Indonesian Journal of Data and Science

Volume 7 Issue 2
ISSN 2715-9936
https://doi.org/10.56705/ijodas.v7i2.459

Research Article

The Performance of the XGBOOST-LSTM and CNN-LSTM Algorithms in the Analysis of Stock Price Prediction Models for the Indonesian Banking Sector

M. Zainal Arifin [1*]; Filbert Chaitra Bessel Kristianto [2]; Fadia Irsania Putri [3]; Agung Bella Putra Utama[4]

[1] Universitas Negeri Malang, Malang 65145, Indonesia, arifin.mzainal@um.ac.id
[2] Universitas Negeri Malang, Malang 65145, Indonesia, filbert.chaitra.2205356@students.um.ac.id
[3] Universitas Negeri Malang, Malang 65145, Indonesia, fadiairsania123.@gmail.com
[4] Universitas Negeri Malang, Malang 65145, Indonesia, agung.bella.2305349@students.um.ac.id

Corresponding Author M. Zainal Arifin: arifin.mzainal@um.ac.id

Received 21 April 2026; Accepted 30 June 2026; Published 31 July 2026

Abstract:
Introduction: Accurate stock price forecasting is important for investors and financial institutions because stock movements reflect dynamic market conditions and influence investment decision-making. This study compares CNN-LSTM and XGBoost-LSTM models for predicting stock prices in the Indonesian banking sector. **Method**: Historical daily closing prices of PT Bank Central Asia Tbk (BBCA), PT Bank Rakyat Indonesia (Persero) Tbk (BBRI), and PT Bank Mandiri (Persero) Tbk (BMRI) were collected from Yahoo Finance, with 1,000 observations for each stock. Data were normalized using Min-Max scaling, transformed using a four-day sliding window to predict the following day, and chronologically divided into 80% training and 20% testing sets. Both models were evaluated using RMSE, MAE, R^2, and MAPE, with each experiment repeated ten times. **Results and Discussion**: CNN-LSTM consistently outperformed XGBoost-LSTM on all three test datasets. For BBCA, BBRI, and BMRI, CNN-LSTM achieved R^2 values of 0.8589, 0.7746, and 0.7409, respectively, compared with 0.8048, 0.7509, and 0.6260 for XGBoost-LSTM. CNN-LSTM also produced lower RMSE, MAE, and MAPE values across all test sets, indicating stronger and more stable generalization. **Conclusion**: CNN-LSTM provides more reliable predictive performance than XGBoost-LSTM for the evaluated Indonesian banking stocks, demonstrating the effectiveness of combining local feature extraction with temporal dependency learning for stock price forecasting.

Keywords: Prediction Stock; XGBoost; CNN; LSTM.

1. Introduction

The stock market is one of the key indicators reflecting a country's economic conditions, as it reflects investors' expectations regarding companies' prospects and future economic growth [1]. Changes in share prices are influenced by various factors, such as a company's fundamentals, market sentiment, economic policies, and global economic dynamics, which make price movements dynamic and difficult to predict [1], [2]. The accuracy of share price forecasts is a crucial aspect, as the results of these forecasts are used as the basis for investment decision-making, the formulation of corporate business strategies, and financial risk analysis [3]. In Indonesia, the banking sector is one of the sectors with the largest market capitalisation [4]; consequently, the ability to predict share price movements in this sector holds strategic value for investors, financial institutions and regulators in supporting more informed decision-making.

Various approaches have been developed to improve the accuracy of share price predictions. Conventional approaches, such as technical analysis, utilise historical price patterns to forecast price movements in subsequent periods [5]. With the advancement of computing technology, methods based on machine learning and deep learning are increasingly being used as they are capable of learning non-linear relationships and complex temporal characteristics in time series data [6]. Various algorithms, such as Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), Long Short-Term Memory (LSTMs), Convolutional Neural Networks (CNNs), and even hybrid models combining several algorithms, have demonstrated improved performance compared to conventional approaches in various stock price forecasting scenarios.

However, there is as yet no consensus on which deep learning architecture delivers the best performance across all characteristics of stock data [7]. Previous research conducted by [5] showed that LSTM models are effective at learning long-term temporal dependencies, but still have limitations in capturing local patterns that emerge in time-series data [8]. On the other hand, CNN-LSTM models are able to enhance feature extraction capabilities through convolutional layers prior to the temporal learning process carried out by the LSTM [9]. Other studies, such as [10], have also developed the XGBoost-LSTM model by utilising the XGBoost algorithm for feature extraction prior to modelling with LSTM. However, the majority of studies have focused solely on developing a specific model or have used different datasets; consequently, the results obtained are difficult to use as a basis for determining the most suitable model for the characteristics of the Indonesian stock market, particularly the banking sector.

Given these circumstances, there is a need for research that conducts a direct comparison between the CNN-LSTM and XGBoost-LSTM models using the same dataset, pre-processing scheme, training procedures, and evaluation metrics. This approach allows for a more objective evaluation process, ensuring that any differences in performance are genuinely attributable to the characteristics of each model architecture, rather than to differences in the dataset or experimental procedures. Furthermore, the use of historical share price data from the banking companies with the largest market capitalisation in Indonesia is expected to provide a more representative picture of the two models' ability to learn share price movement patterns under domestic market conditions.

Based on the above, this study aims to analyse and compare the performance of the CNN-LSTM and XGBoost-LSTM models in predicting share prices in the Indonesian banking sector. The comparison was carried out using closing price data from PT Bank Central Asia Tbk, PT Bank Rakyat Indonesia (Persero) Tbk, and PT Bank Mandiri (Persero) Tbk, applying the same pre-processing steps to both models. Performance evaluation was carried out using the Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and coefficient of determination (R^2) metrics. The main contribution of this study is to provide a comprehensive analysis of the generalisation ability of both models on the Indonesian banking sector stock dataset, thereby serving as a reference for the development of more accurate and reliable stock price prediction models.

2. Method

The research methodology employed in this study was designed to develop and evaluate a stock price prediction model using two deep learning approaches, namely CNN-LSTM and XGBoost-LSTM. The overall research workflow is presented in **Figure 1**, which comprises four main stages: data collection, data pre-processing, model development, and performance evaluation.

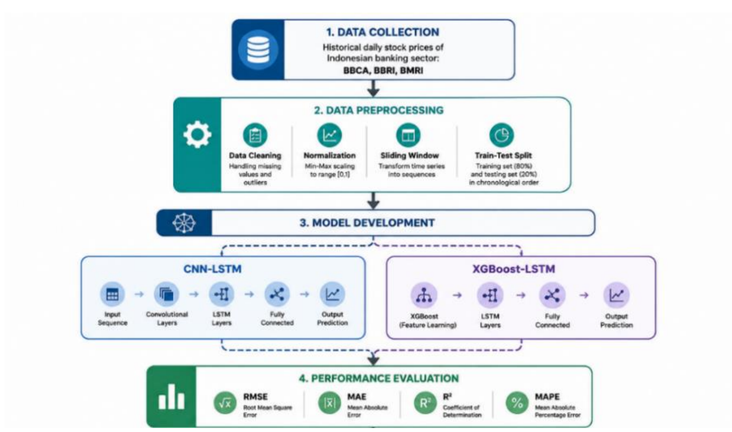

Figure 1. Research Flow

During the model development stage, both approaches were applied to the same dataset, thereby enabling an objective comparison of performance based on evaluation metrics including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), and Mean Absolute Percentage Error (MAPE). This approach is designed to identify the model that offers the best accuracy and generalisation ability in predicting share prices in the banking sector in Indonesia.

Data Collection

This study utilises historical daily share price data for the banking sector obtained via the Yahoo Finance API, using the yfinance library in the Python programming language [11]. The three listed companies selected as the subjects of this study are PT Bank Central Asia Tbk (BBCA.JK), PT Bank Rakyat Indonesia (Persero) Tbk (BBRI.JK), and PT Bank Mandiri (Persero) Tbk (BMRI.JK). These three listed companies were chosen because they are classified as large-cap stocks (blue-chip stocks) with high liquidity and consistently represent the banking sector on the Indonesia Stock Exchange (IDX) [12]. The selection of these three stocks also aims to evaluate the performance of the prediction model across different data characteristics, whilst remaining within the same industry sector.

The data used consists of closing share prices at daily intervals over the period from 7 October 2021 to 2 December 2025. Each dataset comprises 1,000 observations, bringing the total number of observations used in this study to 3,000. All data is stored in Comma-Separated Values (CSV) format to facilitate processing during the data pre-processing and model development stages. This study uses only the 'Closing Price' attribute, as the closing price is considered to represent the final trading value for each day and is the most commonly used variable in time-series-based stock price prediction research. Furthermore, the closing price reflects the final outcome of daily trading activity and thus provides a more stable representation of share price trends compared to intraday prices.

By way of illustration, **Table 1** presents an example of the structure of the daily share price dataset used in this study. The dataset consists of two main attributes: the 'Date' attribute, which represents the trading date, and the 'Closing Price' attribute, which indicates the closing price of the shares on each trading day. The table displays only a few rows of data as an example of the dataset structure, whilst the full dataset is used in the model training and testing processes.

Table 1. Sample Dataset For Pt Bank Central Asia

Date	Closing Price
2021-10-07	6445.874023
2021-10-08	6562.907715
2021-10-11	6531.397949
2021-10-12	6589.916016
2021-10-13	6774.469238
…	…
2025-12-2	…

Based on **Table 1**, each observation is arranged chronologically by trading date, thereby forming a time series. It is important to preserve this characteristic, as the temporal relationships between periods form the basis for the learning process of predictive models [13]. The dataset is subsequently used as input during the data pre-processing stage, for creating data sequences using the sliding window technique, and for the training and testing processes of the CNN-LSTM and XGBoost-LSTM models proposed in this study.

Data Preprocessing

The data pre-processing stage is carried out to prepare the data before it is used in the model training process. This stage aims to improve data quality so that it is suitable for deep learning-based prediction models [14]. In this study, the data pre-processing consists of three main stages, namely data normalisation using the Min-Max Normalisation method, data sequencing (sliding window), and the division of the dataset into training and testing data.

The first stage in data pre-processing is data normalisation. Normalisation is carried out to convert the range of attribute values to a uniform scale so that the model learning process becomes more stable and efficient [15]. If the ranges of values between features differ significantly, the model optimisation process may become less effective, as the model tends to be more influenced by attributes with larger values [16]. Therefore, this study applies the Min-Max Normalisation method, which transforms all data values into a range of 0 to 1.

The normalisation equation used is expressed as follows.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where:

x = the original data value,

x_{min} = the minimum value in the dataset,

x_{max} = the maximum value in the dataset,

x_1 = the result of normalisation.

Through this process, all data are scaled uniformly, thereby accelerating the model convergence process and reducing the dominance of values with a wider range. An example of the data normalisation results is shown in **Table 2**.

Table 2. Data Normalization Results

Closing Price (Actual)	Closing Price (Normalized)
6445.874023	0.006262
6562.907715	0.034459
6531.397949	0.026868
6589.916016	0.040966
6774.469238	0.085431

Table 2 shows an example of the results of normalisation using the Min-Max Normalisation method. It can be seen that all Closing Price values have been successfully mapped to the range 0 to 1 without altering the order or temporal characteristics of the data. This normalisation process aims to standardise the scale of all observations so that the CNN-LSTM and XGBoost-LSTM models can learn price change patterns more effectively. The normalised data is then used as input in the data sequencing (sliding window) stage prior to model training and testing.

Once the normalisation process is complete, the data is then organised into sequences using the sliding window technique. This technique aims to convert time series data into pairs of input and output data [17], so that it can be processed by both the CNN-LSTM and XGBoost-LSTM models.

In this study, a time step size of 4 was used, meaning that the four previous closing prices were used to predict the closing price for the next period. The selection of this time step value was based on research [18], which showed that this size provides a balance between the historical information used and the model's complexity.

By way of illustration, if the following data sequence is available:

$$[x_1, x_2, x_4, x_4, x_5] \quad (2)$$

The data pairs formed are therefore:

Input:

$$[x_1, x_2, x_4, x_4] \quad (3)$$

Target:

$$x_5 \quad (4)$$

This process is repeated until all historical data has been successfully converted into a time-series training dataset. Once the data preparation process using time steps has been completed, the next stage is to split the dataset into a training set and a testing set. The division of the dataset aims to evaluate the model's ability to learn historical patterns whilst also measuring the model's generalisation ability on data that has not been used during the training process [19]. With this separation of data, the model's performance is assessed not only based on its ability to fit the training data, but also on its ability to produce accurate predictions on new data.

In this study, the dataset was split into 80 per cent for the training set and 20 per cent for the testing set. This proportion was chosen as it provides a balance between a sufficient amount of data for the model's learning process and an adequate amount of data for evaluating performance. The training data was used to analyse the temporal relationships between share prices, adjust the model parameters, and optimise the learning process [13]. Meanwhile, the test data was only used after the training process was complete to evaluate the model's ability to make predictions on data it had never seen before.

Unlike tabular data in general, the division of the dataset in this study is carried out chronologically without shuffling [20]. This approach is applied to preserve the temporal order of the time-series data so that historical information is maintained and data leakage is prevented—a condition where information from future data is indirectly used during the training process [21]. Consequently, 80 per cent of the data from the initial period was used as training data, whilst 20 per cent of the data from the final period was used as test data. This approach reflects real-world forecasting conditions, where models always utilise historical information to estimate values for the next period.

Model Development

The model development stage aims to build and implement two deep learning-based stock price prediction approaches, namely Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM) and XGBoost–Long Short-Term Memory (XGBoost-LSTM), using a dataset that has undergone pre-processing. Historical closing price data, which has been normalised, is then transformed into sequence data using the sliding window technique, so that it can represent the temporal relationships between periods and is consistent with the characteristics of time series data. Both models were developed to evaluate the effectiveness of each approach in analysing stock price movement patterns and to compare their predictive capabilities on data with temporal characteristics.

The CNN-LSTM model was developed by integrating the ability of a Convolutional Neural Network (CNN) to extract local patterns in time series data with the ability of Long Short-Term Memory (LSTM) to model both short-term and long-term temporal dependencies [22]. Meanwhile, the XGBoost-LSTM model was constructed using a hybrid approach, in which the XGBoost algorithm is used to extract key information and patterns from historical data before the extracted results are processed by the LSTM network to analyse the temporal relationships underlying changes in share prices [23]. Both models were implemented using the Python programming language, utilising the TensorFlow and Keras libraries for the development of the LSTM networks, as well as the XGBoost library for the implementation of the gradient boosting algorithm.

The first model, CNN-LSTM, was developed by combining the ability of a Convolutional Neural Network (CNN) to extract local patterns from time-series data with that of a Long Short-Term Memory (LSTM) network to learn long-term temporal dependencies [22], [24]. The combination of these two architectures enables the model to capture relationships between share prices over adjacent periods whilst retaining relevant historical information to generate price predictions for the subsequent period [25]. This approach has been widely used in time series forecasting research as it improves feature representation compared to the use of LSTM alone.

In this study, the first layer, Conv1D, is used to extract local patterns from stock price data that has undergone normalisation [24]. The convolutional layer utilises 64 filters with a kernel size of 3 and the Rectified Linear Unit (ReLU) activation function. The ReLU activation function was chosen as it accelerates the convergence process and is effective when applied to data normalised to the range 0–1 [26]. The model's input takes the form of a time series with four days of observations (time step = 4), which is used to predict the share price for the following day.

Next, the feature extraction results from the convolutional layer are processed using 1D MaxPooling with a pool size of 2 to reduce the feature dimension whilst retaining the most representative information characteristics [27]. This process aims to reduce computational complexity whilst minimising information redundancy before the data is passed to the LSTM layers.

The subsequent stage utilises two LSTM layers, each comprising 100 neural units. The first LSTM layer produces sequential outputs (return_sequences=True) so that temporal information can still be processed by the subsequent LSTM layer, whilst the second layer produces a final representation (return_sequences=False) which is used as the basis for the prediction process. To improve the model's generalisation ability and reduce the risk of overfitting, Dropout with a value of 0.2 is applied after each LSTM layer.

In the final stage, the feature representation is passed to a Dense layer comprising 25 neurons with a ReLU activation function, followed by a Dense layer containing a single neuron as the final output in the form of a predicted share price. The complete configuration of the CNN-LSTM model is presented in **Table 3**.

Table 3. CNN-LSTM Model

Layer	Parameters
Conv1D	filters=64, kernel_size=3, activation='relu', input_shape=(4, 1)
MaxPooling1D	pool_size=2
LSTM	units=100, return_sequences=True
Dropout	0.2
LSTM (2)	units=100, return_sequences=False
Dropout (2)	0.2
Dense	units=25, activation='relu'
Dense (2)	units=1

The architectural configuration in Table 3 shows that the CNN-LSTM model is constructed in stages, starting with feature extraction using Conv1D, dimensionality reduction via MaxPooling1D, learning of temporal dependencies using two LSTM layers, and finally generating predicted values via a Dense layer. This configuration is designed to strike a balance between the model's ability to capture historical share price patterns and its ability to generalise to previously unobserved data.

The second model, XGBoost-LSTM, is a hybrid architecture that combines the ability of Extreme Gradient Boosting (XGBoost) to extract non-linear characteristics from data with the ability of Long Short-Term Memory (LSTM) to model temporal dependencies in time-series data [28]. In this study, XGBoost is not used as the final predictive model, but rather as a feature extractor that generates an additional representation of historical share price patterns before they are processed by the LSTM network.

The first stage involved building an XGBoost model using the pre-processed data. The XGBoost parameters were based on a study [29], which demonstrated optimal performance in stock price prediction using the XGBoost-LSTM approach. The parameter configuration used in this study is shown in **Table 4**.

Table 4. XGBoost Algorithm Parameters

Parameters	Value
n_estimators	300
learning_rate	0.03
max_depth	4
random_state	42

The XGBoost model generates predicted values based on historical data patterns, which are then used as additional features. These features are combined with normalised closing price data, so that each input sample has two attributes:

the actual normalised value and the predicted value from XGBoost. Consequently, the input provided to the LSTM has a dimension of (4,2), comprising four-time steps with two features at each time step. This approach aims to enrich the data representation learnt by the LSTM so that the spatial information obtained via XGBoost can be utilised alongside the temporal relationships learnt by the LSTM.

Subsequently, the data is processed using two LSTM layers accompanied by a dropout layer to reduce the risk of overfitting. The final part of the model consists of a fully connected layer that maps the feature representation to the predicted stock price for the next period. To maintain consistency across experiments and ensure that any differences in performance stem from the use of XGBoost as a feature extractor, the LSTM configuration used was made identical to that of the CNN-LSTM model. The complete structure of the LSTM network is shown in **Table 5**.

Table 5. LSTM Algorithm

Layer	**Parameters**
LSTM	units=100, activation='relu', return_sequences=True, input_shape=(4, 2)
Dropout	0.2
LSTM (2)	units=100, return_sequences=False
Dropout (2)	0.2
Dense	units=25, activation='relu'
Dense (2)	units=1

Based on this configuration, the first LSTM layer is responsible for learning the temporal patterns of the two input features at each time step, whilst the second LSTM layer generates the final representation used as input for the dense layer. The use of two LSTM layers enables the model to capture more complex temporal relationships, whilst dropout with a value of 0.2 is applied to improve the model's generalisation ability to previously unobserved data. Subsequently, a dense layer with 25 neurons integrates the feature representations before producing a single output value in the form of a stock price prediction for the following day.

Model Evaluation

Model evaluation was carried out to measure the accuracy of the predictions generated by the CNN-LSTM and XGBoost-LSTM models on the test data. The evaluation process utilised four metrics commonly applied in time series forecasting research, namely Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), and Mean Absolute Percentage Error (MAPE) [30]. The use of multiple evaluation metrics aims to provide a more comprehensive assessment, as each metric has distinct characteristics in measuring the quality of the forecast results. Smaller values of RMSE, MAE and MAPE indicate that the forecast results are closer to the actual values, whilst an R^2 value closer to one indicates that the model is better able to explain the variation in the data.

Root Mean Square Error (RMSE) is used to measure the average magnitude of prediction errors by imposing a greater penalty on errors with high values [31]. Consequently, this metric is highly sensitive to outliers and is frequently used to evaluate regression prediction models.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (5)$$

where:

yi: Actual value at the i-th observation

n = Number of observations

A smaller RMSE value indicates that the predicted results are closer to the actual values.

Furthermore, the Mean Absolute Error (MAE) is used to calculate the average absolute error between the actual value and the predicted value without imposing an additional penalty on large errors [31]. Therefore, the MAE is easier to interpret as the average difference between the prediction and the actual data.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{6}$$

Where:

yi: Actual value at the i-th observation

n = Number of observations

A smaller MAE value indicates a lower level of prediction error in the model.

Furthermore, the Coefficient of Determination (R^2) is used to measure the model's ability to explain the variation in the actual data [32]. The R^2 value ranges from 0 to 1, where a value closer to 1 indicates that the model is better able to represent the patterns in the historical data.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{7}$$

where:

yi = Actual value at the i-th observation

$\hat{y}_i$ = Predicted value at the i-th observation

$\bar{y}$ = Mean of the actual values

n = Number of observations

A higher R^2 value indicates that the model is better able to explain the variation in the actual data.

Mean Absolute Percentage Error (MAPE) is used to measure the average percentage error between the actual and predicted values [33]. As it is expressed as a percentage, this metric facilitates the interpretation of the model's accuracy and allows for the comparison of performance across datasets on different scales.

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{8}$$

Where:

yi = Actual value at the i-th observation

$\hat{y}_i$ = Predicted value at the i-th observation

n = Number of observations

The smaller the MAPE value, the lower the percentage of prediction error, indicating that the model has a higher level of accuracy.

3. Result and Discussion

A performance evaluation was carried out to compare the predictive capabilities of the CNN-LSTM and XGBoost-LSTM models on three Indonesian banking sector share price datasets, namely BBCA, BBRI and BMRI. The performance of both models was measured using four regression evaluation metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), and Mean Absolute Percentage Error

(MAPE) [31]. Lower values of RMSE, MAE and MAPE indicate a lower level of prediction error, whilst an R^2 value closer to one indicates the model's greater ability to explain the variation in the actual data.

To obtain more representative evaluation results and reduce the influence of variations caused by random weight initialisation, each model was trained 10 times with the same parameter configuration. The reported evaluation values are the average of all these training runs, thereby providing a more stable estimate of model performance compared to using a single training run. The evaluation results for the training set are presented in **Table 6**, whilst the evaluation results for the testing set are presented in **Table 6.**

Table 6. Comparison Of Cnn-Lstm And Xgboost-Lstm Performance On The Training Dataset

Model	Stock	RMSE ↓	MAE ↓	R^2 ↑	MAPE ↓
CNN-LSTM	BBCA	0.012273	0.009458	0.9868	0.0116
	BBRI	0.007313	0.005501	0.9836	0.0137
	BMRI	0.012103	0.008754	0.9857	0.0209
XGBoost-LSTM	BBCA	0.013679	0.010716	0.9835	0.0131
	BBRI	0.007146	0.005430	0.9843	0.0136
	BMRI	0.013036	0.009772	0.9849	0.0230

Table 6 shows that both the CNN-LSTM and XGBoost-LSTM models demonstrate excellent ability to learn historical share price patterns during the training process. This is evidenced by R^2 values that consistently exceed 0.98 across the entire dataset, indicating that both models are capable of explaining the majority of the variation in the training data. Furthermore, the relatively low RMSE, MAE and MAPE values indicate that both models produce a low level of prediction error during the learning process.

Although both models demonstrate high training performance, CNN-LSTM shows more consistent performance across most datasets. This model produced lower RMSE, MAE and MAPE values, as well as a higher R^2 value, on the BBCA and BMRI datasets, thus demonstrating a better ability to represent the historical patterns of these two shares. Conversely, on the BBRI dataset, XGBoost-LSTM achieved slightly better results than CNN-LSTM across all evaluation metrics. However, the difference in performance between the two models is relatively small, so both can still be said to have comparable learning capabilities on that dataset.

These findings indicate that both models are equally capable of learning the characteristics of historical data with a high degree of accuracy; however, the CNN-LSTM demonstrates greater consistency in performance as it delivers the best results on two out of the three datasets used. The variation in performance observed on the BBRI dataset also indicates that the characteristics of individual shares may influence the models' effectiveness during the training process.

Table 7. Comparison Of SNN-LSTM And XGBoost-LSTM Performance on the Test Dataset

Model	Stock	RMSE ↓	MAE ↓	R^2 ↑	MAPE ↓
CNN-LSTM	BBCA	0.018442	0.013796	0.8589	0.0165
	BBRI	0.010064	0.007449	0.7746	0.0203
	BMRI	0.015825	0.011839	0.7409	0.0259
XGBoost-LSTM	BBCA	0.021582	0.016518	0.8048	0.0198
	BBRI	0.010649	0.007842	0.7509	0.0213
	BMRI	0.018668	0.014093	0.6260	0.0309

Table 7 presents the evaluation results of the two models on the test data, which was used to assess their generalisation ability to data not involved in the training process. Unlike the training data, which only demonstrates the models' ability to learn historical patterns, the test data provides a more representative picture of the models'

ability to generate predictions on new data. Therefore, the evaluation results on the test data serve as the primary indicator in determining which model has the best predictive performance.

In contrast to the results on the training data, the CNN-LSTM consistently outperformed the XGBoost-LSTM across the entire test dataset. This model produced lower RMSE, MAE and MAPE values, alongside higher R^2 values for the BBCA, BBRI and BMRI shares. This consistency demonstrates that the CNN-LSTM is capable of producing more accurate predictions whilst maintaining a better relationship between predicted values and actual data compared to the XGBoost-LSTM.

The difference in performance between the two models is particularly evident in the BMRI dataset, where the CNN-LSTM achieved a significant improvement in the R^2 value compared to the XGBoost-LSTM, alongside reductions in the RMSE, MAE and MAPE values. Meanwhile, on the BBRI dataset, although the performance gap between the two models was relatively small, CNN-LSTM still maintained its advantage across all evaluation metrics. These findings indicate that the superiority of CNN-LSTM is not limited to specific datasets but is also maintained across the various datasets used in this study.

To complement the quantitative evaluation results presented in **Tables 6** and **7**, a visual analysis was carried out on the prediction results of both models using the BBCA share test data. The selection of the test data aims to evaluate the models' generalisation ability in predicting data that was not used during the training process, whilst BBCA shares were chosen as a representative sample because they demonstrate good predictive performance and have the widest price range amongst the datasets used. Through this visualisation, the degree of alignment between actual share prices and the prediction results from both the CNN-LSTM and XGBoost-LSTM models can be observed directly, making it easier to analyse the differences in the predictive characteristics of the two models.

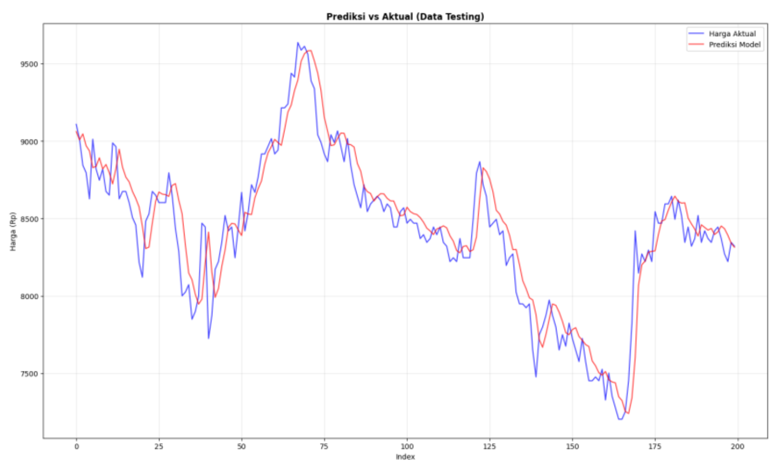

Figure 2. Visualisation of the CNN-LSTM results on the BBCA test data

Figure 2 shows a comparison between actual share prices and the predictions generated using the CNN-LSTM model on the BBCA share test data. Visually, the predicted curve is able to follow the pattern of actual share price movements with a good degree of accuracy throughout the observation period. The model successfully represents the main trends, both during phases of price decline and rise, thereby demonstrating its ability to retain the temporal patterns learnt from the training data when faced with data not previously used during the learning process.

Nevertheless, in certain segments characterised by rapid price changes, deviations were observed between the predicted curve and the actual data. These deviations primarily occurred around turning points, where the predicted curve tended to be smoother than the actual curve. This suggests that the model still has limitations in capturing very rapid price changes, but overall it remains capable of maintaining the direction and pattern of price movements effectively.

Overall, the visualisation results indicate that the CNN-LSTM demonstrates good generalisation ability on the BBCA test data. The closeness between the predicted curve and the actual curve indicates that the model is not only capable of learning the characteristics of the training data, but can also produce consistent predictions on data not used during the training process. This visual observation is consistent with the quantitative evaluation results in Table 6, which show relatively low RMSE and MAE values, as well as a high coefficient of determination (R^2) on the test data.

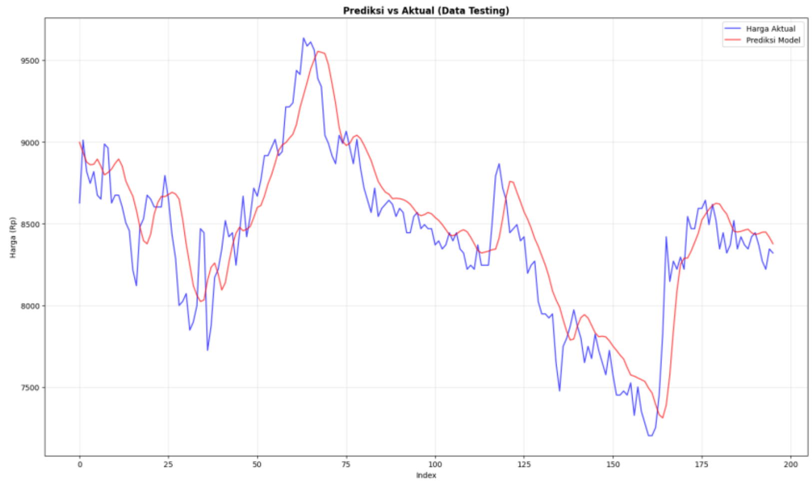

Figure 3. Visualisation of XGBoost-LSTM Results on the BBCA Test Data

Figure 3 shows a comparison between actual share prices and the predicted results using the XGBoost-LSTM model on the BBCA share test data. Visually, the predicted curve is able to follow the pattern of actual share price movements with a good degree of accuracy throughout the observation period. The model successfully captures the main trends, both during phases of rising and falling prices, thereby demonstrating its ability to retain the temporal patterns learnt from the training data when applied to the test data.

In certain segments experiencing rapid price changes, deviations between the predicted curve and the actual curve are evident. These deviations primarily occur around turning points, where the predicted curve tends to be smoother than the actual curve. Consequently, some price peaks and troughs are not fully represented by the model, although the general direction of price movement is still well captured. This characteristic indicates that XGBoost-LSTM is capable of capturing the main trends in the data, but has limitations in tracking more dynamic price fluctuations.

When compared with the CNN-LSTM prediction results in Figure 3, the XGBoost-LSTM prediction curve shows slightly greater deviations during certain periods with relatively sharp price changes. This visual observation is consistent with the quantitative evaluation results in Table 6, where XGBoost-LSTM produces slightly higher RMSE and MAE values, as well as a slightly lower coefficient of determination (R^2), compared to CNN-LSTM on the BBCA share test data. Nevertheless, the degree of alignment between the predicted and actual curves still indicates that the model possesses good generalisation ability in predicting BBCA share price movements.

Discussion

Comparative Analysis of the Performance of CNN-LSTM and XGBoost-LSTM

Based on the results of the quantitative evaluation of the training and test data, the CNN-LSTM model generally demonstrated better predictive performance than XGBoost-LSTM in the majority of test scenarios. This advantage is reflected in the lower RMSE, MAE and MAPE values, as well as the higher coefficient of determination (R^2) values across most datasets. These results indicate that the CNN-LSTM is capable of producing more accurate predictions whilst maintaining its ability to represent variations in share price movements during both the training and testing processes.

These findings are further supported by the visualisation results for the BBCA share test data, which show that the CNN-LSTM prediction curve follows the pattern of actual price movements more consistently than XGBoost-LSTM, particularly during periods characterised by trend reversals or relatively sharp price fluctuations. The alignment between the quantitative evaluation results and visual observations indicates that the CNN-LSTM model is not only capable of minimising prediction errors numerically but also of better preserving the dynamic characteristics of the time-series data.

Nevertheless, the research results also reveal an exception in the BBRI share training data, where XGBoost-LSTM performed slightly better than CNN-LSTM across all evaluation metrics. However, this advantage was not replicated in the test data, where the CNN-LSTM once again delivered superior performance. This discrepancy suggests that excellent performance on training data does not necessarily reflect a model's ability to make predictions on previously unobserved data.

Overall, the research results show that the CNN-LSTM exhibits more stable performance across various data conditions compared to the XGBoost-LSTM. This stability suggests that the combination of local feature extraction by the CNN and temporal dependency modelling by the LSTM is capable of producing a more effective data representation for capturing stock price movement patterns. A further discussion of the factors underlying the performance differences between the two models will be provided in the following subsection.

CNN-LSTM Excels in Predictive Performance

The advantages of the CNN-LSTM in this study can be explained by the characteristics of the model's architecture, which integrates the ability of Convolutional Neural Networks (CNNs) to extract local patterns with the ability of Long Short-Term Memory (LSTM) networks to model long-term temporal dependencies. In the initial stage, the convolutional layer performs feature extraction on stock price movement patterns within a narrow time window, enabling it to identify local characteristics such as short-term trends, changes in the direction of movement, and recurring price fluctuations. These extracted features are then passed to the LSTM layer to analyse more complex temporal relationships, ensuring the model considers not only price changes within a specific period but also their connection to historical patterns over a longer timeframe.

The combination of these two mechanisms enables the CNN-LSTM to construct a more comprehensive data representation compared to the XGBoost-LSTM approach. The feature extraction process by the CNN helps to reduce less relevant information before the data is processed by the LSTM, allowing the LSTM layer to focus more on learning temporal dependencies that genuinely contribute to the prediction process. With a more informative feature representation, the model becomes more adaptive in tracking dynamic changes in share price patterns. This aligns with the evaluation results of this study, in which the CNN-LSTM generally produced lower prediction error rates and better modelling performance on most datasets compared to the XGBoost-LSTM.

Conversely, in the XGBoost-LSTM model, the learning process begins with the XGBoost algorithm, which constructs a data representation using a gradient boosting decision tree approach. This approach is highly effective at capturing non-linear relationships and minimising errors in the training data through a step-by-step optimisation process applied to each decision tree [18]. However, these characteristics also make the model more sensitive to specific patterns present in the training data [3]. In time-series problems with high dynamics, such as share prices, decision-tree-based representations may not necessarily be able to preserve temporal information optimally before it is processed by the LSTM. Consequently, the model's ability to generalise to new data may be less stable compared to the CNN-LSTM.

The results of this study are consistent with various previous studies which state that the integration of CNNs and LSTMs can improve prediction accuracy on time series data because CNNs act as automatic feature extractors, whilst LSTMs are effective at learning long-term temporal relationships. The synergy between these two components produces a richer feature representation, enabling the model to adapt to complex changes in data patterns. Therefore, the superior performance of the CNN-LSTM in this study demonstrates that the combination of local-spatial feature extraction and temporal dependency learning is an effective approach for improving the accuracy of share price predictions.

Analysis of the Model's Generalisability, the BBRI Case Study, and the Limitations of the Research

Based on the research findings, the CNN-LSTM model demonstrated better generalisation ability than XGBoost-LSTM. This is evident from the performance of the CNN-LSTM, which consistently produced lower RMSE, MAE and MAPE values, as well as a higher R^2 value, across all test datasets, namely BBCA, BBRI and BMRI. This consistency indicates that the CNN-LSTM is not only capable of learning historical patterns during the training process but is also able to maintain accuracy when faced with new data patterns. Consequently, the CNN-LSTM strikes a better balance between learning from historical data and adapting to previously unobserved data.

Nevertheless, this study identified one interesting case in the BBRI share training data, where the XGBoost-LSTM model outperformed the CNN-LSTM model slightly across all evaluation metrics. These results suggest that the gradient boosting mechanism in XGBoost is highly effective at adapting to the characteristics of the training data through an iterative optimisation process that minimises prediction errors [28]. However, this advantage did not carry over to the test data, where the CNN-LSTM once again delivered the best performance. This suggests that excellent performance on the training data does not necessarily reflect the model's ability to make predictions on new data.

The difference in performance between the training and test data for XGBoost-LSTM indicates a tendency towards overfitting, a condition where the model adapts too closely to the characteristics of the training data, thereby reducing its generalisation ability [13]. Conversely, CNN-LSTM demonstrates more stable performance across both training and test data. These findings suggest that the evaluation of stock price prediction models should not only consider the error rate on the training data, but also the model's ability to maintain performance on previously unobserved data.

In the context of share price prediction, generalisation ability is a crucial aspect because price movements are influenced by various dynamic factors, such as economic conditions, market sentiment, government policy, and company fundamentals [18]. Consequently, historical share price patterns do not always repeat identically in subsequent periods. Consequently, models capable of maintaining accuracy on new data offer a higher level of reliability for application in forecasting processes and as a tool to support investment decision-making. Based on the results of this study, the CNN-LSTM demonstrates greater potential for use as a stock price forecasting model, as it is capable of delivering more consistent performance across various testing scenarios.

Nevertheless, this study still has several limitations. Firstly, the model evaluation was only conducted on three banking sector shares on the Indonesia Stock Exchange, namely BBCA, BBRI and BMRI; consequently, the study's results cannot yet be directly generalised to other industrial sectors or different market conditions. Secondly, this study utilised only historical closing price data as input variables without considering other information, such as opening prices, high and low prices, trading volume, technical indicators, or external factors such as macroeconomic conditions, market sentiment, and company fundamental information, which may also influence share price movements. Thirdly, this study only compares two hybrid architectures, namely CNN-LSTM and XGBoost-LSTM, and therefore has not explored other deep learning models that might deliver better performance.

Given these limitations, future research could expand the scope of the dataset by including more industrial sectors and different stock indices, integrate technical, fundamental and external factors as additional features, and evaluate more advanced deep learning architectures, such as Transformers, Temporal Fusion Transformers (TFTs) or attention-based models. It is hoped that these approaches will improve both the accuracy and the generalisation ability of the model in predicting share price movements under various market conditions.

4. Conclusion

The comparison carried out in this study utilised share price data for Indonesia's three largest banks: PT Bank Central Asia Tbk, PT Bank Rakyat Indonesia (Persero) Tbk, and PT Bank Mandiri (Persero) Tbk. In this comparison, the CNN-LSTM model generally outperformed the others across all evaluation metrics used, as the CNN algorithm is capable of learning short-term patterns, thereby complementing the strengths of the LSTM algorithm, which is capable of learning long-term patterns. However, the XGBoost-LSTM model demonstrated an advantage in one specific test: the test on the training data for PT Bank Rakyat Indonesia (Persero) Tbk shares. This occurred due to overfitting in the model.

The XGBoost-LSTM model exhibits a significant disparity in performance between the training and test data. Another factor contributing to the XGBoost-LSTM model's superior performance is the low volatility characteristic of the share price data for PT Bank Rakyat Indonesia (Persero) Tbk. The XGBoost algorithm, which performs iterative optimisation, is prone to overfitting and yields more stable results when prices do not exhibit high volatility. When predicting share prices, the CNN-LSTM model outperforms the XGBoost-LSTM model, owing to its higher accuracy and more stable generalisation ability on new data.

References:

[1] M. A. Khan, H. Ali, H. Shabbir, F. Noor, and M. D. Majid, "Impact of Macroeconomic Indicators on Stock Market Predictions : A Cross Country Analysis," vol. 08, no. 01, 2024, doi: https://doi.org/10.56979/801/2024.

[2] P. Meher and R. K. Mishra, "Assessing the influence of factors affecting stock market: an ISM approach," 2026, doi: https://doi.org/10.1108/QRFM-05-2024-0135.

[3] R. Islam and Z. Ikbal, "Impact Of Predictive Data Modeling On Business Decision-Making : A Review Of Studies Across Retail , Finance , And Logistics," vol. 02, no. 02, pp. 33–62, 2022, doi: https://doi.org/10.63125/8hfbkt70.

[4] A. Roosmawarni, D. Fatihudin, and N. Mauliddah, "Market Capitalization and Financial Performance : Evidence from Banking Listed Company in Indonesia," vol. 20, no. 2, pp. 124–136, 2022, doi: https://doi.org/10.31603/bisnisekonomi.v20i2.7835.

[5] G. Sonkavde, D. S. Dharrao, A. M. Bongale, and S. T. Deokate, "Forecasting Stock Market Prices Using Machine Learning and Deep Learning Models : A Systematic Review , Performance Analysis and Discussion of Implications," 2023, doi: https://doi.org/10.3390/ijfs11030094.

[6] B. Ramadevi, "Chaotic Time Series Forecasting Approaches Using Machine Learning Techniques: A Review," 2022, doi: https://doi.org/10.3390/sym14050955.

[7] J. Shah, D. Vaidya, and M. Shah, "Intelligent Systems with Applications A comprehensive review on multiple hybrid deep learning approaches for stock prediction," *Intell. Syst. with Appl.*, vol. 16, no. July, p. 200111, 2022, doi: https://doi.org/10.1016/j.iswa.2022.200111.

[8] H. Song, "Forecasting Stock Market Indices Using the Recurrent Neural Network Based Hybrid Models: CNN-LSTM, GRU-CNN, and Ensemble Models," 2023, doi: https://doi.org/10.3390/app13074644.

[9] J. Zhang, L. Ye, and Y. Lai, "Stock Price Prediction Using CNN-BiLSTM-Attention Model," pp. 1–18, 2023, doi: https://doi.org/10.3390/math11091985.

[10] Z. Shi, Y. Hu, G. Mo, and J. Wu, "Attention-based CNN-LSTM and XGBoost hybrid model for stock prediction," vol. 14, no. 8, pp. 1–7, 2021.

[11] A. Mandal, Y. Shao, and X. Liu, "Automatic Historical Stock Price Dataset Generation Using Python," pp. 1–6.

[12] J. M. Sahombu, "The Effect of Dividend Policy and Market Ratio on Stock Returns of Blue Chip Companies Listed on the Indonesia Stock," vol. 5, no. 06, pp. 1810–1817, 2025, doi: https://doi.org/10.58471/jms.v5i06.

[13] H. Zheng, J. Wu, R. Song, L. Guo, and Z. Xu, "Predicting Financial Enterprise Stocks and Economic Data Trends Using Machine Learning Time Series Analysis," 2024, doi: https://doi.org/10.20944/preprints202407.0895.v1.

[14] Y. Zhao and G. Yang, "Deep Learning-based Integrated Framework for stock price movement prediction," *Appl. Soft Comput.*, vol. 133, p. 109921, 2023, doi: https://doi.org/10.1016/j.asoc.2022.109921.

[15] D. Singh and B. Singh, "Feature wise normalization : An effective way of normalizing data ☆," *Pattern Recognit.*, vol. 122, p. 108307, 2022, doi: https://doi.org/10.1016/j.patcog.2021.108307.

[16] L. Liao, "An Empirical Study of the Impact of Hyperparameter Tuning and Model Optimization on the Performance Properties of," vol. 31, no. 3, 2022, doi: https://doi.org/10.1145/3506695.

[17] Z. Tao, Q. Xu, X. Liu, and J. Liu, "An integrated approach implementing sliding window and DTW distance for time series forecasting tasks," pp. 20614–20625, 2023.

[18] S. Zaheer *et al.*, "A Multi Parameter Forecasting for Stock Time Series Data Using LSTM and Deep Learning Model," *Mathematics*, vol. 11, no. 3, pp. 1–24, 2023, doi: https://doi.org/10.3390/math11030590.

[19] Z. Deng and O. Russakovsky, "Remember the Past : Distilling Datasets into Addressable Memories for Neural Networks," no. NeurIPS, pp. 1–14, 2022.

[20] Z. Zikai, "Enhancing Distributed Machine Learning through Data Shuffling : Techniques , Challenges , and Implications," vol. 03018, pp. 1–8, 2025.

[21] S. Albelali and M. Ahmed, "Hidden Leaks in Time Series Forecasting: How Data Leakage Affects LSTM Evaluation Across Configurations and Validation Strategies".

[22] K. Ullah *et al.*, "Short-Term Load Forecasting : A Comprehensive Review and Simulation Study With CNN-LSTM Hybrids Approach," *IEEE Access*, vol. 12, no. August, pp. 111858–111881, 2024, doi: https://doi.org/10.1109/ACCESS.2024.3440631.

[23] S. Lin, Y. Wang, H. Wei, X. Wang, and Z. Wang, "Hybrid Method for Oil Price Prediction Based on Feature Selection and XGBOOST-LSTM," 2025, doi: https://doi.org/10.3390/en18092246.

[24] M. Ahmadzadeh and S. M. Zahrai, "An integrated deep neural network model combining 1D CNN and LSTM for structural health monitoring utilizing multisensor time-series data," no. 3, 2025, doi: https://doi.org/10.1177/14759217241239041.

[25] J. Wu, K. Xu, X. Chen, S. Li, and J. Zhao, "Price graphs : Utilizing the structural information of financial time series for stock prediction," vol. 588, pp. 405–424, 2022, doi: https://doi.org/10.1016/j.ins.2021.12.089.

[26] X. Wang, H. Ren, and A. Wang, "Smish : A Novel Activation Function for Deep Learning Methods," 2022, doi: https://doi.org/10.3390/electronics11040540.

[27] A. Zafar *et al.*, "A Comparison of Pooling Methods for Convolutional Neural Networks," pp. 1–21, 2022, doi: https://doi.org/10.3390/app12178643.

[28] S. Abed and M. H. Alshayeji, "Hybrid XGBoost-LSTM model for oil well production forecasting using the volve field dataset," *Results Eng.*, vol. 30, no. March, p. 110128, 2026, doi: https://doi.org/10.1016/j.rineng.2026.110128.

[29] H. Oukhouya, H. Kadiri, K. El Himdi, and R. Guerbaz, "Forecasting International Stock Market Trends : XGBoost , LSTM , LSTM-XGBoost , And Backtesting XGBoost Models," vol. 12, no. January, pp. 200–209, 2024, doi: https://doi.org/10.19139/soic-2310-5070-1822.

[30] K. S. Cabuk *et al.*, "Chasing the objective upper eyelid symmetry formula," *Int. Ophthalmol.*, vol. 44, no. 1, pp. 1–9, 2024, doi: https://doi.org/10.1007/s10792-024-03157-y.

[31] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geosci. Model Dev.*, vol. 15, no. 14, pp. 5481–5487, 2022, doi: https://doi.org/10.5194/gmd-15-5481-2022.

[32] J. Gao, "R-Squared (R) – How much variation is explained ?," 2024, doi: https://doi.org/10.1177/26320843231186398.

[33] G. J. Exerc, S. Res, and E. Physiology, "What the mean absolute percentage error (MAPE) should adopt from Bland – Altman analyses," 2025, doi: https://doi.org/10.1007/s12662-025-01084-3.