SINERGI Vol. 30, No. 2, June 2026: 657-670
http://publikasi.mercubuana.ac.id/index.php/sinergi
http://doi.org/10.22441/sinergi.2026.2.027

Bitcoin price prediction using Autoencoder-based GRU and LSTM models

Nurun Nafisah*, Yuni Yamasari, Ervin Yohannes
Department of Informatics, Faculty of Engineering, Universitas Negeri Surabaya, Indonesia

Abstract
The high volatility of Bitcoin prices presents a major challenge in forecasting, as predictive models must capture both short-term fluctuations and long-term trends. Despite extensive studies on deep learning for cryptocurrency prediction, there remains a lack of systematic comparative analysis of autoencoder-based recurrent architectures, particularly AE-LSTM and AE-GRU, across both univariate and multivariate input settings, with statistical validation. This study analyzes Bitcoin price data from 2018 to 2025 to evaluate and compare the performance of two hybrid deep learning models, Autoencoder-based Gated Recurrent Unit (AE-GRU) and Autoencoder-based Long Short-Term Memory (AE-LSTM), in Bitcoin price prediction. The experiments explore the effects of dropout, learning rate, and epochs, using both univariate and multivariate inputs (Open, High, Low, Close). Results show that AE-GRU consistently outperforms AE-LSTM across all configurations, achieving up to 16.5% higher MAPE-based accuracy. The best performance was achieved by the multivariate AE-GRU, dropout = 0.1, learning rate = 0.001, epoch = 100, with RMSE 1667.125, MAE 1145.718, MAPE 2.33%, and R^2 0.995017. Moreover, AE-GRU demonstrates faster training efficiency, requiring 185–195 ms/step, while AE-LSTM takes 208–215 ms/step under the same conditions. AE-GRU's superior accuracy and efficiency are attributed to its simplified gating structure and the feature compression capability of the autoencoder, which enhances learning stability and generalization. Overall, the AE-GRU model offers robust predictive performance and computational efficiency. It is a reliable framework for real-time cryptocurrency forecasting and a promising foundation for advanced deep learning architectures in financial time-series analysis.

Keywords:
Autoencoder;
Bitcoin;
Cryptocurrency;
Forecasting;
Prediction;

Article History:
Received: September 25, 2025
Revised: January 23, 2026
Accepted: February 4, 2026
Published: June 12, 2026

Corresponding Author:
Nurun Nafisah
Department of Informatics, Faculty of Engineering, Universitas Negeri Surabaya, Indonesia
Email:
24051905002@mhs.unesa.ac.id

INTRODUCTION

The massive digital transformation of the last decade has driven major changes in the global financial system. One significant implication of this change is the shift from conventional payment methods to faster and more efficient digital payment systems [1]. This development has also driven the emergence of financial technology, which now plays a strategic role in modern economic system innovation [2]. One of the most revolutionary innovations in fintech development is the emergence of cryptocurrency, a digital or virtual currency developed using blockchain technology and protected by a cryptographic system, making it extremely difficult to counterfeit [3][4]. Blockchain technology stores transaction data in data units called blocks [5], which are interconnected through encryption to form a distributed ledger. This ledger is publicly duplicated across many network nodes in real time, making it virtually unchangeable or manipulable [6].

Of the many types of cryptocurrencies, Bitcoin is the first digital currency and the most widely recognized and traded globally. Bitcoin was first introduced by an anonymous figure named Satoshi Nakamoto [7, 8, 9] in 2008 and launched publicly in 2009 [10]. As a pioneer of decentralized peer-to-peer (P2P) transaction systems [11], Bitcoin eliminates the need for intermediaries such as banks or financial institutions and enables cross-border transactions with a high level of security and transparency [5, 6, 8]. Bitcoin's main advantage lies in its open-access, transparency, and independence from a central authority. This digital currency is globally accessible without geographical restrictions or central regulatory barriers [12]. As the most well-known and widely traded cryptocurrency, Bitcoin is often used as a benchmark for analyzing the dynamics of the overall crypto asset market [8].

Recently, Bitcoin has become increasingly popular as a speculative investment instrument [13]. This trend is driven by significant price spikes and growing public acceptance of digital assets [7]. Several large companies have begun integrating Bitcoin into their portfolio diversification strategies. Furthermore, Bitcoin adoption also has the potential to contribute to addressing specific economic issues, such as inflation [6]. Despite offering the potential for high returns, a key characteristic of Bitcoin and other cryptocurrencies is their extreme level of price volatility. Drastic and unpredictable value changes are influenced by market sentiment, regulatory policies, geopolitical events, technological innovation, and global economic conditions [8][12]. Such uncertainty presents both profit opportunities and significant challenges for investors and researchers in formulating effective investment strategies. Therefore, the development of accurate cryptocurrency price prediction models is essential to minimize risk and optimize decision-making processes [12][14].

Early approaches to predicting cryptocurrency prices generally used traditional statistical methods such as ARIMA and GARCH. While these methods are effective for modeling linear relationships, these methods have limitations in handling nonlinear data and complex volatile patterns typical of crypto price data [15]. In particular, their reliance on linear assumptions restricts their ability to capture long-term dependencies and complex temporal dynamics inherent in Bitcoin price movements. These limitations motivate the adoption of more flexible data-driven approaches capable of modeling nonlinear behavior and long-range temporal dependencies. Along with advances in computing technology, machine learning (ML) and deep learning (DL) approaches have become more adaptive and effective alternative solutions in analyzing extensive data and hidden patterns [6][8]. Among the most relevant deep learning architectures for time-series data prediction are the Recurrent Neural Network (RNN) and its derivatives, namely Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) [16]. Both of these architectures are designed to address the problem of vanishing gradients and can capture long temporal dependencies in sequential data [16][17]. Numerous studies have successfully applied LSTM and GRU models to cryptocurrency price prediction, reporting improved forecasting accuracy compared to traditional statistical techniques.

Recent research also reinforces the importance of developing LSTM and GRU based prediction models for crypto assets. For example, research [18] used historical data from 2017 to 2024 on Bitcoin, Ethereum, and Litecoin and showed that GRU provided better prediction performance than LSTM with lower MAPE values (BTC = 3.54%, ETH = 4.41%, LTC = 8.03%). Meanwhile, research [19] comparing LSTM and GRU on Ethereum prices from 2020–2025 reported that GRU achieved RMSE = 0.0234 and R^2 = 0.9442, confirming its efficiency in capturing nonlinear patterns in crypto time series data. Nevertheless, most existing works focus on standalone LSTM or GRU architectures without incorporating mechanisms for feature compression or noise reduction. To address data redundancy and noise in time-series data, several recent studies have explored the integration of autoencoders with deep learning models. Autoencoders are unsupervised neural networks designed to learn compact and informative latent representations, thereby enhancing feature extraction and reducing noise in high-dimensional data [20].

However, several important research gaps remain. First, there is still a lack of comprehensive comparative analysis between autoencoder-based LSTM (AE-LSTM) and autoencoder-based GRU (AE-GRU) architectures for Bitcoin price prediction. Second, most prior studies predominantly rely on univariate input data, typically using only closing prices, while underutilizing potentially informative multivariate features such as Open, High, Low, and Close (OHLC) prices. Third, only a limited number of studies apply statistical significance testing to rigorously assess whether observed performance differences between competing models are meaningful or merely incidental.

Based on these identified gaps, this study addresses the following research problems: (1) the limited comparative evaluation of AE-LSTM and AE-GRU models for Bitcoin price prediction; (2) the insufficient investigation of univariate versus multivariate input configurations in autoencoder-based recurrent models; and (3) the lack of statistical validation in assessing model performance differences. Accordingly, this study seeks to answer the following research questions:

RQ1: How does the predictive performance of AE-LSTM compare to AE-GRU for Bitcoin price prediction?

RQ2: Does the use of multivariate input data improve prediction accuracy compared to univariate input data?

RQ3: Are the observed performance differences between AE-LSTM and AE-GRU statistically significant?

To address these questions, this study develops and evaluates AE-LSTM and AE-GRU models under both univariate and multivariate input scenarios. Model performance is assessed using RMSE, MAE, R^2, and MAPE metrics, complemented by paired statistical significance tests to validate performance differences. By jointly analyzing predictive accuracy and statistical validity, this research aims to identify the most effective and efficient autoencoder-based recurrent architecture for Bitcoin price forecasting.

The remainder of this paper is organized as follows. Section II describes the dataset and proposed methodology, including the AE-LSTM and AE-GRU architectures. Section III presents the experimental results and discussion. Finally, Section IV concludes the paper and outlines directions for future research.

METHOD

Dataset

This study uses a secondary dataset from the Yahoo Finance platform, consisting of historical Bitcoin (BTC-USD) price data. The dataset covers February 5, 2018, to February 1, 2025, with 2,554 daily data entries. Each entry includes six main features: Open (opening price), High (highest price), Low (lowest price), Close (closing price), Volume (number of assets traded), and Date as a time index. In this study, two input approaches were used to compare the performance of the prediction models: univariate and multivariate.

1. The univariate approach uses the Close feature as the sole input in the modeling. The selection of the closing price is based on the consideration that the closing price represents the final price agreed upon by the market at the end of a trading session, and is therefore considered the most relevant indicator of an asset's actual value in financial analysis and time-series price prediction [19, 21, 22]. Furthermore, previous research has widely applied a similar approach to model and predict crypto asset prices [18][20].
2. The multivariate approach combines Open, High, Low, and Close (OHLC) features. This combination was chosen because it can represent the overall dynamics of price movements within a single trading session. It is often used as a reference in various financial asset price prediction models, such as those implemented in this study [23]. In the context of a multivariate model, additional information from daily fluctuations such as the range between the highest and lowest prices, and the difference between the opening and closing prices can improve the model's prediction accuracy.

AE-LSTM (Autoencoder-based Long Short-Term Memory)

AE-LSTM is an autoencoder architecture that uses Long Short-Term Memory (LSTM) units as the main components in the encoder and decoder sections [24]. This model is designed to recognize long-term dependencies in time series data, common in price fluctuations of digital assets like Bitcoin. It uses three main gates: the Forget Gate, the Input Gate, and the Output Gate [25, 26, 27, 28] allowing the model to retain important information over the long term.

The mathematical formulation of the LSTM unit is defined in (1)–(6) [24][29]:

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

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = tanh(W_C[h_{t-1}, x_t] + b_C) \quad (3)$$

$$C_t = f_t \odot C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t \odot tanh(C_t) \quad (6)$$

where x_t denotes the input vector at time step t, h_t represents the hidden state, C_t is the cell state, W and b are the weight matrices and bias vectors, respectively, σ is the sigmoid activation function.

AE-GRU (Autoencoder-based Gated Recurrent Unit)

AE-GRU is a hybrid architecture that combines an autoencoder with a Gated Recurrent Unit (GRU) in both the encoder and decoder sections [24]. Compared to LSTM, GRU has a simpler structure, yet can still efficiently capture long-term dependencies in sequential data. GRU simplifies the control mechanism by combining the functions of several LSTM gates into two main gates: the Update Gate and the Reset Gate [30][31].

The GRU unit is mathematically formulated in (7)–(10) [24]:

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{7}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{8}$$

$$\tilde{h}_t = tanh\big(W_h x_t + U_h(r_t \odot h_{t-1})\big) \tag{9}$$

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1} \tag{10}$$

where r_t and z_t denote the reset and update gates, respectively, $\tilde{h}_t$ is the candidate hidden state.

Autoencoder Model

In this study, the proposed AE-LSTM and AE-GRU models are implemented as a single end-to-end sequence-to-sequence autoencoder architecture for time-series forecasting. The model does not employ a separate prediction network after the autoencoder stage. The encoder (LSTM/GRU) compresses the input sequence into a latent representation, which is then repeated using a RepeatVector layer and passed directly to the decoder. The decoder reconstructs the temporal representation and is followed by a dense output layer that directly generates the forecasted Bitcoin closing price. Therefore, the forecasting task is fully integrated within the autoencoder framework.

This study implemented two hybrid autoencoder architectures: AE-LSTM and AE-GRU. Figure 1 shows the overall model process flow, including data preprocessing, data splitting, autoencoder model development, and performance evaluation. Algorithm 1 presents the detailed procedure of the proposed autoencoder-based LSTM/GRU forecasting framework.

Algorithm 1 : Autoencoder-based LSTM/GRU for Bitcoin Price Prediction
Input:
  Time-series dataset D
  Window size w
  Input features:
    Univariate: Close
    Multivariate: Open, High, Low, Close (OHLC)

Output:
  Predicted Bitcoin closing price

Steps:
1: Sort data chronologically
2: Normalize features using Min–Max scaling
3: Construct sliding window sequences of length w
4: Split data using time-based split (70% training, 30% testing)
5: Encode input sequences using LSTM or GRU layers
6: Obtain latent representation z
7: Repeat z across time steps
8: Decode using LSTM or GRU layers
9: Generate prediction using a dense layer
10: Train model using Adam optimizer and MSE loss

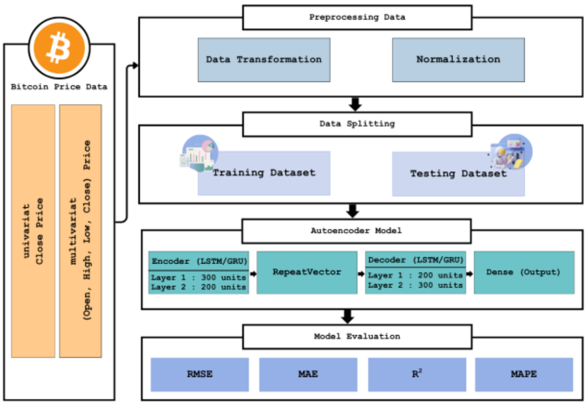

Figure 1. Autoencoder Model Flowchart (AE-LSTM/AE-GRU)

First: Data Preprocessing

The initial data processing stage involves two main steps: data transformation and normalization.

- Data Transformation: Date features are converted to datetime format (YYYY-MM-DD) so they can be sorted and processed chronologically, per the characteristics of time series data. This transformation ensures that the data's chronological order is maintained (time based split), allowing the model to learn historical trend patterns.
- Normalization: All price features (Open, High, Low, Close) are normalized using the MinMaxScaler method to the range [0, 1]. This normalization equalises the scale between features so that no single feature dominates the model's learning process.

The normalized data is then converted into sequential form with a 30-day window size, so that each input represents the price pattern over 30 consecutive days.

Second: Data Splitting

After forming the sequential data, the dataset was divided into training and testing sets using a time-based splitting strategy, where the splits were performed sequentially according to the temporal order of the data. This approach avoids disrupting temporal patterns and ensures that the model learns exclusively from historical trends, thereby preventing information leakage from future observations into the training process. Several split ratios, including 80:20 and 70:30, were explored to determine a suitable balance between training sufficiency and reliable out of sample evaluation. To justify the selection of the data split ratio, a comparative experiment was conducted using a multivariate input setting under fixed hyperparameters (batch size = 32, learning rate = 0.001, dropout = 0.1, and 100 training epochs). The results of this comparison are summarized in Table1. Under identical experimental conditions, the 70:30 split produced lower error values (RMSE and MAE) and higher R^2 scores for both AE-GRU and AE-LSTM models compared to the 80:20 split, indicating more stable and reliable generalization performance. Based on this preliminary analysis, the 70:30 time-based split was selected for all subsequent experiments. This choice is also consistent with previous studies [17][23], which report that a 70:30 split provides a favorable trade-off between model training and testing reliability in time-series prediction tasks. As an additional robustness check, a 60:20:20 configuration was explored to assess potential overfitting and to further confirm the generalizability of the proposed models.

Third: Autoencoder Model Development

This study used two RNN-based hybrid autoencoder architectures: AE-LSTM and AE-GRU. The model structure consists of four main components: an encoder, a repeat vector, a decoder, and a dense output layer, as shown in Figure 1.

- Encoder (LSTM/GRU): Consists of two consecutive layers with 300 and 200 neurons, which transform sequential input into a lower dimensional latent representation. A dropout layer is then applied to reduce the risk of overfitting.
- Repeat Vector: This component replicates the latent representation to be reused as input for the decoding stage.
- Decoder (LSTM/GRU): Consists of two symmetric layers with 200 and 300 neurons, which reconstruct data from the latent representation. A dropout layer is also applied to maintain the model's generalisation ability.
- Dense (Output): The final layer is a single neuron that generates a predicted price value (Close Price).

Several architectural configuration experiments were conducted to achieve optimal performance. Testing with three layers did not significantly improve accuracy and tended to increase computation time.

Table 1. Effect of Time-Based Data Split Ratio on Model Performance (Multivariate, Epoch 100)

Split Ratio	Model	RMSE	MAE	R^2	MAPE
80:20	AE-GRU	1917.532	1324.072	0.991302	2.04%
80:20	AE-LSTM	6815.185	4231.611	0.890124	5.41%
70:30	AE-GRU	1667.125	1145.718	0.995017	2.33%
70:30	AE-LSTM	5910.536	3369.148	0.937361	5.05%

Testing with varying numbers of neurons, namely 128–64, 200–100, and 300–200, showed that a two-layer configuration with 300–200 neurons provided the most stable performance. The model was trained using a combination of parameters, including the number of epochs (70, 100, 150), dropout (0.1 and 0.2), learning rate (0.001, 0.0001, 0.0005), and the Adam optimiser. Each hidden layer used a ReLU activation function to accelerate convergence and avoid vanishing gradients. The training process was carried out with a batch size of 32, which provides a balance between computational speed and stability of weight updates. Several combinations of epochs, dropout rates, and learning rates were intentionally explored to systematically analyze their effects on convergence behavior, prediction accuracy, and model stability. The selected epoch values (70, 100, and 150) represent different training durations to observe underfitting, optimal learning, and overfitting tendencies. Dropout values of 0.1 and 0.2 were chosen to examine the trade-off between regularization strength and information retention, while learning rates of 0.0001, 0.0005, and 0.001 were evaluated to assess convergence speed and optimization stability. This structured exploration enables a fair and reproducible comparison between AE-LSTM and AE-GRU architectures under consistent experimental conditions. All experiments were run using Google Colab with a Tesla T4 GPU (16 GB VRAM) and TensorFlow 2.15. In addition to quantitative evaluation, a paired t-test was conducted to examine the significance of performance differences between AE-LSTM and AE-GRU across various epoch configurations.

Fourth: Model Evaluation.

Model performance was evaluated using four metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), and Mean Absolute Percentage Error (MAPE), which are defined in (11-14) [24][32]:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(a_i - \hat{a}_i)^2} \tag{11}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|a_i - \hat{a}_i| \quad (12)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(a_i - \hat{a}_i)^2}{\sum_{i=1}^{N}(a_i - \bar{a}_i)^2} \quad (13)$$

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{a_i - \hat{a}_i}{a_i}\right| \quad (14)$$

In these equations, a_i denotes the actual observed value at the i^{th} time step, while $\hat{a}_i$ represents the corresponding predicted value generated by the model. The term $\bar{a}$ refers to the mean of all actual values within the dataset, and N indicates the total number of data samples or observations used in the evaluation process. These symbols collectively describe the relationship between the actual and predicted series, allowing the performance metrics to capture both the magnitude and proportion of prediction errors across the dataset.

RESULTS AND DISCUSSION

The evaluation of different parameter combinations is not intended to identify a single optimal setting only, but also to analyze the robustness and sensitivity of each model to training configurations. This analysis provides insights into how AE-LSTM and AE-GRU respond to changes in learning rate, dropout, and training duration, which is critical for understanding their practical applicability in volatile cryptocurrency markets.

Multivariate Input Model Evaluation Results

To ensure a fair and meaningful comparison, it is emphasized that all models reported in Table 2 were trained and evaluated using the same Bitcoin dataset described in the Dataset subsection. All experiments were conducted under identical conditions, including the same data preprocessing procedures, normalization method, input configuration (multivariate OHLC), time-based data splitting strategy (70:30), and evaluation metrics. Consequently, the performance differences observed in Table 2 reflect the comparative predictive capability of the AE-LSTM and AE-GRU architectures.

In Table 2, the performance evaluation of the AE-GRU and AE-LSTM models in the multivariate input scenario (OHLC) was carried out with a window size of 30. The parameter combinations tested included variations in dropout (0.1 and 0.2), learning rate (0.0001, 0.0005, and 0.001), and the number of epochs (70, 100, and 150). The test results show that, in general, AE-GRU has better prediction performance than AE-LSTM, as seen from lower RMSE, MAE, and MAPE values and higher R^2 values for most hyperparameter combinations.

At epoch 70, the AE-GRU configuration with a dropout of 0.1 and a learning rate of 0.0005 was the best, producing an RMSE of 2267.065, MAE of 1538.226, R^2 of 0.990785, and MAPE of 2.73%. Increasing the epoch to 100 resulted in the best performance of all multivariate simulations, namely AE-GRU with a dropout of 0.1 and a learning rate of 0.001, which achieved an RMSE of 1667.125, MAE of 1145.718, MAPE of 2.33%, and R^2 of 0.995017. AE-LSTM with the same configuration produced an RMSE of 5910.536, MAPE of 5.05%, and R^2 of 0.937361. Similar results were seen at epoch 100, where AE-GRU with a dropout of 0.1 and a learning rate of 0.0005 again excelled, recording an RMSE of 1680.964, MAE of 1247.658, and R^2 of 0.994934. Meanwhile, AE-LSTM, despite improvements in some configurations, still lags behind AE-GRU in terms of performance stability and consistency.

Parameter analysis shows that the learning rate significantly impacts the performance of both models. A learning rate of 0.001 consistently produces the best accuracy, especially when combined with a dropout of 0.1. Conversely, a learning rate that is too small (0.0001) causes a drastic decrease in performance, with the RMSE increasing to above 7000 and the R^2 dropping drastically, for example, to 0.868386 in AE-LSTM with a dropout of 0.2 at epoch 100. This indicates that a learning rate that is too low slows down the learning process, thus preventing the model from optimally minimizing the loss. Conversely, learning rates of 0.001 and 0.0005 produce the best results, with 0.001 slightly superior in most AE-GRU configurations.

Dropout variation also impacts performance, with a 0.2 tending to decrease accuracy, particularly in AE-LSTM. For example, at epoch 150, the RMSE jumped to 9729.903. This indicates that excessively large dropout can eliminate important information during training, thus hindering the model's ability to capture price patterns.

Table 2. Model Performance with Multivariate Input

Model	Dropout	Learning_rate	RMSE	MAE	R²	MAPE
			Epoch 70			
AE-GRU	0.1	0.001	3038.911	2202.47	0.983441	3.96%
AE-LSTM	0.1	0.001	3673.709	2861.554	0.975801	5.97%
AE-GRU	0.1	0.0001	5284.278	4198.438	0.949932	7.55%
AE-LSTM	0.1	0.0001	5230.079	3760.304	0.950954	6.31%
AE-GRU	0.1	0.0005	2267.065	1538.226	0.990785	2.73%
AE-LSTM	0.1	0.0005	4202.806	2490.731	0.968329	3.96%
AE-GRU	0.2	0.001	2675.51	1787.612	0.987165	3.01%
AE-LSTM	0.2	0.001	6638.156	3409.621	0.92099	5.07%
AE-GRU	0.2	0.0001	5767.05	4392.056	0.940366	7.48%
AE-LSTM	0.2	0.0001	7125.486	5316.819	0.908963	9.04%
AE-GRU	0.2	0.0005	2998.038	2148.717	0.983884	4.02%
AE-LSTM	0.2	0.0005	4875.404	2720.258	0.95738	4.23%
			Epoch 100			
AE-GRU	***0.1***	***0.001***	***1667.125***	***1145.718***	***0.995017***	***2.33%***
AE-LSTM	0.1	0.001	5910.536	3369.148	0.937361	5.05%
AE-GRU	0.1	0.0001	3044.492	2099.515	0.983381	3.52%
AE-LSTM	0.1	0.0001	6055.197	4654.634	0.934258	8.15%
AE-GRU	0.1	0.0005	1680.964	1247.658	0.994934	2.55%
AE-LSTM	0.1	0.0005	3216.486	1761.002	0.98145	2.79%
AE-GRU	0.2	0.001	2190.542	1492.316	0.991396	2.67%
AE-LSTM	0.2	0.001	7313.832	3997.929	0.904087	5.85%
AE-GRU	0.2	0.0001	5302.456	3909.602	0.949587	6.42%
AE-LSTM	0.2	0.0001	8567.569	6710.633	0.868386	11.82%
AE-GRU	0.2	0.0005	4509.081	3149.3	0.963544	4.98%
AE-LSTM	0.2	0.0005	4734.905	2478.356	0.959801	3.79%
			Epoch 150			
AE-GRU	0.1	0.001	2251.122	1564.959	0.990914	2.79%
AE-LSTM	0.1	0.001	5953.46	2919.108	0.936448	4.24%
AE-GRU	0.1	0.0001	4261.784	3230.089	0.967433	5.62%
AE-LSTM	0.1	0.0001	4912.324	3496.686	0.956732	5.91%
AE-GRU	0.1	0.0005	3101.252	2180.147	0.982755	3.68%
AE-LSTM	0.1	0.0005	5268.642	2731.806	0.950228	4.1%
AE-GRU	0.2	0.001	2182.398	1877.465	0.99146	4.87%
AE-LSTM	0.2	0.001	9729.903	4624.092	0.830252	6.47%
AE-GRU	0.2	0.0001	7473.181	5709.06	0.899862	9.53%
AE-LSTM	0.2	0.0001	5604.434	3774.663	0.943681	5.92%
AE-GRU	0.2	0.0005	4291.091	3176.599	0.966984	5.63%
AE-LSTM	0.2	0.0005	5911.777	2865.781	0.937335	4.18%

In terms of the number of epochs, increasing the dropout rate does not always translate to improved performance. AE-GRU tends to achieve optimal performance at epochs 100, while increasing the dropout rate to epoch 150 sometimes leads to decreased performance due to overfitting. This phenomenon is more pronounced in AE-LSTM, which, in addition to fluctuating performance, is more susceptible to performance degradation at high epochs. Although AE-LSTM achieved competitive results in several settings, such as a dropout of 0.1, a learning rate of 0.0005, and 100 epochs with an RMSE of 3216.486 and an R^2 of 0.98145, these results were still below the best performance of AE-GRU.

Based on multivariate modelling, the most optimal configuration was the combination of AE-GRU with a dropout of 0.1, a learning rate of 0.001, and 100 epochs. This configuration produced predictions with a MAPE of only 2.33% and a near-perfect R^2 0.995017, demonstrating high accuracy and good generalization capabilities.

Univariate Input Model Evaluation Results

Similar to the multivariate experiments, all univariate models reported in Table 3 were trained and evaluated using the same Bitcoin dataset, preprocessing procedures, window size, time-based data splitting strategy (70:30), and evaluation metrics. The only difference lies in the input configuration, where only the Close price was used.

In Table 3, the performance evaluation of the AE-GRU and AE-LSTM models on univariate input scenarios shows that AE-GRU again outperforms AE-LSTM in prediction accuracy for most parameter combinations.

Table 3. Model Performance with Univariate Input

Model	Dropout	Learning_rate	RMSE	MAE	R^2	MAPE
Epoch 70						
AE-GRU	0.1	0.001	1783.991	1419.311	0.994293	3.18%
AE-LSTM	0.1	0.001	4475.276	2329.278	0.964089	3.58%
AE-GRU	0.1	0.0001	5220.698	4259.211	0.95113	8.02%
AE-LSTM	0.1	0.0001	3023.718	1938.961	0.983607	3.27%
AE-GRU	0.1	0.0005	2842.457	1807.786	0.985513	2.9%
AE-LSTM	0.1	0.0005	3897.221	2273.501	0.972767	3.68%
AE-GRU	0.2	0.001	3531.885	2706.999	0.977633	5.25%
AE-LSTM	0.2	0.001	5979.641	2939.063	0.935888	4.3%
AE-GRU	0.2	0.0001	4454.339	3038.456	0.964424	4.72%
AE-LSTM	0.2	0.0001	6830.068	5192.846	0.916355	9.15%
AE-GRU	0.2	0.0005	1936.105	1520.463	0.993279	3.35%
AE-LSTM	0.2	0.0005	5958.294	3468.464	0.936345	5.2%
Epoch 100						
AE-GRU	0.1	0.001	2883.857	1861.243	0.985088	3.14%
AE-LSTM	0.1	0.001	4789.371	2449.14	0.958871	3.79%
AE-GRU	0.1	0.0001	2248.389	1459.068	0.990936	2.61%
AE-LSTM	0.1	0.0001	4241.729	2903.859	0.967739	4.83%
AE-GRU	0.1	0.0005	2187.1	1438.649	0.991423	2.73%
AE-LSTM	0.1	0.0005	5030.461	2776.969	0.954626	4.06%
AE-GRU	0.2	0.001	3345.206	2424.052	0.979935	4.13%
AE-LSTM	0.2	0.001	7976.459	4311.866	0.88592	6.34%
AE-GRU	0.2	0.0001	7335.305	5809.552	0.903523	10.23%
AE-LSTM	0.2	0.0001	7020.217	5089.273	0.911633	8.48%
AE-GRU	*0.2*	*0.0005*	*2101.618*	*1368.014*	*0.992081*	*2.51%*
AE-LSTM	0.2	0.0005	4885.576	2430.152	0.957202	3.59%
Epoch 150						
AE-GRU	0.1	0.001	1861.512	1291.991	0.993787	2.65%
AE-LSTM	0.1	0.001	6992.202	3377.043	0.912337	4.86%
AE-GRU	0.1	0.0001	5769.399	4386.777	0.940317	7.69%
AE-LSTM	0.1	0.0001	3581.125	2150.746	0.977005	3.35%
AE-GRU	0.1	0.0005	2659.621	1854.319	0.987317	3.46%
AE-LSTM	0.1	0.0005	5647.561	3490.205	0.942811	6.11%
AE-GRU	*0.2*	*0.001*	*1717.744*	*1327.605*	*0.994709*	*3%*
AE-LSTM	0.2	0.001	8304.614	4290.819	0.876341	6.31%
AE-GRU	0.2	0.0001	3544.585	2148.99	0.977472	3.29%
AE-LSTM	0.2	0.0001	6598.73	4462.221	0.921926	6.91%
AE-GRU	0.2	0.0005	2794.908	2025.531	0.985994	3.76%
AE-LSTM	0.2	0.0005	7541.276	4114.821	0.898029	5.8%

At epoch 70, the best configuration was obtained by AE-GRU with a dropout of 0.1 and a learning rate of 0.001, resulting in an RMSE of 1783.991, an MAE of 1419.311, a MAPE of 3.18%, and an R^2 of 0.994293. In contrast, AE-LSTM on the same configuration only achieved an R^2 of 0.964089 with a much higher RMSE of 4475.276. At epoch 100, AE-GRU's performance became even more solid, with optimal performance achieved at a dropout of 0.2 and a learning rate of 0.0005, achieving an RMSE of 2101.618, an MAE of 1368.014, a MAPE of 2.51%, and an R^2 of 0.992081. AE-LSTM with the same configuration only achieved an RMSE of 4885.576, an MAE of 2430.152, and an R^2 of 0.957202, with an R^2 and a MAPE of 3.59%. The combination of a low dropout (0.1) and a learning rate 0.0001 also performed well on AE-GRU (RMSE 2248.389 and R^2 0.990936), but still lagged behind the best configuration. A high dropout (0.2) and a low learning rate (0.0001) resulted in the worst performance for both models.

Increasing the number of epochs to 150 indicates that AE-GRU can utilize longer training to improve accuracy. AE-GRU obtained the best overall performance in the univariate scenario with a dropout of 0.2, a learning rate of 0.001, and 150 epochs, which recorded an RMSE of 1717.744, an MAE of 1327.605, an R^2 of 0.994709, and a MAPE of 3.00%. AE-LSTM with the same configuration showed a drastic performance drop with an RMSE of 8304.614 and an R^2 of only 0.876341, indicating higher sensitivity to significant dropouts. Parameter analysis showed that a low learning rate (0.0001) in both models often led to a substantial decrease in accuracy, characterized by a higher RMSE and a lower R^2. A larger dropout (0.2) does not always degrade performance on AE-GRU; in some cases, such as epoch 150, it even produces the best score. Conversely, AE-LSTM tends to be more stable at a dropout of 0.1 and experiences a sharp performance degradation at a dropout of 0.2, especially when combined with a low learning rate.

In general, for both univariate and multivariate inputs, the AE-GRU model consistently demonstrated superior performance compared to the AE-LSTM. In multivariate scenarios, the higher R^2 value of the AE-GRU indicates that adding OHLC variables provides additional contextual information, enhancing the model's ability to recognize price patterns more accurately. Conversely, the AE-LSTM exhibited decreased performance in some configurations, particularly when the dropout and learning rate combination was suboptimal, reflecting the LSTM architecture's sensitivity to variations in training parameters. From an analytical and theoretical perspective, the superiority of the AE-GRU can be attributed to the Gated Recurrent Unit mechanism, which more efficiently addresses the vanishing gradient problem through a simpler architectural structure. The GRU uses only two main gates: the update gate and the reset gate, without the input gate, as in the LSTM. This simplification reduces the number of parameters to be trained, accelerates the backpropagation through time process, and improves convergence efficiency. The GRU gate design also allows more direct control over the flow of temporal information, enabling the model to maintain long-term dependencies with better gradient stability.

From an econometric perspective, these results confirm that cryptocurrency price dynamics are nonlinear and influenced by complex interactions between variables (Open, High, Low, Close). AE-GRU proved more stable in modeling these multivariate relationships than AE-LSTM, which is more susceptible to parameter fluctuations. In addition to its accuracy advantage, AE-GRU also demonstrated higher computational efficiency. The average training time per epoch for AE-GRU ranged from 185–195 ms/step, while AE-LSTM required around 208–215 ms/step. This difference aligns with the GRU's more compact architecture, as it lacks a separate output gate like LSTM. With the same architectural configuration and training parameters (100 epochs, batch size 32), the total training time for AE-GRU was shorter without sacrificing accuracy, confirming its superiority in learning efficiency and prediction performance.

To ensure the robustness of the model and detect potential overfitting or underfitting, this study added additional testing by analyzing the comparison of the training loss and validation loss of the AE-GRU model (window size 30) in data split scenarios 60:20:20. The results showed that in all configurations, the loss curve decreased steadily until it converged, indicating no indication of overfitting or underfitting. The most stable convergence occurred in the 60:20:20 split, with a final training loss value of 2.62×10^{-4} and a validation loss of 1.73×10^{-4}. The slight difference between the two indicates a balanced learning process. In addition, the residual pattern of the model is also randomly distributed around zero, without a significant increasing or decreasing trend, thus indicating a normally distributed error and no systematic bias. This finding is supported by the high evaluation results (R^2 = 0.990; MAPE = 2.45%), indicating that a larger proportion of validation and testing data helps the model recognize complex patterns in crypto price data and improves its generalization ability to new data.

Paired t-test for Multivariate Data

To evaluate whether the performance differences between the AE-GRU and AE-LSTM models were statistically significant, paired t-tests were conducted on the RMSE and MAE metrics across three training epochs (70, 100, and 150). The paired comparisons were performed using results obtained under identical hyperparameter configurations, ensuring a fair and consistent evaluation.

The results in Table 4 show that AE-GRU consistently achieved lower RMSE and MAE values than AE-LSTM across all examined epochs. Statistically significant differences were observed at epochs 70 and 100 for both RMSE and MAE ($p < 0.05$), indicating a clear performance advantage of AE-GRU during earlier and mid-stage training. These results suggest that AE-GRU is able to learn more effective temporal representations and converge more efficiently than AE-LSTM under the same experimental conditions.

However, at epoch 150, the paired t-test results indicate that the performance differences between AE-GRU and AE-LSTM are no longer statistically significant for either RMSE or MAE ($p > 0.05$). This outcome does not imply superior performance of AE-LSTM. Instead, it reflects increased variability in the paired error differences at prolonged training durations. As training progresses, both models tend to approach performance saturation, leading to reduced and less consistent marginal improvements across different hyperparameter settings.

Consequently, the statistical power of the paired t-test decreases at higher epochs, although AE-GRU still shows lower mean error values. Overall, these findings indicate that AE-GRU has a statistically significant advantage over AE-LSTM at epochs 70 and 100, while at epoch 150 both models converge, resulting in similar predictive performance.

Table 4. Paired t-test Results for Multivariate Data

Epoch	Metric	T-stat	P-value	Significant (p<0.05)	Better Model
70	RMSE	-2.8779	0.034673	Yes	AE-GRU
100	RMSE	-3.9835	0.010494	Yes	AE-GRU
150	RMSE	-1.7827	0.134730	No	AE-GRU
70	MAE	-2.5966	0.048446	Yes	AE-GRU
100	MAE	-2.8827	0.034481	Yes	AE-GRU
150	MAE	-0.6932	0.519086	No	AE-GRU

Paired t-test for Univariate Data

A similar analysis was also performed for univariate data, where the model received the Close feature as input. The test results are shown in Table 5. The AE-GRU model showed a lower error value (RMSE) than AE-LSTM across all epochs. This difference was statistically significant at epochs 100 and 150 ($p < 0.05$), while at epoch 70 the difference was not significant. Although AE-GRU had a lower average error for the MAE metric, there was no significant difference across all epoch configurations ($p > 0.05$). These results indicate that in the univariate scenario, AE-GRU significantly outperformed AE-LSTM in RMSE but not in MAE, indicating that the performance improvement of AE-GRU was more dominant in prediction stability (RMSE) than in MAE.

Heatmap Visualization

Figure 2 and Figure 3 present heatmaps showing the performance of the AE-GRU and AE-LSTM models on multivariate and univariate input scenarios with varying combinations of the number of epochs, dropout, and learning rate. The vertical axis represents the model configuration in Architecture–Epoch–Dropout–Learning rate format, while the horizontal axis displays the evaluation metrics RMSE, MAE, R^2, and MAPE. Each row corresponds to a specific configuration, and color gradients indicate performance levels, where darker blue denotes better performance (lower error values and higher R^2), and lighter colors toward yellow indicate performance degradation.

A clear and consistent pattern emerges in Figure 2 (multivariate input), where AE-GRU configurations are predominantly characterized by darker blue regions across error metrics. This visual dominance indicates that AE-GRU achieves lower prediction errors and maintains more stable performance across different hyperparameter settings. In contrast, AE-LSTM exhibits more frequent transitions toward lighter blue and yellow regions, particularly under higher dropout rates (0.2) and lower learning rates (0.0001), suggesting increased sensitivity to hyperparameter selection.

A similar but less pronounced trend is observed in Figure 3 (univariate input). Although AE-GRU still shows relatively better performance, the color contrast between AE-GRU and AE-LSTM is reduced compared to the multivariate case. This indicates that the representational advantage of AE-GRU diminishes when the input space is limited to a single variable, which is consistent with the reduced statistical significance and lower variability captured in the univariate scenario.

Visualization of Model Prediction Graphs

To provide a clearer picture of model performance, a visualization comparing the actual price values with the predicted price values from the two best models was performed: Figure 4 multivariate AE-GRU (Dropout 0.1, Learning rate 0.001, Epoch 100) and Figure 5 univariate AE-GRU (Dropout 0.2, Learning rate 0.001, Epoch 150).

The visualization results show that both models can follow price movement patterns very well, with the prediction line being very close to the actual price line. The multivariate model's prediction fit appears smoother with a lower error rate (RMSE 1667.125, MAPE 2.33%), indicating the model can capture price dynamics with high precision.

Meanwhile, the univariate model also shows excellent performance (RMSE 1717.744, MAPE 3.00%), despite slight deviations at some price peaks and valleys. Overall, this visualization confirms that the AE-GRU model has high predictive ability, with both multivariate and univariate inputs.

Table 5. Paired t-test Results for Univariate Data

Epoch	Metric	T-stat	P-value	Significant (p<0.05)	Better Model
70	RMSE	-1.9791	0.104692	No	AE-GRU
100	RMSE	-3.4976	0.017328	Yes	AE-GRU
150	RMSE	-2.7205	0.041751	Yes	AE-GRU
70	MAE	-0.8582	0.430011	No	AE-GRU
100	MAE	-2.4927	0.054981	No	AE-GRU
150	MAE	-1.9333	0.111016	No	AE-GRU

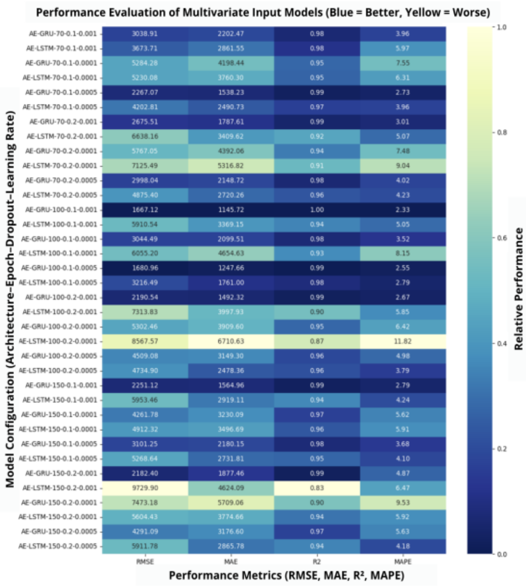

Figure 2. Heatmap of Model Performance with Multivariate Input

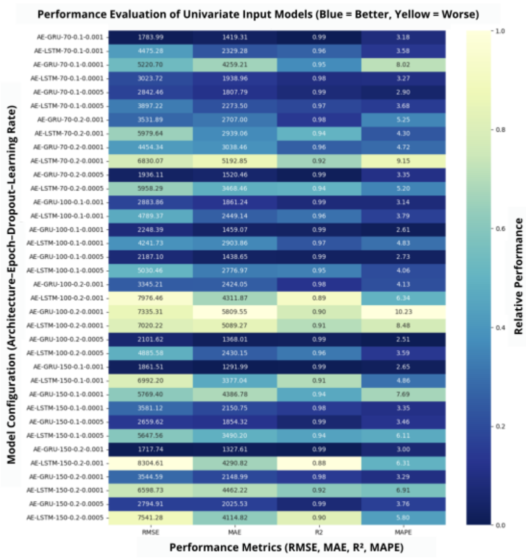

Figure 3. Heatmap of Model Performance with Univariate Input

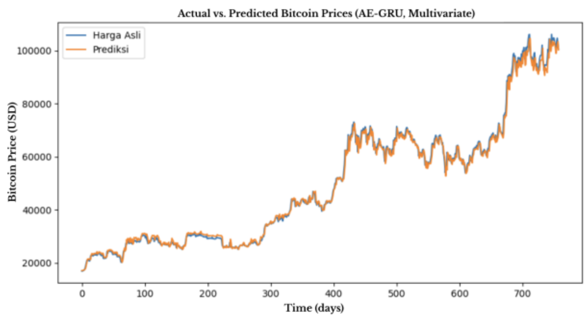

Figure 4. Multivariate AE-GRU – Epoch 100, LR 0.001, Dropout 0.1

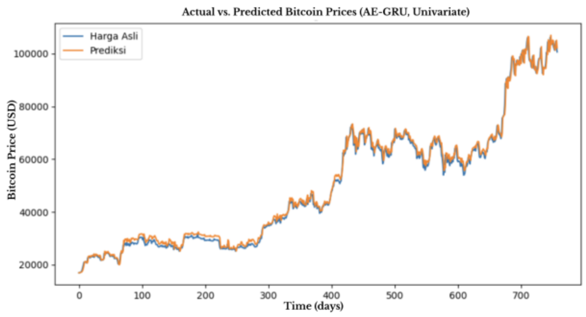

Figure 5. Univariate AE-GRU – Epoch 150, LR 0.001, Dropout 0.2

Discussion

The results show that the AE-GRU model consistently outperforms AE-LSTM in most scenarios, for both multivariate and univariate inputs. In its optimal configuration (multivariate, dropout 0.1, learning rate 0.001, and 100 epochs), AE-GRU achieved an RMSE of 1667.125, an MAE of 1145.718, an R^2 of 0.995017, and a MAPE of only 2.33%. This very high R^2 value indicates that the model is able to explain almost all of the variation in Bitcoin price data, while the low MAPE value indicates minimal prediction error.

A comparison of the results of this study with several previous studies is shown in Table 6. Overall, the proposed AE-GRU model demonstrates significant accuracy improvements compared to previous models. Compared to [21], multivariate AE-GRU with MAPE 2.33% provides a 9.7% accuracy improvement compared to LSTM with 2.58% and 22.8% compared to GRU with 3.02%. Compared to [19], the performance improvement is much more significant, where MAPE decreases from 18.17% to 2.33%, indicating an 87.2% accuracy improvement, with a 5.4% increase in R^2 (from 0.9442 to 0.9950).

Table 6. Comparison of Bitcoin Price Prediction Results with Previous Studies

Research	Model	MAPE	R^2
[21]	LSTM	2.58%	–
	GRU	3.02%	–
	Bi-LSTM	5.15%	–
[19]	LSTM	18.08%	0.9282
	GRU	18.17%	0.9442
[9]	LSTM	3.94%	–
	Bi-LSTM	3.56%	–
	GRU	5.72%	–
[18]	LSTM	9.16%	–
	GRU	3.54%	–
Our Study	AE_GRU (OHLC)	2.33%	0.995017
	AE-GRU	3.00%	0.994709
	AE-LSTM (OHLC)	2.79%	0.981450
	AE-LSTM	3.58%	0.964089

This significant difference means combining an autoencoder and multivariate data (OHLC) contributes significantly to capturing complex price patterns. In [9], the LSTM model has an MAPE of 3.94%, while GRU achieves 5.72%. The proposed model (AE-GRU OHLC) shows a 40.9% error reduction compared to LSTM and 59.3% compared to GRU, strengthening the evidence that feature compression via autoencoder improves learning efficiency. Meanwhile, compared to research [18], the AE-GRU model (2.33%) also showed a 34.2% accuracy improvement over GRU (3.54%).

Overall, these comparative results confirm that AE-GRU consistently outperforms conventional LSTM and GRU models from previous studies in terms of MAPE accuracy and R^2 variance explained. This superiority is primarily due to the autoencoder mechanism, which can extract relevant features from multivariate (OHLC) data, enhancing the model's generalizability. Combining the autoencoder and GRU architectures improved the accuracy and efficiency of cryptocurrency price prediction models. This finding corroborates previous empirical results regarding GRU efficiency and demonstrates that autoencoder integration enriches the input feature representation and significantly improves prediction performance.

CONCLUSION

This study investigated the effectiveness of autoencoder-based GRU and LSTM architectures for cryptocurrency price prediction using univariate and multivariate inputs. The experimental findings consistently indicate that AE-GRU outperforms AE-LSTM across all evaluated configurations, demonstrating superior predictive accuracy and training efficiency. The superiority of AE-GRU becomes more pronounced in the multivariate setting, indicating that the incorporation of OHLC features significantly enhances the model's ability to capture complex temporal dependencies in cryptocurrency price movements. This result confirms that feature representation learned through the autoencoder plays a critical role in improving prediction robustness in highly nonlinear financial time series. In terms of computational efficiency, AE-GRU exhibits faster convergence and lower training time compared to AE-LSTM, which aligns with the architectural simplicity of GRU and its effectiveness in modeling sequential data with fewer parameters. From a scientific perspective, this study contributes to the limited body of research on autoencoder-based recurrent models for cryptocurrency price prediction, particularly in comparing univariate and multivariate input strategies. Furthermore, this study employs a paired t-test to statistically validate the performance differences between AE-GRU and AE-LSTM, ensuring that the observed improvements are statistically significant. This type of statistical verification is still limited in existing cryptocurrency prediction studies. In terms of practical implications, the proposed AE-GRU model demonstrates strong potential for deployment in real-time forecasting systems and investment decision support platforms within highly volatile digital asset markets. Future research may extend this framework by integrating attention mechanisms, hybrid ensemble strategies, or Transformer-based architectures (e.g., Temporal Fusion Transformer or Informer) to further enhance long-term dependency modeling and predictive robustness.

REFERENCES

[1] M. Iqbal, A. Iqbal, A. Alshammari, I. Ali, L. A. Maghrabi, and N. Usman, "Sell or HODL Cryptos: Cryptocurrency Short-to-Long Term Projection Using Simultaneous Classification-Regression Deep Learning Framework," *IEEE Access*, vol. 12, pp. 118169–118184, 2024, doi: 10.1109/ACCESS.2024.3448234.

[2] 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, 2024, doi: 10.3390/forecast6020016.

[3] V. Karnati, L. D. Kanna, T. N. Pandey, and C. K. Nayak, "Prediction and Analysis of Bitcoin Price using Machine learning and Deep learning models," *EAI Endorsed Trans. Internet Things*, vol. 10, pp. 1–8, 2024, doi: 10.4108/eetiot.5379.

[4] N. S. Wen and L. S. Ling, "Evaluation of Cryptocurrency Price Prediction Using LSTM and CNNs Models," *Int. J. Informatics Vis.*, vol. 7, no. 3–2, pp. 2016–2024, 2023, doi: 10.30630/joiv.7.3-2.2344.

[5] M. J. Hamayel and A. Y. Owda, "A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms," *AI*, vol. 2, no. 4, pp. 477–496, 2021, doi: 10.3390/ai2040030.

[6] S. Syed, S. M. Talha, A. Iqbal, N. Ahmad, and M. A. Alshara, "Seeing Beyond Noise: Improving Cryptocurrency Forecasting with Linear Bias Correction," *AI*, vol. 5, no. 4, pp. 2829–2851, 2024, doi: 10.3390/ai5040136.

[7] N. Tripathy *et al.*, "Bitcoin volatility forecasting: a comparative analysis of

conventional econometric models with deep learning models," *Int. J. Electr. Comput. Eng.*, vol. 15, no. 1, pp. 614–623, 2025, doi: 10.11591/ijece.v15i1.pp614-623.
[8] M. C. Lee, "Bitcoin Trend Prediction with Attention-Based Deep Learning Models and Technical Indicators," *Systems*, vol. 12, no. 11, 2024, doi: 10.3390/systems12110498.
[9] 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 Fract.*, vol. 7, no. 2, pp. 1–18, 2023, doi: 10.3390/fractalfract7020203.
[10] T. Awoke, M. Rout, L. Mohanty, and S. C. Satapathy, *Bitcoin Price Prediction and Analysis Using Deep Learning Models*, vol. 134, no. February 2021. Springer Singapore, 2021. doi: 10.1007/978-981-15-5397-4_63.
[11] Y. Sun, X. Tang, and Y. Jiang, "Bitcoin Volatility Forecasting Based on Time Series Decomposition and Deep Learning Model," *Artif. Intell. Human-Computer Interact.*, pp. 194–201, 2025, doi: 10.3233/FAIA250121.
[12] H. M. Fadhil and N. Q. Makhool, "Forecasting Cryptocurrency Market Trends with Machine Learning and Deep Learning," *BIO Web Conf.*, vol. 97, no. April, 2024, doi: 10.1051/bioconf/20249700053.
[13] S. Sujana and B. G. Jairam, "Machine Learning based Framework to Predict the Bitcoin Price," *Grenze Int. J. Eng. Technol.*, vol. 1, pp. 1228–1235, 2024.
[14] P. Boozary, S. Sheykhan, and H. GhorbanTanhaei, "Forecasting the Bitcoin price using the various Machine Learning: A systematic review in data-driven marketing," *Syst. Soft Comput.*, vol. 7, no. August 2024, p. 200209, 2025, doi: 10.1016/j.sasc.2025.200209.
[15] A. Bouteska, M. Z. Abedin, P. Hajek, and K. Yuan, "Cryptocurrency price forecasting – A comparative analysis of ensemble learning and deep learning methods," *Int. Rev. Financ. Anal.*, vol. 92, no. December 2022, p. 103055, 2024, doi: 10.1016/j.irfa.2023.103055.
[16] R. Shirwaikar *et al.*, "Optimized Deep Learning Framework for Cryptocurrency Price Prediction," *SN Comput. Sci.*, vol. 6, no. 1, 2025, doi: 10.1007/s42979-024-03611-9.
[17] M. Yurtsever, "Gold Price Forecasting Using LSTM, Bi-LSTM and GRU," *Eur. J. Sci. Technol.*, vol. 31, pp. 341–347, 2021, doi: 10.31590/ejosat.959405.
[18] R. Kaur *et al.*, "Development of a cryptocurrency price prediction model: leveraging GRU and LSTM for Bitcoin, Litecoin and Ethereum," *Peer J. Comput. Sci.*, vol. 11, pp. 1–30, 2025, doi: 10.7717/peerj-cs.2675.
[19] M. P. A. Saputra, R. A. Ibrahim, and R. S. Saputra, "Comparative Analysis of LSTM and GRU Models for Ethereum (ETH) Price Prediction," *Int. J. Business, Econ. Soc. Dev.*, vol. 6, no. 1, pp. 132–138, 2025, doi: 10.46336/ijbesd.v6i1.887.
[20] C. Raj and M. Chattopadhyay, "Bitcoin price prediction using LSTM autoencoder regularized by false nearest neighbor loss," *Soft Comput.*, vol. 28, no. 21, pp. 12827–12834, 2024, doi: 10.1007/s00500-024-10301-4.
[21] H. A. Yanimaharta, Arya; Santoso, "Performance Evaluation of Deep Learning Models for Cryptocurrency Price Prediction using LSTM, GRU, and Bi-LSTM," *Build. Informatics, Technol. Sci.*, vol. 7, no. 1, pp. 242–251, 2025, doi: DOI 10.47065/bits.v7i1.7353.
[22] M. H. Abdul Hadi, N. A. Ramli, and Q. Ul Islam, "Predicting Bitcoin and Ethereum prices using long short-term memory and gated recurrent unit," *Data Anal. Appl. Math.*, vol. 4, no. 2, pp. 8–17, 2023, doi: 10.15282/daam.v4i2.10195.
[23] M. Z. M. Rafik, N. M. Shah, N. A. Hitam, F. Saeed, and S. Basurra, "Deep Learning Based for Cryptocurrency Assistive System," *Lect. Notes Data Eng. Commun. Technol.*, vol. 179, pp. 204–217, 2023, doi: 10.1007/978-3-031-36258-3_18.
[24] J. Dip Das, R. K. Thulasiram, C. Henry, and A. Thavaneswaran, "Encoder–Decoder Based LSTM and GRU Architectures for Stocks and Cryptocurrency Prediction," *J. Risk Financ. Manag.*, vol. 17, no. 5, 2024, doi: 10.3390/jrfm17050200.
[25] F. Ben Hamadou, T. Mezghani, R. Zouari, and M. Boujelbène-Abbes, "Forecasting Bitcoin returns using machine learning algorithms: impact of investor sentiment," *EuroMed J. Bus.*, vol. 20, no. 1, pp. 179–200, 2025, doi: 10.1108/EMJB-03-2023-0086.
[26] M. Ortu, N. Uras, C. Conversano, S. Bartolucci, and G. Destefanis, "On technical trading and social media indicators for cryptocurrency price classification through deep learning," *Expert Syst. Appl.*, vol. 198, no. March 2023, 2022, doi: 10.1016/j.eswa.2022.116804.
[27] S. Lucchetti and A. Bruno, "A Comparative Analysis of ARIMA and LSTM Models for

Bitcoin Price Prediction," *Proc. Futur. Technol. Conf.*, vol. 1, pp. 516–530, 2026, doi: 10.1007/978-3-032-07986-2_33.

[28] J. Supriyono, I. Mukhlash, M. Iqbal, and D. A. Asfani, "TE-LSTM : winding temperature prediction for induction motors in the oil and gas industry," *Sinergi (Indonesia)*, vol. 29, no. 3, pp. 819–832, 2025, doi: 10.22441/sinergi.2025. 3.022.

[29] L. S. Said, "Cryptocurrency Price Prediction Model Using GRU , LSTM and Bi-LSTM Machine Learning Algorithms," *Int. Conf. Conf. Appliedon Appl. Innov. IT*, vol. 13, no. 3, pp. 271–278, 2025, doi: 10.25673/121021.

[30] S. Ikeda, T. Ito, K. Hasebe, F. Hamakawa, B. B. Biki, and A. Takei, "A Comparative Study on Long-term Cryptocurrency Price Prediction Using LSTM, GRU, and, Bi-LSTM," *J. Robot. Netw. Artif. Life*, vol. 11, no. 1, pp. 49–53, 2025, doi: https://doi.org/10.57417/jrnal.11.1_49.

[31] T. Ito *et al.*, "Predicting High Volatility Cryptocurrency Prices using Deep Learning" *Int. Conf. Artif. Life Robot. (ICAROB2024),* pp. 691–695, 2024, doi: 10.5954/ICAROB.2024.OS22-7.

[32] T. Sari and S. Sakti, "Forecast of sugar demand in retail using SARIMA and decomposition models case study : a retail store in Indonesia," *Sinergi (Indonesia)*, vol. 29, no. 2, pp. 331–346, 2025, doi: 10.22441/sinergi.2025.2.006.