Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

Forecasting Bitcoin Price Prediction with Long Short-Term Memory Networks: Implementation and Applications Using Streamlit

Muhammad Ihsan Fawzi*[1], Taufik Ganesha[2], Priandika Ratmadani Anugrah[3], Maulana Zhahran[4], Faris Akbar Abimanyu[5], Haryo Bimantoro[6]

[1,3,4,5,6]Informatics, Universitas Jenderal Soedirman, Indonesia
[2]Payago, Indonesia

Email: [1]ihsfwz@incomso.com

Received : Jul 23, 2025; Revised : Aug 19, 2025; Accepted : Aug 27, 2025; Published : Oct 16, 2025

Abstract

The rapid growth of cryptocurrency markets, particularly Bitcoin, has highlighted the need for accurate price prediction models to support informed decision-making. While existing studies primarily evaluate machine learning models for price forecasting, few have implemented these models in real-world applications. This paper addresses this gap by developing a Bitcoin price prediction system using Long Short-Term Memory (LSTM) networks, integrated into a user-friendly web-based application powered by Streamlit. The model forecasts Bitcoin prices at 5-minute, 1-hour, and 1-day intervals, demonstrating strong predictive performance. For the 5-minute interval, the model achieved a Mean Squared Error (MSE) of 53,479.86, Mean Absolute Error (MAE) of 150.58, Root Mean Squared Error (RMSE) of 231.26, and Mean Absolute Percentage Error (MAPE) of 0.144%. At the 1-hour interval, errors increased moderately with an MSE of 423,198.24, MAE of 499.93, RMSE of 650.54, and MAPE of 0.505%. For the 1-day interval, the model faced greater variability, reflected in an MSE of 3,089,699.07, MAE of 1,058.88, RMSE of 1,757.75, and MAPE of 2.027%. These results indicate that while predictive precision decreases over longer horizons, the model maintains strong performance across all timeframes. By embedding LSTM predictions into an interactive, real-time forecasting platform, this study demonstrates the practical integration of deep learning into complex financial systems. Beyond cryptocurrency, the approach highlights the potential of intelligent computational models to enhance decision-making processes in data-intensive domains, reinforcing the role of informatics in bridging advanced algorithms with usable technological solutions.

Keywords : *Bitcoin, Price Prediction, Long Short-Term Memory (LSTM), Machine Learning, Cryptocurrency, Web Application, Streamlit*

1. INTRODUCTION

The advancement and adoption of blockchain and cryptocurrency technology have become nearly unstoppable. As a rapidly emerging information technology, blockchain has sparked widespread discussions globally and is seen as a promising solution to existing challenges in the financial industry [1]. The financial markets have an attraction that attracts diverse investors to take part in the game of chance. In recent years, rapid advances in financial technology and artificial intelligence have driven its widespread use in various sectors, including methods such as *machine learning* and *deep learning*, among others [2]. In terms of interpretation and prediction of temporal data, *deep learning* algorithms have made significant advances [3].

The use of quantitative algorithms is increasingly popular in conventional financial markets, such as futures and stock trading, as well as in other financial markets, such as digital currency exchanges [4]. This systematic process involves executing pre-programmed automated trading instructions taking into account factors such as volume, price, and time, in order to maximize profits and minimize risks [5].

Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

In trading in the financial markets, these algorithms use complex mathematical models to improve the efficiency of trading decision-making [6]. By combining advanced algorithms, mathematical models, and human supervision, the system makes trading decisions on the exchange. In addition, according to the Financial Times, quantitative trading is increasingly being used to trade digital currencies (such as Bitcoin) in addition to traditional hedge funds and futures trading [4], [7].

Various quantitative strategies have benefited from the high volatility of digital currencies, which has resulted in increased rates of return. In general, quantitative strategies involve an in-depth analysis of market characteristics, market sentiment, and other diverse information in order to obtain maximum returns with relatively tolerable risks [8]. Based on research, according to the study, the Algorithmic Trading Market was valued at USD 11.66 billion in 2020 and is expected to grow to USD 26.27 billion by 2028, with a compound annual growth rate (CAGR) of 10.7 percent between 2021 and 2028.

In our belief, investment price movements are the result of a dynamic game involving various parties. Although the market is non-linear, we believe there are certain patterns that can be observed. The principle underlying technical analysis indicates that history is a recurring cycle. Many time series in financial markets are found to contain stylized facts of financial markets, and there are many stylized facts of financial market indices that have been discovered, as well as others [9]. Therefore, in this article, we will discuss how to forecast or predict the price of bitcoin.

The time-series *analysis method* is one of the quantitative approaches to forecasting that is now available in the market. The creation and inference of statistical models, as well as the most optimal time series forecasting, control, and filtering, are discussed in detail. Time series analysis can be divided into two forms: deterministic change analysis and random change analysis. The analysis of deterministic changes is more common than the two types. Trend change analysis, cyclical change analysis, and random change analysis are the types of deterministic change analysis, while random change analysis includes AR, MA, and ARMA models [7]. Several forms of random change analysis are discussed in this section.

Traditionally used time series forecasting models, such as ARIMA and the Vector Autoregressive Model (VAR), have some drawbacks because the investment market is a nonlinear and non-homogeneous system [7]. On the other hand, machine learning, thanks to its better performance, can conduct efficient exploration to obtain potential information from the market. In recent years, the application of machine learning in market forecasting has become increasingly important, and in general it has surpassed the forecasting accuracy of classical time series models [10].

Neural network-based price prediction models, such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), multilayer perceptrons (MLPs), and long short-term memory (LSTMs), have become the most widely used approaches in recent years. Abdul Dwiyanto's study uses RNN to predict stock prices to minimize the risk of losses in stock investments [11]. A study also reviewed RNN and LSTM models in hydrological time series prediction [12]. Then studies that predict the price of bitcoin using statistical methods such as Logistic Regression and Linear Discriminant Analysis achieved 66% accuracy for daily price predictions and deep learning models such as LSTM achieved 67.2% accuracy for 5-minute interval price predictions [13]. Ancy John et al., created a Hybrid Recurrent Neural Network (HyRNN) model, which combines the Bi-LSTM, GRU, and LSTM models, resulting in superior accuracy than statistical models and other conventional neural network models in predicting stock prices [14]. Quang Phung Duy used the ARIMA-GARCH model to predict the daily bitcoin price with a timeframe close to the latest bitcoin price data [15]. Patrick Jaquart et al., used machine learning models to predict the daily movements of the 100 largest cryptocurrency markets with binary predictions (up/down), as a result of which the LSTM and GRU models have the highest accuracy of any of the models trained and tested [16]. Huang et al., study shows that machine learning models, particularly LSTM and CNN-LSTM with Markov Transition Field (MTF), outperform traditional

Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 5, October 2025, Page. 2940-2961
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

econometric models like GARCH and HAR in forecasting Bitcoin volatility, especially in short-term horizons such as 7-day predictions [17]. Zidane et al., research evaluates the prediction performance of RNN variants—RNN, LSTM, and GRU—on Bitcoin value time-series data, revealing that LSTM achieves the most accurate results with RMSE of 0.012441 and MSE of 0.000155, albeit with higher computational time. The study also identifies the Tanh activation function as the most optimal choice for RNN models in predicting Bitcoin prices [18]. Esteban et al., study demonstrates that incorporating complex network features from Bitcoin blockchain transactions into machine learning models, such as CNN and LSTM, enhances cryptocurrency portfolio performance during downtrend periods, achieving capital retention improvements of 7.09% and 4.33%, respectively, and offering a promising approach to optimizing investment strategies [19]. Saha, study highlights the effectiveness of using GRU for cryptocurrency price prediction, achieving high accuracy for 1-month forecasts (90.26% for Bitcoin and 90.15% for Ethereum) and moderate accuracy for 6- and 12-month forecasts, providing valuable insights for investors to make informed decisions amidst the high volatility of cryptocurrency prices [20]. Saurabh Singh et al., study highlights the importance of feature selection in Bitcoin price prediction, demonstrating that using block detail features with an LSTM model yields the most accurate results, achieving a mean absolute error rate of 6.38%. This finding emphasizes the strong correlation between Bitcoin prices and block detail data, offering a promising direction for enhancing forecasting accuracy in cryptocurrency markets [21]. Jaquart et al., study examines Bitcoin market predictability across 1 to 60-minute horizons using various machine learning models. It finds that recurrent neural networks and gradient boosting classifiers perform best, with technical features being the most significant. While a long-short trading strategy can yield high returns before transaction costs, it results in negative returns when accounting for transaction costs due to short holding periods [22]. Yujin et al., paper addresses the gap in Bitcoin price prediction research by integrating On-Chain Data from Bitcoin's blockchain with LSTM (Long Short-Term Memory) models, a popular method for time-series forecasting. While previous studies have focused on macro-economic variables and investor sentiment, this approach leverages transaction data to enhance the accuracy of Bitcoin price predictions [23].

These studies are valuable because they establish that LSTM and related deep learning models are effective for financial time-series forecasting. However, their scope is generally limited to proving predictive accuracy through experiments, datasets, and benchmarks without extending the results into practical tools. By contrast, this research not only verifies the predictive performance of LSTM but also implements it into a practical, interactive Bitcoin price prediction application using Streamlit. This approach shifts the focus from theory to real-world usability, enabling traders, investors, and researchers to access forecasts directly through a web-based platform without requiring programming or data science expertise.

LSTM is chosen in this study because previous research has consistently demonstrated its strong performance in predicting stock and cryptocurrency prices [24]. Its ability to capture long-term dependencies in time-series data makes it particularly suitable for modeling the nonlinear and dynamic behavior of Bitcoin prices. Once the model has been trained and tested, it is deployed into an application using Streamlit, a Python-based framework that allows the rapid development of interactive web applications for data science and machine learning. Streamlit simplifies deployment by eliminating the need for manual setup with CSS, HTML, or JavaScript, allowing the focus to remain on model performance and usability [25].

Thus, the novelty of this research lies in bridging the gap between theoretical forecasting models and real-world application, demonstrating how a deep learning model (LSTM) can be operationalized into a usable decision-support tool. This practical integration reflects the growing role of informatics in transforming advanced computational models into accessible, user-oriented technologies.

Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

2. METHOD

The method used in this study, in predicting the price of bitcoin is using the LSTM model. First, bitcoin dataset will be taken, after that, the dataset will be pre-processed. The dataset is then trained using the LSTM model. The model is then saved for use in a bitcoin price prediction application created using the streamlit library.

2.1. Bitcoin Price History Dataset

In this study, bitcoin price predictions will be divided into three types of intervals, the first of which is 5-minute, 1-hour, and 1-day intervals. The prediction itself will be limited to the closing price prediction only for each interval. The period for the 5-minute interval is 5 days, the period for the 1-hour interval is 6 months, and the period for the 1-day interval is 10 years. The dataset itself is taken from the python library api yfinance. Yfinance is a python library API to obtain bitcoin price data based on specific intervals and periods [26].

2.2. *Data Pre-Processing*

The data preprocessing stage is the stage where the data that has been collected is processed first so that it is ready for use and according to what is needed [27]. The main task of data pre-processing is to eliminate data as well as overcome data noise so that the results [28].

2.2.1. Filling in Missing Data

Filling in missing data with valid values on the previous data. One common approach for handling missing values in time series datasets is imputation. Imputation involves filling in the missing values with estimated values based on the observed data. Numerous imputation methods have been proposed, each with its underlying assumptions and algorithms. Some commonly used imputation techniques include mean imputation, forward/backward filling, linear interpolation, and regression-based imputation [29].

2.2.2. Data Normalization

Data normalization, or feature scaling, is the process of forming a data structure so that it can eliminate most of the ambiguity [30]. Data normalization by scaling data is by converting values into values with a range of 0 to 1.

2.2.3. Segmentation

Segmentation is the grouping of data into data that can be used in a model or system. In this study, the length of the data group used the term "*lookback*". The lookback in this study varies based on the interval or period in the historical data. At 5-minute intervals, a half-day or 12-hour lookback is used with the consideration of working time and time at home or rest divided into 2 in one day. At 1-hour intervals use a 12-hour lookback. At 1-day intervals use a 7-day lookback.

2.3. LSTM

LSTM Neural Networks are a variation of recurrent neural networks (RNNs) designed to address the problem of vanishing gradients common to standard RNN [31]. Predictions will be processed by analyzing time-series data on closing prices. For the prediction itself, it will use the LSTM model. Long Short Term Memory (LSTM) network was created with the aim of solving the hidden layer problem. The key in LSTM design is to incorporate non-linear, data-dependent control into RNN cells [32] which can be trained to ensure that the gradient of the destination function by paying attention to the signal (quantity directly proportional to the parameter updates calculated during training by Gradient Descent)

Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

does not disappear. LSTM is used to overcome vanishing gradient or situations where the gradient value is 0 or close to 0 with the gate mechanism [33].

LSTM is another way to calculate hidden state. The memory in the LSTM is called the cells that take the input from the previous state (h_{t-1}) and current inputs (x_t). The set of cells decides what to store in memory and when to delete from memory. The LSTM combines the previous state, the current memory, and the input. LSTMs are very efficient for recording visible long-term dependencies.

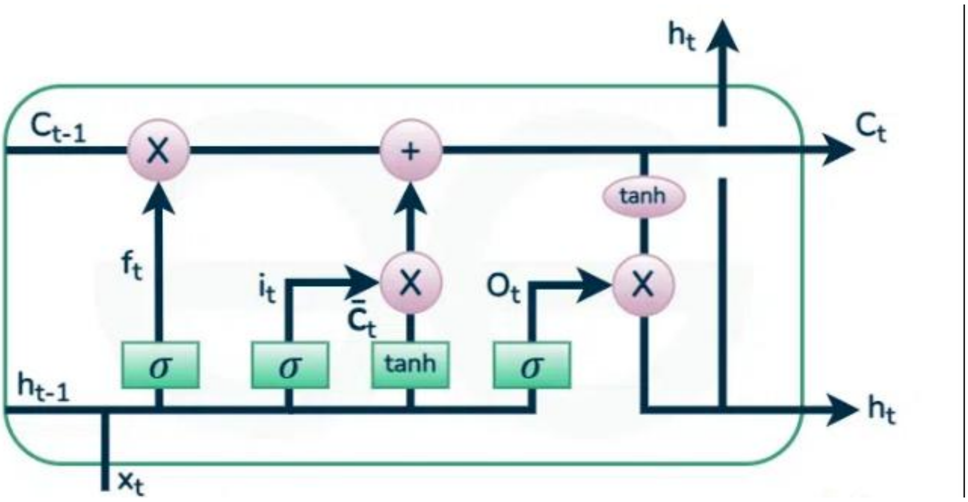

Figure 1. LSTM Architecture

Based on Figure 1, LSTM has 3 gates*, the* first gate is *the forget gate* to determine the information to be removed from the *cell* using *the sigmoid layer* as seen in Equation 8 with the activation function used is Relu. *The second* gate is the *input gate* from which the *sigmoid layer* will be updated and *the tanh layer* will be created as a vector of the updated value [34]. The third gate is *the output gate* to display the contents of the memory cell on the LSTM output.

LSTMs are treated as black boxes that are given *the current input* and *the previous hidden state* to calculate the next *hidden state*. The memory used in LSTM is called *cells* that take *input* from *the previous* state (h_{t-1}) and current inputs (x_t). LSTM is very suitable for learning *long-term dependencies* seen in Equation 1, Equation 2, Equation 3, Equation 4, Equation 5, and Equation 6.

$$\mathrm{i} = \sigma(x_t U^i + s_{t-1} W^i) \quad (1)$$

$$\mathrm{f} = \sigma(x_t U^f + s_{t-1} W^f) \quad (2)$$

$$\mathrm{o} = \sigma(x_t U^o + s_{t-1} W^o) \quad (3)$$

$$\mathrm{g} = \tanh(x_t U^g + s_{t-1} W^g) \quad (4)$$

$$\mathrm{c_t} = c_{t-1} \circ f + g \circ i \quad (5)$$

$$\mathrm{s_t} = \tanh(c_t) \circ o \quad (6)$$

Where is i, f, o*the gate input, forget, output,* g is a hidden *state candidate*, c_t is an internal memory unit which is a combination with the previous memory, c_{t-1} is the previous memory, s_t is the latest *hidden state*, then x_t is the input on each current step *t* and s_{t-1} is *the previous* hidden state.

Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

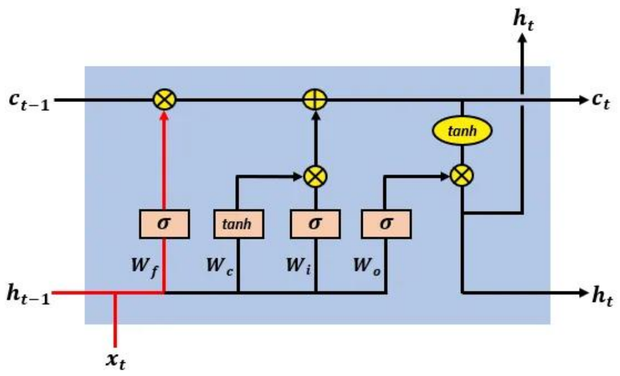

Figure 2. Forget Gate

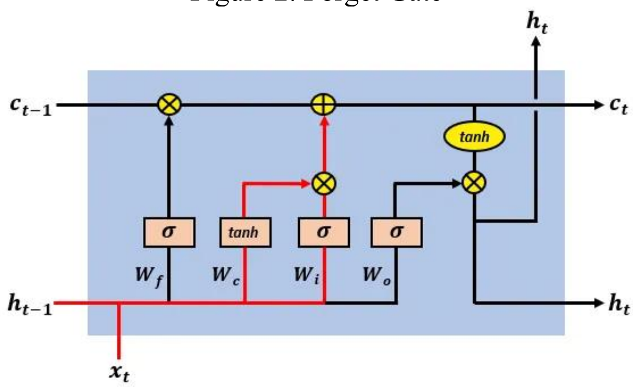

Figure 3. Input Gate

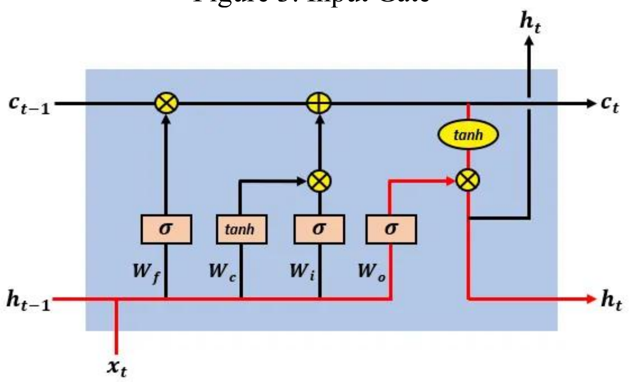

Figure 4. Output Gate

The forget gate as seen in Figure 2 is used to determine the information to be forgotten from *the cell* using *the sigmoid layer*, which can be seen in Equation 7 with the activation function used being Relu which is seen in Equation 8. The process can be seen in Figure 2.

$$\mathrm{f_t} = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{7}$$

$$ReLU\ (\mathrm{x}) = \max(0, x) \tag{8}$$

The input gate as seen in Figure 3 comes from the *sigmoid layer* to be updated and *the tanh layer* to be created as a vector of the updated value. It can be seen in Equations 9 and 10 and the process can be seen in Figure 3.

$$i_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_i) \tag{9}$$

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

$$C_t = \tanh\ (W_C \cdot [h_{t-1}, x_t] + b_c) \qquad (10)$$

Then the *cells* of Equation 8, Equation 9, and Equation 10 will be updated using Equation 11.

$$C_t = f_t * C_{t-1} + i_t * C_t \qquad (11)$$

The gate output as seen in Figure 4 will be calculated based on the cell renewal and *the sigmoid layer* looks like Equations 12 and 13 and the process can be seen in Figure 4.

$$o_t = \sigma(W_o\ \cdot [h_{t-1}, x_t] + b_o) \qquad (12)$$

$$h_t = o_t * \tanh(C_t) \qquad (13)$$

Where is the sigmoid activation function with a range (-1, 1) then tanh is a tangent activation function with a range (-1, 1) while $\sigma W_f, W_t, W_c, W_o$ is the weight matrix, for h_{t-1} is *the previous* hidden state and b_f, b_i, b_c, b_o is a bias vector.

2.4. Model Evaluation

To assess the performance of the proposed LSTM model, four widely used error metrics in time-series forecasting will be applied, namely Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). MSE measures the average squared difference between predicted and actual values, providing an overall indication of model accuracy, while MAE captures the average absolute deviation, showing how far the predictions deviate from the actual data on average. RMSE incorporates the scale of the data and penalizes larger errors more strongly than MAE, making it more sensitive to outliers. Meanwhile, MAPE expresses prediction error as a percentage, allowing easier interpretation of accuracy across different time intervals. The evaluation will be performed on predictions at three different intervals: 5-minute, 1-hour, and 1-day. These intervals are designed to reflect short-term, medium-term, and long-term forecasting performance, where the 5-minute interval aims to capture fine-grained price movements, the 1-hour interval represents intra-day trading patterns, and the 1-day interval evaluates the model’s ability to generalize over longer horizons.

2.5. Model Validation

To rigorously evaluate the predictive capability of the proposed model, validation was performed across multiple temporal resolutions, namely 5-minute, 1-hour, and 1-day intervals. The dataset was partitioned into training and validation subsets, where the training subset was utilized to capture historical price dynamics, while the validation subset served as an independent benchmark for assessing generalization performance. The evaluation compared three key components: the training output, which reflects the model’s capacity to approximate historical data; the validation data, which represents the ground truth; and the prediction output, which corresponds to the model’s estimated values on unseen data. This design enabled a comprehensive assessment of forecasting reliability at different horizons, with the 5-minute interval capturing high-frequency fluctuations, the 1-hour interval representing intra-day market dynamics, and the 1-day interval reflecting broader trend generalization. Through this multi-scale validation process, the robustness, accuracy, and adaptability of the model in addressing both short-term volatility and long-term trend forecasting were systematically examined.

2.6. Implementation

The system flow starts from the system initializing the API, model, and dashboard as seen in Figure 5. Then by default the system will display a historical data dashboard with an interval of 5

Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 5, October 2025, Page. 2940-2961
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

minutes, a 5-day period, and prediction results with a 1-time prediction period which means predictions in the next 5 minutes. Then the user can input the desired prediction interval and period. Once inputted, the system will display a dashboard based on the parameters entered by the user.

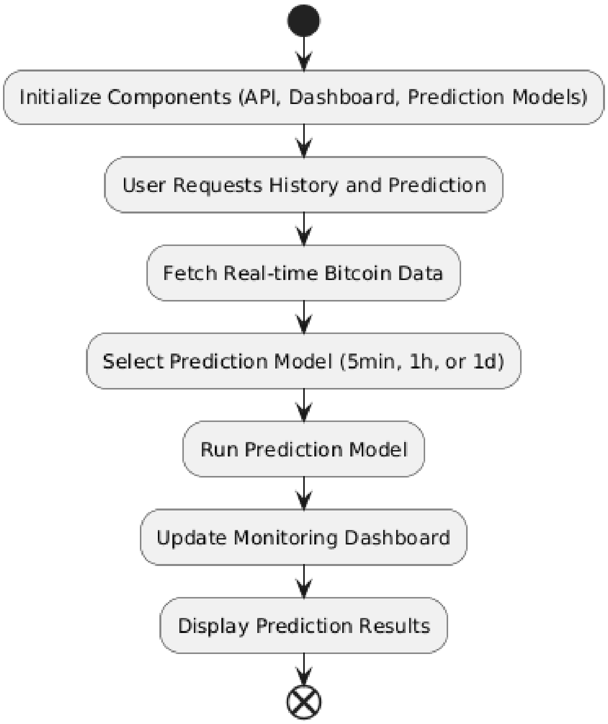

Figure 5. Flowchart

3. RESULT

The research results will be classified into four sub-sections. The first discusses about how bitcoin price history dataset looks like, pre-process the dataset, and then testing results of the model. Next is the evaluation of the model to determine its performance. Then, model validation is conducted to assess its accuracy in predicting Bitcoin prices using new test data. Finally, the implementation of the Bitcoin price prediction application using Streamlit is presented.

3.1. Bitcoin Price History Dataset

Table 1. Bitcoin Price History Dataset

Datetime	Open	Close	...
2025-03-16 00:15:00+00:00	84333.70312	84311.47656	...
2025-03-16 00:20:00+00:00	84297.99219	84295.97656	...
2025-03-20 03:00:00+00:00	86082.70312	86036.28125	...
2025-03-20 03:05:00+00:00	86040.14844	86131.28906	...
...	...	...	...

Table 1 displays a sample of the Bitcoin price history dataset obtained using the yfinance Python library. The dataset is organized by Datetime, which records the timestamp in UTC based on the selected interval. For each timestamp, the table includes the Open and Close prices of Bitcoin within that interval, along with other attributes (not shown here) such as High, Low, and Volume. However, in this study, only the Close attribute is utilized as the predictive target, since it represents the final transaction price within each interval and is widely regarded as the most stable and informative indicator for time series

Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

forecasting. As illustrated in the excerpt, at 00:15:00 on March 16, 2025, Bitcoin opened at 84,333.70 USD and closed slightly lower at 84,311.47 USD. Subsequent rows show the continuation of this time series, highlighting the short-term fluctuations across consecutive intervals. This tabular structure provides the sequential numerical data required to train and evaluate the prediction models for the 5-minute, 1-hour, and 1-day settings.

3.2. Data Pre-Processing

In the data preprocessing stage, the historical cryptocurrency dataset was refined to ensure that it was ready for modeling. Since the primary objective of this study was to predict price movements based on historical data, only the Close attribute was retained from the dataset, just like in Figure 6, as it most directly reflects the market's final valuation within a given time interval and is widely used as a reference in financial forecasting. This selection minimized noise from other attributes such as Open, High, and Low, which could introduce redundant information.

```
# Clean the data for to keep only the required columns
crypto_data_close = crypto_data_hist[["Close"]]
```

Figure 6. Clean Data

3.2.1. Filling in Missing Data

The next step was handling missing values. Missing entries within the Close series were resolved using the forward fill (ffill) method, where each missing record was replaced with the most recent valid observation. This approach preserved the continuity of the time series while avoiding unrealistic volatility, making it particularly suitable for financial data. An example of this imputation process is presented in Figure 7.

```
# Fill missing values
crypto_data_close = crypto_data_close.ffill()
```

Figure 7. Fill Missing Data

3.2.2. Data Normalization

Subsequently, normalization was applied to rescale the Close values to a range between [0, 1] using the Min–Max scaling technique. This transformation ensured that all values contributed proportionally during training, improving model convergence and stability. The results of this normalization process are shown in Figure 3.

```
# Scale into range 0..1
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(crypto_data_close)
```

Figure 8. Scale Data

3.2.3. Segmentation

Finally, the normalized series was converted into supervised learning sequences using the sliding window, or lookback, method. In this step, a sequence of consecutive Close values of fixed length was extracted as input (x), and the immediate subsequent value was designated as the output (y).

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

This segmentation enabled the model to capture temporal dependencies within the data. The structure of this input–output transformation is depicted in Figure 9.

```
lookback = int(24*60/5/2)
x_data = []
y_data = []
for i in range(lookback,len(scaled_data)):
    x_data.append(scaled_data[i-lookback:i,0])
    y_data.append(scaled_data[i,0])

# Converting the x and y values to numpy arrays
x_data, y_data = np.array(x_data), np.array(y_data)

# Reshaping x and y data to make the calculations easier
x_data = np.reshape(x_data, (x_data.shape[0],x_data.shape[1],1))
y_data = np.reshape(y_data, (y_data.shape[0],1))
```

Figure 9. Lookback example for 5-minute interval

3.3. LSTM

Table 2. Hyperparameters for LSTM Model

Hyperparameter	Value	Description
Input Shape	(144, 1)	Sequence length of 144 timesteps with 1 feature (Close price) per timestep.
Number of LSTM Units	50 per layer	Number of neurons in each LSTM layer for sequence learning.
Number of Layers	2	Depth of the LSTM stack to capture temporal dependencies.
Dense Units	1	Final output unit for predicting a single value (next closing price).
Optimizer	Adam	Adaptive optimizer for efficient gradient descent.
Learning Rate	Default (0.001)	Step size used by Adam to minimize loss.
Batch Size	32	Number of training samples used in one iteration.
Number of Epochs	200	Total number of passes through the dataset during training.
Loss Function	Mean Squared Error (MSE)	Measures average squared difference between predicted and actual values.
Activation Function	Tanh (LSTM) / Linear (Dense)	Non-linear transformation inside LSTM and linear output for regression tasks.

The Long Short-Term Memory (LSTM) model employed in this research was designed to effectively capture temporal dependencies in stock price data by leveraging a multi-layered recurrent architecture. As presented in Table 2, the model is composed of two stacked LSTM layers, each configured with 50 hidden units, enabling it to learn both short- and long-term sequential patterns from the input time series. The inclusion of two layers increases the representational capacity of the model while maintaining computational feasibility. The model uses a Dense output layer with a single neuron to generate the final prediction value, corresponding to the next time step in the sequence. Training was conducted with a batch size of 32 and extended over 200 epochs, providing the network with sufficient iterations to refine its internal weights. The Adam optimizer was selected due to its adaptive learning rate mechanism, which enhances convergence efficiency in complex optimization landscapes, while the

Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

mean squared error (MSE) loss function was applied to measure prediction accuracy. Overall, this configuration balances model complexity and computational efficiency, ensuring robust learning performance without excessive overfitting.

3.4. Model Evaluation

The following are on Table 3 is the results of the model evaluation based on its time interval.

Table 3. Model Evaluation

Interval	MSE	MAE	RMSE	MAPE
5-minute	53,479.86	150.58	231.26	0.00144
1-hour	423,198.24	499.93	650.54	0.00505
1-day	3,089,699.07	1,058.88	1,757.75	0.02027

3.4.1. 5-minute Interval Evaluation

The evaluation for the 5-minute interval in Table 3 shows that the Mean Squared Error (MSE) is 53,479.86, indicating the average squared error between the predicted and actual values is relatively low, suggesting good performance for fine-grained predictions. The Mean Absolute Error (MAE) of 150.58 reflects that, on average, the model's predictions deviate from the actual values by 150.58 units. The Root Mean Squared Error (RMSE) is 231.26, which accounts for the scale of the data and is more sensitive to larger errors compared to MAE. Finally, the Mean Absolute Percentage Error (MAPE) is an extremely small 0.00144 (0.144%), highlighting the model's high accuracy in predicting short-term values.

3.4.2. 1-hour Evaluation

The evaluation for the 1-hour interval in Table 3 reveals a Mean Squared Error (MSE) of 423,198.24, indicating a higher error compared to the 5-minute interval, likely due to greater variability over longer time periods. The Mean Absolute Error (MAE) is 499.93, showing larger average prediction errors on an hourly basis. The Root Mean Squared Error (RMSE) of 650.54, being higher than MAE, highlights the increasing impact of larger errors in the model's hourly predictions. Meanwhile, the Mean Absolute Percentage Error (MAPE) is 0.00505 (0.505%), which, although still small, reflects slightly reduced accuracy at this time scale compared to shorter intervals.

3.4.3. 1-day Evaluation

The evaluation for the 1-day interval in Table 3 indicates a significant increase in errors compared to shorter timeframes. The Mean Squared Error (MSE) is 3,089,699.07, reflecting the challenge of capturing day-to-day variability in predictions. The Mean Absolute Error (MAE) is 1,058.88, demonstrating a substantial average deviation between predicted and actual values. The Root Mean Squared Error (RMSE) is 1,757.75, further emphasizing the impact of larger individual errors at this scale. The Mean Absolute Percentage Error (MAPE) stands at 0.02027 (2.027%), which, while still relatively low, highlights the model's diminished precision in long-term forecasts compared to shorter intervals.

3.5. Model Validation

The following are the results of the model validation based on its time interval and its description.

Graph Description :

a) Train (blue line): Practice data that reflects the historical pattern of the bitcoin price
b) Valid (orange line): The validation data used to test the accuracy of the model after training.
c) Predictions (green line): The prediction results generated by the model based on validation data.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871

Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

3.5.1. 5-minute Interval Model Validation

Analysis on Figure 10 :

a) The Predictions (green line) closely follow the Valid (orange line), particularly during stable trends.
b) During a steep upward trend at the end of the timeline, the model successfully captures the price surge, although slight deviations are observed.
c) The shorter 5-minute interval allows the model to respond quickly to price fluctuations, making it suitable for real-time forecasting.

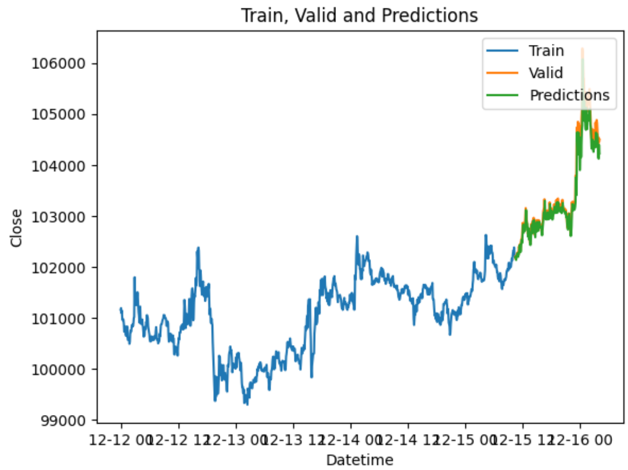

Figure 10. Training and Prediction Results for 5-minute Interval

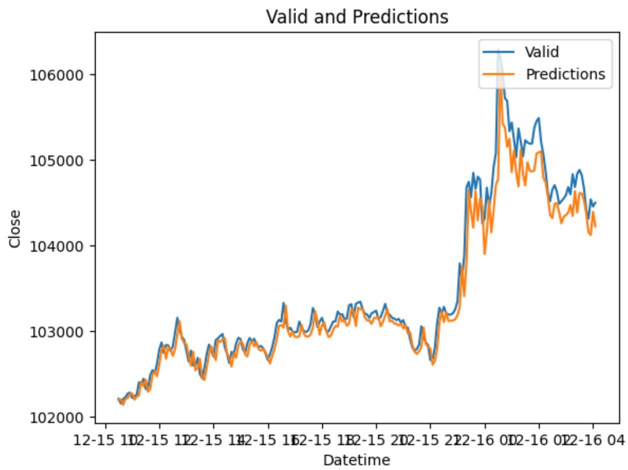

Figure 11. Validation of Prediction Results for 5-minute Interval

Analysis on Figure 11:

a) The Predictions align closely with the actual validation data, indicating strong predictive performance.
b) The model captures both upward and downward movements accurately, with minor deviations during sharp price peaks and troughs.
c) Notably, the model maintains its accuracy even during short-term volatility, demonstrating its ability to generalize well in dynamic conditions.

P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

3.5.2. 1-Hour Interval Model Validation

Analysis on Figure 12:

a) The model effectively follows the overall **upward trend** of Bitcoin prices. The **Predictions** (green line) align closely with the validation data (orange line), showing the model's ability to generalize patterns from the training phase.
b) Although minor deviations exist in some areas, the prediction remains highly accurate, especially during steady upward price movements.
c) The model successfully captures the price surge in the latter part of the timeline, demonstrating its robustness over the 1-hour interval.

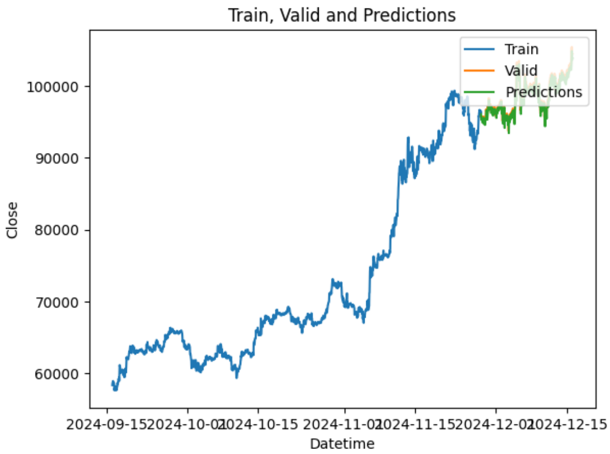

Figure 12. Training and Prediction Results for 1-hour Interval

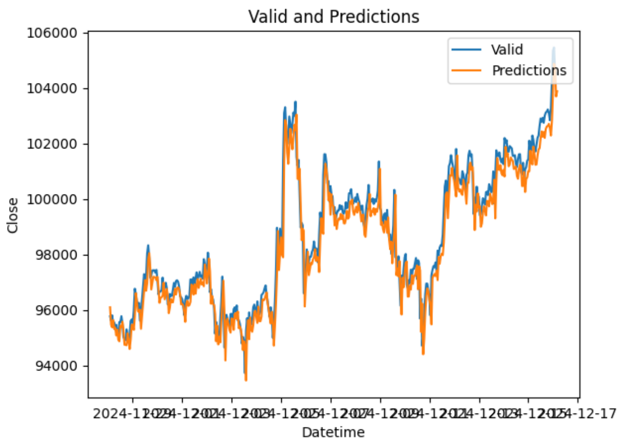

Figure 13. Validation of Prediction Results for 1-hour Interval

Analysis on Figure 13:

a) The Predictions (orange line) closely follow the Valid (blue line), demonstrating strong model accuracy.
b) The model successfully captures the upward and downward movements in price, with only minor deviations during sharp peaks and troughs.
c) This result indicates that the model generalizes well for 1-hour intervals, handling volatility with minimal error.

Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

3.5.3. 1-Day Interval Model Validation

Analysis on Figure 14:

a) The Predictions (green line) closely follow the Valid (orange line), particularly during the upward trends observed in the latter part of the graph.
b) 1-day interval allows the model to focus on macro-level trends, reducing noise caused by short-term volatility.
c) The model captures long-term trends effectively, especially during significant price surges, as seen after 2023.

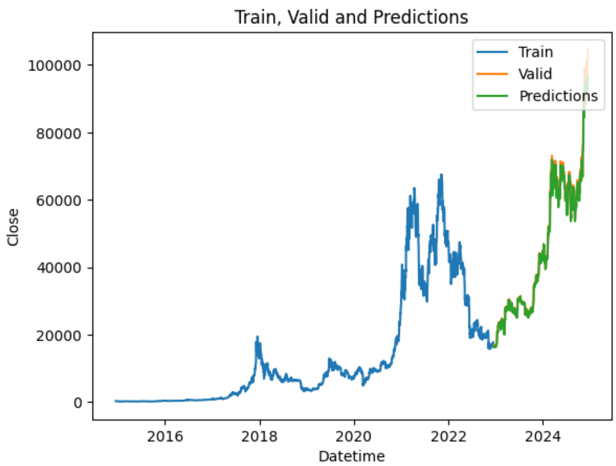

Figure 14. Training and Prediction Results for 1-day Interval

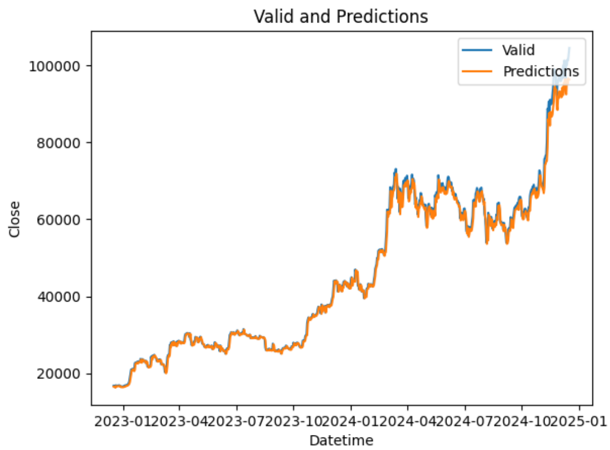

Figure 15. Validation of Prediction Results for 1-day Interval

Analysis on Figure 15:

a) The Predictions (orange line) closely overlap with the Valid (blue line), demonstrating a high level of accuracy.
b) Model captures upward and downward trends in price movements over extended periods with very few deviations.
c) Particularly during the rapid price increase at the end of the graph (2024-2025), the model maintains excellent performance, accurately predicting the trend.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871

Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

3.6. Implementation

The following is a view of the bitcoin price prediction app using a trained and tested model. This app uses the streamlit library.

3.6.1. Dashboard

The Dashboard on Figure 16 consists of two primary sections: the sidebar and the main display area. On the sidebar, users can interact with input features, such as selecting the prediction interval and defining the desired prediction period. For example, a 5-minute interval allows users to observe short-term price movements, while longer intervals provide insight into broader trends. These customizable options enable flexibility based on specific forecasting needs, making the dashboard suitable for both quick real-time analysis and long-term strategic decisions.

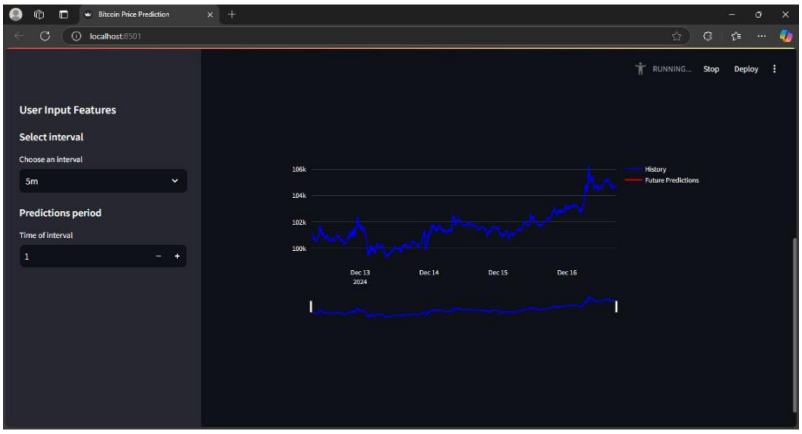

Figure 16. Bitcoin Price Prediction Dashboard Application

The main display area showcases the predicted Bitcoin price with precise details, including the current prediction, the exact time of the forecast, and the current system time, which helps users monitor updates effectively. A small table further displays the prediction results, presenting the date and corresponding predicted price in an organized format. Complementing this, the second section of the dashboard includes a dynamic graph that visualizes historical price data alongside future predictions. The blue line represents past Bitcoin price trends, while the red line shows the forecasted prices, allowing users to observe patterns and assess the model's accuracy visually.

3.6.2. 5-minute Prediction Interval

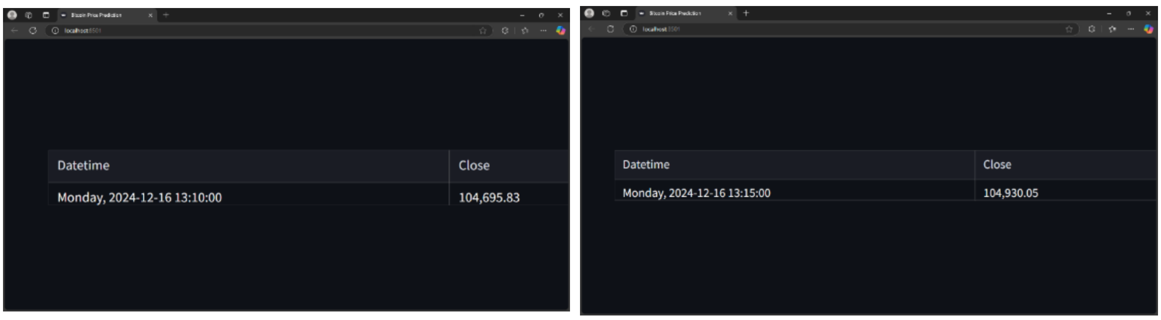

Figure 17. Application 5-minutes Interval Price Prediction Results

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871

Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

Figure 18. 5-minutes Interval Real Time Prices

Table 4. Compare Prediction with Real Price (5-minute Interval)

Datetime	Prediction	Real Price	Different
2024-12-16 13:10	104,695.83	105,155.66	459.83
2024-12-16 13:15	104,930.05	105,176.21	246.16
2024-12-16 13:20	104,858.61	105,137.14	278.53
2024-12-16 13:25	104,858.66	105,062.15	203.49
2024-12-16 13:30	104,728.79	105,209.26	480.47

The Table 4 compares the predicted prices on Figure 17 and the actual real-time prices on Figure 18 for the 5-minute interval price prediction application, highlighting the differences between them. Each row corresponds to a specific timestamp, with the Prediction column showing the price forecasted by the model and the Real Price column displaying the actual observed price. The Difference column calculates the absolute deviation between the predicted and real prices, showing how accurately the model predicts the prices at each 5-minute interval.

For example, at 13:10, the predicted price was 104,695.83, while the real price was 105,155.66, resulting in a difference of 459.83. Similarly, for other intervals, the differences range from 203.49 to 480.47, reflecting the variability in prediction accuracy across the different times. This table provides a clear view of how well the model performs in predicting prices in the short term, with the differences indicating the level of prediction error at each timestamp.

3.6.3. 1-Hour Prediction Interval

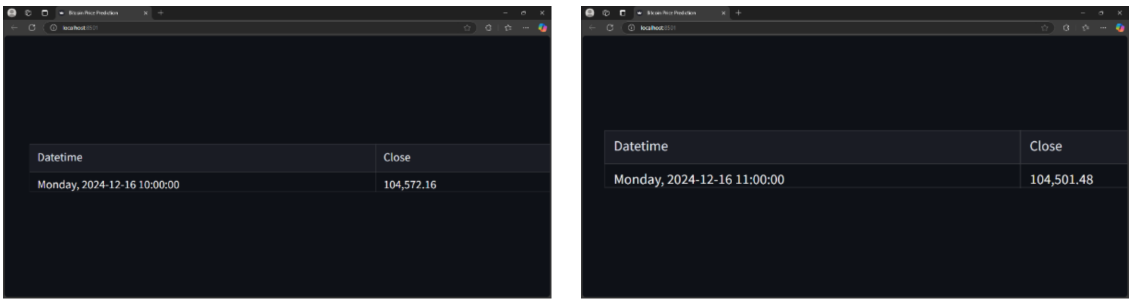

Figure 19. Application 1-hour Interval Price Prediction Results

Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 5, October 2025, Page. 2940-2961
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

Figure 20. 1-hour Interval Real Time Prices

Table 5. Compare Prediction with Real Price (1-hour Interval)

Datetime	Prediction	Real Price	Different
2024-12-16 10:00	104,572.16	104,546.89	-25.27
2024-12-16 11:00	104,501.48	104,648.95	147.47
2024-12-16 12:00	104,481.78	104,993.56	511.78
2024-12-16 13:00	104,428.23	105,113.68	685.45
2024-12-16 14:00	104,340.55	104,955.99	615.44

Table 5 compares the predicted prices on Figure 19 with the real prices on Figure 20 for 1-hour intervals, showing the difference between the two. Each row represents a specific hour, with the Prediction column displaying the price forecasted by the model and the Real Price column showing the actual price observed at that time. The Difference column calculates the difference between the predicted and real prices, indicating the accuracy of the model's predictions for each interval.

For example, at 10:00, the predicted price was 104,572.16, and the real price was 104,546.89, resulting in a small difference of -25.27. At 11:00, the predicted price was 104,501.48, while the real price was 104,648.95, resulting in a positive difference of 147.47. As the intervals progress, the differences increase, with the largest being 685.45 at 13:00. This indicates that the model's accuracy decreases slightly as the time interval grows, with larger discrepancies at later hours.

3.6.4. 1-Day Prediction Interval

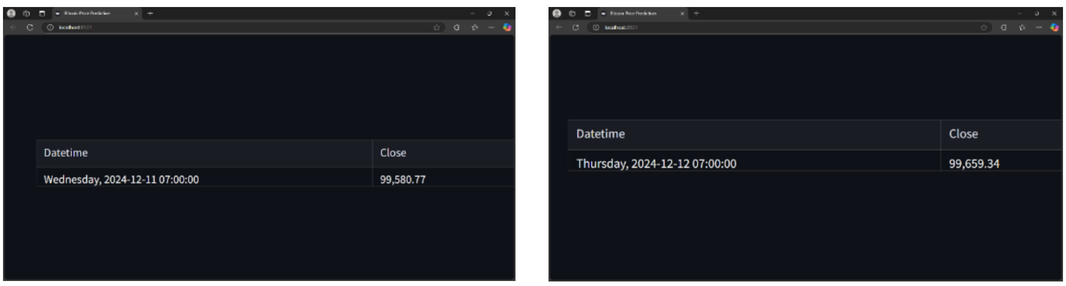

Figure 21. Application 1-day Price Prediction Results

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

Figure 22. 1-hour Interval Real Time Prices

Table 6. Compare Prediction with Real Price (1-day Interval)

Datetime	Prediction	Real Price	Different
2024-12-11	99,580.77	101,176.30	1,595.53
2024-12-12	99,659.34	100,062.38	403.04
2024-12-13	99,657.11	101,434.20	1,777.09
2024-12-14	99,594.41	101,432.34	1,837.93
2024-12-15	99,557.83	104,432.40	4,874.57

Table 6 compares the predicted prices on Figure 21 with the real prices on Figure 22 for 1-day intervals, showing the differences between them. Each row represents a specific date, with the Prediction column displaying the price forecasted by the model and the Real Price column showing the actual price observed on that day. The Difference column indicates the absolute deviation between the predicted and real prices, reflecting the model's prediction accuracy.

For example, on 2024-12-11, the predicted price was 99,580.77, while the real price was 101,176.30, resulting in a difference of 1,595.53. On 2024-12-12, the predicted price was 99,659.34, and the real price was 100,062.38, with a smaller difference of 403.04. As the days progress, the differences between the predicted and real prices increase. The largest discrepancy occurs on 2024-12-15, where the predicted price of 99,557.83 deviates by 4,874.57 from the actual price of 104,432.40. This shows that the model's accuracy decreases over longer time intervals, with larger prediction errors on the later dates.

4. DISCUSSIONS

In recent years, much of the research in the field of financial forecasting, particularly in predicting cryptocurrency prices such as Bitcoin, has primarily focused on the evaluation and optimization of machine learning models. Numerous studies have examined a wide variety of techniques, ranging from statistical methods such as Logistic Regression and Linear Discriminant Analysis, which achieved around 66% accuracy for daily Bitcoin predictions, to deep learning approaches such as LSTM models, which reached 67.2% accuracy for 5-minute interval predictions [13]. More advanced approaches, such as hybrid recurrent neural networks that combine Bi-LSTM, GRU, and LSTM, have demonstrated superior accuracy compared to both statistical models and conventional neural network models [14]. Similarly, Zidane et al. reported that LSTM models achieved the most accurate results on Bitcoin price

Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

forecasting with an RMSE of 0.012441 and an MSE of 0.000155, although at the expense of higher computational time [18]. Other works have also confirmed the robustness of LSTM, CNN-LSTM, and GRU variants in outperforming traditional econometric models such as ARIMA-GARCH and HAR, particularly in short-term horizons [15], [17].

When comparing the results of this study to those of prior works, it can be seen that the proposed LSTM model demonstrates competitive forecasting performance. As shown in Table 2, the model achieved relatively low error values across multiple intervals, with the lowest MAPE of 0.00144 observed for the 5-minute interval, followed by 0.00505 for the 1-hour interval and 0.02027 for the 1-day interval. These error values are comparable to or better than many existing studies, particularly in the context of short-term predictions. For example, the 5-minute interval result (MAPE = 0.00144) suggests a much higher level of precision compared to earlier works that reported accuracy around 67.2% for similar horizons [13]. While Zidane et al. reported lower RMSE and MSE values [18], their study was conducted with a narrower dataset and higher computational demands, whereas our approach demonstrates robust results over extended time horizons (up to 10 years) with lower training complexity. Similarly, the accuracy achieved in this study aligns with observations made by Huang et al. [17], where LSTM-based models consistently outperformed econometric approaches in short-term forecasts.

One of the key advantages of this paper, however, lies beyond raw predictive accuracy. In addition to evaluating the performance of the LSTM model, we extend the contribution by integrating it into an interactive application using the Streamlit framework. Streamlit enables real-time predictions that are accessible to users with limited technical expertise, thereby bridging the gap between research and practice. Unlike many studies that remain confined to offline evaluation [11], [12], [14], [18], this implementation allows end users to directly visualize and interact with predictions across different time intervals (5 minutes, 1 hour, and 1 day), thus enhancing the usability and scalability of the model.

This study therefore demonstrates that while the LSTM model achieves results that are competitive with and in some cases superior to existing research, its true contribution is in the translation of these models into a practical, user-friendly platform. By providing a Streamlit-based web application, the study shows that machine learning models can be operationalized in dynamic markets such as cryptocurrency, offering meaningful insights and accessible forecasting tools to both novice and expert users [35].

5. CONCLUSION

In this study, we have demonstrated the application of machine learning, specifically the LSTM (Long Short-Term Memory) model, for predicting Bitcoin price movements. While many existing studies have focused on evaluating the performance of various predictive models, our research extends beyond the theoretical realm by implementing the LSTM model into a practical web-based application using Streamlit. This development allows users to interact with the model in real-time, providing them with accurate Bitcoin price predictions based on different time intervals, such as 5 minutes, 1 hour, or 1 day.

Our results show that the LSTM model performs well in predicting Bitcoin prices, with high accuracy rates achieved for both short-term and long-term forecasts. By incorporating a user-friendly interface, the web application developed in this study ensures that machine learning models are not only accessible for researchers but also for everyday users who seek to make informed decisions in the cryptocurrency market.

The key contribution of this paper lies in its practical implementation, setting it apart from other studies that typically focus solely on model evaluation. By bridging the gap between theoretical model performance and real-world application, we offer a tool that provides actionable insights to users, thereby enhancing the real-world impact of machine learning in financial markets.

 Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

Beyond its practical utility, this research also contributes to the field of Informatics and Computer Science by showcasing the integration of advanced deep learning techniques with interactive web-based systems. The implementation demonstrates how LSTM can be effectively adapted for real-time time-series forecasting applications, which has broader implications for other domains such as stock market prediction, energy demand forecasting, and intelligent decision support systems. Furthermore, this study highlights the importance of translating theoretical advances in machine learning into accessible software tools, reinforcing the role of Informatics in bridging complex algorithms with practical, user-oriented solutions.

Future work can explore further refinements to the model, including the incorporation of additional features, more advanced algorithms, and the expansion of the web application to support other cryptocurrencies. Additionally, continuous improvements to the application's functionality and user experience can help ensure its scalability and usability in the ever-evolving cryptocurrency market.

CONFLICT OF INTEREST

No conflict of interest between the authors or with research object in this paper.

ACKNOWLEDGEMENT

1. Lembaga Penelitian dan Pengabdian Masyarakat Universitas Jenderal Soedirman

REFERENCES

[1] G. A. Tauda, A. Omara, and G. Arnone, "Cryptocurrency: Highlighting the Approach, Regulations, and Protection in Indonesia and European Union," *BESTUUR*, vol. 11, no. 1, p. 1, 2023, https://doi.org/10.20961/bestuur.v11i1.67125.

[2] G. Kou and Y. Lu, "FinTech: A literature review of emerging financial technologies and applications," *Financ Innov* vol. 11, pp. 1, 2025. https://doi.org/10.1186/s40854-024-00668-6.

[3] I. D. Mienye and T. G. Swart, "A comprehensive review of deep learning: Architectures, recent advances, and applications," *Information,* vol. 15, no. 12, p. 755, 2024, https://doi.org/10.3390/info15120755.

[4] D. Liu, "Quantitative finance and information technologies: A comparative analysis of quantitative trading and cryptocurrency and their regulatory challenges," *Economics* vol. 13, no. 4, pp. 88-99, 2024, DOI: 10.11648/j.eco.20241304.11.

[5] M. D. S. M. Bhuiyan, M. D. AL Rafi, G. N. Rodrigues, M. D. N. H. Mir, A. Ishraq, M.F. Mridha, and J. Shin, "Deep learning for algorithmic trading: A systematic review of predictive models and optimization strategies," *Array*, vol 26, p. 100390, 2025, https://doi.org/10.1016/j.array.2025.100390.

[6] G. Dai, "Analysis on the influence of intelligent algorithm trading in financial market," F*rontiers in Business Economics and Management,* vol. 17, no. 1, pp. 280-283, 2024, DOI: 10.54097/j7r1v853.

[7] X. Zhang, L. Zhang, Q. Zhou, and X. Jin, "A novel bitcoin and gold prices prediction method using an LSTM-P neural network model," *Hindawi Computational Intelligence and Neuroscience* vol. 2022, p. 12, 2022, https://doi.org/10.1155/2022/1643413.

[8] P. Koratamaddi, K. Wadhwani, M. Gupta, and S. G. Sanjeevi, "Market sentiment-aware deep reinforcement learning approach for stock portfolio allocation," *Engineering Science and Technology an International Journal,* vol. 24, no. 4, 2021, DOI: 10.1016/j.jestch.2021.01.007.

[9] H. Ma, D. Prosperino, A. Haluszczynski, and C. Räth, "Linear and nonlinear causality in financial markets," *Chaos*, vol. 34, no. 11, p. 113125, 2024, doi: 10.1063/5.0184267.

[10] M. Muth, M. Lingenfelder, and G. Nufer, "The application of machine learning for demand prediction under macroeconomic volatility: A systematic literature review," *Manag Rev Q* 2024, https://doi.org/10.1007/s11301-024-00447-8.

Vol. 6, No. 5, October 2025, Page. 2940-2961
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5168

[11] M. A. D. Suyudi, E. C. Djamal, and A. Maspupah, “Prediksi Harga Saham menggunakan Metode Recurrent Neural Network,” 2019.

[12] M. Waqas and U. W. Humphries, “A critical review of RNN and LSTM variants in hydrological time series predictions,” *MethodsX*, vol. 13, p. 102946, 2024, doi: 10.1016/j.mex.2024.102946.

[13] Z. Chen, C. Li, and W. Sun, “Bitcoin price prediction using machine learning: An approach to sample dimension engineering,” *Journal of Computational and Applied Mathematics*, vol. 365, p. 112395, 2020, doi: 10.1016/j.cam.2019.112395.

[14] A. John and T. Latha, “Stock market prediction based on deep hybrid RNN model and sentiment analysis,” *Automatika*, vol. 64, no. 4, pp. 981–995, 2023, doi: 10.1080/00051144.2023.2217602.

[15] Q. P. Duy, O. Ng. Thi, P. H. L. Thi, H. D. P. Hoang, K. L. Luong, and K. N. Ng. Thi, “Estimating and forecasting bitcoin daily prices using ARIMA-GARCH models,” *BAJ*, vol. 45, no. 1, pp. 11–23, 2024, doi: 10.1108/BAJ-05-2024-0027.

[16] P. Jaquart, S. Köpke, and C. Weinhardt, “Machine learning for cryptocurrency market prediction and trading,” *The Journal of Finance and Data Science*, vol. 8, pp. 331–352, 2022, doi: 10.1016/j.jfds.2022.12.001.

[17] Z.-C. Huang, I. Sangiorgi, and A. Urquhart, “Forecasting Bitcoin volatility using machine learning techniques,” *Journal of International Financial Markets, Institutions and Money*, vol. 97, p. 102064, 2024, doi: 10.1016/j.intfin.2024.102064.

[18] Z. I. Ramadhan and H. Widiputra, “Comparative Analysis of Recurrent Neural Network Models Performance in Predicting Bitcoin Prices,” *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 8, no. 3, pp. 377–388, 2024, doi: 10.29207/resti.v8i3.5810.

[19] E. W. V. Zuniga, C. M. Ranieri, L. Zhao, J. Ueyama, Y. Zhu, and D. Ji, “Maximizing portfolio profitability during a cryptocurrency downtrend: A Bitcoin Blockchain transaction-based approach,” *Procedia Computer Science*, vol. 222, pp. 539–548, 2023, doi: 10.1016/j.procs.2023.08.192.

[20] V. Saha, "Predicting future cryptocurrency prices using machine learning algorithms," *Journal of Data Analysis and Information Processing,* vol.11, no.4, pp. 409-419, DOI: 10.4236/jdaip.2023.114021.

[21] S. Singh, A. Pise, and B. Yoon, “Prediction of bitcoin stock price using feature subset optimization,” *Heliyon*, vol. 10, no. 7, p. e28415, 2024, doi: 10.1016/j.heliyon.2024.e28415.

[22] P. Jaquart, D. Dann, and C. Weinhardt, “Short-term bitcoin market prediction via machine learning,” *The Journal of Finance and Data Science*, vol. 7, pp. 45–66, 2021, doi: 10.1016/j.jfds.2021.03.001.

[23] Y.-J. An, H.-Y. Oh, and H.-J. Kim, “Forecasting Bitcoin Prices: An LSTM Deep-Learning Approach Using On-Chain Data,” *Adv. sci. technol. eng. syst. j.*, vol. 8, no. 3, pp. 186–192, 2023, doi: 10.25046/aj080321.

[24] R. F. Inaku and J. C. Chandra, “Implementasi data mining dalam prediksi harga saham menggunakan metode long short term memory (LSTM),” *Ticom*, vol. 12, no. 1, pp. 1–7, 2023, doi: 10.70309/ticom.v12i1.99.

[25] Y. Yati, M. Ainol Yaqin, and A. Yusrotun Nadhiroh, “Application of the support vector machine algorithm for timely student graduation prediction based on streamlit web at the Faculty of Informatics Engineering Nurul Jadid University,” *CNAHPC*, vol. 6, no. 3, pp. 1066–1070, 2024, doi: 10.47709/cnahpc.v6i3.3918.

[26] F. Rodrigues and M. Machado, "High-frequency cryptocurrency price forecasting using machine learning models: A comparative study," *Information,* vol. 16, no. 4, p 300, 2025, https://doi.org/10.3390/info16040300.

[27] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A Review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data," *Frontiers in Energy Research,* vol. 9, 2021, DOI: 10.3389/fenrg.2021.652801.

[28] A. E. Karrar, "The effect of using data pre-processing by imputations in handling missing values," *Indonesian Journal of Electrical Engineering and Informatics* (IJEEI), vol. 10, no. 2, pp. 375-384, DOI: 10.52549/ijeei.v10i2.3730.

[29] S. H. Dolatabadi, I. Budinska, and E. Gatial, " Closing the data gap: A comparative study of missing value imputation algorithms in time series datasets," 2024 *Conference: Computational Methods in Systems and Software*, DOI: 10.1007/978-3-031-53552-9_7.

[30] V. Sharma "A study on data scaling methods for machine learning," *International Journal for Global Academic & Scientific Research,* vol. 1, no. 1, DOI: 10.55938/ijgasr.v1i1.4.

[31] I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent neural networks: A comprehensive review of architectures, variants, and applications," *Information,* vol. 15, no. 9, p. 517; 2024, https://doi.org/10.3390/info15090517.

[32] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video Captioning With Attention-Based LSTM and Semantic Consistency," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2045–2055, 2017, doi: 10.1109/TMM.2017.2729019.

[33] A. Pulver and S. Lyu, "LSTM with working memory," *International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 845–851. doi: 10.1109/IJCNN.2017.7965940.

[34] S. A. Mohammad and H. Santoso, "Predict stock prices using the Generative Adversarial Networks," *Sinkron : Jurnal dan Penelitian Teknik Informatika*, vol. 6, no. 2, pp. 560-567, 2022, DOI : https://doi.org/10.33395/sinkron.v7i2.11405.

[35] B. R. Cherukuri, "Building scalable web applications: Best practices for backend architecture," *International Journal of Science and Research (IJSR)*, vol. 13, no. 10, pp. 126–139, Oct. 2024, doi: 10.21275/ES24928085711.