Journal of Applied Data Sciences
Vol. 6, No. 1, January 2025, pp. 683-694

Enhancing Sharia Stock Price Forecasting using a Hybrid ARIMA-LSTM with Locally Weighted Scatterplot Smoothing Regression Approach

Aris Gunaryati[1,*], Achmad Benny Mutiara[2], Sulistyo Puspitodjati[3]

[1,2,3]*Gunadarma University, Jl. Margonda Raya No. 100, Depok 16424, Indonesia*

[1]*Faculty of Technology Communication and Informatics, Nasional University, Jl. Sawo Manila, Pejaten, Jakarta 12520, Indonesia*

(Received: August 14, 2024; Revised: September 15, 2024; Accepted: October 22, 2024; Available online: December 31, 2024)

Abstract

Predicting Sharia stock prices is complex because it has high volatility and non-linear data patterns. To improve the accuracy of the forecast, the right technique is needed according to the existing data pattern. One of the techniques currently developing is integrating (hybrid) two forecasting models. This study proposes a hybrid autoregressive integrated moving average (ARIMA) and long short-term memory (LSTM) model with the locally weighted scatterplot smoothing (lowess) linear regression technique. This model is designed by creating a linear regression between the actual value and the predicted results of the ARIMA and LSTM models using the Lowess technique. The dataset used here is the closing stock prices of four Indonesian Islamic banking companies. The hybrid ARIMA-LSTM model with lowess linear regression significantly outperforms the individual ARIMA and LSTM models because it produces better performance metrics, namely mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), for training and testing datasets. The proposed hybrid model effectively reduces noise, and the model can capture complex patterns in the Sharia stock price dataset, and the prediction results are more accurate. The accuracy values for training data and data testing datasets were respectively 97.6% and 98.3% (BANK. JK), 98.3% and 98.2% (BRIS. JK), 99.4% and 99.5% (BTPN. JK), and 97.7% and 99.3% (PNBS. JK).

Keywords: Stock Price, Forecasting, ARIMA, LSTM, Smoothing Regression, Hybrid Model

1. Introduction

With the increasing interest of investors in investing according to Sharia principles, the sharia stock market in Indonesia has also experienced rapid growth. In general, Islamic stocks show greater performance in crises during the COVID-19 pandemic [1]. Performance of sharia stock indices in Indonesia (JII-70) is higher than the Malaysia (FBMS) during and after the COVID-19 pandemic [2]. Muslim millennials tend to favor stock investments that adhere to their religious values and align with the moral principles and ethical standards upheld by the Muslim community, including in the realm of investing. [3].

Sharia stock prices often experience extreme volatility and irregular changes, making it difficult for investors to predict their movements accurately [4]. To minimize risk and maximize profits in volatile markets, innovations in forecasting techniques are increasingly developing along with technological advances. One of the conventional methods often used is ARIMA, which works well for linear patterns but has difficulty capturing complex nonlinear dynamics in stock prices [5], [6]. Research shows that ARIMA is quite effective for linear data [7], [8], [9], [10], while artificial neural network models such as Long Short-Term Memory (LSTM) are more flexible in handling nonlinear patterns. However, LSTM is less than optimal in capturing short-term patterns [11], [12], [13], [14]. To overcome the weaknesses of these two methods, a hybrid ARIMA-LSTM model was developed, which utilizes the strengths of ARIMA in detecting linear patterns and LSTM in handling nonlinear data [15], [16], [17], [18], [19], [20]. By combining ARIMA residuals into LSTM, this model can improve prediction accuracy [21]. Several studies have developed other hybrid models, such as Wavelet Transform ARIMA-LSTM (WT-ARIMA-LSTM), which uses wavelet transform to analyze data at multiple time scales [22]. A three-stage fusion model combining market sentiment and historical data has also shown better

***Corresponding author: Aris Gunaryati (aris_gunaryati@staff.gunadarma.ac.id)**
DOI: https://doi.org/10.47738/jads.v6i1.514

 ISSN 2723-6471

results than the benchmark model [23]. In addition, a study of the Dhaka stock market using hybrid ARIMA-LSTM successfully handled both linear and nonlinear components in portfolio optimization [24]. Another study combining ARIMA and LSTM with a weighted average smoothing method [25] inspired the development of a new model integrating Lowess regression [26], which can improve the stability and accuracy of predictions.

Based on the literature that has been reviewed, the main objective of this study is to develop a more accurate Sharia stock price prediction model by combining ARIMA and LSTM using the Lowess approach. This study will evaluate the performance of the hybrid model compared to the ARIMA and LSTM models separately and analyze their impact on prediction accuracy. The case study will use daily closing stock price data from four Sharia banks in Indonesia, namely PT Bank Aladin Syariah Tbk, PT Bank Syariah Indonesia Tbk, PT Bank BTPN Syariah Tbk, and PT Bank Panin Dubai Syariah Tbk, for the period from February 1, 2021, to July 31, 2024. It is expected that this study can provide a significant contribution to developing a more accurate predictive model for Sharia stock price forecasting, with the hybrid ARIMA-LSTM method showing its superiority in predicting data with complex patterns and assessing the effectiveness of the Lowess approach in this context.

2. Methodology

To obtain an ARIMA-LSTM hybrid model with the Lowess linear regression approach, it can be done through the stages as presented in figure 1 below:

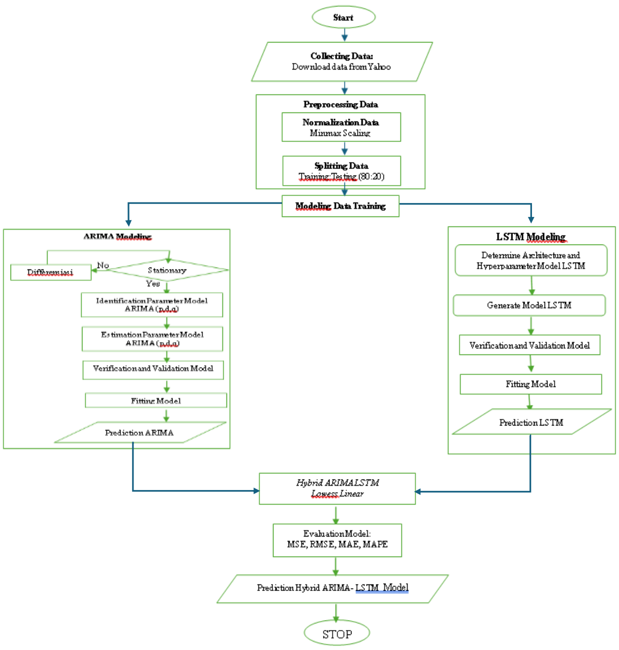

Figure 1. Research Methodology of Sharia Stock Price Forecasting

In figure 1, the proposed hybrid method is a combination of the ARIMA model and LSTM model predictions on a train set by smoothing those predictions' values using locally weighted scatterplot smoothing (lowess) regression. Detailed explanations of each step are explained in the following sub-chapters.

2.1. Collecting Data

The dataset in this study consists of daily stock closing price data from four Sharia banks in Indonesia: PT Bank Aladin Syariah Tbk, PT Bank Syariah Indonesia Tbk, PT Bank BTPN Syariah Tbk, and PT Bank Panin Dubai Syariah Tbk. The data were obtained from Yahoo Finance for the period from February 1, 2021, to July 31, 2024, with tickers BANK.JK, BRIS.JK, BTPN.JK, and PNBS.JK. The total data collected was 844 entries for each bank.

ISSN 2723-6471

2.2. Preprocessing Data

Preprocessing data steps include cleaning data and normalization data using Minmax Scaling and splitting the data into 80% training data and 20% testing data. The formula of the Minmax Scaling is used to normalize the data so that each value is within a certain range, generally between 0 and 1. The formula is as follows:

$$\mathrm{X_{scaled}} = \frac{\mathrm{X - X_{min}}}{\mathrm{X_{max} - X_{min}}} \tag{1}$$

Note: X is the original value of the data to be normalized; X_{min} is the minimum value of the dataset; X_{max} is the maximum value of the dataset.

2.3. Data Modelling

2.3.1. Integrated Autoregressive Moving Average (ARIMA) Model

There are three variations of the Box-Jenkins (ARIMA) model, i.e., mixed ARIMA (combining AR and MA), autoregressive (AR), and moving average (MA). It is organized according to the ARIMA (p, d, q) structure, in which q is the number of moving averages, d is the number of lag observations, and p is the number of lag observations needed to render the time series stationary. Conditional least squares and sample data are used to estimate the autoregressive and moving average operators, Ø(B) and θ(B). This study automates the process of selecting the best ARIMA parameters and producing precise forecasts using Grid Search and Python's pmdarima package [27]. This formula is presented in equation (2), (3), and (4).

$$\emptyset(\mathrm{B})\nabla^{\mathrm{d}}\mathrm{y_t} = \theta(\mathrm{B})\mathrm{e_t} \tag{2}$$

$$\emptyset(\mathrm{B}) = 1 - \emptyset_1\mathrm{B} - \emptyset_2\mathrm{B}^2 - \cdots - \emptyset_{\mathrm{p}}\mathrm{B}^{\mathrm{p}} \tag{3}$$

$$\theta(\mathrm{B}) = 1 - \theta_1\mathrm{B} - \theta_1\mathrm{B}^2 - \cdots - \theta_{\mathrm{q}}\mathrm{B}^{\mathrm{q}} \tag{4}$$

Note: y_t is the actual value at period t; e_t is random error at period t, the noise component of the stochastic model and is assumed to be identically distributed with a mean of zero and a constant variance; $\emptyset(B)$ is the autoregressive operator of order p with parameters $\phi_1, \phi_2, \phi_3, \ldots, \phi_p$; $\theta(B)$ is the moving average operator of order q with parameters $\theta_1, \theta_2, \theta_3, \ldots, \theta_q$; $\phi_1, \phi_2, \phi_3, \ldots, \phi_p$ and $\theta_1, \theta_2, \theta_3, \ldots, \theta_q$ are unknown coefficients that were estimated from sample data using the conditional least squares approach; B and $\nabla = 1 - B$ are backward shift operators that can be defined as $B^m y_t = y_{t-m}$; $\Delta y_t = y_t - y_{t-1}$ with $\nabla^d = \nabla\nabla^{d-1}$

2.3.2. Long Sort-Term Memory (LSTM) Model

An enhanced variety of recurrent neural networks (RNN) called LSTM is intended to manage long-term dependencies and resolve the vanishing gradient issue. Three gates—input, forget, and output gates—control the flow of information in it [6]. The equation for these gates is:

$$\mathrm{I_t} = \sigma(\mathrm{W_{xi}} \cdot \mathrm{x_t} + \mathrm{W_{hi}} \cdot \mathrm{H_{t-1}} + \mathrm{b_i}) \tag{5}$$

$$\mathrm{F_t} = \sigma(\mathrm{W_{xf}} \cdot \mathrm{x_t} + \mathrm{W_{hf}} \cdot \mathrm{H_{t-1}} + \mathrm{b_f}) \tag{6}$$

$$\mathrm{O_t} = \sigma(\mathrm{W_{xo}} \cdot \mathrm{x_t} + \mathrm{W_{ho}} \cdot \mathrm{H_{t-1}} + \mathrm{b_o}) \tag{7}$$

$$\tilde{\mathrm{C}}_\mathrm{t} = \tanh\,(\mathrm{W_{xc}} \cdot \mathrm{x_t} + \mathrm{W_{hc}} \cdot \mathrm{H_{t-1}} + \mathrm{b_c}) \tag{8}$$

$$\mathrm{C_t} = \mathrm{F_t} \odot \mathrm{C_{t-1}} + \mathrm{I_t} \odot \tilde{\mathrm{C}}_\mathrm{t} \tag{9}$$

$$\mathrm{H_t} = \mathrm{O_t} \odot \tanh(\mathrm{C_t}) \tag{10}$$

Note: I_t = Input gates; F_t = Forget gates; O_t = Output gates; $\tilde{C}_t$ = new candidate vectors; C_t = current cell state; and H_t = current hidden state; $W_{xi}, W_{xf}, W_{xo}, W_{hi}, W_{hf}, W_{ho}, W_{xc}$ and W_{hc} = learnable weight parameters; b_i, b_f, b_o, and b_c = learnable bias parameters.

Here, $\odot$ is elementwise multiplication. To generate a single output, the study used a two-layer LSTM and a single dense layer. The model's settings, which were determined to be ideal for the datasets, are 100 epochs, 32 batches, the 'Adam' optimizer, and 'MSE' as the loss function.

2.3.3. New Hybrid ARIMA-LSTM Models with Locally Weighted Scatterplot Smoothing (Lowess) Regression Approach

Lowess is a non-parametric regression method used to detect patterns in non-linear data by performing local linear regression on small portions of the data rather than applying a single model to the entire dataset. It assigns weights based on the distance between data points and the point of interest. The equation is:

$$\hat{Y} = \beta^{T}X \tag{11}$$

Extending this concept to using weights is quite simple and the normal equation just needs an extra term:

$$\beta = \left(X^{T}WX\right)^{-1}X^{T}WY \tag{12}$$

In this experiment, Lowess linear regression was used to smooth and combine predictions from the ARIMA (X) and LSTM (X_2) models as independent variables. The actual value (Y) was the dependent variable to produce a more accurate forecast by combining the predictions from both models.

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 \tag{13}$$

Note: $\hat{Y}$ is the prediction of the dependent variable Y ; $\hat{\beta}_0$ is intercept; $\hat{\beta}_1$ and $\hat{\beta}_2$ are the estimation of regression coefficients for X_1 (output ARIMA) and X_2 (output LSTM); X_1 and X_2 are independent variables (prediction of ARIMA and LSTM)

In standard linear regression, each data point contributes equally to the coefficient values β_0, β_1 and β_2. In the weighted linear regression, each data point i has a weight w_i, which determines how much influence the point has on the model. The goal is to minimize the Weighted Sum of Squared Errors (WSSE), which is:

$$\text{WSSE} = \sum_{i=1}^{n} w_i \left(Y_i - \hat{Y}_i\right)^2 \tag{14}$$

Note: w_i : The weight for the data point i; Y_i : The actual value for the data point i; $\hat{Y}_i$: The predicted value for the data point i is calculated from the regression model.

Combine X_1 and X_2 into a matrix X with shape $(n, 2)$ where n is the number of data points and a column of ones to account for β_0 (the intercept):

$$X = \begin{pmatrix} 1 & X_{1,1} & X_{2,1} \\ 1 & X_{1,2} & X_{2,2} \\ \vdots & \vdots & \vdots \\ 1 & X_{1,n} & X_{2,n} \end{pmatrix} \tag{15}$$

Note: $X_{1,1}, X_{1,2}, \dots, X_{1,n}$: The predictions from the ARIMA model for data-1 to data-n; $X_{2,1}, X_{2,2}, \dots, X_{2,n}$: The predictions from the LSTM model for data-1 to data-n; n: The amount of data predicted during the training period; The Lowess linear regression computes the Euclidean distances between the new data points (X_{1new}, X_{2new}) and all the other points in the dataset as follows:

$$d_i = \sqrt{(X_{1,i} - X_{1new})^2 + (X_{2,i} - X_{2new})^2} \tag{16}$$

Then, apply a Tricube Kernel function to compute the weights w_j based on the distance d_j and the bandwidth h.

$$w_i = \left(1 - \left(\frac{d_i}{h}\right)^3\right)^3 \text{ if } d_i < h \text{ and } w_i = 0 \text{ if } d_i \geq h \tag{17}$$

Note: d_i: The Euclidean distance between a new point and all points in a dataset in two-dimensional space; h: The bandwidth is taken from the t-th percentile frac of the distance distribution; w_i: The weight for the data point i; After the weights are calculated, construct the diagonal weight matrix W of the shape. (n, n), $W = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{pmatrix}$

Solve the weighted least squares regression by minimizing the weighted sum of squared errors in equation (14). The equation to compute the coefficients β_0, β_1 and β_2 is given by:

$$\hat{\beta} = \left(X^T W X\right)^{-1} X^T W Y \tag{18}$$

where $\hat{\boldsymbol{\beta}} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \end{pmatrix}$ is a vector of the parameters estimate (intercept β_0) and regression coefficients (β_1 and β_2) estimate and $\boldsymbol{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}$ is a vector of actual values Y_i ; X^T is the transpose of the matrix X in equation (15).

2.4. Metric Evaluation

The performance of the hybrid model can be evaluated using standard error metrics, such as:

$$\text{Mean Squared Error (MSE): MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{y}_i)^2) \tag{19}$$

$$\text{Root Mean Square Error (RMSE): RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{y}_i)^2} \tag{20}$$

$$\text{Mean Absolute Error (MAE): MAE} = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{y}_i| \tag{21}$$

$$\text{Mean Absolute Percentage Error (MAPE): MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i - \hat{y}_i|}{Y_i} \text{ x } 100\% \tag{22}$$

The smaller the percentage error value in MAPE, the more accurate the forecast results are MAPE values can be interpreted into 4 categories, here is the explanation table 1:

Table 1. MAPE Target (Lewis, 1982)

MAPE	< 10%	10% -19%	20% - 49%	≥ 50%
Accuracy	Very Accurate	Good	Fair	Not Accurate

These metrics help in comparing the predictive accuracy of the hybrid model against individual ARIMA and LSTM models.

3. Results and Discussion

The following are the results of modeling simulations for the dataset using the ARIMA, LSTM, and hybrid ARIMA-LSTM models with the Lowess linear regression approach according to the stages in the methodology.

3.1. Summary of Models and Parameters

The summary table for each model includes important details such as input data, best model structure, and parameter models, which can be seen in table 2 below:

Table 2. Summary of Models and Parameters Models

Dataset	Model	Best Model Structure	Parameter Model
BANK.JK	ARIMA	ARIMA (2,2,1)	AR. L1: 0.0848 (p = 0.001) AR. L2: -0.0970 (p = 0.000) MA. L1: -0.9711 (p = 0.000)
	LSTM	1 LSTM Layer (50 units), 1 Dense Layer	Epochs: 100, Batch Size: 32, Seq_length: 1, optimizer: Adam, loss: mse

	Hybrid ARIMA-LSTM	ARIMA (2,2,1) + 1 LSTM Layer (50 units)	ARIMA (2,2,1) + 1 LSTM Layer (50 units), frac: 0.1
BRIS.JK	ARIMA	ARIMA (0,1,0)	None
	LSTM	1 LSTM Layer (50 units), 1 Dense Layer	Epochs: 100, Batch Size: 32, Seq_length: 1, optimizer: Adam, loss: mse
	Hybrid ARIMA-LSTM	ARIMA (0,1,0) + 1 LSTM Layer (50 units)	ARIMA (0,1,0) + 1 LSTM Layer (50 units), frac: 0.1
BTPN.JK	ARIMA	ARIMA (1,1,2)	AR. L1: -0.5394 (p = 0.000) MA. L1: 0.2842 (p = 0.044) MA. L2: -0.2878 (p = 0.000)
	LSTM	1 LSTM Layer (50 units), 1 Dense Layer	Epochs: 100, Batch Size: 32, Seq_length: 1, optimizer: Adam, loss: mse
	Hybrid ARIMA-LSTM	ARIMA (1,1,2) + 1 LSTM Layer (50 units)	ARIMA (1,1,2) + 1 LSTM Layer (50 units), frac: 0.1
PNBS.JK	ARIMA	ARIMA (3,1,3)	AR. L1: -0.2340 (p= 0.252) AR. L2: -0.5582 (p = 0.000) AR. L3: -0.7682 (p = 0.000) MA. L1: 0.2243 (p = 0.292) MA. L2: 0.5746 (p = 0.000) MA. L3: 0.7167 (p = 0.001)
	LSTM	1 LSTM Layer (50 units), 1 Dense Layer	Epochs: 100, Batch Size: 32, Seq_length: 1, optimizer: Adam, loss: mse
	Hybrid ARIMA-LSTM	ARIMA (3,1,3) + 1 LSTM Layer (50 units)	ARIMA (3,1,3) + 1 LSTM Layer (50 units), frac: 0.1

Table 2 shows the ARIMA model for the bank. JK dataset is ARIMA (2, 1). The BRIS.JK dataset is ARIMA (0,1,0), for BTPN. JK dataset is ARIMA (1,1,2) and for PNBS.JK dataset is ARIMA (3,1,3). Meanwhile, the PNBS.JK dataset features a more complex model with multiple AR and MA terms, suggesting significant lagged influences. Each model demonstrates a relatively low p-value for residual autocorrelation, indicating a good fit. The LSTM model is characterized by 50 units in the LSTM layer, and one dense layer with the number of epochs = 100, batch size = 32, sequence length = 1, optimizer = Adam, and loss function = mse are also specified. Hybrid ARIMA-LSTM parameters show the combined structure and model parameters used for the hybrid models. The frac parameter represents a fraction or proportion of the data used in the combination process. For example, for BANK.JK, the hybrid model is a combination of ARIMA (2, 2, 1) with one LSTM layer (50 units) and frac set to 0.1.

3.2. Performance Models

The prediction results are essential for evaluating a model's performance. By comparing them with actual data, the model's accuracy can be assessed using various evaluation metrics.

3.2.1. Comparison Prediction and Actual Value

Plotting actual versus predicted values visually compares the model's ability to capture data trends and values. A well-performing model shows the prediction line closely following the actual line. Smaller residuals, or the differences between actual and predicted values, indicate better performance. Figure 2 illustrates the ARIMA, LSTM, and hybrid ARIMA-LSTM with the Lowess model's predictions compared to actual values for both training and testing datasets.

Figure 2. Graphic of Model Prediction Results vs Actual Value for Training Data and Testing Data

Figure 2 presents that the ARIMA model performs well during the training period, closely matching actual stock prices. However, in the test period, there is a clear divergence between the model's flat, upward predictions and the actual, more volatile stock prices. This indicates that the ARIMA model overfits the training data and struggles to generalize to new, unseen data, resulting in poor predictions during the test phase. The LSTM model performs well for BANK.JK and BRIS.JK in both training and testing phases, though slight deviations occur in predicting price fluctuations during testing. BTPN.JK shows good performance in training but faces challenges in testing, while PNBS.JK experiences larger deviations, especially towards the end of the test period. The main difference between training and testing results is the higher accuracy in training due to familiarity with the data, while testing predictions tend to have larger deviations, particularly with volatile price swings.

The hybrid ARIMA-LSTM model with lowess linear regression performs well during the training phase for all datasets (BANK.JK, BRIS.JK, BTPN.JK, and PNBS.JK), accurately capturing patterns and trends. However, during the testing phase, the model shows mixed results. It follows the general trends but struggles with high volatility, particularly for Bank.JK and BTPN.JK, leading to larger discrepancies. BRIS.JK and PNBS.JK datasets show better accuracy, although some deviations occur during volatile periods. Overall, the model effectively captures stock price trends but requires fine-tuning to handle volatility in unseen data. The comparison of ARIMA, LSTM, and hybrid ARIMA-LSTM models with Lowess linear regression shows that all models closely match actual values during the training period. However, in the testing period, the ARIMA model struggles with non-linear patterns, showing significant discrepancies. The LSTM model performs better, capturing non-linear data more effectively. The hybrid ARIMA-LSTM model with lowess linear regression provides the best and most stable predictions, closely resembling actual values in both training and testing periods, outperforming both individual models.

3.2.2. Comparison Model with Evaluation Metrics

To evaluate and compare model performance, we used the evaluation metrics MSE, RMSE, MAE, and MAPE. The following table 3 summarizes the metrics evaluation values of each model for each dataset and provides a comparison of evaluation metrics (e.g., MSE, RMSE, MAE, MAPE) for ARIMA, LSTM, and Hybrid ARIMA-LSTM models across four stock datasets (BANK.JK, BRIS.JK, BTPN.JK, and PNBS.JK). It highlights the performance of each model on both training and test data, allowing for an assessment of prediction accuracy and effectiveness for each stock data set. For the BANK.JK dataset, the ARIMA model shows high errors, especially in the test data, indicating poor performance in predicting unseen data. The LSTM model improves on ARIMA, with lower errors in the test set. However, the hybrid ARIMA-LSTM model performs best, reducing errors significantly in both training and test data, demonstrating that the combination of both models captures data patterns more effectively. For the BRIS.JK dataset, the ARIMA model performs poorly on the test data with high errors, while the LSTM model improves over ARIMA but still has relatively high errors. The hybrid model shows significant improvement, lowering errors on the test set. For the BTPN.JK dataset, ARIMA performs well in training but struggles in testing, leading to higher errors.

Table 3. Models Evaluation Metrics

Dataset	Model	Evaluation Metrics Train Data				Evaluation Metrics Test Data			
		MSE	RMSE	MAE	MAPE	MSE	RMSE	MAE	MAPE
BANK.JK	ARIMA	9979.84034	99.899151	67.067642	0.0376337	28628.783	169.20042	149.23	0.1510357
	LSTM	4865.90184	69.756016	48.425642	0.02849	2449.2015	49.489408	42.721388	0.041939
	Hybrid ARIMA-LSTM	3956.93127	62.904144	42.952694	0.0239167	849.63896	29.148567	17.963431	0.0165498
BRIS.JK	ARIMA	7891.29574	88.832965	48.468796	0.0266517	651001.34	806.84654	718.06548	0.2996836
	LSTM	2420.96052	49.203257	32.146191	0.0178547	4317.6002	65.708449	46.50483	0.0201369
	Hybrid ARIMA-LSTM	2067.49511	45.469716	30.14391	0.0167153	3495.2904	59.120981	41.620812	0.0181552
BTPN.JK	ARIMA	1570.10331	39.624529	23.654598	0.0088684	24644.22	156.98478	111.76651	0.0471029
	LSTM	876.145746	29.599759	20.715084	0.0077564	790.60114	28.11763	21.590034	0.008839
	Hybrid ARIMA-LSTM	631.331458	25.12631	17.075474	0.0064231	319.29293	17.86877	12.217379	0.0049396
PNBS.JK	ARIMA	25.1625212	5.0162258	2.9068259	0.0324418	20.654355	4.5447063	4.0154733	0.0790223
	LSTM	12.4706597	3.5313821	2.2758014	0.0260684	0.6638049	0.8147422	0.6843723	0.0130736
	Hybrid ARIMA-LSTM	10.8980394	3.3012179	2.0114493	0.0226353	0.3574968	0.5979104	0.3983082	0.0074789

The LSTM model outperforms ARIMA with lower errors in both phases, while the hybrid model achieves the best performance, particularly in reducing test data errors. In the PNBS.JK dataset, ARIMA performs decently but has higher test errors, LSTM improves significantly, and the hybrid model excels with minimal errors across both training and testing. (See figure 3 and figure 4).

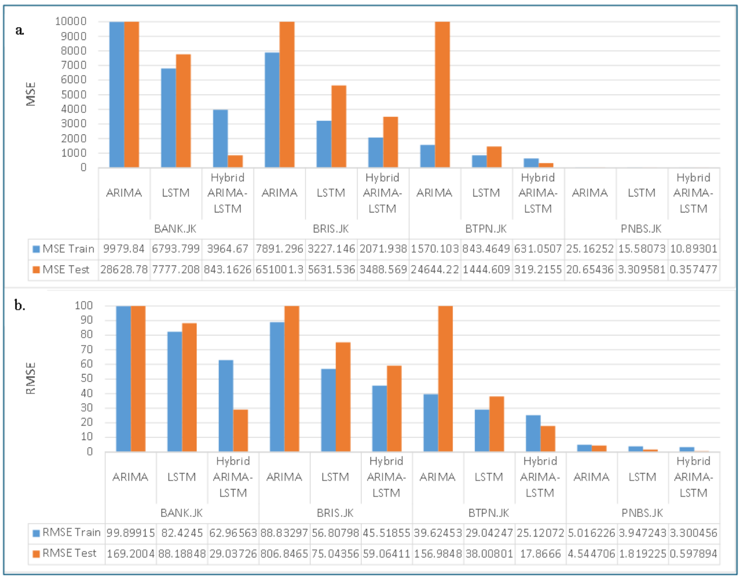

Figure 3. The value of MSE (a) and RSME(b) for each model of each dataset

Figure 3(a) shows that the MSE values for the hybrid ARIMA-LSTM model are the lowest compared to the ARIMA and LSTM models in both training and testing datasets, indicating its greater consistency and accuracy. The highest MSE occurs in the Bank.JK dataset for the ARIMA model during training and in the BRIS.JK dataset during testing, while the lowest MSE is in the PNBS.JK dataset, indicating a better representation of actual values. Similarly, figure 3(b) illustrates that the RMSE values for the hybrid ARIMA-LSTM model are also the smallest, reinforcing its superior performance. The highest RMSE values are found in the same datasets (BANK.JK during training and BRIS.JK during testing), with the lowest RMSE in the PNBS.JK dataset, further suggesting better model accuracy for this dataset.

Figure 4. The value of MAE (a) and MAPE (b) for each model of each dataset

Figure 4(a) shows that the MAE values for the hybrid ARIMA-LSTM model are the lowest compared to the individual MAE values of the ARIMA and LSTM models in both training and testing datasets, indicating superior consistency and accuracy. The highest MAE occurs in the bank. JK dataset with the ARIMA model during training and in the BRIS.JK dataset during testing. In contrast, the lowest MAE is found in the PNBS.JK dataset for both training and testing periods, suggesting a better representation of actual values. Figure 4(b) highlights that the MAPE values for the hybrid ARIMA-LSTM model are also the lowest compared to the LSTM model in the BANK. JK dataset during training and the ARIMA model during testing. The smallest MAPE value is observed in the PNBS.JK dataset for both training and testing periods, reinforcing that this model better captures actual values compared to the others.

ISSN 2723-6471

4. Conclusion

Predicting Sharia stock prices presents a significant challenge due to the inherent volatility and non-linear patterns in the data. In this study, we demonstrated that the hybrid ARIMA-LSTM model, combined with lowess linear regression, consistently outperforms both the ARIMA and LSTM models in forecasting the closing prices of stocks for four Indonesian Islamic banking companies. By leveraging the strengths of both ARIMA and LSTM, the hybrid model effectively captures complex time-series patterns and improves prediction accuracy. The evaluation metrics—MSE, RMSE, MAE, and MAPE—confirm the superior performance of the hybrid ARIMA-LSTM model across all datasets. Specifically, the model exhibits the lowest error values for both training and testing datasets, which underscores its ability to generate reliable predictions even when faced with previously unseen data. The comparison between the ARIMA and LSTM models further highlights the advantages of combining traditional statistical approaches with deep learning methods, as ARIMA alone struggles to predict accurately, particularly in the test data, while LSTM performs better but still falls short of the hybrid model's performance.

This hybrid approach demonstrates a robust capacity to generalize and adapt to various data complexities, as evidenced by its performance on the four datasets: BANK.JK, BRIS.JK, BTPN.JK, and PNBS.JK. The significant reduction in error rates across all performance metrics validates that integrating ARIMA's time-series forecasting capabilities with LSTM's ability to model long-term dependencies yields more accurate and stable predictions. This is particularly crucial in volatile markets such as those associated with Sharia stock prices, where precision is critical for both investors and financial analysts. Furthermore, the analysis of individual evaluation metrics, such as MSE, RMSE, MAE, and MAPE, solidifies the hybrid model's consistency. The model consistently achieves the lowest error values compared to both standalone models across the training and testing phases. This further emphasizes the practicality of this hybrid approach for stock price prediction tasks in real-world financial scenarios, where forecasting accuracy can directly influence investment strategies.

In conclusion, the hybrid ARIMA-LSTM model is an effective solution for predicting Sharia stock prices, demonstrating its ability to reduce prediction errors significantly while providing more reliable results compared to traditional and deep learning models. Its application can be extended to various financial forecasting problems, potentially enhancing decision-making processes in the financial markets. Future research could explore further enhancements to the hybrid model, such as incorporating additional financial indicators or experimenting with other machine learning techniques, to improve the robustness and accuracy of predictions in different market conditions.

5. Declarations

5.1. Author Contributions

Conceptualization: A.G., A.B.M., S.P.; Methodology: S.P.; Software: A.G.; Validation: A.G., S.P., and A.B.M.; Formal Analysis: A.G., S.P., and A.B.M.; Investigation: A.G.; Resources: S.P.; Data Curation: S.P.; Writing Original Draft Preparation: A.G., S.P., and A.B.M.; Writing Review and Editing: S.P., A.G., and A.B.M.; Visualization: A.G.; All authors have read and agreed to the published version of the manuscript.

5.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

5.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

5.4. Institutional Review Board Statement

Not applicable.

5.5. Informed Consent Statement

Not applicable.

 ISSN 2723-6471

5.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. R. Arifin, A. S. Fauzi, S. Runtiningsih, F. Aligarh, and Arif Nugroho, “The Performance of Islamic Stocks and Conventional Stocks During the COVID-19 Shock: Evidence from Indonesian Stock Market”, *IQTISHODUNA: Jurnal Ekonomi Islam*, vol. 13, no. 1, pp. 265–284, Apr. 2024.

[2] F. Azizah, S. S. Handayani, Riskayanto, and M. R. Rizaldy, “Comparative Analysis of Indonesia and Malaysia Sharia Stock Index Performance Using Sharpe, Treynor, and Jensen Methods”, *IJIBEC*, vol. 8, no. 1, pp. 116–126, Jun. 2024.

[3] S. Sudirman, “Sharia Stocks and Muslim Millennials Investors in Indonesia: Between Religious and Economic Motives”, *J. Professional Business Review*, vol. 8, no. 5, p. e01829, May 2023.

[4] S. Suripto, "Decision-making model to predict auto-rejection: an implementation of ARIMA for accurate forecasting of stock price volatility during the COVID-19," *Decision Science Letters*, vol. 12, no. 1, pp. 107–116, 2023.

[5] W. A. Woodward, B. P. Sadler, and S. D. Robertson, *Time Series for Data Science: Analysis and Forecasting*. Chapman and Hall/CRC Texts in Statistical Science Series, 2022, pp. 275–380.

[6] M. Joseph, *Modern Time Series Forecasting with Python*, Packt Publishing Ltd., 2023, pp. 87, 302–303.

[7] S. T. Wahyudi, "The ARIMA model for the Indonesia stock price," *International Journal of Economics and Management*, vol. 11, no. S1, pp. 223–236, 2017.

[8] B. K. Meher, I. T. Hawaldar, C. M. Spulbar, and F.R. Birau, "Forecasting stock market prices using mixed ARIMA model: a case study of Indian pharmaceutical companies," *Investment Management and Financial Innovations*, vol. 18, no. 1, pp. 42–54, 2021.

[9] Dong, Yichen; Li, Siyi; Gong, Xueqin. “Time series analysis: An application of arima model in stock price forecasting,” *in: 2017 International Conference on Innovations in Economic Management and Social Science (IEMSS 2017). Atlantis Press*, vol. 2017, no. 1, pp. 703-710, 2017.

[10] A. Subakkar, S. Graceline Jasmine, L. Jani Anbarasi, J. Ganesh, and C.M. Yuktha Sri, "An analysis on Tesla’s stock price forecasting using ARIMA model," in *Proceedings of the International Conference on Cognitive and Intelligent Computing (ICCIC 2021), Volume 2*, Springer, 2023, vol. 2, no. 1, pp. 83–89.

[11] A. A. Adebiyi, A. O. Adewumi, and C. K. Ayo, "Comparison of ARIMA and artificial neural networks models for stock price prediction," *Journal of Applied Mathematics*, vol. 2014, no. 1, pp. 1–7, 2014.

[12] P. R. Low and E. Sakk, "Comparison between autoregressive integrated moving average and long short-term memory models for stock price prediction," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 4, pp. 1828–1835, 2023.

[13] D. Kobiela, D. Krefta, W. Król, P. Weichbroth ., "ARIMA vs LSTM on NASDAQ stock exchange data," *Procedia Computer Science*, vol. 207, pp. 3836–3845, 2022.

[14] H. Dong, "Stock Price Prediction Using ARIMA and LSTM," 2022 6th Annual International Conference on Data Science and Business Analytics (ICDSBA), Changsha, China, vol. 6, no. 1, pp. 195-201, 2022

[15] S. Siami-Namini, N. Tavakoli and A. Siami Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series," 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, vol. 17, no. 1, pp. 1394-1401, 2018.

[16] K. Chen, Y. Zhou and F. Dai, "A LSTM-based method for stock returns prediction: A case study of China stock market," *2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA*, vol. 2015, no. 1, pp. 2823-2824, 2015, doi: 10.1109/BigData.2015.7364089.

[17] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 2017, pp. 1643-1647, doi: 10.1109/ICACCI.2017.8126078.

[18] E. Dezsi and I. A. Nistor, "Can deep machine learning outsmart the market? A comparison between econometric modeling and long-short-term memory," *Rom. Econ. Bus. Rev.*, vol. 11, no. 4.1, pp. 54–73, 2016.

[19] A. J. P. Samarawickrama and T. G. I. Fernando, "A recurrent neural network approach in predicting daily stock prices an application to the Sri Lankan stock market," 2017 IEEE International Conference on Industrial and Information Systems (ICIIS), Peradeniya, Sri Lanka, 2017, pp. 1-6, doi: 10.1109/ICIINFS.2017.8300345.

[20] M. Hiransha, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "NSE stock market prediction using deep-learning models," *Procedia Comput. Sci.*, vol. 132, pp. 1351–1362, 2018.

[21] S. Kulshreshtha, et al., "An ARIMA-LSTM hybrid model for stock market prediction using live data," *J. Eng. Sci. Technol. Rev.*, vol. 13, no. 4, pp. 117-123, 2020.

[22] J. Zhang, H. Liu, W. Bai, X. Li, , "A hybrid approach of wavelet transform, ARIMA, and LSTM model for the share price index futures forecasting," *The North American Journal of Economics and Finance*, vol. 69, no. 102022, 2024.

[23] R. Kumar, P. Kumar, and Y. Kumar, "Three-stage fusion for effective time series forecasting using Bi-LSTM-ARIMA and improved DE-ABC algorithm," *Neural Comput. Appl.*, vol. 34, no. 21, pp. 18421–18437, 2022.

[24] R. Alam, R. I. Arnob, and A. E. Alam, "An ARIMA-LSTM Correlation Coefficient Based Hybrid Model for Portfolio Management of Dhaka Stock Exchange," in *Emerging Technologies in Computing. iCETiC 2020*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 332, M. H. Miraz, P. S. Excell, A. Ware, S. Soomro, and M. Ali, Eds. Cham, Switzerland: Springer, 2020, pp. 214–226.

[25] C. Ma, J. Wu, H. Hu, Y.N Chen, and J.Y. Li, "Predicting stock prices using hybrid LSTM and ARIMA model," *IAENG Int. J. Appl. Math.*, vol. 54, no. 3, 2024.

[26] W. S. Cleveland and S. J. Devlin, "Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting," *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 596–610, 1988.

[27] J. Herzen, F. Lässig, S. G. Piazzetta, T. Neuer, L. Tafti, G. Raille, T. V. Pottelbergh, M. Pasieka, A. Skrodzki, N. Huguenin, M. Dumonal, J. Kościsz, D. Bader, F. Gusset, M. Benheddi, C. Williamson, M. Kosinski, M. Petrik, and G. Grosch, "Darts: user-friendly modern machine learning for time series," *J. Mach. Learn. Res.,* vol. 23, no. 1, Art. no. 124, pp. 1–6, Jan. 2022.