Journal of Applied Data Sciences
Vol. 6, No. 3, September 2025, pp. 1850-1864
ISSN 2723-6471
1850

Integrating Moving Average Indicators with Long Short-Term Memory Model in Bitcoin Price Forecasting

Phung Duy Quang[1,*], Nguyen Hoang Duy[2], Pham Quang Khoai[3], Bui Duc Duong[4]

[1]*Faculty of Technology and Data Science, Foreign Trade University, Hanoi, Vietnam*

[2]*Institute of Economics and International Business, Foreign Trade University, Hanoi, Vietnam*

[3]*Faculty of Development Economics, VNU University of Business and Economics, Vietnam*

[4]*Luong The Vinh Secondary and High School, Hanoi, Vietnam*

(Received: January 20, 2025; Revised: March 1, 2025; Accepted: April 20, 2025; Available online: July 10, 2025)

Abstract

Bitcoin price forecasting remains a challenging task due to the market's high volatility and complex nonlinear dynamics. This study proposes a novel forecasting framework by integrating Long Short-Term Memory (LSTM) networks with Moving Average (MA) indicators—specifically Simple Moving Average (SMA), Exponential Moving Average (EMA), and Weighted Moving Average (WMA)—as auxiliary input features to enhance model accuracy. The objective is to examine the frequency-specific effectiveness of these hybrid models across daily and high-frequency datasets. Using historical Bitcoin data from Bitstamp between January 2021 and December 2024, we conducted experiments at four epoch levels (50, 100, 150, 200) to determine optimal model configurations. Empirical results reveal that, on daily data, LSTM combined with a 10-period WMA achieves the lowest Mean Absolute Percentage Error (MAPE) of 2.1661% at 150 epochs, while for high-frequency data, the combination with a 10-period SMA yields superior performance with a MAPE of 0.4895%. Furthermore, increasing epochs beyond the optimal point led to performance degradation, indicating overfitting. Compared to the standalone LSTM model, our integrated approach demonstrates significantly improved adaptability to short-term fluctuations and heightened forecasting precision. This research contributes a comprehensive comparative analysis of MA-enhanced deep learning models for cryptocurrency price prediction, and offers practical insights for algorithmic traders, financial analysts, and decision-support systems in volatile digital asset markets.

Keywords: Bitcoin Price Prediction, LSTM, Deep Learning, Moving Average Indicators, SMA, EMA, WMA, High-Frequency Trading, Cryptocurrency Forecasting, Time-Series Analysis

1. Introduction

Bitcoin, introduced as the pioneering decentralized cryptocurrency [1], presents significant forecasting challenges due to its high price volatility, influenced by factors including market sentiment [2], regulatory changes, and global economic events [3]. While traditional time-series models like ARIMA and GARCH often struggle with Bitcoin's strong nonlinearities, deep learning models, particularly LSTM networks, offer a more powerful alternative for capturing complex sequential patterns. However, LSTMs require careful configuration to mitigate overfitting risks, especially with volatile financial data.

This study explores enhancing LSTM-based Bitcoin price forecasting by integrating common technical indicators: SMA, EMA, and WMA. We hypothesize that incorporating these indicators as input features can improve predictive accuracy by providing smoothed trend information and potentially easing difficulties associated with raw price volatility. This study aims to explore how the incorporation SMA, EMA and WMA can improve the forecasting accuracy of LSTM models in predicting Bitcoin prices. Furthermore, the research seeks to determine the optimal combination of moving average periods and training epochs for both daily and high-frequency data. Finally, this study will investigate how different moving average indicators influence prediction performance and what practical implications can be drawn for investors and financial practitioners.

*Corresponding author: Phung Duy Quang (quangpd@ftu.edu.vn)
DOI: https://doi.org/10.47738/jads.v6i3.766

To investigate these questions, this study develops a hybrid forecasting model that integrates LSTM with moving average indicators. Through extensive experimentation and evaluation, we aim to provide actionable insights into improving Bitcoin price prediction accuracy. The paper is structured as follows: Section 2 presents a review of related literature, highlighting key works on LSTM and technical indicators. Section 3 describes the research methodology, including data collection, preprocessing, and model design. Section 4 discusses the experimental results and performance analysis. Section 5 focuses on the discussion and interpretation of the findings. Finally, Section 6 concludes with major findings and practical implications for the financial sector.

2. Literature Review

Bitcoin price forecasting has been an area of significant research interest due to the cryptocurrency's unique price characteristics. Early studies predominantly applied traditional statistical models, such as ARIMA and GARCH, which were successful in handling stationary time-series data but exhibited limitations when applied to nonlinear, volatile cryptocurrency markets [4]. The inability of these models to capture complex dependencies in Bitcoin price data motivated the shift towards machine learning and deep learning approaches.

2.1. Traditional Forecasting Methods

ARIMA (Auto-Regressive Integrated Moving Average) and GARCH (Generalized Auto-Regressive Conditional Heteroskedasticity) are two popular statistical time-series forecasting models. ARIMA, specifically, is well-adapted to univariate time series and performs well when data is in a linear trend [5]. Nevertheless, [6] points out that ARIMA is not able to effectively predict Bitcoin prices because of the high volatility and nonlinear nature of the asset. Similarly, GARCH models, despite accounting for changing variances, have shown limited adaptability to sudden market fluctuations in cryptocurrencies [7].

2.1.1. Deep Learning Models

Deep learning models, specifically LSTM networks, have been increasingly popular due to their long-term dependency and non-linear dynamics capabilities. LSTM networks are the gradient-based variant of Recurrent Neural Networks (RNNs) that have been specifically engineered to effectively tap into the temporal relationship of sequential data, showing excellent traditional time-series forecasting [8]. Besides, Tripathy found that LSTM outperforms traditional models, as they showed better forecasting accuracy using metrics like Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) [9]. The selection of data frequency and timeframe is a crucial factor in Bitcoin price forecasting as well. In [10] high-frequency data (5-minute intervals) was compared with daily data and it was found that deep learning models like LSTM perform better with high-frequency data, while traditional statistical methods are more effective for daily data. This highlights LSTM's adaptability to frequent price changes, making it particularly valuable in high-frequency trading environments.

2.2. The Role of Moving Average Indicators in Enhancing Forecasting

SMA, EMA, and WMA have been classic indicators to discover trends and market signals from long ago by traders. In [11], combining deep learning models with technical indicators significantly improves forecasting performance for both short-term and long-term price movements. These indicators help smooth out frequent price fluctuations, making it easier for models to extract meaningful patterns. In particular, SMA and EMA have proven effective in reducing noise, especially in high-frequency trading environments. As shown by Bakar and Rosbi [12] in their study, WMA is especially suitable for high-frequency trading, where it gives more weight to the last data points. Their research revealed that when WMA is integrated with deep learning models, the error on forecasting decreases as the most recent market behaviors matter more in these volatile environments. Based partly on the above results, our investigation is to push further on incorporating SMA, EMA and WMA with LSTM networks for optimal bitcoin price prediction.

2.3. Combining LSTM with Moving Average Indicators

Several studies have demonstrated that incorporating moving averages into LSTM models can improve predictive performance. In [13] LSTM models were combined with moving averages using daily Bitcoin data with different lookback periods (3, 5, 10, 15 days). Their results showed that WMA was the most effective indicator for daily data with the lowest MAPE, as it assigns greater weight to recent prices, making it more responsive to short-term trends.

2.4. Research Gaps and Proposed Approach

Although there has been a lot of research on Bitcoin deep learning forecasting, there are still some research gaps. To begin with, the literature in this field does not provide full exploration of distinguishing indicators from using 5-period, 10-period and 20-period in high-frequency and daily data. Secondly, not all previous studies are based on a comparison across timeframes regarding the best (SMA/EMA/WMA) indicator pairings with LSTM. To address these gaps, our study proposes integrating moving average indicators (SMA, EMA, WMA) with the LSTM model for both hourly and daily datasets. Furthermore, we assess the effectiveness of different indicator periods (5-period, 10-period, 20-period) to provide a comprehensive evaluation of how LSTM performs when combined with moving average technical indicators. Not only does this approach bridge the gap between theoretical research and real-world applications, it provides a scientific basis needed to develop effective cryptocurrency trading strategies.

3. Methodology

The method used in this study is explained in this section. The research workflow includes data collection, data preparation, model deployment, and performance evaluation. It is briefly shown in figure 1. The overall research process follows a standard workflow, illustrated in figure 1, encompassing stages from data acquisition and preparation through model training, validation, and final performance evaluation on the test set.

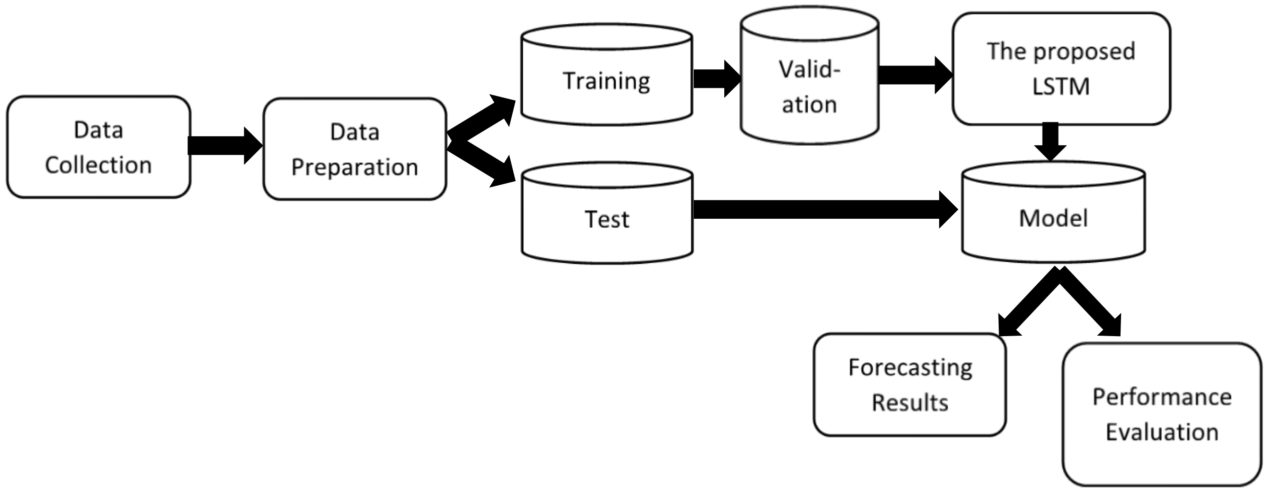

Figure 1. Flowchart illustrating the key stages of the research methodology

3.1. Data Collection

This study utilizes Bitcoin closing prices sourced from Bitstamp from January 1, 2021, to December 29, 2024, comprising 1,459 daily and 35,016 hourly observations. Closing prices were selected as reliable indicators of final market sentiment within each timeframe. Figures 2 and figure 3 illustrate the data's characteristic high volatility. Both daily and hourly data frequencies were intentionally selected to evaluate the proposed model's performance across different temporal resolutions relevant to financial analysis; daily data typically informs longer-term trend analysis and investment strategies, while hourly data captures intraday dynamics crucial for shorter-term trading activities.

Figure 2. Bitcoin hourly data from January 1st 2021 to December 29th 2024

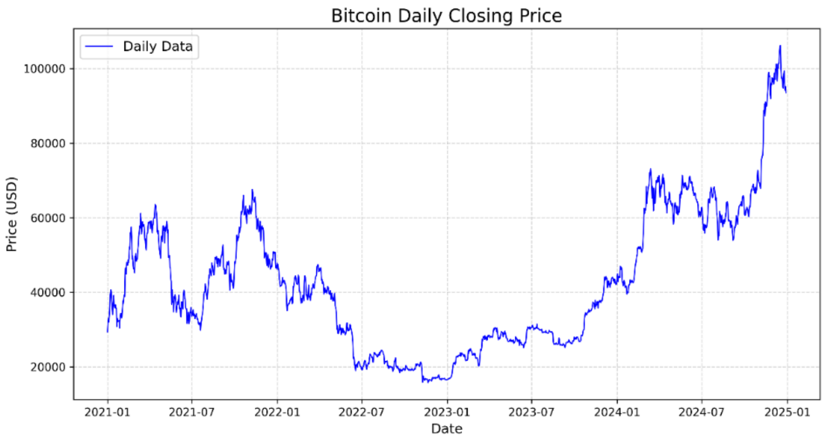

Figure 3. Bitcoin daily data from January 1st 2021 to December 29th 2024

The result of table 1 reveals that the average Bitcoin price during the hourly timeframe is 42,527.40 USD, while for the daily timeframe, it is 42,557.71 USD, indicating no significant difference and suggesting consistency in the mean price across both periods. Next, the price volatility, measured by the standard deviation, is 18,720.81 USD for hourly data and 18,733.36 USD for daily data, demonstrating high volatility in both timeframes. In the hourly timeframe, Bitcoin's closing price reached a maximum of 108,276 USD and a minimum of 15,578 USD, compared with 106,187 USD and 15,766 USD in the daily timeframe.

Table 1. The Descriptive Statistics of Bitcoin Closing Prices

	Hourly Price	Daily Price		Hourly Price	Daily Price
Mean	42527.403	42557.709	**Skewness**	0.785	0.783
Median	39728.975	39741.12	**Kurtosis**	3.352	3.343
Maximum	108276	106187	**Jarque-Bera**	3776.256	156.293
Minimum	15632	15766	**Probability**	0	1.15E-34
Std. dev.	18720.814	18733.357	**Sum**	1489139528	62091697.99
Observations	35016	1459	**Sum Sq. dev.**	1.2272E+13	5.1202E+11

Additionally, the price distribution is slightly right-skewed, with skewness values of 0.7850 (hourly) and 0.7831 (daily). The kurtosis values are 3.3518 (hourly) and 3.3435 (daily), both close to a normal distribution level. However, the Jarque-Bera test indicates that Bitcoin's price distribution does not follow a normal distribution, possibly due to market-driven external factors. These results emphasize the high volatility inherent in the Bitcoin market.

ISSN 2723-6471

3.2. Data Preparation

3.2.1. Data Split

In this study, the dataset is divided into three parts: a training set, a validation set, and a test set. Specifically, data from January 1st, 2021 to October 18th, 2023, accounting for approximately 70% of the total research period, was used for model training. The validation set contains data from October 19th, 2023 to May 24th, 2024, covering approximately 15% of the total period. The remaining data constitutes the test set, used for final performance evaluation. This chronological partitioning ensures that model evaluation is performed on unseen data. Figure 4 visually depicts this allocation of the dataset into distinct training (70%, 1021 days), validation (15%, 219 days), and test (15%, 219 days) periods.

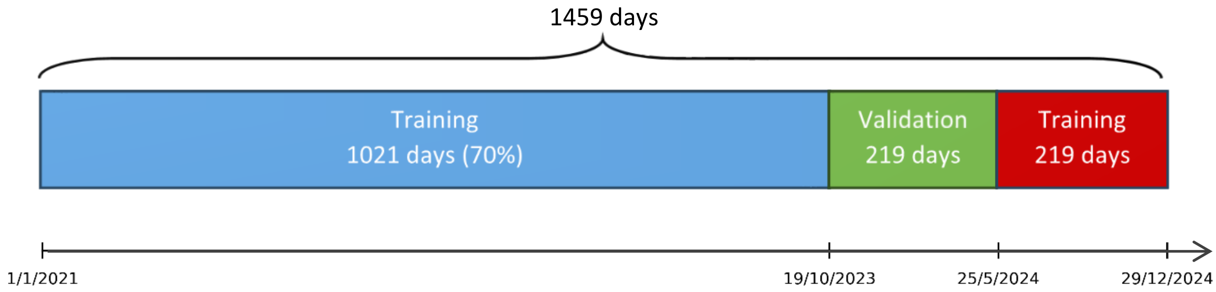

Figure 4. Data split in this study

The partitioning of the dataset into training, validation, and test sets is a crucial step in building reliable time-series models. As highlighted in [14], a dedicated validation set is crucial for optimizing hyperparameters and minimizing overfitting while keeping the training data focused on learning general patterns. Similarly, [15] emphasized that a properly allocated test set ensures unbiased evaluation and enhances the model's ability to generalize to unseen data. In this study, we split the data into 70% training, 15% validation, and 15% test, providing a balanced approach, aligning with established practices and ensuring robust forecasting performance.

3.2.2. Missing values and outlier detection

Handling missing values and detecting outliers is a crucial preprocessing step in time-series data to ensure the quality and reliability of model training. As missing data can lead to biased model outputs, while undetected outliers can distort predictions and affect the model's generalizability [16]. Therefore, we need to identify and clean up these anomalies before normalization. In this study, the missing values and outlier detection and handling are applied only in the training set and then imputation and cleaning strategies are used the same way on the validation and test sets, respectively. This approach aligns with [17], they emphasize that applying data correction directly to the training set ensures that the model learns consistent patterns without data leakage or bias from the test set. Applying this correction only to the training set prevents the model from "seeing" information that could influence future predictions.

3.2.2.1. Missing values detection

The total number of missing values in the dataset D = {x1, x2, …, xn} is calculated as:

$$M = \sum_{i=1}^{n} 1(x_i = NA) \quad (1)$$

M is the total number of missing values

3.2.2.2. Outlier detection

In this study, outliers are detected by using the interquartile range (IQR) method:

$$IQR = Q3 - Q1 \quad (2)$$

The lower bound L and upper bound U are defined as:

$$L = Q1 - 1.5IQR \tag{3}$$

$$U = Q3 + 1.5IQR \tag{4}$$

Values outside this range ($x_i < L \ V \ x_i > U$) are classified as outliers. The result of table 2 shows that there are no missing values and outliers in the Training set, so we conduct further calculation. Plus, as no anomalies were detected, no imputation or outlier treatment was applied.

Table 2. Results of data checking on the training dataset

	Hourly Price	**Daily Price**
Number of missing values	0	0
Number of outliers	0	0

3.2.3. Calculation of Moving Average Indicators

Moving averages are a method used to calculate the average of prices over a specified period, helping to reduce short-term volatility and provide clearer trading signals. The moving average value is continuously updated as new data becomes available, ensuring that the indicator reflects current market trends. Moving averages (MA) are one of the most important tools in technical analysis, helping smooth out price fluctuations and identify market trends [18]. In this study, MA values are calculated using standard time periods of 5-, 10-, and 20-intervals, applied to both 5-minute and daily timeframes.

3.2.3.1. SMA

The simple moving average at time t is calculated as the average of the closing prices over a fixed period n. The formula is defined as:

$$SMA_t = \frac{1}{n} \sum_{i=0}^{n-1} P_{t-i} \tag{5}$$

SMA_t is the simple moving average at time t; n is the number of periods used for the average and P_{t-i} is the closing price at t – i periods.

3.2.3.2. EMA

The EMA assigns more weight to recent prices, ensuring that the indicator responds more quickly to new market data. The formula is defined as:

$$EMA_t = P_t \times \alpha + EMA_{t-1} \times (1 - \alpha) \tag{6}$$

EMA_t is the exponential moving average at time t; P_t is the closing price at time t; $\alpha = \frac{2}{n+1}$ is the smoothing factor; EMA_{t-1} is the EMA value from the previous period.

3.2.3.3. WMA

WMA at time t is calculated by assigning higher weights to recent prices, while older prices are assigned lower weights. The formula is given as:

$$WMA_t = \frac{\sum_{i=1}^{n} w_i \times P_{t-i+1}}{\sum_{i=1}^{n} w_i} \tag{7}$$

WMA_t is the weighted moving average at time t; P_{t-i+1} is the closing price at time $t - i + 1$; w_i is the weight assigned to each price, where w_i = I (i.e., more recent prices have higher weights); n is the number of periods used for the calculation.

3.2.4. Data Normalization

Following the calculation of moving averages, data normalization was performed using Min-Max Scaling to bring all features into a common range [0, 1], enhancing model stability and learning consistency, as defined in the equation:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{8}$$

X' is the normalized value; Xmin and Xmax are the minimum and maximum values of the dataset, respectively.

While addressing potential non-stationarity is crucial, this study leverages the inherent capability of LSTM networks to model complex sequential dependencies, including trends often found in non-stationary financial data, unlike traditional models requiring explicit transformations (e.g., differencing). Consistent with common applications of LSTMs in financial forecasting that prioritize sequence learning capacity, explicit stationarity-inducing transformations were therefore not applied before model training. The consequences of this methodological choice are further discussed alongside the study's limitations.

3.3. Model Deployment

3.3.1. Model Selection

In this study, we use the LSTM model to forecast Bitcoin price fluctuations. LSTM is a variant of the RNN with the ability to retain long-term information through a series of gated units, including input gates, forget gates, and output gates. This model effectively handles dynamic changes and the high nonlinearity of financial time series data [19]. The overview of LSTM architecture is shown in figure 5, which provides a schematic representation of the LSTM cell structure employed in this study.

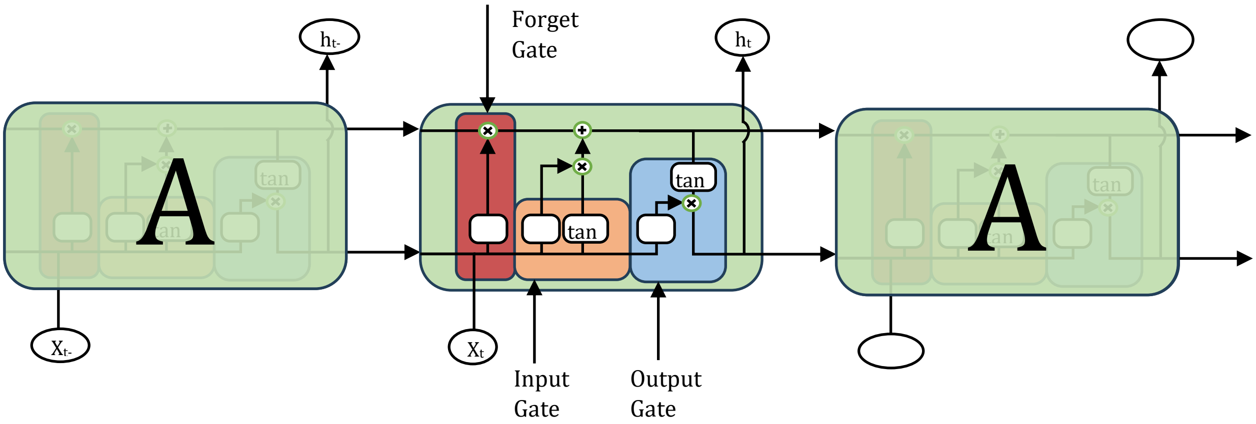

Figure 5. The overview of LSTM architecture [10]

The core components include the forget gate (f_t), input gate (i_t), and output gate (o_t), which work in concert with the cell state (C_t) to selectively retain or discard information over time, enabling the network to learn long-term dependencies as described in the following equations:

$$X = \begin{bmatrix} Bitcoin\ price\ at\ time\ t \\ Moving\ average\ indicators\ at\ time\ t \end{bmatrix} \tag{9}$$

The LSTM cell operates as follows: Forget gate: $f_t = \delta(W_f.X + b_f)$; Input gate: $i_t = \delta(W_i.X + b_i)$; Output gate: $o_t = \delta(W_o.X + b_o)$; Candidate cell state: $\tilde{C}_t = tanh(W_c.[h_{t-1}, x_t] + b_c)$; Update cell state: $C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$; Output: $h_t = o_t \odot \tanh(C_t)$. δ is the sigmoid activation function; W represents weight matrices, and b represents bias terms; $\odot$ denotes the element-wise product.

3.3.2. Detailed Model Structure

In this study, the LSTM model used is designed with a specific architecture consisting of an input layer, two LSTM layers, and one output layer. The input layer contains the feature set, which includes Bitcoin prices along with three moving average indicators: SMA, EMA, and WMA, calculated with time periods of 5, 10, and 20. Following this are two LSTM layers, each comprised of 64 neurons utilizing the tanh activation function; this structure allows the model to capture long-term dependencies inherent in the time series data. Finally, the architecture concludes with an output layer, which is a Dense layer containing a single neuron responsible for predicting the Bitcoin price at time t+1.

To support these configurations, we employed a manual hyperparameter tuning strategy guided by iterative validation and established literature. The batch size of 64 was selected for its balance between training efficiency and gradient stability [20], while the sequence length of 30 captured one full trading month—commonly used in short-term trend modeling [21]. It is pertinent to note that no Dropout layers were implemented within this specific architecture.

3.3.3. Loss Function and Model Optimization

In this study, we used the MSE as the loss function to measure the average squared difference between actual and predicted Bitcoin prices. Since MSE is sensitive to large errors, the penalty for big discrepancies helps the model focus on minimizing major forecasting mistakes. Additionally, MSE ensures smooth gradient descent, aiding in stable optimization during training.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_i - \hat{y}_i)^2 \tag{10}$$

y_i is the actual Bitcoin price at time i; $\hat{y}_i$ is the predicted price at time i; n is the total number of observations

For optimization, the Adam (Adaptive Moment Estimation) algorithm was employed, leveraging its benefits of momentum and adaptive learning rates. In this study, the learning rate was explicitly set to 10^{-3} as among the learning rates tested (0.01, 0.001, 0.0005), the value of 0.001 yielded the most consistent convergence and lowest validation error, aligning with [22] .While other Adam hyperparameters utilized the default values provided by the TensorFlow/Keras library ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\in = 10^{-7}$). The batch size was configured to 64, and models were trained for specified epoch lengths (50, 100, 150, and 200) to facilitate comparative analysis of training duration effects. Notably, an early stopping mechanism was not utilized in these experiments; training proceeded for the complete designated number of epochs for each experimental run to observe performance evolution and potential overfitting across the full training spectrum. Adam dynamically adjusts the learning rate based on the average of past gradients, ensuring faster convergence and preventing oscillation during optimization. The parameter update rule is expressed as:

$$\theta = \theta - \eta \cdot \nabla \mathrm{MSE} \tag{11}$$

θ represents the model parameters (weights and biases); η is the learning rate; ∇MSE is the gradient of the MSE loss

3.4. Performance Evaluation

In this study, we use three key metrics: MAPE, Root Mean Square Error (RMSE), and the Coefficient of Determination (R^2) to evaluate the forecasting performance of the Bitcoin price prediction model. Among these, MAPE measures the average percentage deviation between predicted and actual values; RMSE measures the standard deviation of the prediction errors and is particularly sensitive to large errors due to its squared component; while the coefficient of determination R2 measures the proportion of the variance in the target variable explained by the model relative to the variance around the mean. The mathematical formulas for these evaluation metrics are presented in table 3.

Table 3. Success Criterion (Correlogram) formulas

Name	Formulas
MAPE	$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left\lvert\frac{P_t - \hat{P}_t}{P_t}\right\rvert \times 100$

RMSE	$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(P_t - \widehat{P}_t)^2}{n}}$
R²	$R^2 = 1 - \frac{\sum_{t=1}^{n}(P_t - \widehat{P}_t)^2}{\sum_{t=1}^{n}(P_t - \overline{P}_t)^2}$

P_t represents the actual value; $\widehat{P}_t$ represents the predicted value; $\overline{P}_t$ is the mean of the actual values; n denotes the number of forecast periods.

The model with the lowest values for MAPE, RMSE and the highest values for R^2 is considered the most suitable. Plus, it should be noted that the performance metrics reported in this study are based on a single execution run for each model configuration, using the default random seed initialization provided by the TensorFlow/Keras library. Variability due to random weight initialization was not explicitly controlled across multiple runs.

3.5. Software Used

In this study, Python is utilized as the primary programming language for implementing and training the Bitcoin price prediction model, with the entire code execution process carried out on the Google Colab platform. Several key libraries were employed to facilitate this process: TensorFlow/Keras was used for developing and training the LSTM model; Pandas and NumPy were utilized for data processing and manipulation; Matplotlib and Seaborn were employed for data visualization; and Scikit-learn was used for data normalization and evaluating model performance.

4. Results

This section details the performance of the LSTM models, both with and without the incorporation of technical indicators, using daily and hourly Bitcoin closing price data. The results are organized by training epochs (50, 100, 150, and 200) to systematically analyze the evolution of model performance. As expected, models combined with indicators generally outperform the baseline LSTM model, showing improvements in terms of RMSE, MAPE, and R^2.

4.1. Performance on Daily Closing Price Data

4.1.1. Epochs = 50

Initial training at 50 epochs established a performance baseline for all model configurations. The empirical results from this phase are detailed in table 4.

Table 4. Performance of LSTM models using daily Bitcoin closing price data (Epochs = 50)

Model	MSE	RMSE	MAPE	R²
LSTM	14631261	3825.083	3.7821%	93.9203%
LSTM + SMA5	23643503	4862.459	4.7858%	90.1755%
LSTM + SMA10	22500172	4743.435	4.7394%	90.6505%
LSTM + SMA20	32083359	5664.217	6.1919%	86.6684%
LSTM + EMA5	19954107	4467.002	4.3078%	91.7085%
LSTM + EMA10	11209799	3348.104	3.3019%	95.3420%
LSTM + EMA20	22455976	4738.774	4.5084%	90.6689%
LSTM + WMA5	11962919	3458.745	3.4021%	95.0291%
LSTM + WMA10	6338510	2517.640	2.5122%	97.3662%
LSTM + WMA20	38368621	6194.241	6.3667%	84.0567%

For daily data at 50 epochs, the 10-period WMA model performs best, with a MAPE of 2.5122% and an R^2 of 97.3662%. Its RMSE (2517.64) shows that while it effectively tracks short-term fluctuations, the model is not yet fully optimized for capturing long-term trends.

4.1.2. Epochs = 100

As the epoch count increases to 100, significant performance improvements become evident, with models demonstrating a better balance between accuracy and generalization. The results are presented in table 5.

Table 5. Performance of LSTM models using daily Bitcoin closing price data (Epochs = 100)

Model	MSE	RMSE	MAPE	R^2
LSTM	9376188	3062.056	2.8077%	96.1039%
LSTM + SMA5	10757976	3279.935	2.9089%	95.5298%
LSTM + SMA10	8473559	2910.938	2.6679%	96.4790%
LSTM + SMA20	14146239	3761.149	3.8118%	94.1218%
LSTM + EMA5	9612720	3100.439	3.0885%	96.0056%
LSTM + EMA10	8075008	2841.656	2.7243%	96.6446%
LSTM + EMA20	12162310	3487.450	3.0521%	94.9462%
LSTM + WMA5	12859493	3586.014	3.5581%	94.6565%
LSTM + WMA10	12548633	3542.405	3.5425%	94.7857%
LSTM + WMA20	16129450	4016.149	3.4111%	93.2978%

For daily data, the 10-period SMA model slightly edged ahead, achieving a MAPE of 2.6679% and an R^2 of 96.4790%. Its RMSE (2910.938) is relatively higher than that of the 10-period WMA at 50 epochs, but the SMA model compensates with its ability to smooth fluctuations and adapt better to evolving trends.

4.1.3. Epochs = 150

At 150 epochs, the models generally exhibited their best performance metrics according to the evaluation criteria applied in this study. The details are shown in table 6.

Table 6. Performance of LSTM models using daily Bitcoin closing price data (Epochs = 150)

Model	MSE	RMSE	MAPE	R^2
LSTM	7489768	2736.744	2.5498%	96.8878%
LSTM + SMA5	8640841	2939.531	2.7926%	96.4095%
LSTM + SMA10	19553902	4421.979	4.3893%	91.8748%
LSTM + SMA20	13095518	3618.773	3.3692%	94.5584%
LSTM + EMA5	21375043	4623.315	4.3106%	91.1181%
LSTM + EMA10	9402438	3066.340	2.7696%	96.0930%
LSTM + EMA20	35541330	5961.655	5.7932%	85.2316%
LSTM + WMA5	9064449	3010.722	2.7456%	96.2335%
LSTM + WMA10	4836934	2199.303	2.1661%	97.9901%
LSTM + WMA20	6477707	2545.134	2.4375%	97.3083%

The 10-period WMA model achieved a MAPE of 2.1661% and the lowest RMSE (2199.303), with an impressive R^2 of 97.9901%. This indicates that the model effectively handles both short-term fluctuations and long-term trends, making it highly reliable for forecasting in volatile market environments.

4.1.4. Epochs = 200

At 200 epochs (see table 7), signs of overfitting appear, particularly in the daily data models.

Table 7. Performance of LSTM models using daily Bitcoin closing price data (Epochs = 200)

Model	MSE	RMSE	MAPE	R^2
LSTM	27082992	5204.132	4.0981%	88.7462%
LSTM + SMA5	13919132	3730.835	3.3011%	94.2162%
LSTM + SMA10	9764589	3124.834	2.7971%	95.9425%
LSTM + SMA20	13804922	3715.498	3.5271%	94.2637%

LSTM + EMA5	12618219	3552.213	3.3973%	94.7568%
LSTM + EMA10	14759523	3841.812	3.2661%	93.8670%
LSTM + EMA20	8964927	2994.149	2.8918%	96.2748%
LSTM + WMA5	5480240	2340.991	2.3698%	97.7228%
LSTM + WMA10	5668812	2380.927	2.8758%	97.6444%
LSTM + WMA20	31576626	5619.308	4.5150%	86.8790%

Although the 5-period WMA model maintained a relatively low MAPE (2.3698%) and a high R^2 (97.7228%) , the marginal improvement over the 150-epoch level indicates that further training is no longer beneficial. The RMSE (2340.991) slightly increases, confirming that extending training leads to diminishing returns.

4.2. Performance on Hourly Closing Price Data

This section analyzes the model performance on high-frequency (hourly) data, organized by epoch.

4.2.1. Epochs = 50

The initial performance metrics for models trained for 50 epochs on the hourly data are detailed in table 8.

Table 8. Performance of LSTM models using hourly Bitcoin closing price data (Epochs = 50)

Model	MSE	RMSE	MAPE	R^2
LSTM	1920159	1385.698	0.9943%	99.0857%
LSTM + SMA5	1169020	1081.212	0.7693%	99.4434%
LSTM + SMA10	1420743	1191.949	0.9589%	99.3235%
LSTM + SMA20	1053169	1026.240	0.7772%	99.4985%
LSTM + EMA5	1939377	1392.615	1.1206%	99.0765%
LSTM + EMA10	3040809	1743.791	1.4306%	98.5521%
LSTM + EMA20	829936.6	911.0086	0.7646%	99.6048%
LSTM + WMA5	3098622	1760.290	1.2543%	98.5246%
LSTM + WMA10	1084324	1041.309	0.7345%	99.4837%
LSTM + WMA20	593531.6	770.410	0.6131%	99.7174%

For hourly data, the 20-period WMA model emerged as the best performer at this stage, achieving the lowest MAPE (0.6131%), the lowest RMSE (770.41), and the highest R^2 (99.7174%). This indicates that a longer-period WMA effectively captures short-term fluctuations in high-frequency data while maintaining strong overall explanatory power.

4.2.2. Epochs = 100

Upon extending the training duration to 100 epochs, the resulting performance metrics were recorded, as presented in table 9.

Table 9. Performance of LSTM models using hourly Bitcoin closing price data (Epochs = 100)

Model	MSE	RMSE	MAPE	R^2
LSTM	1912144	1382.803	0.9083%	99.0895%
LSTM + SMA5	2176285	1475.224	0.9783%	98.9637%
LSTM + SMA10	1346299	1160.301	0.7992%	99.3589%
LSTM + SMA20	1363511	1167.695	0.7815%	99.3508%
LSTM + EMA5	3776481	1943.317	1.3891%	98.2018%
LSTM + EMA10	674961.1	821.5602	0.6318%	99.6786%
LSTM + EMA20	2358411	1535.712	1.1249%	98.8770%
LSTM + WMA5	517980.7	719.7088	0.5540%	99.7534%
LSTM + WMA10	2860552	1691.317	1.1741%	98.6379%
LSTM + WMA20	2081283	1442.665	1.0531%	99.0090%

In the hourly data analysis, the 10-period WMA model emerged as the best performer at 100 epochs, with a MAPE of 0.5540%, an R^2 of 99.7534%, and an RMSE of 719.7088.

4.2.3. Epochs = 150

The models achieved their peak performance at the 150-epoch mark. A comprehensive breakdown of these optimal results for the hourly dataset is provided in table 10.

Table 10. Performance of LSTM models using hourly Bitcoin closing price data (Epochs = 150)

Model	MSE	RMSE	MAPE	R^2
LSTM	2470889	1571.906	1.0996%	98.8235%
LSTM + SMA5	548696.4	740.7405	0.5688%	99.7387%
LSTM + SMA10	343174.7	585.8112	0.4895%	99.8366%
LSTM + SMA20	2365877	1538.141	1.2987%	98.8735%
LSTM + EMA5	3374277	1836.920	1.2535%	98.3933%
LSTM + EMA10	1376443	1173.219	1.1914%	99.3446%
LSTM + EMA20	2321294	1523.579	1.1104%	98.8947%
LSTM + WMA5	2531340	1591.019	1.0985%	98.7947%
LSTM + WMA10	1744866	1320.934	0.8712%	99.1692%
LSTM + WMA20	1538410	1240.327	0.8279%	99.2675%

For hourly data, the 10-period SMA model achieved outstanding results, with the lowest MAPE (0.4895%) and an R^2 of 99.8366%. Its superior performance demonstrates its capability to capture rapid changes in high-frequency data while maintaining low prediction errors.

4.2.4. Epochs = 200

Finally, the results from the 200-epoch training phase, which suggested the onset of diminishing returns, are summarized in table 11.

Table 11. Performance of LSTM models using hourly Bitcoin closing price data (Epochs = 200)

Model	MSE	RMSE	MAPE	R^2
LSTM	3813678	1952.864	1.2124%	98.1841%
LSTM + SMA5	1552767	1246.101	1.0414%	99.2606%
LSTM + SMA10	2067266	1437.799	1.0339%	99.0157%
LSTM + SMA20	3191260	1786.410	1.1090%	98.4805%
LSTM + EMA5	2030210	1424.854	0.9231%	99.0333%
LSTM + EMA10	513170.8	716.359	0.5593%	99.7556%
LSTM + EMA20	941730.4	970.428	0.6816%	99.5516%
LSTM + WMA5	682648.9	826.225	0.6181%	99.6750%
LSTM + WMA10	853263.1	923.722	0.6661%	99.5937%
LSTM + WMA20	1040399	1020.000	0.8022%	99.5046%

For hourly data, the 10-period EMA model maintained strong performance with a MAPE of 0.5593% and the highest R^2 (99.7556%). However, similar to the daily data models, further training did not yield significant improvements.

4.3. Summary and Visualization of Optimal Results

Overall, 150 epochs were the optimal training duration for both daily and hourly data in this study. For daily data, the 10-period WMA was the best model, respectively the 10-period SMA in the hourly data. The comparisons between its predicted price and actual price are shown in figure 6 and figure 7.

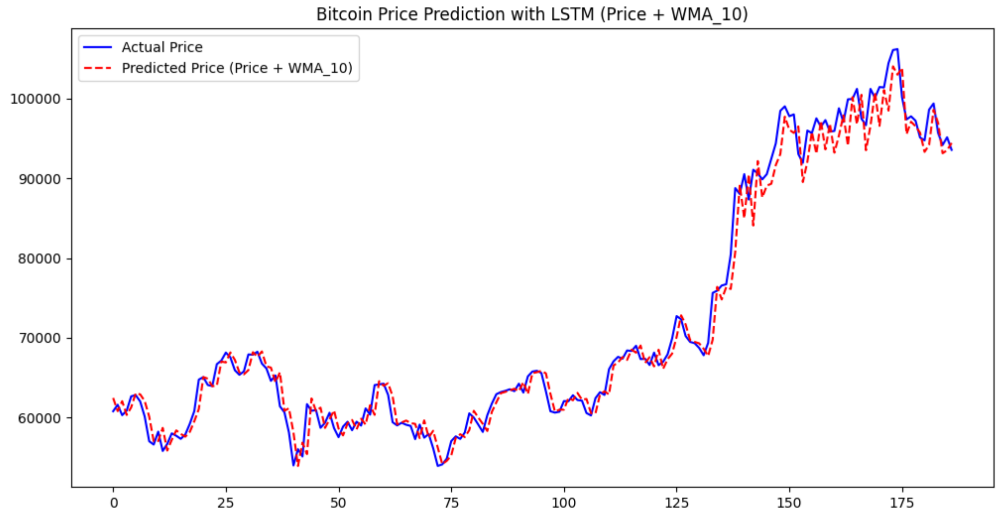

Figure 6. Comparison between actual price and predicted price using daily price (LSTM integrated with10-period WMA indicator)

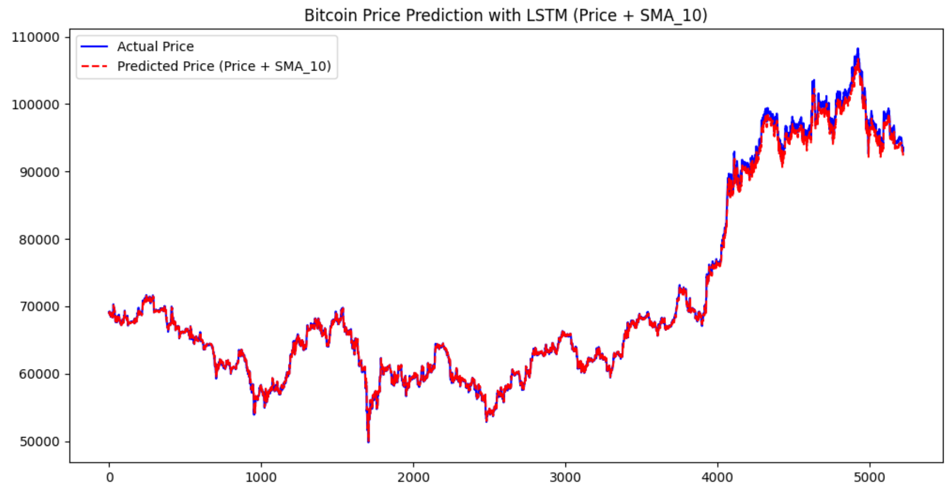

Figure 7. Comparison between actual price and predicted price using daily price (LSTM integrated with 10-period SMA indicator)

5. Conclusion

This research confirms the value of optimizing training epochs and integrating technical indicators (SMA, EMA, WMA) with LSTM models for Bitcoin price forecasting. Findings indicate 150 epochs as optimal in these experiments, with performance degrading afterward due to overfitting, highlighting the need for mitigation strategies like early stopping in practical applications. Incorporating moving averages generally improved accuracy compared to standalone LSTM by smoothing volatility and providing clearer trend signals. However, the best indicator depended on data frequency: 10-period WMA excelled for daily data (MAPE 2.1661%), while 10-period SMA was superior for hourly data (MAPE 0.4895%). This suggests WMA's recency weighting benefits lower frequencies, while SMA's smoothing effectively handles high-frequency noise. These results offer practical guidance: tailor indicator choice (WMA for daily, SMA for hourly) and epoch count (around 150 here) to the specific trading context.

While aligning with prior work on the benefits of dynamic information [13] and moving averages [12], this study adds nuance regarding the frequency-specific effectiveness of different MAs (finding SMA better for hourly, contrasting with the emphasis on WMA for high-frequency in [12]). Key limitations include relying solely on price/indicator data (excluding exogenous factors like sentiment or macroeconomic news), not formally addressing non-stationarity, performing limited hyperparameter tuning, and deriving results from single experimental runs without statistical significance testing. Future work addressing these limitations could yield more robust and comprehensive forecasting models. In conclusion, carefully tuning epochs and selecting frequency-appropriate moving average indicators significantly improves LSTM-based Bitcoin forecasting, offering valuable tools for analysts and investors.

6. Declarations

6.1. Author Contributions

Conceptualization: P.D.Q., N.H.D.; Methodology: P.D.Q., P.Q.K.; Software: P.D.Q.; Validation: N.H.D., B.D.D.; Formal Analysis: P.D.Q.; Investigation: P.D.Q.; Resources: N.H.D., P.Q.K.; Data Curation: P.D.Q.; Writing – Original Draft Preparation: P.D.Q.; Writing – Review and Editing: N.H.D., P.Q.K., B.D.D.; Visualization: P.D.Q.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

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

6.3. Funding

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

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

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

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," *bitcoin.org*, vol. 2008, no. 1, pp. 1–9, Jan. 2008.

[2] A. Vakil, M. Jain, S. Joshi, and Prof. A. Raghtate, "Bitcoin Sentiment Analysis using Machine Learning Algorithms," *international J. Sci. Res. Eng. Manag*., vol. 08, no. 008, pp. 1–14, Sep. 2024, doi: 10.55041/IJSREM37305.

[3] R. Sakariyahu, R. Lawal, R. Adigun, A. Paterson, and S. Johan, "One crash, too many: Global uncertainty, sentiment factors and cryptocurrency market," *J. Int. Financ. Mark. Inst. Money*, vol. 94, no. 1, pp. 1–16, Jul. 2024, doi: 10.1016/j.intfin.2024.102028.

[4] N. Latif, J. D. Selvam, M. Kapse, V. Sharma, and V. Mahajan, "Comparative Performance of LSTM and ARIMA for the Short-Term Prediction of Bitcoin Prices," *Australas. Account. Bus. Finance J.*, vol. 17, no. 1, pp. 256–276, 2023, doi: 10.14453/aabfj.v17i1.15.

[5] M. F. Rizvi, "ARIMA Model Time Series Forecasting," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 12, no. 5, pp. 3782–3785, May 2024, doi: 10.22214/ijraset.2024.62416.

[6] R. Yang, "Bitcoin price and return prediction based on LSTM," *Theor. Nat. Sci.*, vol. 26, no. 1, pp. 74–80, Dec. 2023, doi: 10.54254/2753-8818/26/20241021.

[7] A. García-Medina and E. Aguayo-Moreno, "LSTM–GARCH Hybrid Model for the Prediction of Volatility in Cryptocurrency Portfolios," *Comput. Econ.*, vol. 63, no. 4, pp. 1511–1542, Apr. 2024, doi: 10.1007/s10614-023-10373-8.

[8] A. Khumaidi, P. Kusmanto, and N. Hikmah, "Optimizing Bitcoin Price Predictions Using Long Short-Term Memory Algorithm: A Deep Learning Approach," *Ilk. J. Ilm.*, vol. 16, no. 1, pp. 38–45, Apr. 2024, doi: 10.33096/ilkom.v16i1.1831.38-45.

[9] N. Tripathy, S. Hota, D. Mishra, P. Satapathy, and S. K. Nayak, "Empirical Forecasting Analysis of Bitcoin Prices: A Comparison of Machine Learning, Deep Learning, and Ensemble Learning Models," *Int. J. Electr. Comput. Eng. Syst.*, vol. 15, no. 1, pp. 21–29, 2024.

[10] Z. Chen, C. Li, and W. Sun, "Bitcoin price prediction using machine learning: An approach to sample dimension engineering," *J. Comput. Appl. Math.*, vol. 365, no. 1, pp. 1–14, Feb. 2020, doi: 10.1016/j.cam.2019.112395.

[11] M.-C. Lee, "Bitcoin Trend Prediction with Attention-Based Deep Learning Models and Technical Indicators," *Systems*, vol. 12, no. 11, pp. 1–17, Nov. 2024, doi: 10.3390/systems12110498.

[12] N. A. Bakar and S. Rosbi, "Weighted Moving Average of Forecasting Method for Predicting Bitcoin Share Price using High Frequency Data: A Statistical Method in Financial Cryptocurrency Technology," *Int. J. Adv. Eng. Res. Sci.*, vol. 5, no. 1, pp. 64–69, 2018, doi: 10.22161/ijaers.5.1.11.

[13] L. Boongasame and P. Songram, "Cryptocurrency price forecasting method using long short-term memory with time-varying parameters," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 30, no. 1, pp. 435–443, Apr. 2023, doi: 10.11591/ijeecs.v30.i1.pp435-443.

[14] T. Jerez and W. Kristjanpoller, "Effects of the validation set on stock returns forecasting," *Expert Syst. Appl.*, vol. 150, no. 1, pp. 1–9, Jul. 2020, doi: 10.1016/j.eswa.2020.113271.

[15] N. Pai and V. Ilango, "LSTM Neural Network Model with Feature selection for Financial Time series Prediction," *in 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India: IEEE*, vol. 2020, no. 10, pp. 672–677, Oct. 2020, doi: 10.1109/I-SMAC49090.2020.9243376.

[16] S. Jain, N. Choudhary, and K. Jain, "Outlier detection and imputation of missing data in stock related time series mulitivariate data using LSTM autoencoder," *J. Integr. Sci. Technol.*, vol. 12, no. 3, pp. 1–7, Jan. 2024, doi: 10.62110/sciencein.jist.2024.v12.761.

[17] T. Burzykowski, M. Geubbelmans, A.-J. Rousseau, and D. Valkenborg, "Validation of machine learning algorithms," *Am. J. Orthod. Dentofacial Orthop.*, vol. 164, no. 2, pp. 295–297, Aug. 2023, doi: 10.1016/j.ajodo.2023.05.007.

[18] J. J. Murphy, "Moving Averages," *in Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. New York: New York Institute of Finance*, vol. 1999, no. 1, pp. 195–196, 1999.

[19] S. S. D. Souza and J. E. Silva, "Application of LSTM recurrent neural networks for bitcoin price prediction," *Rev. DELOS*, vol. 17, no. 61, pp. 1–21, Nov. 2024, doi: 10.55905/rdelosv17.n61-062.

[20] Y. Zheng, L. Tang, Y. Wang, K. W. Chau, and S.-Y. Cho, "Time-Sensitive Index Future Trading Trend Prediction with Deep Learning methods," *preprint, Research Square*, vol. 2023, no. 9, pp. 1–21, Sep. 2023, doi: 10.21203/rs.3.rs-3339532/v1.

[21] J. Xiao, "Stock Prediction using LSTM model," *Appl. Comput. Eng.*, vol. 8, no. 1, pp. 74–79, Aug. 2023, doi: 10.54254/2755-2721/8/20230084.

[22] Y. Jia, A. Anaissi, and B. Suleiman, "ResNLS: An improved model for stock price forecasting," *Comput. Intell.*, vol. 40, no. 1, pp. 1–20, Feb. 2024, doi: 10.1111/coin.12608.