Performance Comparison of LSTM, XGBoost, and Residual-Correction Hybrid LSTM–XGBoost Models for Bitcoin Price Forecasting

Ihsan Maulana Anwas[1], Imam Marzuki Shofi[1], Feri Fahrianto[1], Ajif Yunizar Pratama[2]
[1]Universitas Islam Negeri Syarif Hidayatullah, Jakarta, Indonesia
[2]Kyushu Institut of Technology, Fukuoka, Japan

Article Info

Article history:

Received December 04, 2025
Revised January 23, 2026
Accepted January 31, 2026

Keywords:

Bitcoin;
Hybrid LSTM-XGBoost;
LSTM;
Price Forecasting;
Time Series;
XGBoost.

ABSTRACT

The objective of this study is to systematically compare the predictive performance of Long Short-Term Memory (LSTM), Extreme Gradient Boosting (XGBoost), and a Hybrid LSTM–XGBoost model for next-day Bitcoin (BTC–USD) closing-price forecasting. The research method employs a quantitative time-series modeling approach using a decade-long daily Bitcoin price dataset. A strictly chronological train–test split and a one-step-ahead forecasting scheme are applied to prevent look-ahead bias and ensure experimental validity. Model performance is evaluated using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), symmetric Mean Absolute Percentage Error (sMAPE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination R^2 on the original price scale. The results demonstrate that the Hybrid LSTM–XGBoost model consistently outperforms the standalone LSTM and XGBoost models across all evaluation metrics, indicating superior predictive accuracy and robustness under high market volatility. The contribution of this study lies in providing a controlled, uniform, and methodologically rigorous head-to-head comparison of deep learning, machine learning, and hybrid architectures for Bitcoin price forecasting, thereby enriching the empirical literature and offering a reliable foundation for the development of adaptive decision-support systems in volatile cryptocurrency investment environments.

Corresponding Author:

Feri Fahrianto, +62 813-8333-1796,
Departement of Sains and technology, Informatics Engineering,
Universitas Islam Negeri Syarif Hidayatullah, Jakarta, Indonesia
Email: feri.fahrianto@uinjkt.ac.id

How to Cite:
I. M. Anwas, F. Fahrianto, I. M. Shofi, and Ajif Yunizar Pratama, ”Performance Comparison of LSTM, XGBoost, and Residual-Correction Hybrid LSTM–XGBoost Models for Bitcoin Price Forecasting”, *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 25, No. 2, pp. 251-262, March, 2026.

1. INTRODUCTION

Bitcoin, the first blockchain-based cryptocurrency introduced by Nakamoto in 2008, has evolved from a niche digital payment system into a globally traded financial asset with substantial market capitalization. Over the past decade, Bitcoin has attracted increasing participation from both retail and institutional investors, strengthening its role as an alternative investment instrument and digital store of value [1–3]. Despite this growing adoption, Bitcoin prices remain highly volatile and are influenced by regulatory developments, market sentiment, and macroeconomic conditions, making short-horizon price forecasting both challenging and valuable for decision support, risk management, and trading applications [4, 5].

To address the complexity of Bitcoin price dynamics, numerous studies have investigated machine learning and deep learning approaches for cryptocurrency forecasting. Traditional statistical models often struggle to capture the nonlinear and non-stationary behavior of cryptocurrency markets, whereas learning-based methods have shown competitive predictive performance for major digital assets, including Bitcoin [6, 7]. Consequently, research has increasingly focused on comparing representative model families under consistent evaluation settings to determine which approaches are most reliable for short-term forecasting [8].

Among deep-learning approaches, Long Short-Term Memory (LSTM) networks have become particularly popular because they can capture long-range temporal dependencies and nonlinear patterns in sequential financial data while mitigating the vanishing-gradient problem common in recurrent architectures [9–11]. Prior work frequently reports that LSTM-based models achieve strong performance on short-term Bitcoin price prediction and can outperform classical time-series benchmarks across various settings [12, 13].

In addition to deep sequence models, tree-based ensemble learners-especially boosted decision trees-are widely used in financial prediction because they are computationally efficient, robust to heterogeneous predictors, and effective at modeling nonlinear interactions among engineered lag and rolling-window features [14, 15]. In cryptocurrency forecasting, boosted-tree approaches have been reported to achieve competitive accuracy relative to deep-learning baselines, particularly when relevant market dynamics are captured through feature engineering [7, 8]. This makes boosted tree learning a strong standalone benchmark and a meaningful comparator for evaluating the practical value of sequence-based predictors for Bitcoin price forecasting [16].

A further line of research explores hybrid strategies that combine the complementary strengths of different learners. One practical hybrid mechanism is residual correction, in which a base predictor (e.g., LSTM) produces an initial forecast and a secondary model learns the remaining systematic error patterns to apply an additive correction. Hybrid LSTM–XGBoost frameworks have demonstrated promising results across multiple forecasting domains by leveraging the complementary strengths of temporal representation learning and gradient-boosting error modeling [17–19].

Nevertheless, truly uniform "head-to-head" comparative studies that evaluate LSTM, boosted-tree learning (e.g., XGBoost/boosted trees), and residual-correction Hybrid LSTM–XGBoost specifically for Bitcoin under a strictly consistent experimental protocol remain relatively limited [20, 21]. Differences in dataset span, market period, splitting strategy (random vs. chronological), feature engineering, and reported evaluation metrics often make it difficult to attribute performance differences to modeling choices rather than experimental design [8, 22, 23]. Therefore, this study aims to comprehensively evaluate the performance of LSTM, XGBoost, and residual-correction Hybrid LSTM–XGBoost for Bitcoin price forecasting using RMSE, MAE, sMAPE, MAPE, and R^2. This evaluation contributes to the Bitcoin-forecasting literature by providing a more like-for-like comparison across deep, tree-based, and hybrid approaches, and supports the development of investment decision-support systems that are more adaptive to volatility in cryptocurrency markets.

2. RESEARCH METHOD

The research method outlines the structured stages for analyzing, modeling, and evaluating Bitcoin price prediction using LSTM, XGBoost, and a hybrid residual-correction framework. This section explains how raw data is transformed through preprocessing, modeling, and evaluation so that each step contributes meaningfully to the final comparison among the three approaches. The complete workflow is illustrated in Figure 1, which visually summarizes the sequential steps performed throughout the research.

Figure 2 presents the workflow of the residual-correction Hybrid LSTM $\longrightarrow$ XGBoost model employed in this study. The process begins with historical Bitcoin closing prices, which are preprocessed through chronological splitting, Min-Max normalization, and the construction of 60-day sliding windows to preserve temporal causality. In the first stage, an LSTM model is trained on these sequences to learn long-term temporal dependencies and produce a base next-day price forecast. The prediction errors (residuals) are then computed as the difference between the actual prices and the LSTM outputs. In the second stage, an XGBoost regressor is trained to model these residuals using lagged prices, rolling statistical features, and residual lags derived strictly from information available up to time t-1. During inference, the predicted residual is added to the LSTM forecast to obtain the final hybrid prediction, enabling the model to combine the sequence-learning capability of LSTM with the local nonlinear error-correction strength of gradient boosting.

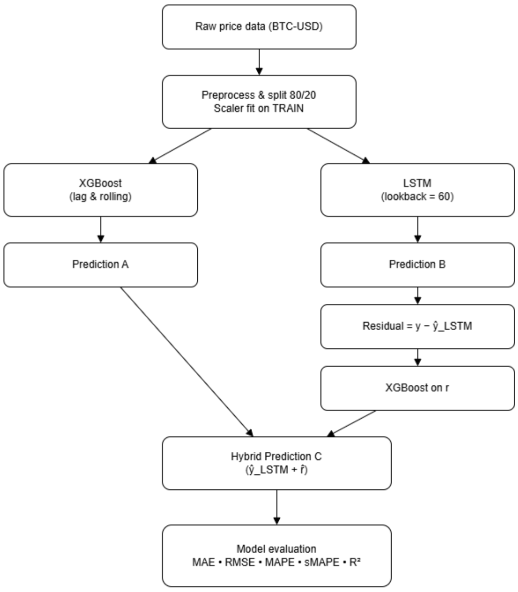

Figure 1. Research Method

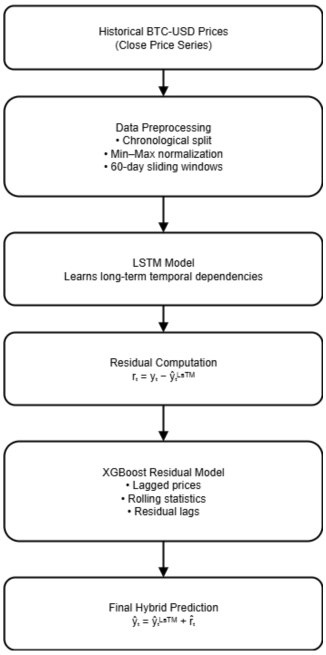

Figure 2. Hybrid LSTM $\longrightarrow$ XGBoost model

2.1. Dataset and Split

The dataset comprises 3,653 daily Bitcoin closing prices (BTC-USD) spanning 22 December 2014 to 21 December 2024, obtained from Yahoo Finance. The analysis focuses on the Close series, which is preserved in chronological order to maintain temporal causality. The data are partitioned chronologically into an 80% training set (2,922 days; 22-12-2014 to 21-12-2022) and a 20% test set (731 days; 22-12-2022 to 21-12-2024), without shuffling. This setup ensures that test-set predictions are strictly out-of-sample in time, so that the models are evaluated in a realistic forward-looking scenario [1, 2].

2.2. Data Preparation and Feature Engineering

The analysis uses only the Close series as the target variable. The data are first screened for missing values and obvious anomalies. Values are then normalized to [0,1] using a Min-Max scaler fitted on the training set only and subsequently applied to both training and test partitions to avoid look-ahead bias. Supervised learning samples are generated via sliding windows with a 60-day look-back horizon [11, 24]. To maintain strict temporal causality, the first test window is seeded with the last 60 days of the training span (overlap), ensuring that the prediction for day t uses information only up to t-1.

For the LSTM model, these windows are reshaped into three-dimensional sequence tensors suitable for recurrent modelling. For the tree-based model (XGBoost), lagged and aggregated features are constructed from the same normalized series. Specifically, Section 2.3 details the set of lagged prices and rolling statistics used as predictors in the gradient-boosting and hybrid configurations.

2.3. Models

LSTM. The first model is a univariate Long Short-Term Memory (LSTM) network. It consists of an LSTM layer with 64 units, followed by a Dropout layer with a rate of 0.2, a Dense layer with 32 units and ReLU activation, and a final output Dense layer with a single neuron for next-day price prediction. The network is trained using the Adam optimizer with mean squared error (MSE) loss. Early stopping with a patience of 20 epochs is applied based on validation loss, with a maximum of 300 epochs and a batch size of 32. The last 20% of the training set is used as the validation split, and shuffle=False is enforced to preserve temporal order.

XGBoost. The second model uses Extreme Gradient Boosting (XGBoost) for regression, applied directly to the price level. The input feature space comprises lagged prices yt-k for k=1, . . . ,30 and rolling means over 7, 14, and 30 days, each computed only from information available up to time t-1. The main hyperparameters are: n_estimators = 600, max_depth = 5, learning_rate = 0.05, subsample = 0.9, colsample_bytree = 0.8, and reg_lambda = 1.0. This configuration balances model flexibility, regularization strength, and computational efficiency.

Hybrid (LSTM $\longrightarrow$ XGBoost, residual correction). The third model is a residual-correction hybrid in which XGBoost is trained to correct the errors of the LSTM. In the first stage, the LSTM is trained as described above and produces initial predictions $\hat{y}t^{LSTM}$. on the training period. Residual targets are then computed as shown in Equation 1.

$$rt = yt - \hat{y}tLSTM \tag{1}$$

In the second stage, an XGBoost regressor is trained on these residuals using price-lag features (as in the standalone XGBoost model) and residual lags (1-7) together with rolling statistics. During testing, residual features and rolling statistics are computed solely from information available up to time t-1. The final hybrid prediction for day t is obtained by adding the residual prediction $\hat{r}$t to the LSTM prediction, as shown in Equation 2. Equation 1 thus defines the hybrid LSTM $\longrightarrow$ XGBoost model as a residual-correction architecture, where the sequence model captures long-term temporal structure and the tree-based learner focuses on correcting remaining systematic errors.

$$\hat{y}Hyb = \hat{y}tLSTM + \hat{r}t \tag{2}$$

2.4. Experimental Setup

All experiments were conducted in Python 3 on Google Colab, using TensorFlow 2 for the LSTM implementation, XGBoost for gradient-boosting, and scikit-learn utilities for preprocessing and evaluation. Random seeds were set across NumPy, Python, TensorFlow, and XGBoost to improve reproducibility. The code used in this study is available from the authors upon reasonable request.

The experimental protocol follows the chronological 80/20 split described in Section 2.1, with no shuffling at any stage. Preprocessing retains only the Close series, applies Min-Max normalization fitted to the training set, and forms 60-day sliding windows; the first test window is seeded with the last 60 days of the training span to ensure strictly causal evaluation. LSTM training preserves temporal order (shuffle=False) and employs early stopping (patience = 20) with a maximum of 300 epochs and a batch size of 32, as described in Section 2.3. Evaluation adopts a one-step-ahead (teacher-forcing) protocol on the test set, where each prediction for day t uses only information available up to t-1. Section 2.5 details the reported metrics and the naïve lag-1 baseline.

2.5. Evaluation Metrics

Evaluation uses a one-step-ahead setup: the prediction for day t only uses information available up to day t-1. All performance scores are computed on the original USD scale after reversing any normalization or scaling transformations, so that errors are directly interpretable in monetary terms. This setup mirrors a realistic trading or risk-management scenario in which future prices are not used to form current predictions. In addition, a simple naïve lag-1 baseline is included, defined as shown in Equation 3.

$$\hat{y}t = yt - 1 \tag{3}$$

Model performance is summarized using mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE), symmetric mean absolute percentage error (sMAPE), and the coefficient of determination (R2). MAE quantifies the average absolute deviation in dollars, while RMSE is similar but penalizes larger errors more strongly. MAPE reports the average percentage error, whereas sMAPE provides a more balanced measure when values vary widely over time. R2 indicates the proportion of the observed variance in Bitcoin prices that the model explains. For MAE, RMSE, MAPE, and sMAPE, lower values indicate better performance; for R2, higher values are preferred. In addition, directional accuracy-the percentage of days for which the predicted sign of the price change matches the actual sign-is reported to evaluate the models' ability to predict the direction of price movements, which is crucial for many trading and risk-management applications [1, 11, 8, 16].

3. RESULT AND ANALYSIS

3.1. Experimental Overview

The experiments use 3,653 daily Bitcoin (BTC-USD) closing prices covering the period from 22 December 2014 to 21 December 2024, obtained from Yahoo Finance. All observations are ordered by date and split chronologically into 80% training data (2,922 days; 22-12-2014 to 21-12-2022) and 20% test data (731 days; 22-12-2022 to 21-12-2024), without shuffling, to preserve time-series causality. As described in Section 2, preprocessing focuses on the Close series, applies Min–Max normalization fitted on the training data only, and generates supervised samples using 60-day sliding windows. The first test window is seeded with the last 60 days of the training span to ensure that the prediction for day t uses information only up to t-1.

Three models are compared under this setup: a univariate LSTM, an XGBoost regressor using lag and rolling features, and a residual-correction Hybrid LSTM $\longrightarrow$ XGBoost in which XGBoost learns to model the residuals of the LSTM, as defined in Equation 1. The evaluation protocol adopts a one-step-ahead (teacher-forcing) scheme on the test set and reports MAE, RMSE, MAPE, sMAPE, and R2 on the original USD scale. A simple naïve lag-1 benchmark, $\hat{y}$t=yt-1, is used in the analysis to provide context for model performance, although its detailed scores are not tabulated.

3.2. Quantitative Results

Table 1. Quantitative Results

Model	MAE	RMSE	MAPE (%)	sMAPE (%)	R^2
Hybrid (LSTM+XGB-Residual)	**1,372.22**	**2,260.52**	2.52	2.56	0.9886
LSTM	**1,444.78**	**2,385.95**	2.54	2.58	0.9873
XGBoost	**4,144.81**	**9,326.96**	6.26	6.90	0.8065

In Table 1 (Quantitative Results), the Hybrid model delivers the best results across all metrics. Compared with the standalone LSTM, the Hybrid reduces RMSE by about 5.3% (from 2,385.95 to 2,260.52) and MAE by about 5.0% (from 1,444.78 to 1,372.22), while also slightly improving MAPE (from 2.54% to 2.52%) and sMAPE (from 2.58% to 2.56%) and increasing R2 from 0.9873 to 0.9886. Compared with pure XGBoost, the Hybrid shows substantially larger gains: RMSE drops by roughly three quarters, MAPE is reduced by more than half, and R2 improves from 0.8065 to 0.9886.

From a practical perspective, an MAPE of approximately 2.5% indicates that the Hybrid model closely tracks Bitcoin's daily movements despite the asset's high volatility. The relatively small gap between MAE and RMSE for the Hybrid also suggests that very large errors are relatively infrequent. Overall, the quantitative evidence supports the conclusion that residual correction meaningfully enhances short-term Bitcoin price forecasting compared with standalone LSTM and XGBoost.

3.3. Visual Comparasion

In the visual comparison over the test period, 22 December 2022 to 21 December 2024, each model's prediction curve is aligned with the actual price series to assess path closeness, phase lag, and the ability to follow peaks and troughs. Overall, these visualizations complement the quantitative findings in Section 3.2 by illustrating how each approach responds to rapid regime changes in Bitcoin prices, particularly during the sharp uptrend in 2024.

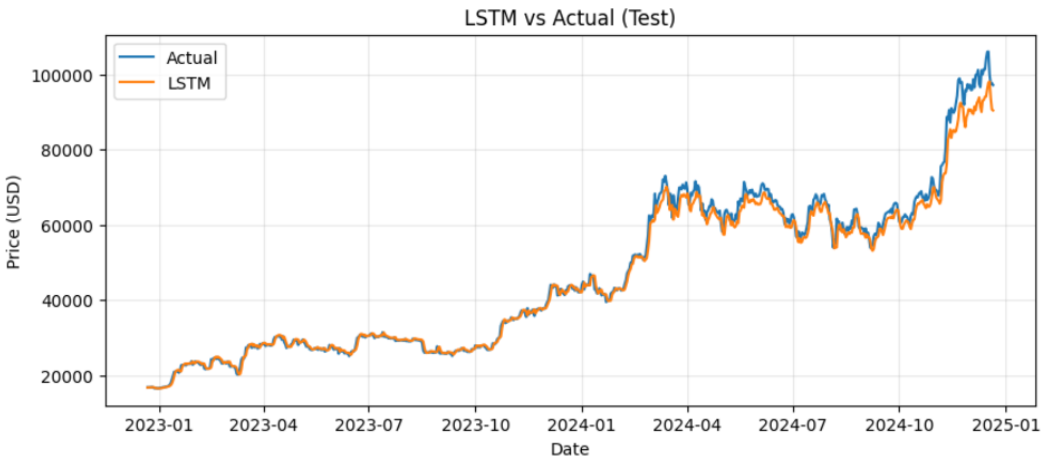

Figure 3. LSTM vs Actual

In Figure 3 (LSTM vs Actual), the LSTM model follows the long-term trend well and maintains a relatively close distance to the actual curve across most of the test horizon. However, a noticeable response lag appears when slope changes are very rapid, particularly during the strong rally phases in early to mid-2024, when predictions tend to lag and occasionally underestimate near-local peaks. After these surge phases, the LSTM curve re-converges toward the actual series and remains reasonably accurate during correction phases.

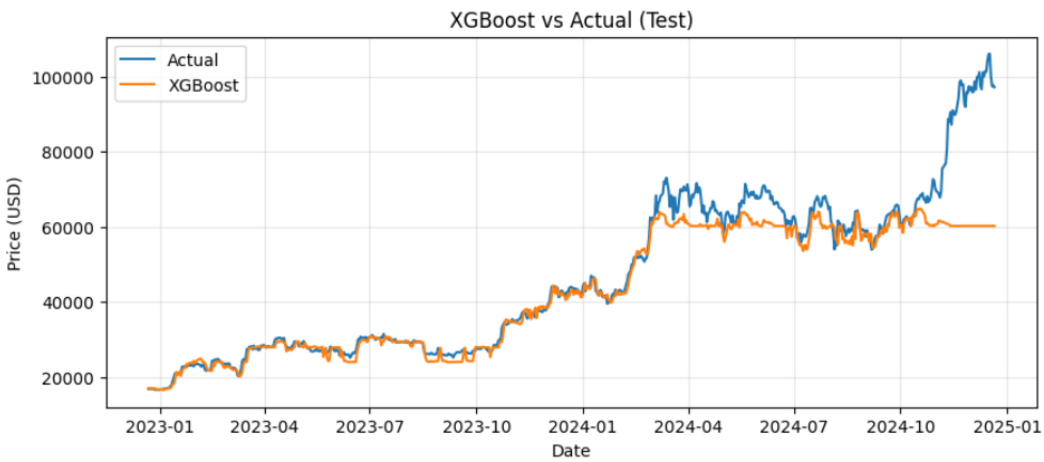

Figure 4. XGBoost vs Actual

In Figure 4 (XGBoost vs Actual), the XGBoost predictions appear smoother but tend to flatten around certain price levels, indicating a tendency to revert toward a local average when lag and rolling features do not fully capture acceleration dynamics. During the rapid uptrend in 2024, the prediction curve appears "flat" relative to the actual series, resulting in wide deviations around peaks and during sharp reversals. This visual behavior highlights the limitations of the pure XGBoost model in tracking fast, large-amplitude movements in Bitcoin prices.

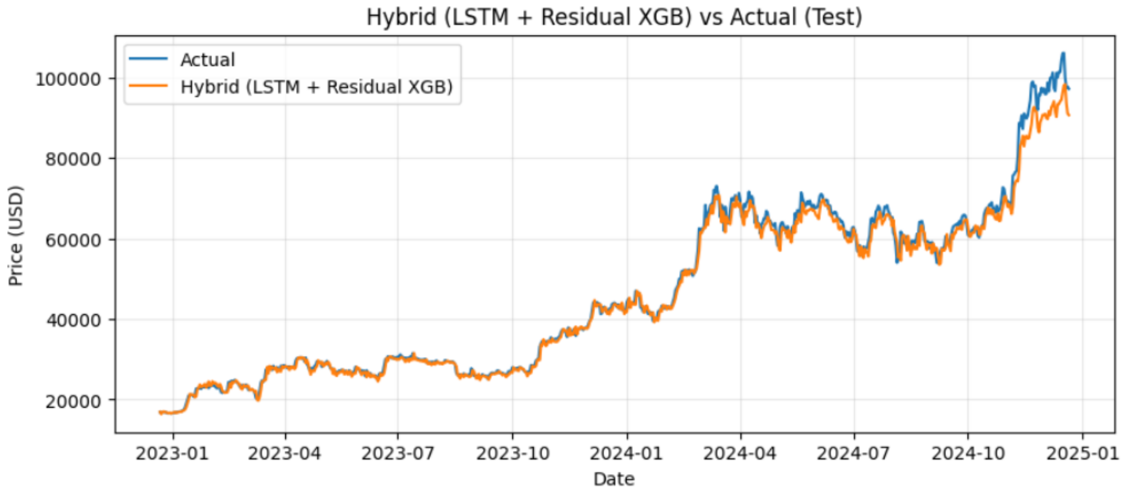

Figure 5. Hybrid LSTM-Residual XGBoost vs Actual

In Figure 5 (Hybrid LSTM–Residual XGBoost vs Actual), the hybrid curve most closely follows the actual data, with narrower deviations across the test window. Residual correction via XGBoost effectively reduces LSTM bias in turbulent segments, thereby enabling more accurate tracking of peaks and troughs without excessive smoothing. Phase lag also appears reduced relative to the standalone LSTM, particularly around major breakouts and pullbacks in 2024. Visually, these results confirm that the hybrid approach successfully combines the long-term dependency modelling of LSTM with the local non-linear sensitivity of gradient boosting on residuals.

3.4. Error Behavior

An analysis of error behavior provides additional insight into where and when the models tend to deviate from the actual values. Overall, the residual distribution for the Hybrid model is the tightest around zero across the test horizon, indicating relatively small and stable deviations relative to the other two models. This pattern is consistent with the metrics in Table 1: once LSTM captures long-term dependencies, residual correction via XGBoost reduces remaining local non-linear errors and narrows the error band.

Temporally, the largest residual spikes tend to coincide with sharp regime shifts in 2024. In these segments, the LSTM model is occasionally delayed in responding to strong acceleration, leading to underestimation near peaks, whereas pure XGBoost is pulled toward the local mean, producing flatter curves and larger errors around breakouts and pullbacks. The Hybrid mitigates both sources of bias: residual correction lifts predictions near peaks and lowers them near troughs without over-smoothing, thereby making the predicted path more closely adhere to the actual curve.

In terms of the error scale, there is evidence of heteroskedasticity: absolute errors increase as the price level rises. This is natural for level-based metrics such as MAE and RMSE, whereas relative metrics such as MAPE and sMAPE remain approximately 2.5% for the best model. Methodologically, if an analysis objective demands variance stability across price levels, a log-price transformation or modelling in terms of returns/differenced series can be considered to produce a more homogeneous error spread.

Visual inspection of residual patterns also suggests that a portion of the remaining error structure is associated with sudden market shocks (e.g., macroeconomic news or rapid shifts in crypto sentiment). This creates opportunities for improvement by incorporating exogenous features, such as volume, on-chain indicators, or event markers, into the correction component, thereby capturing error patterns not fully explained by historical prices alone. Overall, the error behavior supports the conclusion that residual correction functions effectively as a bias corrector for LSTM, particularly during periods of high volatility.

3.5. Discussion

The results in Sections 3.2–3.4 show that the Hybrid LSTM $\longrightarrow$ XGBoost residual-correction approach consistently outperforms single models on the one-step-ahead horizon [22, 23]. Conceptually, this performance can be explained by a clear division of roles: LSTM captures long-term dependencies and trend/seasonality information within the 60-day window, while XGBoost serves as a nonlinear corrector of the residuals not captured by the sequence structure. When the market enters a phase of rapid regime change-such as the sharp rally throughout 2024-LSTM alone tends to under-estimate near the peaks because the rate of change exceeds its generalization capacity. The residual-boosting component adds the local sensitivity needed to correct this bias without sacrificing the ability to track long-run trends.

In contrast, pure XGBoost, which relies solely on lag and rolling features of the price level, shows a tendency to flatten around local averages when acceleration dynamics increase [8, 24, 16]. This pattern is consistent with the visually flatter curves in the second half of 2024 and with the lower quantitative metrics. The implication is that direct regression to the price level with boosted trees requires richer feature design (e.g., log-price, returns, rolling volatility, or time-of-day/time-of-year markers) to respond quickly to changes in slope. Even without aggressive feature expansion, however, the hybrid strategy already exhibits a better bias–variance trade-off: the burden of modelling long-term dynamics is assigned to LSTM, while XGBoost focuses on the error structure, which is typically lower-dimensional and more local.

From an evaluation perspective, the one-step-ahead scheme ensures that there is no look-ahead bias: predictions for day t use information only up to t-1, normalization is fitted solely on training data, and LSTM training is performed with shuffle=False and early stopping. This design ensures a fair inter-model comparison and that the reported performance is accountable. Taken together, the findings affirm that combining sequence models with residual-boosting correctors is an effective and practical strategy for highly volatile daily Bitcoin price prediction, and they provide empirical support for using such hybrids as building blocks in decision-support systems for trading and risk management [25, 26].

These findings are consistent with previous studies on Bitcoin price forecasting that report superior performance of hybrid and ensemble models over single learners [23, 25]. In particular, studies such as Bouteska et al. [8] and Omole and Enke [23] show that hybrid deep-learning approaches tend to achieve lower prediction errors under high market volatility. Compared with earlier hybrid architectures such as CNN–LSTM or GRU-based models [22–24, 27], the residual-correction LSTM $\longrightarrow$ XGBoost approach used in this study offers a simpler two-stage structure while achieving competitive or lower MAE and MAPE values on a longer daily BTC–USD dataset. This suggests that explicit residual modeling is an effective and parsimonious strategy for improving short-horizon forecasts of Bitcoin.

From a practical standpoint, the proposed forecasting framework is well aligned with real-world trading and decision-support systems. The one-step-ahead setup reflects a realistic daily trading scenario in which predictions generated after market close can inform next-day positioning. The relatively low percentage errors achieved by the Hybrid model indicate its potential as a forecasting engine to support risk-aware decisions, such as trend confirmation, position sizing, or stop-loss calibration. While the model is not intended as a standalone trading strategy, it can be integrated into trading dashboards or automated pipelines alongside technical rules and risk-management modules.

3.6. Reproducibility and Limitations

All experiments are designed to be consistently replicable. The data are ordered chronologically and split by time into 80% training and 20% testing; Min–Max normalization is fit only on the training data and then applied back to both training and test sets. For LSTM, time-series windowing uses a 60-day look-back, with the test window initiated by the tail of the training data to preserve causality. LSTM training is conducted without shuffling (shuffle = False), using early stopping with patience 20, batch size 32, and a maximum of 300 epochs. On the boosting side, XGBoost is trained with price-lag features ($k = 1 \ldots 30$) and rolling means (7/14/30 days), all computed from information $\leq$ t-1 in the hybrid model, XGBoost models LSTM residuals on the training data and at test time always uses residual lags up to t-1. Seeds are set in NumPy, Python, TensorFlow, and XGBoost to improve result consistency, with the caveat of small nondeterminism on GPU execution.

A key limitation of this study lies in its univariate design, which relies exclusively on the Bitcoin closing price as the input variable. Although close-only modeling is widely used in prior studies [8, 22, 23, 28] and allows for fair comparison across models, it does not account for exogenous factors such as trading volume, volatility indicators, on-chain metrics, macroeconomic variables, or market sentiment. Consequently, sudden price movements driven by external events may not be fully captured by historical price information alone. In addition, modeling price levels rather than transformed variables (e.g., log-prices or returns) introduces heteroskedasticity, as absolute prediction errors tend to increase at higher price levels, which is a common characteristic of cryptocurrency markets.

The evaluation protocol follows a one-step-ahead (teacher forcing) scheme, meaning the prediction on day t only uses historical information up to t-1. All metrics-MAE, RMSE, MAPE, sMAPE, and R^2-are calculated on the original (USD) scale to maintain practical interpretability. As a simple comparator, a naïve lag-1 baseline is prepared to provide context for how far the models learn beyond a strategy that merely shifts the previous day's value. With this design, look-ahead bias is avoided and inter-model comparisons are made under fair conditions.

The main limitations of this study lie in the scope of the evaluation, which is limited to the one-step-ahead horizon. The results do not yet reflect multi-step performance without updates (e.g., projecting several days ahead at once without observing the latest daily actuals). In addition, the modeling is univariate-using only the closing price-so exogenous variables such as volume, implied volatility, on-chain indicators, and macro event markers are not yet considered. The volatility characteristics of Bitcoin also imply heteroskedasticity (absolute errors tend to grow at higher price levels), which is natural for level-based metrics and can affect error stability across market regimes. Finally, although hyperparameters are chosen reasonably, a wider search space (or a Bayesian optimization approach) could find more optimal configurations.

To increase robustness and the breadth of findings, the following development directions are suggested: (i) strict walk-forward evaluation for multi-step forecasting without updates with residuals projected recursively; (ii) feature expansion via variance-stabilizing transformations (e.g., log-price, differencing/returns, and rolling volatility), as well as the inclusion of exogenous variables (volume, on-chain indicators, and event markers); (iii) time-series cross-validation (TimeSeriesSplit) accompanied by comparative significance testing—such as the Diebold–Mariano test-to assess performance differences across models; and (iv) rolling re-training or adaptive learning strategies to address rapid market-regime shifts. With these steps, reproducibility is maintained while strengthening external validity and model resilience in more challenging scenarios.

4. CONCLUSION

This study evaluated next-day Bitcoin (BTC–USD) closing-price forecasting under a strictly chronological, one-step-ahead evaluation protocol by comparing representative deep sequential, tree-based boosting, and residual-correction hybrid approaches. The results indicate that residual correction provides a practical performance benefit for short-horizon Bitcoin forecasting, with the Hybrid LSTM $\longrightarrow$ XGBoost model achieving the best overall accuracy on the test period. Specifically, the Hybrid model attains MAE = 1,372.22, RMSE = 2,260.52, MAPE = 2.52%, sMAPE = 2.56%, and R^2 = 0.9886, demonstrating close tracking of daily price movements despite high market volatility. The comparative evidence further supports that the Hybrid approach consistently outperforms the standalone baselines across the reported metrics under the same experimental setting. These results are consistent with the proposed hypothesis, indicating that residual correction via Hybrid LSTM $\longrightarrow$ XGBoost improves forecasting accuracy over the standalone baselines under a unified protocol.

From a methodological perspective, this work contributes a like-for-like benchmark for Bitcoin forecasting by enforcing a unified experimental protocol (chronological splitting, causal one-step-ahead evaluation, and metric reporting on the original USD scale) and by demonstrating an interpretable two-stage residual-learning design that is straightforward to integrate into forecasting dashboards and decision-support pipelines.

Several limitations should be acknowledged. The experiments focus on a single asset (Bitcoin) and a single temporal granularity (daily closing prices), and model performance may vary across other cryptocurrencies, different sampling frequencies, or alternative market regimes. Moreover, the evaluation emphasizes point-forecast accuracy; uncertainty quantification and transaction-cost-aware backtesting are not included, so real trading utility is not directly measured. Future work should extend the study to multivariate inputs (e.g., volume, technical indicators, macro variables, or sentiment signals), adopt walk-forward validation across market regimes, and incorporate probabilistic forecasting and backtesting to better reflect deployment-oriented decision-making requirements.

5. ACKNOWLEDGEMENTS

Thanks to the Faculty of Science and Technology, the Informatics Engineering Study Program, and the supervisors from the Informatics Study Program.

6. DECLARATIONS

AI USAGE STATEMENT

During manuscript preparation, the authors used ChatGPT (OpenAI) to polish the language and improve text clarity. Following the use of this tool, the authors carefully reviewed, revised, and verified the final wording, and they take full responsibility for all content in the publication.

AUTHOR CONTIBUTION

I.M.A., I.M.S., F.F., and A.Y.P. contributed to conceptualization, methodology design, data curation and preprocessing, machine-learning implementation, visualization, validation, and the analysis and interpretation of results. I.M.A., I.M.S., F.F., and A.Y.P. also drafted the initial manuscript, carried out revisions, and approved the final version of the manuscript.

FUNDING STATEMENT
This research received no specific funding from public, commercial, or non-profit funding agencies. All research activities were carried out with support from the authors' institution.

COMPETING INTEREST
The authors declare that there are no conflicts of interest related to the research, authorship, or publication of this article. The study was conducted independently without any financial, commercial, or personal relationships that could be perceived as influencing the results. No external funding sources or sponsoring organizations had any role in the design, data collection, analysis, interpretation, or decision to publish the findings. The authors affirm that the research was carried out with full academic integrity and transparency. Furthermore, all authors have reviewed and approved the final version of the manuscript and agree to take responsibility for its content..

REFERENCES

[1] J. Chen, "Analysis of bitcoin price prediction using machine learning," *Journal of Risk and Financial Management*, vol. 16, no. 1, pp. 1–25, Jan. 2023, https://doi.org/10.3390/jrfm16010051.

[2] T. Kim, H. Jo, W. Choi, and B.-G. Jang, "Bitcoin price direction forecasting and market variables," *Journal of Futures Markets*, vol. 45, no. 10, pp. 1579–1600, Oct. 2025, https://doi.org/10.1002/fut.70010.

[3] M. Paskaleva and I. Vasenska, "The btc price prediction paradox through methodological pluralism," *Risks*, vol. 13, no. 10, pp. 1–43, Oct. 2025, https://doi.org/10.3390/risks13100195.

[4] A. Bouteska, S. Mefteh-Wali, and T. Dang, "Predictive power of investor sentiment for bitcoin returns evidence from covid-19 pandemic," *Technological Forecasting and Social Change*, vol. 184, no. 1, pp. 1–13, Nov. 2022, https://doi.org/10.1016/j.techfore.2022.121999.

[5] A. Mikhaylov, H. Dincer, S. Yuksel, G. Pinter, and Z. A. Shaikh, "Bitcoin mempool growth and trading volumes:integrated approach based on qrof multi-swara and aggregation operators," *Journal of Innovation Knowledge*, vol. 8, no. 3, pp. 100 378–100 388, Jul. 2023, https://doi.org/10.1016/j.jik.2023.100378.

[6] E. Akyildirim, A. Goncu, and A. Sensoy, "Prediction of cryptocurrency returns using machine learning," *Annals of Operations Research*, vol. 297, no. 1-2, pp. 3–36, Feb. 2021, https://doi.org/10.1007/s10479-020-03575-y.

[7] A. A. Oyedele, A. O. Ajayi, L. O. Oyedele, S. A. Bello, and K. O. Jimoh, "Performance evaluation of deep learning and boosted trees for cryptocurrency closing price prediction," *Expert Systems with Applications*, vol. 213, no. 3, pp. 119 233–119 246, Mar. 2023, https://doi.org/10.1016/j.eswa.2022.119233.

[8] A. Bouteska, M. Z. Abedin, P. Hajek, and K. Yuan, "Cryptocurrency price forecasting a comparative analysis of ensemble learning and deep learning methods," *International Review of Financial Analysis*, vol. 92, no. 12, p. 103055, Mar. 2024, https://doi.org/10.1016/j.irfa.2023.103055.

[9] A. Dutta, S. Kumar, and M. Basu, "A gated recurrent unit approach to bitcoin price prediction," *Journal of Risk and Financial Management*, vol. 13, no. 2, pp. 23–33, Feb. 2020, https://doi.org/10.3390/jrfm13020023.

[10] P. Lamothe Fernandez, D. Alaminos, and Lamothe-Lopez, "Deep learning methods for modeling bitcoin price," *Mathematics*, vol. 8, no. 8, pp. 1245–1255, jul 2020, https://doi.org/10.3390/math8081245.

[11] P. L. Seabe, C. R. B. Moutsinga, and E. Pindza, "Forecasting cryptocurrency prices using lstm, gru, and bi-directional lstm : A deep learning approach," *Fractal and Fractional*, vol. 7, no. 2, pp. 203–213, Feb. 2023, https://doi.org/10.3390/fractalfract7020203.

[12] 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," *Australasian Accounting, Business and Finance Journal*, vol. 17, no. 1, pp. 256–276, 2023, https://doi.org/10.14453/aabfj.v17i1.15.

[13] M. Ula, V. Ilhadi, and Z. M. Sidek, "Comparing long short-term memory and random forest accuracy for bitcoin price forecasting," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 23, no. 2, pp. 259–272, Jan. 2024, https://doi.org/10.30812/matrik.v23i2.3267.

[14] S. Ranjan, P. Kayal, and M. Saraf, "Bitcoin price prediction:a machine learning sample dimension approach," *Computational Economics*, vol. 61, no. 4, pp. 1617–1636, Apr. 2023, https://doi.org/10.1007/s10614-022-10262-6.

[15] A. M. Rather, "A new method of ensemble learning: Case of cryptocurrency price prediction," *Knowledge and Information Systems*, vol. 65, no. 3, pp. 1179–1197, Mar. 2023, https://doi.org/10.1007/s10115-022-01796-0.

[16] N. Kiranmai Balijepalli and V. Thangaraj, "Prediction of cryptocurrency's price using ensemble machine learning algorithms," *European Journal of Management and Business Economics*, vol. 34, no. 4, pp. 1–17, Jan. 2025, https://doi.org/10.1108/EJMBE-08-2023-0244.

[17] L. Semmelmann, S. Henni, and C. Weinhardt, "Load forecasting for energy communities: A novel lstm-xgboost hybrid model based on smart meter data," *Energy Informatics*, vol. 5, no. 1, pp. 24–34, Sep. 2022, https://doi.org/10.1186/s42162-022-00212-9.

[18] Z. Ye, Y. Wu, H. Chen, Y. Pan, and Q. Jiang, "A stacking ensemble deep learning model for bitcoin price prediction using twitter comments on bitcoin," *Mathematics*, vol. 10, no. 8, pp. 1307–1320, Apr. 2022, https://doi.org/10.3390/math10081307.

[19] F. Rodrigues and M. Machado, "High-frequency cryptocurrency price forecasting using machine learning models:a comparative study," *Information*, vol. 16, no. 4, pp. 300–310, Apr. 2025, https://doi.org/10.3390/info16040300.

[20] M. Liu, G. Li, J. Li, X. Zhu, and Y. Yao, "Forecasting the price of bitcoin using deep learning," *Finance Research Letters*, vol. 40, no. 1, pp. 1–7, May 2021, https://doi.org/10.1016/j.frl.2020.101755.

[21] P. Boozary, S. Sheykhan, and H. GhorbanTanhaei, "Forecasting the bitcoin price using the various machine learning : A systematic review in data-driven marketing," *Systems and Soft Computing*, vol. 7, no. 12, Dec. 2025.

[22] Y. Li and W. Dai, "Bitcoin price forecasting method based on cnn-lstm hybrid neural network model," *The Journal of Engineering*, vol. 2020, no. 13, pp. 344–347, Jul. 2020, https://doi.org/10.1049/joe.2019.1203.

[23] O. Omole and D. Enke, "Deep learning for bitcoin price direction prediction: Models and trading strategies empirically compared," *Financial Innovation*, vol. 10, no. 1, pp. 117–127, Aug. 2024, https://doi.org/10.1186/s40854-024-00643-1.

[24] I. S. Kervanci, M. F. Akay, E. Ozceylan, G. . T. IT Department, Gaziantep University, A. . T. Computer Engineering Department, Cukurova University, and G. . T. Industrial Engineering Department, Gaziantep University, "Bitcoin price prediction using lstm, gru and hybrid lstm-gru with bayesian optimization, random search, and grid search for the next days," *Journal of Industrial and Management Optimization*, vol. 20, no. 2, pp. 570–588, 2024, https://doi.org/10.3934/jimo.2023091.

[25] Y. Li, S. Jiang, X. Li, and S. Wang, "Hybrid data decomposition-based deep learning for bitcoin prediction and algorithm trading," *Financial Innovation*, vol. 8, no. 1, pp. 31–42, Apr. 2022, https://doi.org/10.1186/s40854-022-00336-7.

[26] A. Golnari, M. H. Komeili, and Z. Azizi, "Probabilistic deep learning and transfer learning for robust cryptocurrency price prediction," *Expert Systems with Applications*, vol. 255, no. 12, pp. 124 404–124 414, Dec. 2024, https://doi.org/10.1016/j.eswa.2024.124404.

[27] A. Ladhari and H. Boubaker, "Deep learning models for bitcoin prediction using hybrid approaches with gradient-specific optimization," *Forecasting*, vol. 6, no. 2, pp. 279–295, Apr. 2024, https://doi.org/10.3390/forecast6020016.

[28] R. Kaur, M. Uppal, D. Gupta, S. Juneja, S. Y. Arafat, J. Rashid, J. Kim, and R. Alroobaea, "Development of a cryptocurrency price prediction model: Leveraging gru and lstm for bitcoin, litecoin and ethereum," *PeerJ Computer Science*, vol. 11, no. 3, pp. 2675–2685, Mar. 2025, https://doi.org/10.7717/peerj-cs.2675.

[This page is intentionally left blank.]