
Journal of Informatics and Web Engineering

Vol. 5 No. 3 (October 2026)

eISSN: 2821-370X

Comparative Analysis of Time Series Forecasting Models for Stock Price Prediction: ARIMA, LSTM, and Prophet

Muhammad Syahid Pebriadi^{1*}, Tino Kemal Fattah², Putriana Salman³, Rizky Amelia⁴

¹Accounting Information Systems, Politeknik Negeri Banjarmasin, Banjarmasin, South Kalimantan, Indonesia

^{2,3,4}Accounting, Politeknik Negeri Banjarmasin, Banjarmasin, South Kalimantan, Indonesia

*corresponding author: (m.syahid@poliban.ac.id; ORCID: 0000-0001-7403-2142)

Abstract - Stock price forecasting with a uniform approach is a challenging process due to nonlinearity, regime changes, and seasonality. The following paper would like to address the practical problem of selecting an appropriate model through comparing three widely used models such as Auto Regressive Integrated Moving Average (ARIMA), Long Short-Term Memory (LSTM), and Prophet based on how they are applied to different attributes of data and forecast-related tasks. The structured direct comparison will be our proposal, as to be able to identify the strengths and shortcomings in abiding by each model. Each of the model is methodologically trained and tested on the same data stock price series and parameterized by the time of standard errors: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and the Mean Absolute Percent Error (MAPE). Experimental design is founded on the canonical setups and reasonable hyperparameters to an algorithm and the results are measured based on the hold-out samples, which have volatile intervals. The results show that under the conditions of intricate dynamics and large volatility, LSTM delivers the greatest general results with MAE = 277.50, MSE = 165,244.61, RMSE = 406.50 and MAPE = 3.09%. ARIMA performs well on stable, approximately linear and stationary segments but struggles with nonlinearity and long-term dependencies. Prophet is effective when strong seasonal patterns are present and is robust to missing values and outliers. We conclude that ARIMA suits stable series, LSTM is preferable for complex and rapidly changing markets, and Prophet is advantageous under pronounced seasonality. These findings offer practical guidance for model selection aligned with data properties and forecasting goals.

Keywords— Stock Price Prediction, ARIMA, LSTM, Prophet, Time Series Forecasting

Received: 4 August 2025; Accepted: 12 November 2025; Published: 16 October 2026

This is an open access article under the [CC BY-NC-ND 4.0](#) license.



1. INTRODUCTION

One of the biggest problems in finance is predicting stock prices. This is because prices change a lot and are affected by many outside and inside factors that are hard to predict. Many different ways have been created to try to predict stock prices, from technical and fundamental analysis to methods that use artificial intelligence. Time series

forecasting is one of the most common ways to model and predict stock prices. It uses past data to guess how prices will move in the future [1]-[3].

There are a number of algorithms that can be used to predict stock prices, such as Auto Regressive Integrated Moving Average (ARIMA), Long Short-Term Memory (LSTM), and Prophet. ARIMA is a classic method for predicting time series data. It's often used to predict stock prices because it's easy to use and can handle linear data [4]. But ARIMA can't handle nonlinear data or the market's complicated dynamics very well [5]-[7]. ARIMA uses patterns and trends in stationarized historical time-series data to model stock prices. It then uses the estimated model to predict future prices. This method lets investors and analysts use structured and systematic time-series data [8]-[10] to predict short-term changes in the stock market.

LSTM, a kind of neural network used in deep learning, has shown that it can predict time series data very well by finding long-term and nonlinear patterns in the data [11]-[13]. LSTM is particularly suitable for use with data that has long-term dependencies, such as stock prices, which are often influenced by external factors requiring an understanding of context over long periods [14]. LSTM networks process stock-market data as time-ordered sequences, using memory cells and three gates (input, forget, and output) that learn to retain or discard information, enabling the model to capture nonlinear patterns and long-term dependencies while mitigating the vanishing-gradient problem [11], [15], [16].

Prophet, developed by Facebook, is a forecasting method designed to handle time series with clear seasonality and nonlinear trends [17]. This method offers flexibility in handling data with seasonal patterns and special events that affect stock prices. Prophet models stock series as the sum of a trend (piecewise linear/logistic with automatic changepoints), seasonality (weekly/annual/daily via Fourier terms), event effects (earnings, ex-dividend, rate decisions, holidays), and noise [18], [19].

ARIMA was used on stationary, linear and one season series because it is easy to understand whereas uncertainty was calibrated [16]. Prophet has been chosen when there are over one seasonal pattern, structural breaks or known events to us because its trend-seasonality-holiday structure can be decomposed [20]. The use of LSTM models happened when the dynamics of the residual were nonlinear or temporal characteristics were correlated at longer distances [21]. This step-by-step process, which had undergone the classical models up to the deep-learning models, guaranteed that the two approach fulfilled the statistical properties of the data, in addition, to addressing an objective with regards to clear inference, and long-term forecasting.

Nevertheless, regardless of the pros and cons of both algorithms, there is too much evidence lacking direct discussions of how they are effective or not in predicting stock prices. In an endeavour to provide a more detailed insight on the effectiveness of each algorithm in the volatile stock market environment, this paper will seek to compare the ARIMA, LSTM and Prophet algorithms in terms of their accuracy and predictive ability in stock pricing.

2. LITERATURE REVIEW

During the last years, comparisons between various time series forecasting models, such as ARIMA, LSTM, and the Prophet, have attracted the attention of people in a wide range of areas, including finance. Each of the models has advantages and disadvantages that can develop them as more or less productive by the data and by the case.

The most popular and the best are the ARIMA methods used to predict the time series. The results are easier to comprehend. The model can use its Autoregressive (AR) and moving-average terms to determine the trend and seasonal variations in the data. There are several studies where ARIMA outperformed more complicated models [9], [10], [20], [21]. For example, Ahammad et al. [22] showed that ARIMA produced the lowest RMSE and MAE values for stock market forecasting compared to LSTM and in emergency health department indicator forecasting, ARIMA was a strong competitor to the Prophet [23].

LSTM networks possess the ability to discern intricate patterns and nonlinear relations within the data. It is most preferred when the data is in large volumes and shows a long-term dependency. For example, Masum et al. [24] showed that an LSTM model outperformed an ARIMA model for multi-step electricity load forecasting but may need more training time and computational requirements [25]. Maulenkul et al. [26] demonstrated that LSTM produced smoother and more stable long-term forecasts than ARIMA and Prophet, whereas ARIMA and Prophet exhibited greater sensitivity to short-term fluctuations in time-series data.

Prophet is made to work with time series data exhibiting pronounced seasonal changes. It is easy to use, and it can consider the holidays or events that affect the forecast. Menculini et al. [27] found that although Prophet is fast and easy to implement, it is less accurate than ARIMA and LSTM models, but other studies indicated the Prophet model surpassed the ARIMA and LSTM models for long-term electricity load forecasting with the lowest error [18], [28]. Fattouh et al. [29] demonstrated that the Prophet forecasting model achieved high predictive accuracy for hospital bed occupancy, with MAPE values ranging from 3.21% to 3.53% across forecasting horizons of 30–180 days, while incorporating holiday effects and COVID-19 indicators as external factors.

The comparative performance of these models depends on the characteristics of the data. Chen et al. showed that although the LSTM model produced the most accurate results when predicting dengue cases with climate data, the Prophet model was best suited for long-term forecasting [19]. That means the model selection may depend on the forecasting task and data characteristics. While ARIMA is still a golden standard for time series forecasting, the LSTM model is suitable for capturing the complex patterns and dependencies, and the Prophet model is easy to use when dealing with seasonal data. Therefore, model selection depends on the forecasting tasks' characteristics, computational resources, and the required accuracy.

3. RESEARCH METHODOLOGY

3.1 Data Collection

The data used is the Indonesian Stock Price All Time dataset [30]. This dataset consists of historical trading data for 926 listed companies on the Indonesia Stock Exchange (IDX) as of May 25, 2024. Each file in the dataset denotes a singular issuer, with information obtained from Yahoo Finance.

3.2 Data Preprocessing

This stage is performed to check for any missing data (missing values) or Null values. Missing data would be replaced with the mean or median values. If this data is left unchanged, it could interfere with the modeling process. After the preprocessing stage, the dataset is divided into training and testing sets using an 80:20 ratio. The first 80% of the observations are used as the training data to develop and fit the model, while the remaining 20% are used as the testing data to evaluate the model's forecasting performance on unseen data. Since the dataset consists of time-series stock price data, the data are divided chronologically without random shuffling in order to preserve the temporal sequence and prevent information leakage from future observations into the training process.

3.3 Model Creation

3.3.1 ARIMA Model

The ARIMA model has three parameters that can be seen in the decomposition graph and the plotting of the ACF and PACF graphs, which are **p** (order of autoregression): This is used to determine the necessary lag variables; **d** (order of integration): This is used to determine the number of differences; and **q** (order of moving average): This is used to determine the window size for the Moving Average (MA). The ARIMA model is divided into 3 groups: the **AR** model in (1), **Integration (I)** in (2), **MA** in (3), and the combined **ARIMA** model (see Equations (1) to (3)).

- AR

$$X_t = \mu' + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + e_t \quad (1)$$

Where:

μ' = constant

ϕ_p = AR parameter of p

e_t = error value at time t

- I

$$Z_t = X_t - X_{t-1} \quad (2)$$

Where:

X_t = current data value

X_{t-1} = past data value

- MA

$$X_t = \mu' + e_t - \phi_1 e_{t-1} - \phi_2 e_{t-2} - \dots - \phi_q e_{t-k} \quad (3)$$

Where:

μ' = constant

ϕ_1 to ϕ_q are the parameter of *moving average*

e_{t-k} = error value at time $t - k$

3.3.2 LSTM Model

The LSTM model is employed for forecasting time series data and is specifically engineered to handle sequential information. This model requires consideration of several criteria, including the quantity of hidden layers and suitable input hyperparameters. This model's architecture comprises three layers: the input layer, the hidden layer, and the output layer.

The stages in the LSTM model process are shown in the following Equations (4) to (9).

- *Forget gate* (f_t)

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (4)$$

Where:

f_t = *forget gate*

σ = sigmoid activation function

W_f = weight of *forget gate*

h_{t-1} = *hidden state cell value*

x_t = *input value*

b_f = bias of *forget gate*

- *Input gate* (i_t)

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (5)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (6)$$

Where:

i_t = *input gate*

σ = sigmoid activation function

W_i = weight of *input gate*

b_i = bias of *input gate*

$\tilde{C}_t$ = *hidden state cell value*

$\tanh$ = *tanh activation function*

W_c = weight of *candidate gate*

b_c = bias of *candidate gate*

- *Cell state* (c_t)

$$c_t = (i_t \cdot \tilde{C}_t + f_t \cdot c_{t-1}) \quad (7)$$

Where:

c_t = *cell gate*

i_t = *input gate*

$\tilde{C}_t$ = *candidate gate*

f_t = *forget gate*

c_{t-1} = *previous cell state value*

- *Output gate* (o_t)

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (8)$$

$$h_t = o_t \cdot \tanh(c_t) \quad (9)$$

Where:

o_t = *output gate*

σ = sigmoid activation function

W_o = weight of *output gate*

h_{t-1} = *hidden state value*

c_t = *cell gate*

b_o = bias of *output gate*

3.3.3 Prophet Model

Facebook has developed the model named Prophet that is involved in prediction. This model is based on an additive model and nonlinear of trend in which predictions are made based on the yearly, weekly, daily, and holiday seasons. The model works quite well in the analysis of time series data whose seasonal pattern is multiple and rather acceptable examines missing values and outliers. The Prophet model works in the following Equations (10) to (13).

$$y_{(t)} = g_{(t)} + s_{(t)} + h_{(t)} + \epsilon_{(t)} \quad (10)$$

Where:

$y_{(t)}$ = predicted value

$g_{(t)}$ = *piecewise linear* or *logistic* growth curve for modelling non-periodic changes in time series.

$s_{(t)}$ = periodic changes.

$h_{(t)}$ = holiday effects with irregular schedules.

$\epsilon_{(t)}$ = errors in the time series.

$$g_{(t)} = \frac{C}{1 + \exp(-k(t - m))} \quad (11)$$

Where:

C = *Carry capacity*

k = *Growth rate*

m = *Offset parameter*

$$s_{(t)} = \sum_{n=1}^N \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right) \quad (12)$$

Where:

P = time period

$$h_{(t)} = [1(t \in D_1, \dots, t \in D_L)]k \quad (13)$$

Where:

D = set of past and future holiday dates

t = time during the holiday

k = change in holiday days during prediction

3.4 Model Performance Comparison

This step compares the ARIMA, LSTM, and Prophet models. The Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and the Mean Absolute Percent Error (MAPE) are the performance measures of these models.

- MAE = the mean of the values of error numbers between the actual numbers and the predicted numbers. The forecasts would be more appropriate because the value of MAE is less-see Equation (14).
- MSE = the average of the squared dispersion between the factual value and the forecasted value. The estimates would be more accurate as the value of MSE decreases with less observed Equation (15).
- RMSE is the square root of the mean squared deviations between the actual values of the data and the values that are predicted. The value of RMSE would be less and hence the predictions would be more accurate-see Equation (16).
- MAPE is the average value of the percentage differences between the predicted results and the actual data values. The predictions would be more accurate as the MAPE value gets smaller-see Equation (17).

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \tilde{Y}_i| \quad (14)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \tilde{Y}_i)^2 \quad (15)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \tilde{Y}_i)^2} \quad (16)$$

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left(\frac{Y_i - \tilde{Y}_i}{Y_i} \right) \quad (17)$$

Where:

n = number of data sample

Y_i = actual value

$\tilde{Y}_i$ = predicted value

4. RESULTS AND DISCUSSIONS

4.1 Data Description

The data used is the Indonesian Stock Price All Time dataset [30]. This dataset consists of historical stock data for a company with the stock code AALI, referring to Astra Agro Lestari. The dataset contains 7 columns with a total of 5763 rows of data (see Table 1). The data columns contain the following information:

- Date: Transaction date (data type: Time).
- Open: Stock opening price (data type: Numeric).
- High: Stock's highest price (data type: Numeric).
- Low: Stock's lowest price (data type: Numeric).
- Close: Stock closing price (data type: Numeric).
- Adj Close: Adjusted closing price based on certain factors (data type: Numeric).
- Volume: The volume of stocks traded on that day (data type: Numeric).

Table 1. Sample of AALI Stock Historical Data

Date	Open	High	Low	Close	Adj Close	Volume
2001-04-05	571.710632	571.710632	547.889343	547.889343	248.808212	57721
2001-04-06	547.889343	547.889343	547.889343	547.889343	248.808212	83433
2001-04-09	547.889343	547.889343	547.889343	547.889343	248.808212	10494
...	...	...	...	...	...	...
2024-05-20	6125.0	6225.0	6125.0	6200.0	6200.0	513700
2024-05-21	6200.0	6225.0	6100.0	6100.0	6100.0	505200
2024-05-22	6150.0	6175.0	6075.0	6175.0	6175.0	474100

Figure 1. AALI Stock Data Visualizations shows the price of closing stock AALI from the early 2000s until 2024. The x-axis shows the dates, while the y-axis shows the price of the stock. There are several peaks from the first year until the most recent year. There is one peak that is quite sharp, around 2008. This peak was certainly caused by a significant event. The price of stocks has decreased since the 2010s. The price of AALI's stock is similar to the consolidation pattern.

4.2 Data Preprocessing

This step is performed to check for any missing data (missing values) or Null values. Missing data would be replaced with the mean or median values. The missing data verification is performed using Python code as shown in Figure 2. The results from running the code (see Figure 3) show that there is no missing data in any of the columns, allowing this data to be used for further analysis. It is also important to note that for the purpose of stock price forecasting analysis over a specific period, only the Close column is used.

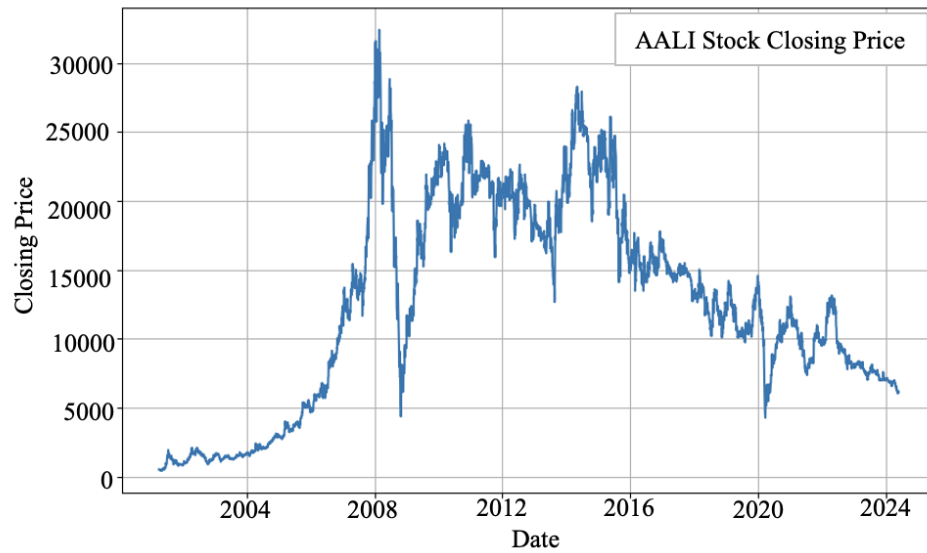


Figure 1. AALI Stock Data Visualization

```
import pandas as pd

# Load the dataset
file_path = 'AALI_historical_data.csv'
data = pd.read_csv(file_path)

# Check for missing values
missing_values = data.isnull().sum()

# Display the result of missing values
print("Missing Values Summary:")
print(missing_values)

# Optionally, display the percentage of missing values
missing_percentage = (missing_values / data.shape[0]) * 100
print("\nMissing Values Percentage:")
print(missing_percentage)
```

Figure 2. Python Code for Checking Missing Values in Data

```
Missing Values Summary:
Date      0
Open      0
High      0
Low       0
Close     0
Adj Close 0
Volume    0
dtype: int64

Missing Values Percentage:
Date      0.0
Open      0.0
High      0.0
Low       0.0
Close     0.0
Adj Close 0.0
Volume    0.0
dtype: float64
```

Figure 3. Program Code Execution Result

4.3 Model Creation

4.3.1 ARIMA Model

The first stage in constructing the ARIMA model is to ascertain the stationarity of the data. This analysis adheres to the notable alpha threshold of 0.05 [31]. If the data is not stationary, differencing is required. The inspection is performed using the Augmented Dickey-Fuller (ADF) test, and the results can be seen in Table 2. Based on the ADF test results, the p-value is greater than the alpha value, so the data needs to be differenced. The different data can be seen in Figure 4.

The ARIMA model has three parameters: **p** (order of autoregression), which is used to determine the required lag variables; **d** (order of integration), which is used to determine the number of differencing; and **q** (order of moving

average), which is used to determine the window size for MA. To determine the value of each parameter, the **pmdarima** library is used in the Python programming language (see Figure 5).

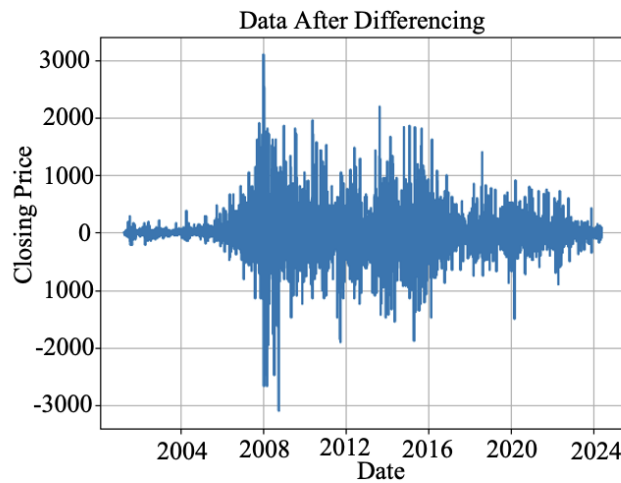


Figure 4. Stock Data After Differencing

Table 2. ADF Test Results

ADF Statistic	P-Value	Symbol	Alpha
-2.0543	0.2633	>	0.05

```
#import pmdarima library
import pmdarima as pm
```

Figure 5. Import Library Pmdarima

The use of this library to determine the automatic ARIMA model can be seen in Figure 6. The code **pm.auto_arima(close_data, seasonal=False, stepwise=True, trace=True)** uses the **auto_arima** function from the **pmdarima** library to automatically select the best ARIMA model based on the provided **close_data** (which contains stock closing prices). The parameter **seasonal=False** indicates that the data does not have a seasonal component, while **stepwise=True** enables stepwise search to efficiently select the best parameter combination (**p, d, q**). The **trace=True** setting allows the process of searching and evaluating parameters during model selection to be displayed. This function will search for the optimal parameters using criteria such as **AIC** or **BIC** and produce the best ARIMA model for the given data, without the need for manual parameter determination.

```
model = pm.auto_arima (close_data,
                        seasonal=False,
                        stepwise=True,
                        trace=True)
```

Figure 6. Examples of using the Pmdarima Library

Figure 7 shows the results of the search and evaluation of the optimal parameters for this study. The SARIMAX model displayed in this output has the ARIMA (5, 1, 3) parameters, with an AIC of 67960.889 and a BIC of 68018.812, which are used to evaluate the model fit. The AR coefficients for lags 1, 2, and 3 are all significant, with very small p-values, indicating a strong influence of past values on the model. The MA (Moving Average) coefficients for the

five lags are also significant; although the coefficients for lags 4 and 5 are smaller, they still contribute to the model's predictions. The sigma2 result of $1.458e+05$ indicates the variance of the residuals (prediction errors). The LjungBox test indicates the residual value to be autocorrelated and Jarque Berra test indicates that the model values are not normally distributed which would be a weakness to the validity of the model. Nevertheless, the test of heteroskedasticity does not indicate anything concerning non- constant residual variance. Generally, this model is applicable to the data, but the problem of non-normal residual distribution should be discussed more. In this study though no further analysis would be done to correct the problem of non-normal residual values, since it does not form the subject of the study.

SARIMAX Results						
=====						
Dep. Variable:	y	No. Observations:	4611			
Model:	SARIMAX(5, 1, 3)	Log Likelihood	-33971.444			
Date:	Sun, 27 Jul 2025	AIC	67960.889			
Time:	04:46:52	BIC	68018.812			
Sample:	0	HQIC	67981.274			
	- 4611					
Covariance Type:	opg					
=====						
	coef	std err	z	P> z	[0.025	0.975]

ar.L1	0.9730	0.053	18.267	0.000	0.869	1.077
ar.L2	0.3543	0.096	3.698	0.000	0.167	0.542
ar.L3	-0.8769	0.060	-14.682	0.000	-0.994	-0.760
ar.L4	0.1407	0.013	11.021	0.000	0.116	0.166
ar.L5	-0.0467	0.010	-4.608	0.000	-0.067	-0.027
ma.L1	-0.8602	0.053	-16.287	0.000	-0.964	-0.757
ma.L2	-0.4615	0.091	-5.094	0.000	-0.639	-0.284
ma.L3	0.8066	0.053	15.130	0.000	0.702	0.911
sigma2	1.479e+05	1419.528	104.212	0.000	1.45e+05	1.51e+05
=====						
Ljung-Box (L1) (Q):	0.02	Jarque-Bera (JB):	12370.64			
Prob(Q):	0.88	Prob(JB):	0.00			
Heteroskedasticity (H):	12.97	Skew:	0.16			
Prob(H) (two-sided):	0.00	Kurtosis:	11.02			
=====						

Figure 7. Results of the Search and Evaluation of the Best Model Parameters

Figure 8 indicates the forecast of the closing prices of AALI stock through ARIMA. The date (X-axis) is between approximately 2020 and 2024 whereas, the Y (Closing Price) is between 4000 and 14 000. The orange line is the actual historical price closing data and the red line is the forecast price obtained by the ARIMA model. The plot above shows that the red line is more likely to plateau beyond sometimes, implying that ARIMA model does not indicate the actual fluctuations of stock prices, giving forewarned similar values over the next period of time. This implies that the ARIMA model is not able to cope with trends or the volatility of the stock prices, which is more complex. Some potential causes of this include the failure of the model to represent very volatile data, ARIMA poorly chosen parameters, or external effects not reflected by the model.



Figure 8. Comparison of Actual Data with ARIMA Model Predicted Data

Table 3 shows the performance analysis figures of a forecasting model. MAE which is 1911.49 represents the average size of deviations between the expected and actual values without considering the direction of error. The lesser the MAE number, the better is the performance of the model. The MSE 5,135,610.20 is used to measure the average squared difference between the predicted values and the actual values with a higher penalty on large errors. The RMSE of 2266.19 is the square root of the MSE, offering an error measurement in the same units as the original data. A lower RMSE value indicates superior model performance. The MAPE of 23.85% indicates that, on average, the model's predictions deviate by around 23% from the actual values.

Table 3. ARIMA Model Forecasting Accuracy Results

MAE	MSE	RMSE	MAPE
1911.49	513560.20	2266.19	23.85%

The ARIMA forecast (red line) collapses to an almost constant $\approx 10k$ while the actual AALI price swings and trends down toward $\sim 6k$; accordingly, MAE 1,911 is the typical day-to-day miss, RMSE 2,266 ($\sim 19\%$ above MAE) signals occasional large errors on shock/trending periods, and MAPE 23.85% shows the mistakes are big relative to price—especially when the level is low. This happens because multi-step ARIMA on differenced data tends to revert to the mean/last level, underfitting regime shifts and volatility clustering; forecasting levels (not returns), omitting event/calendar regressors, and not re-estimating in a rolling fashion all exacerbate the flat forecast.

4.3.2 LSTM Model

Figure 9 shows a line graph juxtaposing the actual closing price of AALI stock with the projected closing price derived from the LSTM model. The X-axis (Date) represents the time frame from January 2020 to July 2024, whereas the Y-axis (Closing Price) indicates stock prices fluctuating between around 4000 and 14,000. LSTM model gives a forecast of the prices of the closing given by the red line and the real price of the closing which is seen in the yellow line. This is a chart that calculates the comparison between stock prices and predicted outcomes. This graph indicates that the red and yellow lines coincide in a parallel form thus means that the LSTM model is quite accurate in forecasting the prices of stocks despite the minor deviations in certain periods.

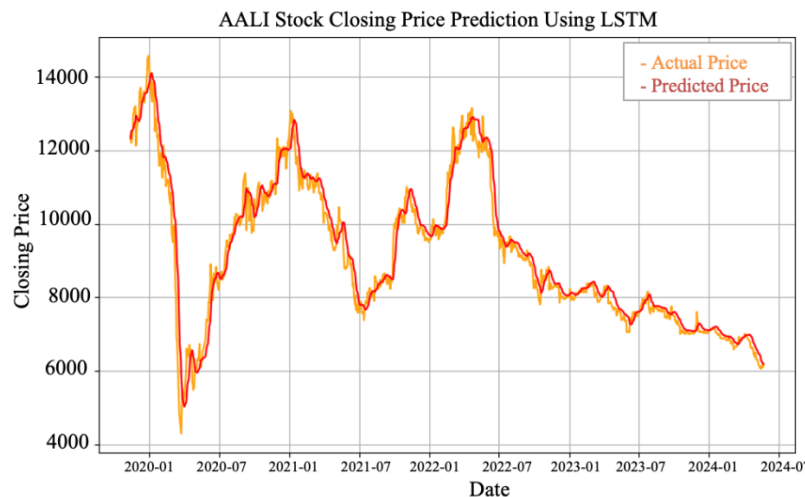


Figure 9. Comparison of Actual Data with LSTM Model Predicted Data

Table 4 shows the forecasting metrics with the LSTM model. The ratio of the MAE of 277.50 shows that the mean difference between the forecasted and the actual values is about 277.50 points. It demonstrates that the value of 165244.61 when the Mean Squared Mistake (MSE) equals it reveals that the total error between the forecasts and data is 165244.61, which implies that there is an error but at a smaller value than before. RMSE= 406.50 represents the

average error of prediction that is in the same units as the original data and hence represents the average error as being 406.50 points on average. The MAPE of 3.09 implies that the mean forecast error would cause the analyses to deviate by an average of 3.09 percent of the actual price. Such results suggest that the prediction model is far more precise with the reasonably smaller errors compared to the earlier discoveries.

Table 4. LSTM Model Forecasting Accuracy Results

MAE	MSE	RMSE	MAPE
277.50	165244.61	406.50	3.09%

The LSTM tracks AALI closely: MAE 277.50 means the typical daily miss; RMSE 406.50 ($\approx 46\%$ above MAE) signals occasional larger mistakes on sharp jumps; MAPE 3.09% shows errors are small relative to the price level. Lower MAE/RMSE would mean fewer and smaller absolute misses, while a lower MAPE would mean better scale-adjusted accuracy across high/low price regimes. These patterns occur because LSTM captures nonlinear momentum/volatility from rolling windows but—optimized for MSE—tends to smooth sudden, news-driven moves (earnings, policy shocks), which inflate squared errors during spikes.

4.3.3 Prophet Model

Figure 10 shows the predicted closing price of AALI stock using the Prophet model. The orange line on the plot represents the actual closing price data of AALI stock, showing significant price fluctuations, along with increases and decreases in AALI's stock closing price. Meanwhile, the red line on the plot represents the closing price of AALI stock predicted using the Prophet model, which does not show significant stock price fluctuations, appearing much smoother and aligning with the existing trend. The predicted closing price of AALI stock, displayed by the red line, is below the peak of the actual closing price, indicating that the Prophet model represents much smoother fluctuations and follows the existing trend. All in all, this plot illustrates that Prophet is less sensitive to short-term changes in the stock price data, but it is more sensitive to the price trends over the long term.

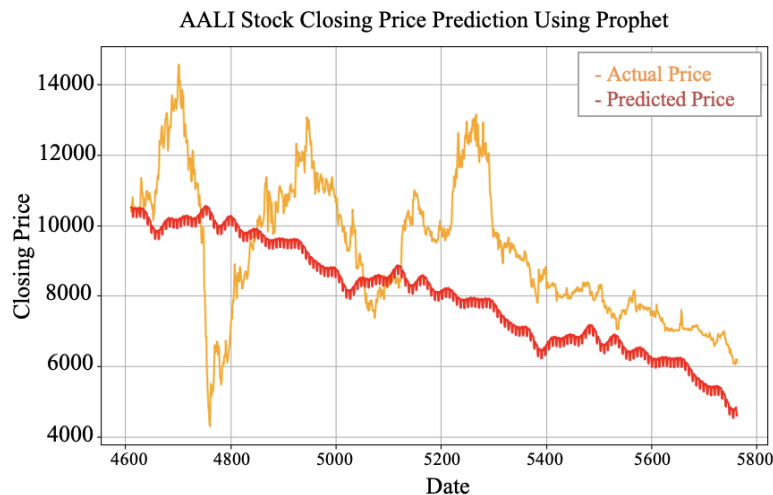


Figure 10. Comparison of Actual Data with Prophet Model Predicted Data

Table 5 presents the prediction with Prophet model. MAE of 1749.01 indicates that, on average, the predicted AALI stock price has an error of about 1749.01 points. Meanwhile, MSE of 4353815.49 indicates a significant error, as MSE gives more weight to larger errors. RMSE of 2086.58 provides an overview that the average prediction error in stock price units is around 2086.58 points. Finally, MAPE of 19.32% shows that the average prediction error is about 19.32%

of the actual stock price. Overall, although the Prophet model provides fairly good results, there are significant errors that need to be addressed to improve the prediction accuracy.

Table 5. Prophet Model Forecasting Accuracy Results

MAE	MSE	RMSE	MAPE
1749.01	4353815.49	2086.58	19.32%

Prophet fits AALI with a smoothed downward path, so MAE 1,749 is the typical daily miss, while RMSE 2,086 ($\approx 19\%$ above MAE; $MSE \approx 4.35 \times 10^6$) shows occasional large errors—usually on sharp rebounds or news shocks—and MAPE 19.32% means errors average nearly one-fifth of price. This happens because Prophet prioritizes decomposable trend/seasonality and, without explicit event/calendar regressors (earnings, rate decisions) or a return-based target, it lags regime shifts and over-smooths volatility-clustered moves. Adding event regressors, using multiplicative seasonality or modeling returns, and refitting in a rolling manner would lower those numbers.

4.4 Comparison of Model Performance

Table 6 compares the evaluation results of three stock price prediction models: ARIMA, LSTM, and Prophet, based on four metrics: MAE, MSE, RMSE, and MAPE. The ARIMA model shows relatively large errors with an MAE of 1911.49, MSE of 513560.20, RMSE of 2266.19, and MAPE of 23.85%, indicating that its predictions are less accurate. In contrast, LSTM shows the best results with an MAE of only 277.50, MSE of 165244.61, RMSE of 406.50, and the lowest MAPE of 3.09%, indicating highly accurate predictions. Prophet performs better than ARIMA, with an MAE of 1749.01, MSE of 4353815.49, RMSE of 2086.58, and MAPE of 19.32%, although it is not as accurate as LSTM. Overall, the LSTM model provides the best prediction results, followed by Prophet and ARIMA, with lower accuracy.

Table 6. Comparison of Accuracy for ARIMA, LSTM, & Prophet Models

Model	MAE	MSE	RMSE	MAPE
ARIMA	1911.49	513560.20	2266.19	23.85%
LSTM	277.50	165244.61	406.50	3.09%
PROPHET	1749.01	4353815.49	2086.58	19.32%

4.5 Discussion

The most widely used time series forecasting models are ARIMA, Prophet and LSTM. All the models possess unique strengths in relation to the type of time series data. ARIMA is an effective and easy model of statistics because it predicts pure, inexplicable trends and seasonality over time. It fits well few anomalies, in stable and short-term predictions. But when data is nonlinear, have high variance or have a long-term dependency, then it is not effective. This can be demanding in time and operation to identify the correct parameters by hand, time-consuming, requires extensive knowledge in statistics, and cannot withstand sudden changes in the data, making it not applicable to volatile markets, like those in the stock market price patterns.

Prophet is a smaller model to grasp seasonal and nonlinear trends. It handles missing values, outliers, and abrupt changes in trends (change points). Thus, it is well-suited for businesses such as sales or commodity price predictions. However, its restricted ability to capture variable interactions and long-term dependency makes the forecast less accurate than LSTM for volatile and dynamic data such as stock predictions.

LSTM is a deep learning model suited for capturing nonlinear, anomaly, and long-term dependent data. It achieves low error rates, especially in stock price predictions but requires large data and computation time which makes it less accessible for non-technical users. Ultimately, the model chosen depends on the type of data and forecasting needs. LSTM is the best performing model for complex data at the expense of higher computation needs.

5. CONCLUSION

This paper did a good job of making stock price prediction models with ARIMA, LSTM, and Prophet. LSTM was the best of the three. The LSTM model had the fewest mistakes and did a better job of modeling the complicated and nonlinear trends in historical stock prices, especially when the prices were very unstable. The most widely used time series forecasting models are ARIMA, Prophet and LSTM. All the models possess unique strengths in relation to the type of time series data. ARIMA is an effective and easy model of statistics because it predicts pure, inexplicable trends and seasonality over time. It fits well few anomalies, in stable and short-term predictions. But when data is nonlinear, have high variance or have a long-term dependency, then it is not effective. This can be demanding in time and operation to identify the correct parameters by hand, time-consuming, requires extensive knowledge in statistics, and cannot withstand sudden changes in the data, making it not applicable to volatile markets, like those in the stock market price patterns. Prophet is a smaller model to grasp seasonal and nonlinear trends.

ACKNOWLEDGEMENT

The authors would like to thank Politeknik Negeri Banjarmasin and all parties involved in this research study and publication.

FUNDING STATEMENT

The authors received funding from Politeknik Negeri Banjarmasin under an internal grant scheme for the research and publication of this article.

AUTHOR CONTRIBUTIONS

Muhammad Syahid Pebriadi: Conceptualization, Methodology, Software, Data Curation, Formal Analysis, Validation, Visualization, Investigation, Writing – Original Draft Preparation;
Tino Kemal Fattah: Conceptualization, Resources, Data Selection, Interpretation of Results, Writing – Review & Editing, Writing – Original Draft Preparation;
Putriana Salman: Project Administration, Supervision, Methodology, Validation, Writing – Review & Editing, Final Approval of the Version to be Published;
Rizky Amelia: Proofreading and Structuring.

CONFLICT OF INTERESTS

No conflict of interests was disclosed.

ETHICS STATEMENTS

Our publication ethics follow The Committee of Publication Ethics (COPE) guideline. <https://publicationethics.org/>

DATA AVAILABILITY

The data that support the findings of this study are available from Kaggle (<https://www.kaggle.com/datasets/nigelimmanuel/indonesian-stock-price-all-time>). Restrictions apply to the availability of these data, which were used under license for this study.

REFERENCES

- [1] T. Shen, 'Machine Learning Applications in Stock Price Prediction', *Advances in Economics, Management and Political Sciences*, vol. 94, pp. 257–267, Oct. 2024, doi: 10.54254/2754-1169/94/2024OX0211.
- [2] Y. L. Tan, Y. H. Pang, S. Y. Ooi, W. H. Khoh, and F. S. Hiew, 'Stacking Ensemble Approach for Churn Prediction: Integrating CNN and Machine Learning Models with CatBoost Meta-Learner', *Journal of Engineering Technology and Applied Physics*, vol. 5, no. 2, pp. 99–107, Sep. 2023, doi: 10.33093/jetap.2023.5.2.12.
- [3] H. H. Mohd Zaki, 'The Determinants of House Prices in Malaysia', *International Journal of Management, Finance and Accounting*, vol. 2, no. 1, pp. 1–18, Feb. 2021, doi: 10.33093/ijomfa.2021.2.1.1.
- [4] S. Tiwary and P. K. Mishra, "Time-series forecasting of stock prices using ARIMA: A case study of TESLA and NIO," *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 2022, doi: 10.46254/in02.20220427.
- [5] C. Pereira da Veiga, C. R. Pereira da Veiga, F. M. Giroto, D. A. B. Marconatto, Z. Su, et al., "Implementation of the ARIMA model for prediction of economic variables: evidence from the health sector in Brazil," *Humanities and Social Sciences Communications*, vol. 11, no. 1, pp. 1068, 2024, doi: 10.1057/s41599-024-03023-3.
- [6] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PLOS ONE*, vol. 13, no. 3, pp. e0194889, 2018, doi: 10.1371/journal.pone.0194889.
- [7] A. I. E. Nensi, M. Al Maida, K. A. Notodiputro, Y. Angraini, and L. N. A. Mualifah, "Performance analysis of ARIMA, LSTM, and Hybrid ARIMA-LSTM in forecasting the composite stock price index," *CAUCHY Journal of Pure and Applied Mathematics*, vol. 10, no. 2, pp. 588–604, 2025, doi: 10.18860/cauchy.v10i2.33379.
- [8] I. Sardar, M. A. Akbar, V. Leiva, A. Alsanad, and P. Mishra, "Machine learning and automatic ARIMA/Prophet models-based forecasting of COVID-19," *Stochastic Environmental Research and Risk Assessment*, vol. 37, no. 1, pp. 345–359, 2023, doi: 10.1007/s00477-022-02307-x.
- [9] M. Y. Wijaya, N. Fitriyati, and N. R. Sandika, "Assessing ARIMA model performance in hierarchical time series forecasting of tourist arrivals: The role of data normalization and bottom-up strategy," *Journal of Modern Applied Statistical Methods*, vol. 24, no. 2, pp. 4, 2025, doi: 10.53941/jmasm.2025.100004.
- [10] H. A. Adam, F. Raditionsyah, M. R. Imani, M. Faris, and A. R. Lubis, "Comparison of ARIMA and LSTM Models in Stock Price Forecasting: A Case Study of GOTO.JK," —
- [11] X. Pan, "Stock price prediction system LSTM based on deep learning," *Journal of Economics and Public Finance*, vol. 10, no. 4, 2024, doi: 10.22158/jepf.v10n4p235.
- [12] O. V. Johnson, X. Chew, K. W. Khaw, and Z. L. Chong, "CRATSM: An effective hybridization of deep neural models for customer retention prediction in the telecom industry," *Journal of Engineering Technology and Applied Physics*, vol. 6, no. 2, pp. 66–78, 2024, doi: 10.33093/jetap.2024.6.2.10.
- [13] C. Pereira da Veiga, C. R. Pereira da Veiga, F. M. Giroto, D. A. B. Marconatto, Z. Su, et al., "Decade-long analysis of sustainable development goals compliance and financial performance tiers in all banking companies listed on Bursa Malaysia," *International Journal of Management, Finance and Accounting*, vol. 6, no. 1, pp. 32–86, 2025, doi: 10.33093/ijomfa.2025.6.1.2.
- [14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [15] A. Sunki, C. Satya Kumar, G. Surya Narayana, V. Koppera, and M. Hakeem, "Time series forecasting of stock market using ARIMA, LSTM and FB Prophet," *MATEC Web of Conferences*, vol. 392, pp. 01163, 2024, doi: 10.1051/mateconf/202439201163.

- [16] S. Chaturvedi, E. Rajasekar, S. Natarajan, and N. McCullen, "A comparative assessment of SARIMA, LSTM RNN and FB Prophet models," *Energy Policy*, vol. 168, p. 113097, 2022, doi: 10.1016/j.enpol.2022.113097.
- [17] S. J. Taylor and B. Letham, 'Forecasting at Scale', *The American Statistician*, vol. 72, no. 1, pp. 37–45, Jan. 2018, doi: 10.1080/00031305.2017.1380080.
- [18] C. Long, C. Yu, and T. Li, "Prophet-based medium and long-term electricity load forecasting research," *Journal of Physics: Conference Series*, vol. 2356, no. 1, pp. 012002, 2022, doi: 10.1088/1742-6596/2356/1/012002.
- [19] Z. Chen, C. Wang, L. Lv, L. Fan, S. Wen, and Z. Xiang, "Research on peak load prediction of distribution network lines based on Prophet-LSTM model," *Sustainability*, vol. 15, no. 15, pp. 11667, 2023, doi: 10.3390/su151511667.
- [20] G. Alsheheri, "Comparative analysis of ARIMA and NNAR models for time series forecasting," *Journal of Applied Mathematics and Physics*, vol. 13, no. 1, pp. 267–280, 2025, doi: 10.4236/jamp.2025.131012.
- [21] C. Jiang, 'Comparative Analysis of ARIMA and Deep Learning Models for Time Series Prediction', in *Proceedings of the 2nd International Conference on Data Analysis and Machine Learning*, Kuala Lumpur, Malaysia: SCITEPRESS - Science and Technology Publications, 2024, pp. 306–310. doi: 10.5220/0013516200004619.
- [22] I. Ahammad, S. K. Roy, M. Hasan, et al., "Advancing stock market predictions with time series analysis including LSTM and ARIMA," *Cloud Computing and Data Science*, pp. 226–241, 2024, doi: 10.37256/ccds.5220244470.
- [23] D. Duarte and J. Faerman, "Comparison of time series prediction of Healthcare Emergency Department Indicators with ARIMA and Prophet," *9th International Conference on Computer Science, Engineering and Applications (ICCSEA 2019)*, pp. 123–133, 2019, doi: 10.5121/csit.2019.91810.
- [24] S. Masum, Y. Liu, and J. Chiverton, "Multi-step time series forecasting of electric load using machine learning models," Springer, pp. 148–159, 2018, doi: 10.1007/978-3-319-91253-0_15.
- [25] A. S. Pandit, A. H. B. Jalal, A. K. Toma, and P. Nachev, 'Analyzing historical and future acute neurosurgical demand using an AI-enabled predictive dashboard', *Sci Rep*, vol. 12, no. 1, p. 7603, May 2022, doi: 10.1038/s41598-022-11607-9.
- [26] T. Maulenkul, T. Aimyshev, A. Oshibayeva, A. Yazici, et al., 'Forecasting the future burden of major diseases in Kazakhstan using global burden of disease and time series models (ARIMA, Prophet, LSTM), 2019-2032', *Electronic Journal of General Medicine*, vol. 23, no. 1, p. em710, Jan. 2026, doi: 10.29333/ejgm/17734.
- [27] L. Menculini, A. Marini, M. Proietti, A. Garinei, et al., "Comparing Prophet and deep learning to ARIMA in forecasting wholesale food prices," *Forecasting*, vol. 3, no. 3, pp. 40, 2021, doi: 10.3390/forecast3030040.
- [28] U. T. Prodjinotho, P. K. Chetangny, M. B. Agbomahena, V. Zogbochi, L. Medewou, G. Barbier, and D. Chamagne, "Long term forecasting of peak demand and annual electricity consumption of the West African power pool," *International Journal of Energy and Power Engineering*, vol. 13, no. 2, 2024, doi: 10.11648/j.ijepe.20241302.11.
- [29] M. Fattouh, L. Lyssenko, F. Heilmeyer, T. Ball, and Ch. Haverkamp, 'Forecasting hospital bed occupancy: a time series approach with prophet', *BMC Medical Informatics and Decision Making*, vol. 26, no. 1, p. 162, May 2026, doi: 10.1186/s12911-026-03542-w.
- [30] N. Immanuel, "Indonesia Stock Exchange Historical Data," Kaggle Dataset, 2020.
- [31] I. G. T. Suryawan, I. K. N. Putra, P. M. Meliana, and I. G. I. Sudipa, 'Performance Comparison of ARIMA, LSTM, and Prophet Methods in Sales Forecasting', *Sinkron*, vol. 8, no. 4, pp. 2410–2421, Oct. 2024, doi: 10.33395/sinkron.v8i4.14057.

BIOGRAPHIES OF AUTHORS

	Muhammad Syahid Pebriadi is a dedicated lecturer at Politeknik Negeri Banjarmasin, where he contributes to the development of computer science education. He earned his bachelor's degree in Computer Science from Lambung Mangkurat University in 2014 and went on to pursue a master's degree in the same field at Bogor Agricultural University, graduating in 2017. His academic and research interests focus on Artificial Intelligence, Computer Vision, and Financial Technology. He is reachable via email at: m.syahid@poliban.ac.id.
	Tino Kemal Fattah got his bachelor's degree in Accounting from STIESIA Surabaya, where he focused on Financial Accounting. He got his Master's degree in Accounting from UPN "Veteran" East Java, where he focused on Public Sector Accounting. He started his career as an accounting consultant and then worked as an accountant for a mining contractor company. He is currently a lecturer in Accounting at Politeknik Negeri Banjarmasin. He has a lot of experience teaching public sector accounting and computerized accounting applications. You can reach him tino@poliban.ac.id.
	Putriana Salman is a lecturer at Politeknik Negeri Banjarmasin. Her studies are mostly about financial technology, digital accounting for small and medium-sized businesses, and financial accounting. She got her bachelor's degree in accounting from Lambung Mangkurat University in 2011 and her master's degree in accounting from Brawijaya University in 2014. She also takes part in other scientific activities that have to do with the growth of accounting and finance. You can get in touch with her at putrianasalman@poliban.ac.id.
	Rizky Amelia is a lecturer at Politeknik Negeri Banjarmasin with a Master's degree in Education. She specializes in English language teaching and educational technology. One of her notable works is developing a digital learning module using a virtual lab to make science learning more engaging for elementary teacher education students. She is also active in research and has published works on English teaching and instructional material development. You can find more on her Google Scholar profile. She can be contacted at rizky.amelia@poliban.ac.id.