SINTECHCOM: Science, Technology and Communication Journal
Vol. 6, No. 3, June 2026, pp. 207-218, DOI: 10.59190/stc.v6i3.377

Robust cryptocurrency price forecasting using a Bayesian-optimized CNN–LSTM hybrid model

Abdul Aziz Sulton, Fitri Insani*, Okfalisa, Lestari Handayani, Nazruddin Safaat Harahap
Department of Informatics Engineering, UIN Sultan Syarif Kasim Riau, Pekanbaru 28293, Indonesia

ABSTRACT

The rapid growth of cryptocurrency has caused the price movements of digital assets such as Bitcoin (BTC) and Ethereum (ETH) to become highly volatile and difficult to predict. This study aims to develop a cryptocurrency price prediction model using a hybrid convolutional neural network–long short-term memory (CNN-LSTM) architecture optimized with Bayesian Optimization. The data used in this study consisted of daily historical data for Bitcoin and Ethereum from January 1, 2018, to December 31, 2025, obtained from Yahoo Finance. The research stages included data preprocessing, normalization using Min-Max Scaling, sequence generation using the sliding window method (window sizes of 30, 60, and 90), CNN-LSTM model development, hyperparameter optimization using Bayesian Optimization (30, 50, and 100 trials), and evaluation using regression metrics including MSE, RMSE, MAE, MAPE, and R^2. The results showed that the hybrid CNN-LSTM model outperformed the standalone CNN and LSTM models, with RMSE reductions of 27% – 59% for BTC and 18% – 19% for ETH. For Bitcoin data, the best model was obtained using 30 trials with a window size of 30, achieving an RMSE of $2,588.33, MAE of $2,004.23, MAPE of 1.99%, and R^2 of 0.9468. Meanwhile, for Ethereum data, the best model was obtained using 50 trials with a window size of 60, achieving an RMSE of $138.56, MAE of $99.75, MAPE of 3.27%, and R^2 of 0.9737. These results indicate that the combination of CNN-LSTM and Bayesian Optimization is effective for predicting cryptocurrency prices with non-linear and volatile characteristics.

ARTICLE INFO

Article history:

Received Jun 2, 2026
Revised Jun 7, 2026
Accepted Jun 8, 2026

Keywords:

Bayesian Optimization
CNN-LSTM
Cryptocurrency Price
Deep Learning
Time Series Forecasting

* **Corresponding Author**
E-mail address: fitri.insani@uin-suska.ac.id

1. INTRODUCTION

The rapid development of financial technology has led to the emergence of digital investment instruments in the form of cryptocurrencies, where Bitcoin (BTC) and Ethereum (ETH) have become the assets with the largest market capitalization and trading volume worldwide. The selection of Bitcoin and Ethereum in this study is based on their dominance as the primary market leaders. Bitcoin serves as the pioneer of digital assets, while Ethereum is recognized as the largest smart contract platform, and together they represent the sentiment and liquidity of the majority of the global cryptocurrency market [1]. Unlike traditional stock markets, the cryptocurrency market operates continuously 24 hours a day, resulting in unique price dynamics [2]. These market characteristics create extreme price volatility, which is often influenced by investor sentiment and rapidly changing public speculation [3].

The main challenge in predicting cryptocurrency prices lies in the complex characteristics of the data. Cryptocurrency price data are time-series data that are non-linear and non-stationary [4]. This complexity makes the price movement patterns of Bitcoin and Ethereum difficult to predict accurately using conventional statistical approaches alone. Recent studies have shown that artificial neural network–based models are empirically more robust in handling complex digital asset data structures than conventional methods [5]. In this regard, Deep Learning (DL) approaches have been widely adopted because of their ability to model complex non-linear relationships without requiring strict statistical assumptions [6].

One of the most effective DL algorithms for time-series data is Long Short-Term Memory (LSTM). The primary advantage of LSTM lies in its ability to learn long-term dependencies and overcome memory limitations in sequential data [7]. However, the use of a standalone LSTM often encounters difficulties when dealing with highly noisy data. Without a robust feature extraction process, LSTM struggles to distinguish irrelevant short-term fluctuations from the main market trends [8]. To address this issue, Convolutional Neural Networks (CNNs) are integrated due to their advantages in automatic feature extraction and effective local pattern recognition [9]. CNNs are capable of reducing the complexity of input data through convolutional layers before further processing [10, 11].

The integration of a hybrid CNN-LSTM architecture has emerged as a comprehensive solution for handling market volatility. In this architecture, CNN acts as a feature extractor to filter noise and capture important features from price fluctuations [12]. The extracted features are then passed to the LSTM layer to analyze long-term sequential dependencies based on temporal sequences [11]. This integration effectively overcomes the limitations of standalone CNN models in temporal memory understanding and the limitations of standalone LSTM models in extracting complex spatial features [4, 13].

The advantages of the hybrid CNN–LSTM architecture have been validated by various empirical studies. [9] found that the combination of CNN and LSTM produced more stable performance with lower prediction errors compared to single models. In their study, the hybrid CNN-LSTM model achieved an MAE of 209.89 and an RMSE of 258.31, outperforming the standalone CNN model with an MAE of 215.98 and RMSE of 261.90, as well as the LSTM model with an MAE of 229.78 and RMSE of 297.97. [14] also reported that the hybrid model was effective in reducing the risk of overfitting while handling mixed cryptocurrency datasets more efficiently. [8] demonstrated the significance of this model in reducing prediction errors, where the implementation of the hybrid CNN-LSTM reduced the Root Mean Square Error (RMSE) to 144 on validation data, significantly lower than the pure LSTM model, which produced errors above 288 on highly volatile Bitcoin data. Furthermore, [5] showed that neural network-based models were able to outperform conventional models, achieving prediction results 9.77% better than the Heterogeneous Autoregressive Model (HAR) for short-term forecasting over a 7-day horizon.

However, the performance of Deep Learning models is highly influenced by the selection of hyperparameters, such as learning rate, number of CNN filters, and number of LSTM units. Determining hyperparameters manually or using conventional methods such as Grid Search and Random Search is considered inefficient because it requires high computational costs and does not necessarily produce optimal parameters [15]. Therefore, this study applies Bayesian Optimization (BO) to optimize the hyperparameters of the CNN-LSTM architecture. BO employs a probabilistic approach to identify the optimal combination of parameters more efficiently than conventional search methods [15].

Based on these considerations, this study focuses on developing Bitcoin and Ethereum price prediction models using a hybrid CNN-LSTM architecture optimized with Bayesian Optimization to generate accurate, stable, and efficient predictions for complex and non-linear cryptocurrency data.

2. RESEARCH METHODS

The research methodology consists of a sequence of systematically designed steps used as guidelines to achieve the research objectives. Figure 1 illustrates the research stages discussed in this chapter.

2.1. Data Collection

The data used in this study consist of daily historical time-series data of Bitcoin (BTC) and Ethereum (ETH) cryptocurrency assets. All data were obtained from the Yahoo Finance platform through the yfinance library interface in the Python programming language [7, 16]. The selection of this data source was based on its accuracy and completeness of historical price records, which have become standard in various digital asset prediction studies.

The dataset covers the observation period from January 1, 2018, to December 31, 2025, to provide broad market trend coverage [17]. The attributes collected in each daily record include the opening price (Open), highest price (High), lowest price (Low), closing price (Close), and trading volume (Volume).

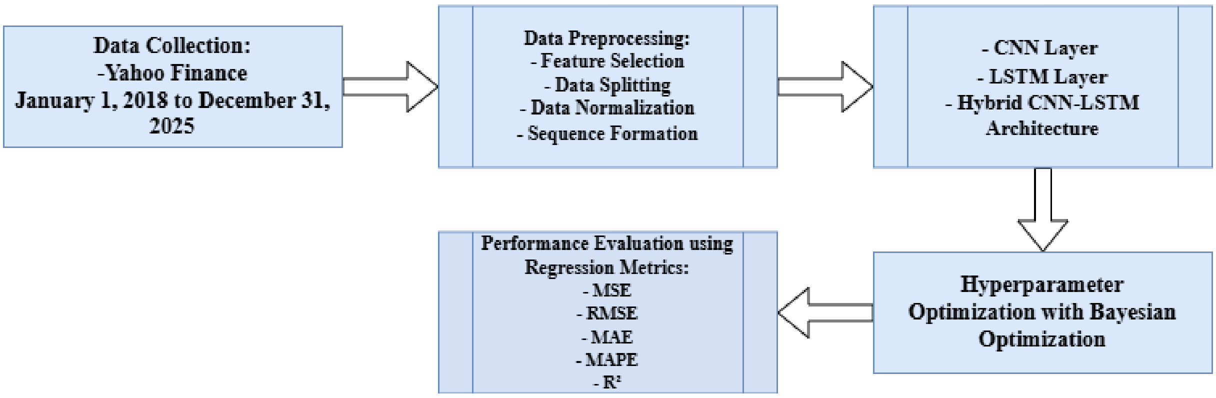

Figure 1. The flow of the research stages.

2.2. Data Preprocessing

The data preprocessing stage is a crucial step to ensure that the dataset is in optimal condition before the prediction model training process. This process aims to improve data quality, standardize feature scales, and prepare the data structure according to the requirements of the CNN-LSTM architecture [2, 18]. Data processing in this study was carried out through four main stages: feature selection, data splitting, data normalization, and sequence formation.

2.2.1. Feature Selection

Feature selection was performed to determine the input variables that significantly influence the prediction target. Based on previous studies [2, 9], the Open, High, Low, and Volume variables were selected as the main input features. Meanwhile, the Close variable was defined as the prediction target. The selection of these features was based on their strong correlation with closing prices and their common use in cryptocurrency price prediction literature.

2.2.2. Data Splitting

The dataset used consists of 2,922 daily records for both Bitcoin and Ethereum assets. Following the methodology applied in previous studies [19, 20], the dataset was chronologically divided into three subsets to support the training, validation, and evaluation processes. The data distribution proportions were 70% for training with 2,045 records, 15% for validation with 438 records, and 15% for testing with 439 records.

2.2.3. Data Normalization

To address significant scale differences among features such as Volume values being much larger than Open prices data normalization was performed [4, 21]. The normalization method used was Min-Max Scaling, which transforms all values into a range between 0 and 1. The equation used is as follows:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

The normalization process ensures that the deep learning model can converge faster during the training process.

2.2.4. Sequence Formation

The final stage of preprocessing is transforming the normalized data into a sequence structure so that it can be processed by the CNN-LSTM architecture. This model requires input in a three-dimensional format, namely (number of samples, time-steps, number of features), with four input features consisting of Open, High, Low, and Volume [4, 18]. The sequence generation process is performed using the sliding window method with window size variations of 30, 60, and 90 days. For example, in the 30-day window size, the model learns patterns from the previous 30 days of price data to predict the price on the 31st day.

2.3. Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of artificial neural network specifically designed to process data with a grid-like structure. Although CNN was initially widely used in image processing (2D data), it has also proven to be highly effective for processing one-dimensional time-series data through the 1D Convolutional Neural Network (Conv1D) architecture [9]. According to [22], the main advantage of CNN in time-series analysis is its ability to learn internal representations and automatically extract important features from raw data. This capability makes CNN highly effective for mapping sequential time-window inputs to prediction targets.

In the context of cryptocurrency price prediction, CNN is implemented as a Conv1D network to extract local features and hidden patterns from historical price data, such as price fluctuations, lowest prices, trading volume, and other related variables. The primary operation in CNN is the convolution process, in which a small kernel (or filter) is applied to the input data (time series) to generate a feature map. Mathematically, the convolution operation in CNN is formulated as follows:

$$C_i = f\left(\sum_{k=1}^{K} X_{i+k-1} \cdot W_k + b\right) \quad (2)$$

Description:

C_i : The result of the convolution operation at the i-th position (feature map);
F : A non-linear activation function (such as ReLU) applied after the summation operation;
K : The kernel size or the length of the filter window;
X_{i+k-1} : The input value within a specific time window being processed;
W_k : The weight of the kernel at the k-th position learned during the training process;
B : The bias value added to optimize the model.

In the hybrid architecture, CNN plays a crucial role as a powerful preprocessing layer before the data are passed to the memory layer (LSTM). By extracting rich and compact feature representations from financial time-series data, CNN reduces the input dimensionality while improving the quality of the information to be processed [14].

2.4. Long Short-Term Memory

Long Short-Term Memory (LSTM) is an extension of the Recurrent Neural Network (RNN) designed to overcome the vanishing gradient problem, enabling it to learn long-term dependencies in sequential data [23]. In cryptocurrency price prediction, LSTM is widely used because it can effectively recognize trend patterns and price fluctuations in non-linear and volatile time-series data [24].

LSTM consists of a memory cell structure controlled by three main gates, namely the forget gate, input gate, and output gate. These gates determine which information should be retained, updated, or discarded from the network memory [25]. Mathematically, the operations of LSTM are formulated in Equations (3) – (8).

$$Forget\ Gate: f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad (3)$$

$$Input\ Gate: i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (4)$$

$$Candidate\ \widetilde{Cell}\ State: C_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (5)$$

$$Output\ Gate\ Activation: o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (6)$$

$$Hidden\ State: h_t = o_t \odot \tanh(C_t) \quad (7)$$

$$Cell\ State: C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (8)$$

Description:

x_t	: Vector input on the step time t;
h_{t-1}	: Hidden state on the time step t-1;
C_{t-1}	: Cell state in step time t-1;
h_t	: Hidden state on the time step t;
C_t	: Cell state in step time t;
W_f, W_i, W_C, W_o	: Weight matrices are studied;
b_f, b_i, b_C, b_o	: Bias vectors (bias vectors) studied;
σ	: Sigmoid activation function $\sigma(z) = 1/(1+e^{-z})$;
tanh	: Hyperbolic tangent activation function;
$\odot$	: Element-by-element multiplication (Hadamard product);
$[h_{t-1}, x_t]$	: Vector concatenation.

In the hybrid architecture, LSTM plays an important role in processing temporal features that have previously been extracted by the CNN layer, resulting in more accurate price predictions that are adaptive to market dynamics [24, 25].

2.5. CNN-LSTM Hybrid

The CNN-LSTM hybrid model is an integration of two deep learning architectures designed to combine the strengths of each model in processing time-series data optimally. In this architecture, the Convolutional Neural Network (CNN) functions as an automatic feature extractor responsible for capturing local or spatial patterns, while the Long Short-Term Memory (LSTM) network is used to model long-term temporal dependencies [14]. This combination has been proven to be more effective in capturing complex cryptocurrency market dynamics compared to the use of a single model [18]. Visually, the integration flow of these components is illustrated in Figure 2.

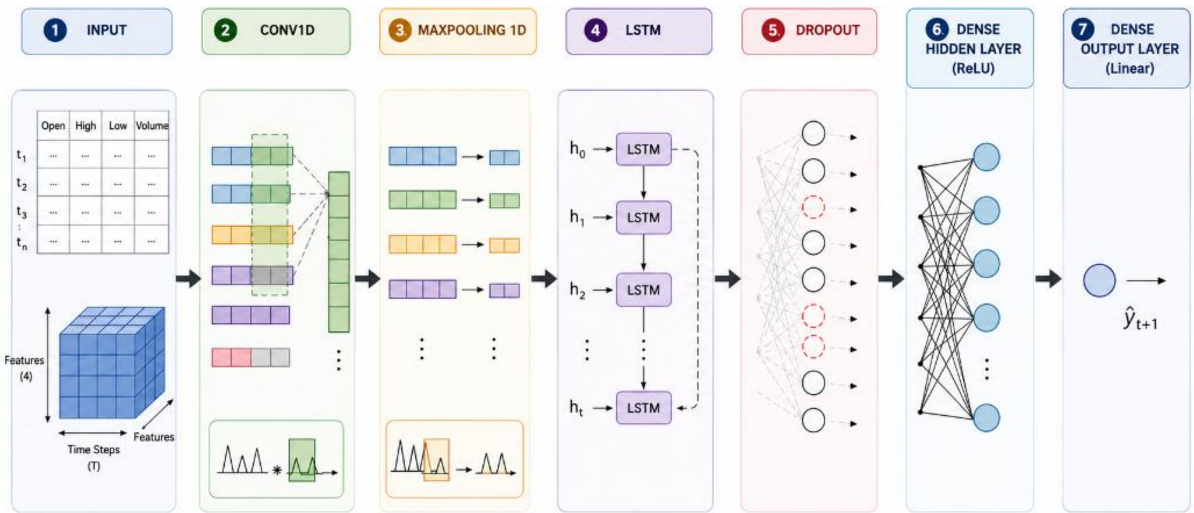

Figure 2. CNN-LSTM hybrid model architecture.

The process of applying the model in this study is systematically arranged as follows:

- Input Data: The input data consist of multivariate time-series data including the Open, High, Low, and Volume features. These data are transformed into three-dimensional sequences to meet the input requirements of the convolutional layer [3].

- 1D Convolutional Layer (Conv1D): This layer acts as the initial processing stage that detects local patterns and short-term trend fluctuations in daily price data. The convolution operation enables the model to filter noise and highlight important features from the raw data [14].
- MaxPooling 1D: After feature extraction, a pooling layer is used to reduce the dimensionality of the data. This process is essential for preserving the most dominant features while accelerating computation time and minimizing the risk of overfitting [18].
- Long Short-Term Memory (LSTM): The features extracted and summarized by the CNN layer are then passed to the LSTM layer. At this stage, the model learns chronological relationships and temporal dependencies that influence future price movements [3].
- Dropout Layer: As a regularization mechanism, a dropout layer is applied to randomly deactivate neurons during the training process. This forces the model to learn more robust data representations and improves its generalization capability on unseen data [14].
- Dense Hidden Layer: A fully connected layer with a ReLU activation function is used to perform non-linear mapping of the processed features, thereby enhancing the model's understanding of the data structure [18].
- Dense Output Layer: The final stage of the architecture is an output layer consisting of a single neuron with a linear activation function, which produces a single predicted value, namely the closing price (Close price) for the next period [18].

The synergy between CNN's capability in feature extraction and LSTM's capability in temporal memory enables the hybrid model to generate more accurate and stable predictions in handling the high volatility of digital assets [14, 18].

2.6. Hyperparameter Optimization with Bayesian Optimization

Hyperparameter optimization aimed to determine the best parameter combination to minimize prediction errors in volatile data [20]. This study applied Bayesian Optimization (BO) due to its efficiency in modeling the relationship between hyperparameters and the objective function through a probabilistic model, enabling the identification of optimal parameters with fewer evaluations compared to conventional methods [15, 25]. The optimization process used Root Mean Squared Error (RMSE) on the validation dataset as the objective function, while the optimized hyperparameters included the number of filters, kernel size, LSTM units, dense units, learning rate, and batch size. The hyperparameter search space is presented in Table 1, whereas the optimization process was conducted using variations in the number of iterations (30, 50, and 100 trials) and windowing scenarios (30, 60, and 90 days) [7, 18].

Table 1. Hyperparameter search spaces.

Hyperparameter	Symbols	Value Range
Number of Conv1D filters	Filters	16 – 128
Kernel size	kernel_size	2 – 7
Number of LSTM units	lstm_units	32 – 128
Number of dense units	dense_units	16 – 128
Learning eate	learning_rate	0.0001 – 0.01
Batch size	batch_size	16 – 64

2.7. Performance Evaluation Metrics

The evaluation of model performance in this study was conducted using regression metrics to measure the level of accuracy between the predicted values ((y_i)ˆ)and the actual values (y_i) [26]. The evaluation metrics used include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2). MSE and RMSE are used to measure the magnitude of prediction errors, MAE is used to calculate the average absolute error, MAPE measures the percentage of prediction errors, while R^2evaluates the model's ability to explain the variability of the actual data. The use of multiple evaluation metrics aims to provide a comprehensive assessment of the model's prediction error from different perspectives [17]. The evaluation metrics are defined as follows:

$$\mathrm{MSE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{y_i} - \widehat{y_\imath})^2 \tag{9}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_\imath})^2} \tag{10}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \widehat{y_\imath}| \tag{11}$$

$$\mathrm{MAPE} = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{y_i - \widehat{y_\imath}}{y_i}\right| \tag{12}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \widehat{y_\imath})^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{13}$$

Description:
N : The total number of data samples or observations;
y_i : Actual value (real market price) at the i-th observation;
$\hat{y}_i$: Predicted value generated by the model at the i-th observation;
$\bar{y}$: The average value of all actual data y;
i : Data sequence index.

3. RESULTS AND DISCUSSIONS

This section discusses hyperparameter optimization results and CNN-LSTM performance on Bitcoin (BTC) and Ethereum (ETH) data. The optimization used Bayesian Optimization with varying trial numbers (30, 50, and 100) and window sizes (30, 60, and 90). Model performance was evaluated using regression metrics, including MSE, RMSE, MAE, MAPE, and R^2 [26].

3.1. Hyperparameter Optimization With Bayesian Optimization

Bayesian optimization was used to find the optimal CNN-LSTM hyperparameters by minimizing validation RMSE. The optimization used varying trial numbers and window sizes [4, 7].

Table 2. Bayesian optimization results for BTC.

Parameters	Trials	Window 30	Window 60	Window 90
Filters	30	16	121	121
	50	128	121	66
	100	124	128	45
Kernel size	30	2	2	2
	50	2	2	2
	100	2	2	3
LSTM units	30	128	127	127
	50	119	127	71
	100	81	82	126
Dense units	30	16	85	85
	50	16	85	16
	100	16	16	46
Learning rate	30	0.001195	0.001672	0.001672
	50	0.000779	0.001672	0.000717
	100	0.002626	0.000812	0.000464
Batch size	30	16	16	16
	50	16	16	19
	100	28	16	16
Best validation RMSE	30	0.024328	0.026085	0.026056
	50	**0.024256**	0.025283	0.025486
	100	0.024396	0.025369	0.025270

3.1.1. Hyperparameter Optimization Results for BTC

The results presented in Table 2 demonstrate that the CNN-LSTM model trained on the BTC dataset achieved the lowest validation RMSE of 0.024256 using a configuration of 50 trials and a window size of 30. This finding suggests that a shorter window size was more effective in capturing the highly dynamic and volatile characteristics of cryptocurrency price movements compared to larger window sizes of 60 and 90. The optimal hyperparameter configuration consisted of 128 filters, a kernel size of 2, 119 LSTM units, 16 dense units, a learning rate of 0.000779, and a batch size of 16.

3.1.2. Hyperparameter Optimization Results for ETH

The results presented in Table 3 demonstrate that the CNN-LSTM model trained on the ETH dataset achieved the lowest validation RMSE of 0.020006 using a configuration of 50 trials and a window size of 30. This performance was better than that obtained for the BTC dataset, indicating that ETH price patterns tended to be more stable and easier for the model to predict. The optimal hyperparameter configuration consisted of 58 filters, a kernel size of 2, 107 LSTM units, 16 dense units, a learning rate of 0.000983, and a batch size of 16.

Table 3. Bayesian optimization results for ETH.

Parameters	Trials	Window 30	Window 60	Window 90
	30	128	128	128
Filters	50	58	16	16
	100	105	16	16
	30	2	2	2
Kernel size	50	2	2	2
	100	2	2	2
	30	100	128	99
LSTM units	50	107	128	128
	100	128	78	70
	30	128	16	35
Dense units	50	16	78	16
	100	16	76	16
	30	0.010000	0.003171	0.010000
Learning rate	50	0.000983	0.001565	0.000197
	100	0.002295	0.010000	0.001407
	30	38	39	64
Batch size	50	16	34	16
	100	16	31	16
	30	0.021261	0.021481	0.022443
Best validation RMSE	50	**0.020006**	0.020949	0.022064
	100	0.020699	0.020638	0.021274

3.2. Evaluate the CNN-LSTM Model with Regression Metrics

After obtaining the optimal hyperparameters, the CNN-LSTM model is evaluated on the test data (Testing Set) that had never been seen before. Evaluation used five regression metrics: MSE, RMSE, MAE, MAPE, and R^2 [26].

3.2.1. Test Results for BTC

The performance evaluation in Table 4 indicates that the CNN-LSTM model on BTC data achieved the best performance using the configuration of 30 trials and a window size of 30, with an MSE of \$6,699,458.31, RMSE of \$2,588.33, MAE of \$2,004.23, MAPE of 1.99%, and R^2 of 0.9468. The model achieved an accuracy level of 98.01%, indicating that it was highly accurate in predicting Bitcoin prices. Theoretically, the combination of CNN for local feature extraction and LSTM for temporal modeling successfully captured the highly volatile movement patterns of BTC prices.

Visually, Figure 3 presents a comparison between the actual and predicted Bitcoin prices on the best testing configuration (30 trials, window size 30). The predicted line (red) was observed to closely follow the actual line (blue), indicating that the model was able to capture BTC price movement patterns effectively. Several sharp price spikes were also successfully predicted with relatively small errors.

Table 4. Test results for BTC.

Trials	Window size	MSE (USD²)	RMSE (USD)	MAE (USD)	MAPE (%)	R^2
30	**30**	**6,699,458.31**	**2,588.33**	**2,004.23**	**1.99**	**0.9468**
	60	23,184,148.32	4,814.99	4,160.04	3.98	0.8270
	90	8,198,775.96	2,863.35	2,267.71	2.21	0.9425
50	30	11,098,729.07	3,331.48	2,710.99	2.63	0.9119
	60	10,281,398.78	3,206.46	2,596.74	2.52	0.9233
	90	26,895,981.97	5,186.13	4,556.07	4.40	0.8114
100	30	28,786,801.38	5,365.33	4,675.08	4.47	0.7714
	60	19,727,636.65	4,441.58	3,783.31	3.62	0.8528
	90	13,933,052.26	3,732.70	3,065.63	2.97	0.9023

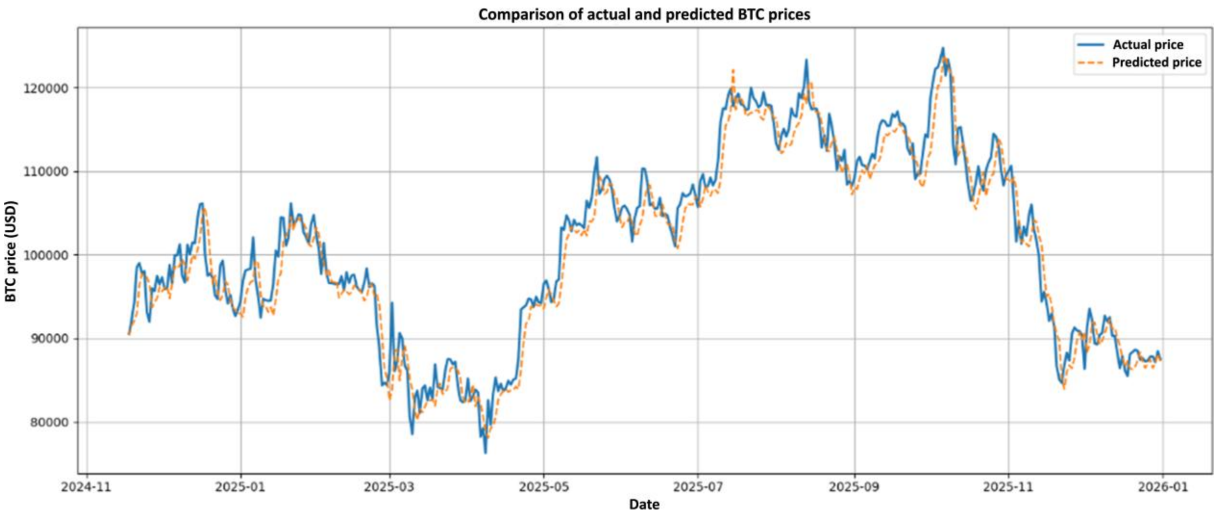

Figure 3. Comparison of actual prices and Bitcoin predictions on configuration 30 trials with window size 30.

3.2.2. Test Results for ETH

The performance evaluation in Table 5 indicates that the CNN-LSTM model on the ETH dataset achieved the best performance using a configuration of 50 trials and a window size of 60, resulting in an MSE of $19,198.85, an RMSE of $138.56, an MAE of $99.75, a MAPE of 3.27%, and an R^2value of 0.9737. The model achieved an accuracy level of 96.73%, indicating strong predictive capability for Ethereum price movements. Interestingly, the optimal configuration for ETH differed from that of BTC, suggesting that each cryptocurrency possesses distinct volatility patterns and temporal characteristics that influence model performance.

Table 5. Test result for ETH.

Trials	Window size	MSE (USD²)	RMSE (USD)	MAE (USD)	MAPE (%)	R^2
30	30	20,653.02	143.71	105.20	3.37	0.9706
	60	19,681.69	140.29	100.05	3.27	0.9730
	90	20,639.44	143.66	103.40	3.39	0.9736
50	30	20,054.05	141.61	102.66	3.32	0.9715
	60	**19,198.85**	**138.56**	**99.75**	**3.27**	**0.9737**
	90	22,343.49	149.48	107.23	3.58	0.9714
100	30	19,274.94	138.83	100.79	3.25	0.9726
	60	21,075.38	145.17	105.07	3.40	0.9711
	90	21,701.92	147.32	105.05	3.48	0.9722

Figure 4 presents a comparison between the actual and predicted Ethereum prices using the best configuration (50 trials, window size 60). Although ETH exhibited relatively high volatility, the CNN-LSTM model was still able to follow the price movement patterns effectively. The model predictions tended to lag slightly during extremely sharp price spikes; however, the overall major movement patterns were successfully captured with satisfactory accuracy.

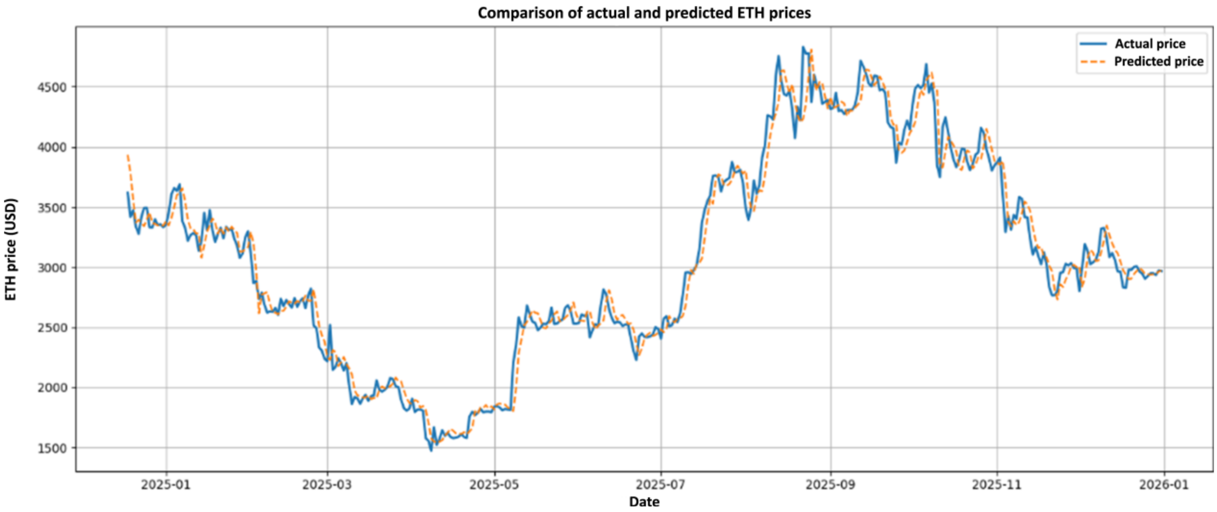

Figure 4. Ethereum actual price comparison and prediction on configuration 50 trials with window size 60.

3.3. Comparison with Single CNN and LSTM Models

To validate the superiority of the proposed hybrid architecture, additional experiments were conducted using standalone CNN and standalone LSTM models with the best hyperparameter configurations obtained from the BTC test (30 trials, window size 30) and ETH test (50 trials, window size 60). Table 6 presents the performance comparison of the three models for both BTC and ETH.

Table 6. Performance comparison of CNN, LSTM, and CNN-LSTM models for BTC.

Assets	Model	MSE (USD²)	RMSE (USD)	MAE (USD)	MAPE (%)	R²
BTC	Single CNN	12,651,788.31	3,556.94	2,948.92	2.88	0.8995
BTC	Single LSTM	38,939,702.11	6,240.17	5,501.06	5.25	0.6908
BTC	**Hybrid CNN-LSTM**	**6,699,458.31**	**2,588.33**	**2,004.23**	**1.99**	**0.9468**
ETH	Single CNN	28,617.21	169.17	126.55	4.12	0.9608
ETH	Single LSTM	26,596.89	163.09	118.78	3.83	0.9635
ETH	**Hybrid CNN-LSTM**	**19,198.85**	**138.56**	**99.75**	**3.27**	**0.9737**

Based on Table 6, the hybrid CNN-LSTM model consistently outperformed both standalone models across all evaluation metrics. For the BTC dataset, the hybrid model reduced the RMSE by 27% compared to the standalone CNN model and by 59% compared to the standalone LSTM model. For the ETH dataset, the hybrid model reduced the RMSE by approximately 18% – 19% compared to the standalone models. These results demonstrate that the integration of CNN as a feature extractor and LSTM as a temporal modeling component significantly improved prediction accuracy, particularly for highly volatile assets such as Bitcoin.

4. CONCLUSION

This study successfully developed a cryptocurrency price prediction model using a hybrid CNN-LSTM architecture optimized with Bayesian Optimization. The proposed model proved effective in handling the non-linear and volatile characteristics of Bitcoin (BTC) and Ethereum (ETH) data. Based on testing results, the hybrid CNN-LSTM model outperformed the standalone CNN and LSTM models, reducing RMSE by 27% – 59% for BTC and 18% – 19% for ETH. In this architecture, CNN functioned as a feature extractor to capture important features and short-term patterns while reducing noise in the data, whereas LSTM was utilized to learn long-term sequential dependencies from time-series data. In addition, Bayesian Optimization efficiently identified optimal hyperparameter combinations, leading to improved model performance.

The experimental results demonstrated that the CNN-LSTM model achieved excellent predictive performance for both cryptocurrency assets. For BTC, the best model was obtained using 30 trials with a window size of 30, achieving an accuracy of 98.01% with an MSE of \$6,699,458.31, RMSE of \$2,588.33, MAE of \$2,004.23, MAPE of 1.99%, and R^2 of 0.9468. Meanwhile, for ETH, the best model was achieved using 50 trials with a window size of 60, resulting in an accuracy of 96.73%, MSE of \$19,198.85, RMSE of \$138.56, MAE of \$99.75, MAPE of 3.27%, and R^2 of 0.9737. These findings indicate that the model was capable of capturing cryptocurrency price movement patterns with relatively low and stable prediction errors.

Overall, this study demonstrates that the combination of CNN-LSTM and Bayesian Optimization is an effective approach for cryptocurrency price prediction. The resulting model exhibited good generalization capability and has the potential to be implemented as a decision-support system for digital asset market analysis. Future research may improve the model by incorporating external variables such as technical indicators, market sentiment, and social media data to enhance predictive performance under extreme market conditions.

ACKNOWLEDGMENTS

The author would like to thank the Sultan Syarif Kasim State Islamic University Riau, Indonesia, for their academic support during the implementation of this research. Appreciation is also expressed to the supervisors, examiners, research colleagues, as well as anonymous reviewers and editorial teams for their constructive input and suggestions that have helped improve this article.

REFERENCES

[1] John, D. L., Binnewies, S., & Stantic, B. (2024). Cryptocurrency price prediction algorithms: A survey and future directions. *Forecasting*, **6**(3), 637–671.

[2] Wen, N. S. & Ling, L. S. (2023). Evaluation of cryptocurrency price prediction using LSTM and CNNs models. *JOIV: International Journal on Informatics Visualization*, **7**(3-2), 2016–2024.

[3] Afrinanda, R., Efrizoni, L., Agustin, W., & Rahmiati, R. (2023). Hybrid model for sentiment analysis of Bitcoin prices using deep learning algorithm. *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, **22**(2), 309–324.

[4] Badar, W., Ramzan, S., Raza, A., Fitriyani, N. L., Syafrudin, M., & Lee, S. W. (2025). Enhanced interpretable forecasting of cryptocurrency prices using autoencoder features and a hybrid CNN-LSTM model. *Mathematics*, **13**(12), 1908.

[5] Huang, Z. C., Sangiorgi, I., & Urquhart, A. (2024). Forecasting Bitcoin volatility using machine learning techniques. *Journal of International Financial Markets, Institutions and Money*, **97**, 102064.

[6] Sudriyanto, S., Faid, M., Malik, K., & Supriadi, A. (2024). Evaluasi Model Jaringan Saraf Tiruan Berbasis LSTM dalam Memprediksi Fluktuasi Harga Bitcoin. *Journal of Advanced Research in Informatics*, **2**(2), 15–22.

[7] Alvi, J., Nizam, K., Jafri, S. M., Rehan, M., & Ali, M. M. (2024). Cryptocurrency predictive analytics: A comparative study of LSTM, CNN, and GRU models. *Pakistan Business Review*, **26**(3), 255–287.

[8] Zahilah, R., Hajar, S., & Stiawan, D. (2023). CNN-LSTM Hybrid Model for Improving Bitcoin Price Prediction Results. *Applied Mathematics and Computational Intelligence (AMCI)*, **12**(4), 13–26.

[9] Li, Y. & Dai, W. (2020). Bitcoin price forecasting method based on CNN-LSTM hybrid neural network model. *The Journal of Engineering*, **2020**(13), 344–347.

[10] Wanto, A., Yuhandri, Y., & Okfalisa, O. (2023, August). Optimization accuracy of CNN model by utilizing CLAHE parameters in image classification problems. 2023 International Conference on Networking, Electrical Engineering, Computer Science, and Technology (IConNECT), 253–258.

[11] Mounika, S., Yadav, P. A., Yashaswi, T., Krishna, C. Y., & Reddy, V. K. (2021). Cryptocurrency Price prediction using CNN and LSTM models. *International Journal for Research in Applied Science and Engineering Technology*, **9**(3), 107–114

[12] Gao, M. (2025). Cryptocurrency Price Prediction Based on CNN-BiLSTM-AM Model. *Proceedings of the 1st International Conference on E-Commerce and Artificial Intelligence (ECAI 2024)*, 516–523.

[13] Septiana, R., Putri, A. R., Maharani, W. S., Gultom, G. D., Simanullang, E. D., Tania, K. D., Meiriza, A., & Rifai, A. (2025). Prediksi Harga Emas Indonesia Menggunakan Model CNN-LSTM. *Infomatek*, **27**(1), 131–138.

[14] Livieris, I. E., Kiriakidou, N., Stavroyiannis, S., & Pintelas, P. (2021). An advanced CNN-LSTM model for cryptocurrency forecasting. *Electronics*, **10**(3), 287.

[15] Kervanci, I. S., Akay, M. F., & Özceylan, E. (2024). Bitcoin price prediction using LSTM, GRU and hybrid LSTM-GRU with bayesian optimization, random search, and grid search for the next days. *Journal of Industrial and Management Optimization*, **20**(2), 570–588.

[16] Nair, M., Marie, M. I., & Abd-Elmegid, L. A. (2023). Prediction of cryptocurrency price using time series data and deep learning algorithms. *International Journal of Advanced Computer Science and Applications*, **14**(8).

[17] Hiskiawan, P., William, J., & Jansel, L. F. T. (2025). A Hybrid Data Science Framework for Forecasting Bitcoin Prices using Traditional and AI Models. *Journal of Applied Informatics and Computing*, **9**(5), 2089–2101.

[18] Regine, D. & Zabarnyi, A. (2025). Comparison of CNN-LSTM hybrid and CNN methods for Ethereum (ETH) to US dollar (USD) exchange rate prediction. *International Journal Artificial Intelligent and Informatics*, **3**(1), 16–23.

[19] Wanto, A., Yuhandri, Y., & Okfalisa, O. (2024). RetMobileNet: A New Deep Learning Approach for Multi-Class Eye Disease Identification. *Revue d'Intelligence Artificielle*, **38**(4).

[20] Tripathi, B. & Sharma, R. K. (2022). Modeling bitcoin prices using signal processing methods, bayesian optimization, and deep neural networks. *Computational Economics*, **1**.

[21] Ula, M., Ilhadi, V., & Sidek, Z. M. (2024). Comparing Long Short-Term Memory and Random Forest Accuracy for Bitcoin Price Forecasting. *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, **23**(2), 259–272.

[22] Hasan, S. H., Hasan, S. H., Ahmed, M. S., & Hasan, S. H. (2022). A Novel Cryptocurrency Prediction Method Using Optimum CNN. *Computers, Materials & Continua*, **71**(1).

[23] Seabe, P. L., Moutsinga, C. R. B., & Pindza, E. (2023). Forecasting cryptocurrency prices using LSTM, GRU, and bi-directional LSTM: a deep learning approach. *Fractal and Fractional*, **7**(2), 203.

[24] Pandey, P. & Sharma, G. (2025). Effective Price Prediction of Cryptocurrencies using CNN-Based Dual Directional Model. *Science & Technology Asia*, 201–219.

[25] Febriyanti, A. Y., Prasetya, D. A., & Trimono, T. (2025). Stock Price Prediction and Risk Estimation Using Hybrid CNN-LSTM and VaR-ECF. *Jurnal Teknik Informatika (Jutif)*, **6**(3), 1539–1554.

[26] Khoiri, S. A. & Wahid, A. (2024). Analisis kinerja algoritma machine learning dalam prediksi harga cryptocurrency. *JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia)*, **9**(2), 133–141.