400	3427
		15	0,0417	3815
		20	0,0574	3847
	10	5	0,0411	6599
		10	0,0537	6943
		15	0,0484	7512
		20	0,0544	7897
	15	5	0,0518	10521
		10	0,0506	10636
		15	0,0416	10960
		20	0,0525	10985
	20	5	0,0402	13581
		10	0,0520	14483
		15	0,0559	15418
		20	0,0536	14679
Currency Exchange Rate (USD/IDR)	5	5	0,0335	3506
		10	0,0536	3677
		15	0,0500	3836
		20	0,0546	3822
	10	5	0,0338	6798
		10	0,0540	7398
		15	0,0556	7898
		20	0,0512	7898
	15	5	0,0285	10598
		10	0,0610	14583
		15	0,0568	15463
		20	0,0506	11380
	20	5	0,0364	13202
		10	0,0587	14716
		15	0,0530	17317
		20	0,0600	14959

Table 4 shows the fold and lag experiments on the BBCA stock dataset and the US Dollar to Indonesian Rupiah exchange rate dataset. In the BBCA stock dataset, the impact of fold and lag on the error rate varies, and increasing the fold and lag values does not always affect the error rate. Smaller fold and lag configurations can provide better results.

In the dataset of the US Dollar to Indonesian Rupiah exchange rate, the results vary. Increasing the fold and lag does not always affect the error rate; however, an increase in the number of folds and lags will increase the execution time. The results also show that the number of folds and lags can produce a better error rate. The results of the experiment in Table 3 also provide information that larger variations in folds and lags will result in longer execution times. This information is needed because, besides the error rate, execution time will also be considered in the selection of the best model.

The next experiment is to compare the performance of the transformer model and the LSTM model per fold. This test uses both datasets with all fold variations from the previous tests, while for the lag variation, only the lag value that results in the minimum error rate in Table 4 is used. Table 5 below shows the performance comparison of the transformer model and the LSTM model on the BBCA stock dataset.

Table 5. Comparison of the Performance of Transformer and LSTM Models on the BBCA Stock Dataset with fold=5 and lag-5

Fold	MAPE per Fold (Transformer Model)	MAPE per Fold (LSTM Model)
Fold 1	0,0390	0,0190
Fold 2	0,0329	0,0226
Fold 3	0,0207	0,0148
Fold 4	0,0554	0,0148
Fold 5	0,0423	0,0111

Table 5 shows the comparison of the performance of the transformer model and the LSTM model with a fold=5 and lag=5 configuration. Figure 10 below shows the comparison of MAPE per fold of the transformer model and the LSTM model in graphical form.

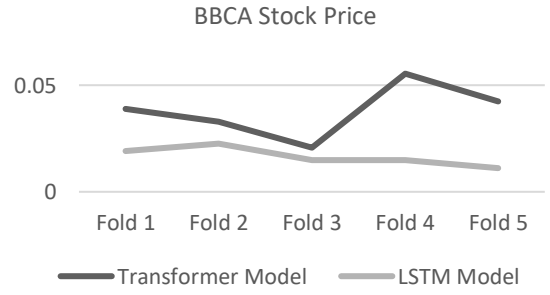

Figure 10. "Comparison graph of MAPE on BBCA stock with fold=5 and lag=5

Based on the observations in the graph in Figure 10, it shows that all folds in the LSTM model have a smaller error rate compared to the transformer model. The transformer model shows a fluctuating pattern, while the LSTM model shows a tendency to decrease the error rate.

The next test was conducted using the dataset of the US Dollar to Indonesian Rupiah exchange rate. The results of this test are shown in Table 6 below.

Table 6. Comparison of the Performance of Transformer and LSTM Models on the USD/IDR Exchange Rate Dataset with fold=5 and lag=5

Fold	MAPE per Fold (Transformer Model)	MAPE per Fold (LSTM Model)
Fold 1	0,0273	0,0043
Fold 2	0,0431	0,0065
Fold 3	0,0498	0,0025
Fold 4	0,0236	0,0035
Fold 5	0,0237	0,0073

Table 6 shows the comparison of the performance of the transformer model and the LSTM model with a fold=5 and lag=5 configuration. Figure 11 below shows the comparison of MAPE per fold on the transformer model and the LSTM model in graphical form.

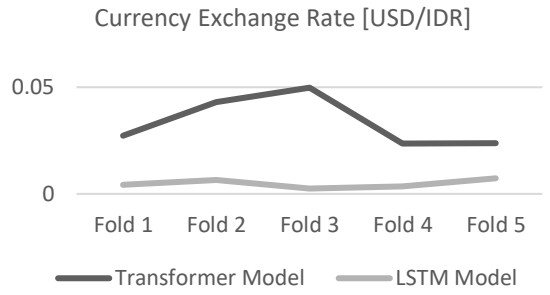

Figure 11. Comparison graph of MAPE on the USD/IDR exchange rate dataset with fold=5 and lag=5

Based on the observations through the graph shown in Figure 11, the error rate on both models shows a fluctuating pattern; however, the LSTM model has a less significant difference in error rate compared to the transformer model. The next test will be conducted by increasing the number of folds as shown in Table 7 below.

Table 7. Comparison of the Performance of Transformer and LSTM Models on the BBCA Stock Dataset with fold=10 and lag=5

Fold	MAPE per Fold (Transformer Model)	MAPE per Fold (LSTM Model)
Fold 1	0,1023	0,0214
Fold 2	0,0225	0,0183
Fold 3	0,0697	0,0299
Fold 4	0,0593	0,0204
Fold 5	0,0284	0,0210
Fold 6	0,0299	0,0182
Fold 7	0,0137	0,0114
Fold 8	0,0261	0,0163
Fold 9	0,0366	0,0144
Fold 10	0,0229	0,0126

Table 7 shows the comparison of the performance of the transformer model and the LSTM model with a fold=10 and lag=5 configuration. Figure

12 below shows the comparison of MAPE per fold on the transformer model and the LSTM model in graphical form.

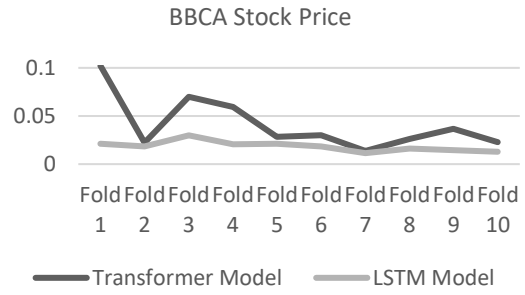

Figure 12. Comparison graph of MAPE on BBCA stock with fold 10 and lag=5

Based on the observations made on the graph shown in Figure 12, both the transformer model and the LSTM model show a tendency to decrease the error rate. The next test is conducted on the dataset of the US Dollar to Indonesian Rupiah exchange rate as shown in Table 8 below.

Table 8. Comparison of the Performance of Transformer and LSTM Models on the USD/IDR Exchange Rate Dataset with fold=10 and lag=5

Fold	MAPE per Fold (Transformer Model)	MAPE per Fold (LSTM Model)
Fold 1	0,0232	0,0115
Fold 2	0,0111	0,0038
Fold 3	0,0468	0,0054
Fold 4	0,0266	0,0064
Fold 5	0,0529	0,0026
Fold 6	0,0684	0,0029
Fold 7	0,0267	0,0032
Fold 8	0,0570	0,0032
Fold 9	0,0178	0,0036
Fold 10	0,0075	0,0081

Table 8 shows the comparison of the performance of the transformer model and the LSTM model with a fold=10 and lag=5 configuration. The graph in Figure 13 below shows the comparison of MAPE per fold on the transformer model and the LSTM model.

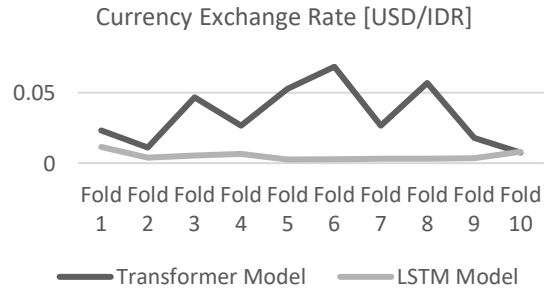

Figure 13. Comparison graph of MAPE on the USD/IDR exchange rate dataset with fold =10 and lag=5

Based on the graph shown in Figure 13, both the transformer model and the LSTM model show a fluctuating pattern in the error rate. Although both show a fluctuating pattern, the LSTM model does not show significant changes compared to the transformer model. In the next test, the number of folds will be increased again, as shown in Table 9 below.

Table 9. Comparison of the Performance of Transformer and LSTM Models on the BBCA Stock Dataset with fold=15 and lag=15

Fold	MAPE per Fold (Transformer Model)	MAPE per Fold (LSTM Model)
Fold 1	0,0535	0,0263
Fold 2	0,0882	0,0260
Fold 3	0,0297	0,0274
Fold 4	0,0496	0,0336
Fold 5	0,0715	0,0525
Fold 6	0,0459	0,0281
Fold 7	0,0432	0,0188
Fold 8	0,0321	0,0165
Fold 9	0,0376	0,0171
Fold 10	0,0429	0,0293
Fold 11	0,0189	0,0258
Fold 12	0,0379	0,0168
Fold 13	0,0300	0,0158
Fold 14	0,0180	0,0127
Fold 15	0,0255	0,0140

Table 9 shows the comparison of the performance of the transformer model and the LSTM model with a fold=15 and lag=15 configuration. The performance comparison graph of the transformer model and the LSTM model is shown in Figure 14 below.

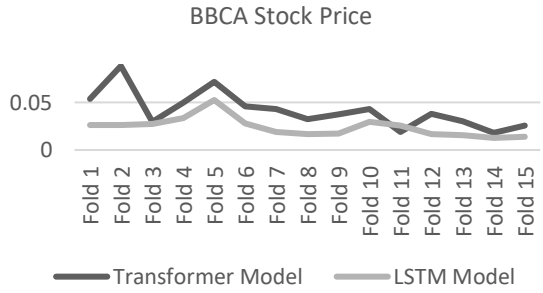

Figure 14. Comparison graph of MAPE on the BBCA stock dataset with fold=15 dan lag=15

The graph in Figure 14 shows that the error rate in both models tends to decrease. The next test is conducted on the dataset of the US Dollar to Indonesian Rupiah exchange rate as shown in Table 10 below.

Table 10. Comparison of the Performance of Transformer and LSTM Models on the USD/IDR Exchange Rate Dataset with fold=15 and lag=5

Fold	MAPE per Fold (Transformer Model)	MAPE per Fold (LSTM Model)
Fold 1	0,0165	0,0062
Fold 2	0,0069	0,0086
Fold 3	0,0073	0,0042
Fold 4	0,0070	0,0040
Fold 5	0,0542	0,0080
Fold 6	0,0185	0,0076
Fold 7	0,0462	0,0033
Fold 8	0,0349	0,0014
Fold 9	0,0532	0,0040

Fold 10	0,0396	0,0050
Fold 11	0,0569	0,0011
Fold 12	0,0208	0,0037
Fold 13	0,0157	0,0040
Fold 14	0,0328	0,0045
Fold 15	0,0176	0,0052

Table 10 shows the comparison of the performance of the transformer model and the LSTM model with a fold=15 and lag=15 configuration. The performance comparison graph of the transformer model and the LSTM model is shown in Figure 15 below.

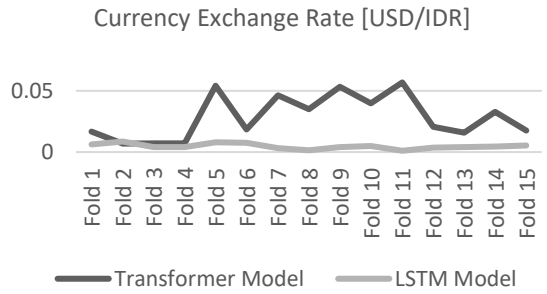

Figure 15. Comparison graph of MAPE on the USD/IDR exchange rate dataset with fold=15 and lag=5

Based on the graph shown in Figure 15, both models show a fluctuating pattern; however, the LSTM model does not show significant changes in error rate, which is in contrast to the transformer model. The final test in this study will increase the number of folds again, as shown in Table 11 below.

Table 11. Comparison of the Performance of Transformer and LSTM Models on the BBCA Stock Dataset with fold=20 and lag=5

Fold	MAPE per Fold (Transformer Model)	MAPE per Fold (LSTM Model)
Fold 1	0,0198	0,0154
Fold 2	0,1276	0,0184
Fold 3	0,0299	0,0131
Fold 4	0,0185	0,0154
Fold 5	0,0345	0,0268
Fold 6	0,0402	0,0266
Fold 7	0,0536	0,0346
Fold 8	0,0388	0,0200
Fold 9	0,0503	0,0178
Fold 10	0,0419	0,0171
Fold 11	0,0272	0,0217
Fold 12	0,0303	0,0190
Fold 13	0,0151	0,0105
Fold 14	0,0253	0,0110
Fold 15	0,0597	0,0171
Fold 16	0,0276	0,0192
Fold 17	0,0233	0,0121
Fold 18	0,0398	0,0111
Fold 19	0,0584	0,0128
Fold 20	0,0429	0,0204

Table 11 shows the comparison of the performance of the transformer model and the LSTM model with a fold=20 and lag=5 configuration. The graphical presentation of this table is shown in Figure 16 below.

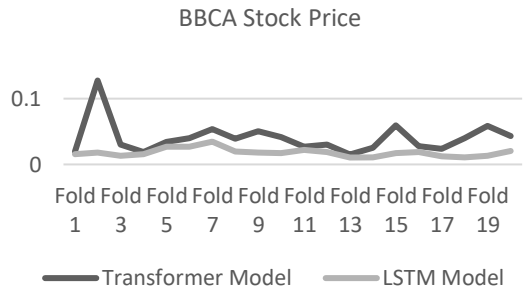

Figure 16. Comparison graph of MAPE on the BBCA stock dataset with fold=20 and lag=5

Figure 16 shows that the error rate in both models has a fluctuating pattern, but the LSTM model has a less significant difference in error rate compared to the transformer model. The next test uses the dataset of the US Dollar to Indonesian Rupiah exchange rate as shown in Table 12 below.

Table 12. Comparison of the Performance of Transformer and LSTM Models on the USD/IDR Exchange Rate Dataset with fold=20 and lag=5

Fold	MAPE per Fold (Transformer Model)	MAPE per Fold (LSTM Model)
Fold 1	0,0203	0,0083
Fold 2	0,0200	0,0074
Fold 3	0,0482	0,0051
Fold 4	0,0332	0,0111
Fold 5	0,0176	0,0045
Fold 6	0,0149	0,0023
Fold 7	0,0681	0,0101
Fold 8	0,0144	0,0081
Fold 9	0,0275	0,0054
Fold 10	0,0274	0,0025
Fold 11	0,0443	0,0016
Fold 12	0,0728	0,0064
Fold 13	0,0651	0,0067
Fold 14	0,0436	0,0035
Fold 15	0,0484	0,0029
Fold 16	0,0171	0,0049
Fold 17	0,0404	0,0107
Fold 18	0,0246	0,0064
Fold 19	0,0588	0,0055
Fold 20	0,0208	0,0042

Table 12 shows the comparison of the performance of the transformer model and the LSTM model. The configuration of both models also uses fold=20 and lag=5. The graphical presentation of this table is shown in Figure 17 below.

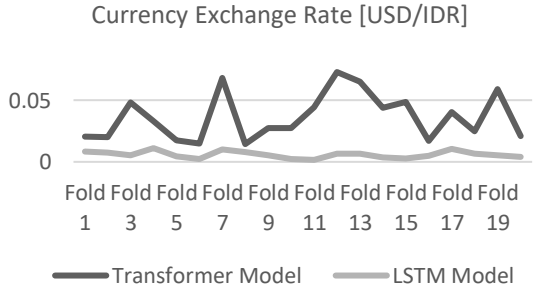

Figure 17. Comparison graph of MAPE on the USD/IDR exchange rate dataset with fold=20 and lag=5

Based on the experiments conducted on the best models from all fold variations, the transformer model does not show good performance compared to the LSTM model in all experiments. The LSTM model has more consistent performance with relatively stable error rates in almost all folds, while the transformer model shows a fluctuating pattern with significant changes. This indicates that the transformer model is less effective in capturing long-term temporal patterns.

This is because the basic architecture of the transformer relies more on self-attention. The self-attention mechanism tends to be less effective in handling data with long-term temporal dependencies and lacks components that can be explicitly used for sequential processing.

The following will show the prediction results of the best model per fold on the BBCA stock dataset and the US Dollar to Indonesian Rupiah exchange rate dataset with fold=5 and lag=5 configuration. The prediction results are in the form of time series cross-validation, which is the model validation technique used in this study. Additionally, the error rate graph per fold for each model will also be shown. Figure 18 below shows the visualization of the test results on the transformer model on the BBCA dataset.

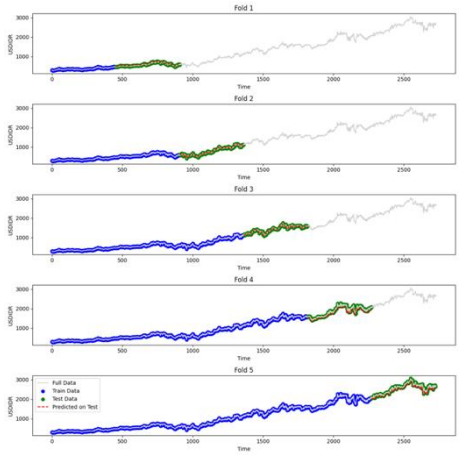

Figure 18. "Prediction visualization on the BBCA stock dataset

Figure 18 is a visualization of the prediction results with a fold=5 and lag=5 configuration, while the visualization of MAPE per fold is shown in Figure 19.

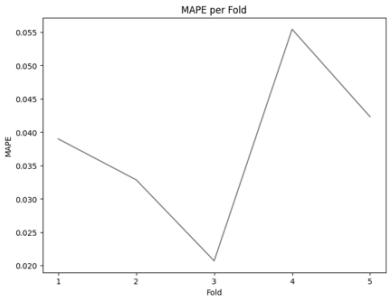

Figure 19. MAPE visualization on the BBCA stock dataset

The following is the visualization of the test results on the US Dollar to Indonesian Rupiah exchange rate dataset, as shown in Figure 20 below.

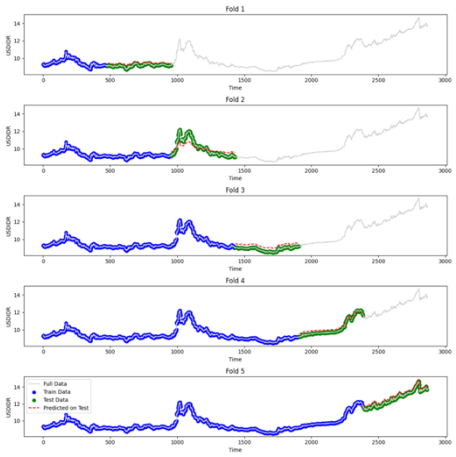

Figure 20. Prediction visualization on the currency exchange rate dataset USD/IDR

Figure 20 shows the visualization of the prediction results with a fold=5 and lag=5 configuration, while the visualization of MAPE per fold is shown in Figure 21 below.

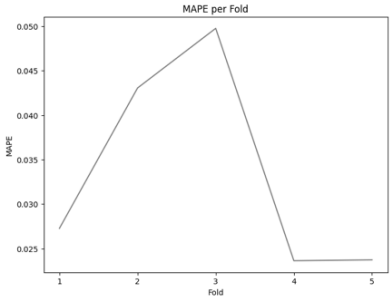

Figure 21. MAPE visualization on the currency exchange rate dataset USD/IDR

4. CONCLUSION

The results of this study show that the transformer model with *lagged features* and *time series cross-validation* has not yet been able to provide its best performance in predicting long-term time series data. In all experiments and tests, the transformer model has a relatively higher error rate compared to the LSTM model. The transformer model also shows a fluctuating pattern in each fold, with no tendency for error rates to decrease from the initial fold to the final fold. This means that the transformer model is not yet optimal in capturing long-term temporal patterns in the data. This is because the self-attention mechanism in transformers is not specifically designed to handle data with long-term temporal dependencies and also the size of the data used in this study is still limited. Developing effective solutions requires a mechanism specifically developed to handle data with long-term temporal dependencies. Additionally, the architecture of transformers needs to be engineered and simplified to be specifically designed for long-term time series forecasting by reducing computational complexity and increasing the size of the sequence data further for the validation process. Therefore, it can be concluded that the transformer model with *lagged features* and time series cross-validation cannot be considered an

effective solution for handling long-term data dependency forecasting.

Acknowledgment

We would like to express our deepest gratitude to the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia for providing funding support for this research through the Penelitian Dosen Pemula Program. Our thanks also go to the research team for their dedication and cooperation. We are also grateful to the entire Universitas Cendekia Mitra Indonesia for providing support and enabling this research to be completed.

5. REFERENCE

[1] A. K. Sharma and D. P. S. Shanmugaboopathi, "Digital Revolution and Its Nature and Extent in the Contemporary World," *Technoarete Trans. Adv. Soc. Sci. Humanit.*, vol. 2, no. 4, pp. 27–32, 2022, doi: 10.36647/ttassh/02.04.a005.

[2] C. Stach, "Data Is the New Oil–Sort of: A View on Why This Comparison Is Misleading and Its Implications for Modern Data Administration," *Futur. Internet*, vol. 15, no. 2, pp. 1–49, 2023, doi: 10.3390/fi15020071.

[3] A. W. Saputra, A. P. Wibawa, U. Pujianto, A. B. P. Utama, and A. Nafalski, "LSTM-based Multivariate Time-Series Analysis: A Case of Journal Visitors Forecasting," *Ilk. J. Ilm.*, vol. 14, no. 1, pp. 57–62, 2022, doi: 10.33096/ilkom.v14i1.1106.57-62.

[4] D. P. Khairunnisa, N. Halwatunnissa, and D. H. Nurdiansyah, "Analysis Forecasting of Operational Expense of PT. Bank Rakyat Indonesia (Persero) Tbk," *Eqien - J. Ekon. dan Bisnis*, vol. 9, no. 2, pp. 480–487, 2022.

[5] C. Ubal, G. Di-Giorgi, J. E. Contreras-Reyes, and R. Salas, "Predicting the Long-Term Dependencies in Time Series Using Recurrent Artificial Neural Networks," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1340–1358, 2023, doi: 10.3390/make5040068.

[6] S. M. Al-Selwi, M. F. Hassan, S. J. Abdulkadir, and A. Muneer, "LSTM Inefficiency in Long-Term Dependencies Regression Problems," *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 30, no. 3, pp. 16–31, 2023, doi: 10.37934/araset.30.3.1631.

[7] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are Transformers Effective for Time Series Forecasting?," *Proc. 37th AAAI Conf. Artif. Intell. AAAI 2023*, vol. 37, pp. 11121–11128, 2023, doi: 10.1609/aaai.v37i9.26317.

[8] X. Ma, Z. Liu, M. Zheng, and Y. Wang, "Application and exploration of self-attention mechanism in dynamic process monitoring," *IFAC-PapersOnLine*, vol. 55, no. 6, pp. 139–144, 2022, doi: 10.1016/j.ifacol.2022.07.119.

[9] R. Akbar, R. A. Siroj, M. Win Afgani, and Weriana, "Experimental Reseacrch Dalam Metodologi Pendidikan," *J. Ilm. Wahana Pendidik.*, vol. 9, no. 2, pp. 465–474, 2023, [Online]. Available: https://jurnal.peneliti.net/index.php/JIWP/article/view/3165

[10] S. Sukmawati, S. Sudarmin, and S. Salmia, "Development of Quality Instruments and Data Collection Techniques," *J. Pendidik. dan Pengajaran Guru Sekol. Dasar*, vol. 6, no. 1, pp. 119–124, 2023, doi: 10.55215/jppguseda.v6i1.7527.

[11] R. R. Salam, M. F. Jamil, Y. Ibrahim, R. Rahmaddeni, S. Soni, and H. Herianto, "Analisis Sentimen Terhadap Bantuan Langsung Tunai (BLT) Bahan Bakar Minyak (BBM) Menggunakan Support Vector Machine," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 1, pp. 27–35, 2023, doi: 10.57152/malcom.v3i1.590.

[12] M. R. Ismail, A. Zain, J. Jamaludin, F. Dewantoro, and D. Pratiwi, "Perhitungan Data Curah Hujan yang Hilang dengan Menggunakan Metode Interpolasi Linier," *J. Tek. Sipil Sendi*, vol. 4, no. 2, pp. 60–66, 2023.

[13] M. R. Suherlan, A. Asriyanik, and A. Pambudi, "UMMIBOT sebagai Media Layanan Informasi Penerimaan Mahasiswa Baru Universitas Muhammadiyah Sukabumi," *J. Inform. Terpadu*, vol. 9, no. 2, pp. 82–91, 2023.

[14] E. Sokolova, O. Ivarsson, A. Lillieström, N. K. Speicher, H. Rydberg, and M. Bondelind, "Data-driven models for predicting microbial water quality in the drinking water source using E. coli monitoring and hydrometeorological data," *Sci. Total Environ.*, vol. 802, 2022, doi: 10.1016/j.scitotenv.2021.149798.

[15] N. H. Setiawan and Z. Zulkarnain, "Forecasting Palm Oil Production using Long Short-term Memory (LSTM) with Time Series Cross Validation (TSCV)," *Int. J. Soc. Serv. Res.*, vol. 04, no. 05, pp. 1237–1251, 2024.

[16] W. Gu *et al.*, "Predicting Stock Prices with FinBERT-LSTM: Integrating News Sentiment Analysis," in *Proceedings of the 2024 8th International Conference on Cloud and Big Data Computing*, 2024, pp. 67–72. doi: 10.1145/3694860.3694870.