Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics e-ISSN: 2656-8624
Homepage: https://ijeeemi.org/; Vol. 7, No. 4, pp. 687-700, November 2025

RESEARCH PAPER OPEN ACCES

Hybrid Autoencoder Architectures with LSTM and GRU Layers for Bitcoin Price Prediction

Nurun Nafisah[1], Yuni Yamasari[2], and Ervin Yohannes[3]
Department of Informatics, Surabaya State University, Surabaya, Indonesia

Abstract

The high volatility of cryptocurrency markets, particularly Bitcoin, poses significant challenges for accurate price forecasting. To address this issue, this study evaluates the performance of four autoencoder-based deep learning architectures: AE-LSTM, AE-GRU, AE-LSTM-GRU, and AE-GRU-LSTM. The models were developed and tested using a univariate approach, where only the closing price was used as input, and two different window sizes (30 and 60) were applied to analyse the effect of historical sequence length on prediction accuracy. Several parameter configurations, including the number of epochs, dropout rate, and learning rate, were explored to determine the optimal model performance. The dataset comprises Bitcoin's daily closing prices from 2018 to 2025, encompassing diverse market phases, including both bullish and bearish trends. Model performance was assessed using four evaluation metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), the coefficient of determination (R^2), and Mean Absolute Percentage Error (MAPE). The results indicate that the AE-LSTM-GRU consistently achieved the best overall performance across all configurations. For a window size of 30, it achieved an RMSE of 1.53067 and a MAPE of 1.98%, while for a window size of 60, the best performance recorded was an RMSE of 1.55217 and a MAPE of 2.09%. The hybrid structure combining LSTM's capability to capture long-term dependencies with GRU's efficiency in information decoding demonstrated strong robustness in modelling highly volatile time series. This study contributes to financial time series forecasting by presenting hybrid autoencoder architectures that strike a balance between predictive accuracy and computational efficiency, providing practical insights for researchers and practitioners in financial technology and cryptocurrency analytics.

Paper History
Received August 10, 2025
Revised October 10, 2025
Accepted October 20, 2025
Published November 30, 2025

Keywords
Cryptocurrency;
Univariate;
Encoder;
Decoder;
Bitcoin

Author Email
24051905002@mhs.unesa.ac.id
yuniyamasari@unesa.ac.id
ervinyohannes@unesa.ac.id

I. Introduction

In recent years, cryptocurrencies have experienced exponential popularity and market capitalization, making them one of the most exciting digital assets of the modern era [1], [2]. Cryptocurrency is a digital or virtual currency built using blockchain technology [3], [4], [5] and secured with cryptographic techniques to prevent data forgery and manipulation [6], [7]. Blockchain technology acts as a public record-keeping system that stores transactions through interconnected encrypted blocks, creating a secure, transparent, and resistant infrastructure to interference or hacking [8], [9]. Unlike traditional financial systems, cryptocurrencies enable direct peer-to-peer transactions [10] without intermediaries and can be conducted 24/7 without operational time limits [11], [12], [13]. Furthermore, the decentralized nature of cryptocurrencies gives users full control, making them attractive to investors seeking a more open, independent financial system free from central authorities or central banks [7], [14].

Among the various types of cryptocurrencies in circulation, bitcoin (BTC), the first and most widely recognized cryptocurrency, plays a central role in the digital asset ecosystem. Introduced by an anonymous figure named Satoshi Nakamoto in 2009 [15]Bitcoin became a milestone in the digital financial revolution and a significant reference in analyzing crypto market trends [14], [16]. The popularity of Bitcoin not only created new investment opportunities but also presented considerable challenges, especially related to the market's highly volatile and speculative nature [17], [18]. Bitcoin prices can change drastically quickly, influenced by various factors such as market sentiment, regulatory policies, trading volume, technological advances, and social media activity [13], [19], [20].

This volatility poses unique challenges for price prediction models. Extreme price changes over a short period can trigger model instability, distinguishing between short-term random patterns and more meaningful long-term trends. Furthermore, sharp fluctuations have the potential to cause the model to overfit to the immediate pattern, thereby reducing its ability to generalise and predict future prices. Heteroscedastic and non-stationary price data also increase the complexity of analysis, requiring predictive models to be highly adaptable. Therefore, developing more robust and adaptive predictive models is crucial to addressing the uncertainty of the crypto market. Reliable price prediction accuracy is needed to assist investors in

Corresponding author: Yuni Yamasari, yuniyamasari@unesa.ac.id, Department of Informatics, Surabaya State University, Surabaya, Indonesia.
Digital Object Idenfier (DOI): https://doi.org/10.35882/ijeeemi.v7i4.132
Copyright © 2025 by the authors. Published by Jurusan Teknik Elektromedik, Politeknik Kesehatan Kemenkes Surabaya Indonesia. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

making informed decisions and for financial analysts in minimizing risk amidst constantly changing market dynamics [1], [14].

Various approaches have been used to predict financial asset prices, including traditional models such as ARIMA, GARCH, and Support Vector Regression (SVR) [21]. However, these approaches often fail to capture complex and nonlinear patterns in non-stationary and highly volatile data, making them less accurate in the dynamic cryptocurrency market [13], [17]. These limitations have prompted many researchers to explore machine learning-based approaches, including deep learning, which have proven superior in analyzing complex and dynamic patterns in crypto market data [1], [13], [22].

Recurrent Neural Network (RNN) models, particularly Long Short-Term Memory (LSTM) [21] and Gated Recurrent Unit (GRU) variants, are capable of addressing the vanishing gradient problem and model temporal relationships within sequential or time series data [2], [18], [23]. Empirical studies have also demonstrated the effectiveness of LSTM and GRU in the crypto context. For example, [24] evaluated LSTM, GRU, and Bi-LSTM models on Bitcoin, Ripple, and Dogecoin. The results showed that LSTM excelled in predicting Bitcoin (MAPE 2.58%) and Ripple (MAPE 4.33%), while GRU performed better on Dogecoin (MAPE 4.12%). These findings confirm that LSTM and GRU have complementary strengths worthy of further exploration in crypto asset prediction.

Several studies have begun exploring the integration of RNN models, such as LSTM and GRU, into autoencoder architectures with an encoder–decoder structure to enhance predictive performance. Some studies implement LSTM as both the encoder and decoder layers (AE-LSTM) [25], [26]. Meanwhile, another study used GRU as the basis for the encoder and decoder (AE-GRU)[27]. Research [25], [26] proposed a regularized LSTM Autoencoder model with False Nearest Neighbor (FNN) loss to predict Bitcoin price. Experimental results showed that the model outperformed standard LSTM, SVM, and Logistic Regression. On the other hand, [27] also proposed the development of two Autoencoder-based encoder–decoder models, namely AE-LSTM and AE-GRU, to predict the S&P stock index and bitcoin, and the evaluation results showed that AE-GRU consistently produced the highest accuracy. However, these studies are still limited to homogeneous architectures (either LSTM or GRU). To the best of our knowledge, no prior work has examined hybrid autoencoder architectures that combine LSTM and GRU in an encoder–decoder setting.

Based on this foundation, this study proposes two new models: (1) AE-LSTM-GRU, which combines LSTM as the encoder and GRU as the decoder, and (2) AE-GRU-LSTM, with GRU as the encoder and LSTM as the decoder. Both models are designed to capture sequential patterns in Bitcoin price data, and their performance will be compared with existing AE-LSTM and AE-GRU models in the literature. Although other advanced architectures exist, such as transformer or hybrid models involving attention mechanisms, this study focuses on LSTM and GRU. This choice was based on several considerations. Transformer models typically require significantly larger datasets, higher computational resources, and more complex hyperparameter tuning to function optimally. Meanwhile, the LSTM and GRU structures are relatively simpler, computationally less demanding, and consistently proven in previous literature for limited time series data. Therefore, using LSTM and GRU is considered more appropriate for this study to explore and compare various RNN-based autoencoder architectures for Bitcoin price prediction.

Specifically, this study aims to answer the following research questions: (RQ1) Can the hybrid AE-LSTM-GRU model provide higher accuracy than pure LSTM and GRU-based autoencoder models? (RQ2) Which hybrid architecture is more effective, AE-LSTM-GRU or AE-GRU-LSTM?. Performance evaluation is conducted using RMSE, MAE, R^2, and MAPE metrics to measure the accuracy and consistency of each model's predictions. Through this experimental approach, the research is expected to identify the optimal autoencoder architecture for predicting Bitcoin prices, particularly in the face of high volatility in the cryptocurrency market. Furthermore, this study aims to enrich the academic literature on cryptocurrency price forecasting.

II. Materials and Method

A. Dataset

This study utilizes secondary data from the Yahoo Finance platform, including historical daily Bitcoin (BTC-USD) price data in time series format. The dataset covers the period from February 5, 2018, to February 1, 2025. Each entry contains six main features: Open: daily opening price, High: daily highest price, Low: daily lowest price, Close: daily closing price, Volume: total trading volume, Date: time information as an index.

This study uses a univariate input approach, utilizing only the closing price as the sole input variable. This feature selection is based on previous studies that consistently use the closing price. It is considered the most relevant representation of market value at the end of a trading session. The closing price reflects the final market consensus after all trading activity has occurred, making it often used as a primary indicator in asset value analysis [24]. Furthermore, this price is considered the most stable because it reflects market conditions that have reached equilibrium after intraday fluctuations [28]. Other studies also confirm that close price is a consistent reference point in time series modeling [24] and has been widely used in crypto asset price prediction [25], [29], [30]. Therefore, in the context of time series-based predictive modeling, the close price is considered representative, computationally efficient, and proven effective in various previous studies.

B. Autoencoder

An autoencoder (AE) is a type of neural network designed for unsupervised learning. This architecture consists of two main components: an encoder and a decoder [31]. The main objective is to learn a compressed representation of the input data while minimizing the

Corresponding author: Yuni Yamasari, yuniyamasari@unesa.ac.id, Department of Informatics, Surabaya State University, Surabaya, Indonesia.
Digital Object Idenfier (DOI): https://doi.org/10.35882/ijeeemi.v7i4.132
Copyright © 2025 by the authors. Published by Jurusan Teknik Elektromedik, Politeknik Kesehatan Kemenkes Surabaya Indonesia. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

difference between the reconstructed output and the original input. In general, the autoencoder process can be expressed by the following Eq. (1) [27]:

$$x \rightarrow h = e(x) \rightarrow \hat{x} = d(h) \quad (1)$$

In this formula, $e(x)$ is the encoder function that transforms the input x into a hidden representation h. The decoder then works through the function $d(h)$ to reshape the data into $\hat{x}$ [27]. This process aims to generate a simplified form of the original data while retaining important information, and then restore it as closely as possible to its original form while minimizing reconstruction error. In this study, the reconstruction error was minimized using the Mean Squared Error (MSE) loss function, mathematically defined by Eq. (2) [27]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (2)$$

where y_i represents the original input data, and $\hat{y}_i$ is the reconstructed output from the decoder. The lower MSE the value, the better the autoencoder captures the underlying data pattern while retaining key information.

decoder layers [25], [26], as well as AE-GRU [27]. However, most previous studies focused solely on using the same layer type for the encoder and decoder. In this study, the authors propose two hybrid autoencoder architectures: AE-LSTM-GRU (LSTM as encoder and GRU as decoder) as the primary model, and AE-GRU-LSTM (GRU as encoder and LSTM as decoder) as a comparison model. Furthermore, the AE-LSTM and AE-GRU models were retested using adjusted layer configurations to ensure fair comparisons.

To guide the methodological design, this study formulates two research questions and a testable hypothesis as follows: (RQ1) Can the hybrid AE-LSTM-GRU model produce higher prediction accuracy than pure AE-LSTM and AE-GRU? (RQ2) Which hybrid architecture is more effective between AE-LSTM-GRU and AE-GRU-LSTM? Thus, the hypothesis tested is that combining LSTM and GRU layers in one autoencoder architecture can improve the accuracy and stability of Bitcoin price prediction compared to a homogeneous autoencoder architecture.

In Table 1. and Fig. 1, the previous study by [27], AE-LSTM and AE-GRU were constructed using three neuron layers (200-100-80). In this study, the authors adopted a two-layer approach with 256–128 neurons for all models, including AE-LSTM-GRU and AE-GRU-LSTM.

Table 1. Autoencoder Model Architecture and Configuration

No	Model	Encoder	Decoder	Neurons	Description
A.	Reference model from previous research				
1	AE-LSTM	LSTM	LSTM	200-100-50	Dip Das et al. (2024)
2	AE-GRU	GRU	GRU	200-100-50	Dip Das et al. (2024)
B.	Models used in this research				
1	AE-LSTM	LSTM	LSTM	256-128	Retested with new layer configuration
2	AE-GRU	GRU	GRU	256-128	Retested with new layer configuration
3	AE-LSTM-GRU	LSTM	GRU	256-128	Proposed Model
4	AE-GRU-LSTM	GRU	LSTM	256-128	Proposed Comparison Model

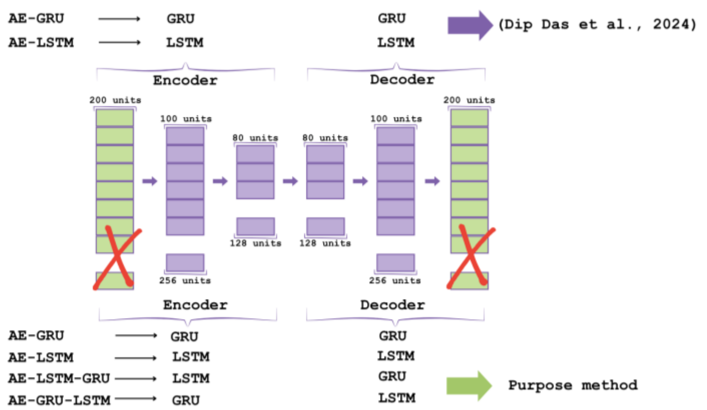

Fig. 1. Illustration of the Architecture and Novelty of the Autoencoder Model Used

C. Model Architectures Autoencoder

The high fluctuation of Bitcoin prices has become a significant concern for investors and market players, thus driving increasing interest in predicting its price movements. Several previous studies have implemented AE-LSTM, which uses LSTMs as both encoder and

The two-layer configuration was selected through a step-by-step experiment. Initially, the researchers tested

Corresponding author: Yuni Yamasari, yuniyamasari@unesa.ac.id, Department of Informatics, Surabaya State University, Surabaya, Indonesia.
Digital Object Idenfier (DOI): https://doi.org/10.35882/ijeeemi.v7i4.132
Copyright © 2025 by the authors. Published by Jurusan Teknik Elektromedik, Politeknik Kesehatan Kemenkes Surabaya Indonesia. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

various layer configurations, namely two and three layers, to assess their impact on model performance. Test results showed that adding a third layer did not significantly improve performance but increased the computational load. After the two-layer structure was deemed more stable and efficient, the experiment continued with combinations of neuron numbers, such as 128–64, 200–100, and 256–128. The 256–128 configuration yielded the most optimal performance from these trials and was used as the standard for all models in this study.

The following is a brief description of each model:

1. AE-LSTM

The AE-LSTM (Autoencoder-based Long Short-Term Memory) is a type of hybrid neural network that integrates an autoencoder framework, utilizing LSTM layers as the core components within both its encoder and decoder sections [27]. This model leverages the LSTM's capability to capture long-term dependencies within sequential data through its three main gates: input, forget, and output gates. Mathematically, the operations of an LSTM unit can be represented by Eq. (3) [21]:

$$\begin{gathered} ft = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i \\ \tilde{C}_t = \tanh\,(W_C \cdot [h_{t-1}, x_t] + b_C \\ C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \\ o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o \\ h_t = o_t * \tanh\,(C_t) \end{gathered} \quad (3)$$

where x_t is the input at time t, h_t is the hidden state, C_t is the cell state, σ denotes the sigmoid activation function, and $tanh$ is the hyperbolic tangent activation function. W and b represent trainable weight matrices and bias vectors, respectively.

2. AE-GRU

The AE-GRU (Autoencoder-based Gated Recurrent Unit) is a hybrid neural network model that merges an autoencoder design with GRU layers serving as the primary units in both the encoder and decoder parts [27]. GRU simplifies the gating mechanism to two gates, update and reset, allowing faster training while maintaining comparable accuracy to LSTM.

The GRU unit can be defined by Eq. (4) [21]:

$$\begin{gathered} z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z \\ r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r \\ \tilde{h}_t = \tanh\,(W_h \cdot [r_t * h_{t-1}, x_t] + b_h \\ h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \end{gathered} \quad (4)$$

where z_t and r_t are the update and reset gates, h_t is the hidden state, and other parameters follow the same definitions as in the LSTM formulation.

3. AE-LSTM-GRU

AE-LSTM-GRU is a hybrid autoencoder model proposed in this study, combining LSTM as an encoder and GRU as a decoder. LSTM was chosen because of its ability to capture long-term dependencies through three main gates, making it effective in filtering important information from complex historical sequences. Meanwhile, GRU has a simpler structure with two gates, making it more computationally efficient and speeding up the decoding process without significantly sacrificing accuracy [27].

4. AE-GRU-LSTM

AE-GRU-LSTM is an autoencoder variant that uses a GRU as the encoder and an LSTM as the decoder. It acts as a comparison model to the main proposed architecture.

D. Autoencoder Model Development Process Flow

Developing an autoencoder-based Bitcoin price prediction model requires a series of structured steps, from data collection to the final evaluation of model performance. Each stage is designed to ensure optimal data processing, enabling the model architecture to capture essential patterns in historical price data. This process flow encompasses not only initial processing stages such as data transformation and normalization, but also data windowing, dataset splitting, encoder-decoder architecture design, and model performance testing using specific evaluation metrics.

Fig. 2. presents a flowchart that illustrates all these stages concisely and in a structured manner, making it easier for readers to understand the interrelationships between the processes involved in this research. The diagram is structured systematically to ensure a clear understanding of the workflow.

Stage 1: Data Preprocessing

Before entering the model architecture, the obtained historical data is transformed into a structured and analysis-friendly format [6]. The dataset covers the period from February 5, 2018, to February 1, 2025, with a daily frequency. Each record originally consists of five attributes: closing price (Close), opening price (Open), highest price (High), lowest price (Low), and trading volume. However, this study adopts a univariate approach, where only the closing price is used as the predictive input, in line with the research scope.

The historical Bitcoin price data is converted to conform to the standard date format (YYYY-MM-DD). Next, a data quality check was performed. The results showed that there were no missing values in any column, so no data imputation was necessary. To detect extreme values, a boxplot of closing prices was used. Several values were identified as outliers, but they were retained because they reflect the natural volatility of Bitcoin prices. Therefore, the data was not subjected to further cleaning to maintain historical integrity. After validation, the data is normalized using the MinMaxScaler method, which transforms feature values into the range [0, 1] to maintain learning stability and equalize feature scale, as shown in Eq. (5) [24].

$$X\,norm = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (5)$$

Stage 2: Windowing

The normalized data is windowed to adapt the data for processing by the deep learning model. This method divides the historical data into sequential chunks of a

Corresponding author: Yuni Yamasari, yuniyamasari@unesa.ac.id, Department of Informatics, Surabaya State University, Surabaya, Indonesia.
Digital Object Idenfier (DOI): https://doi.org/10.35882/ijeeemi.v7i4.132
Copyright © 2025 by the authors. Published by Jurusan Teknik Elektromedik, Politeknik Kesehatan Kemenkes Surabaya Indonesia. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

specific length (window size), which are then used as input to the model. This study used two window size configurations: 30 and 60.

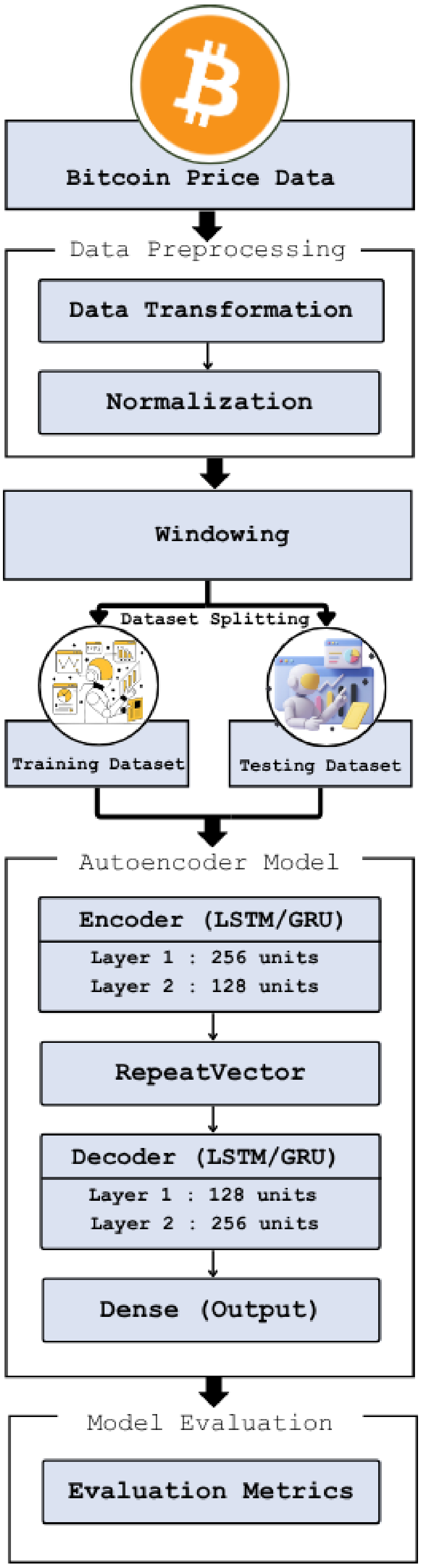

Fig. 2. Autoencoder Model Development Process Flow

The choice of window size is not arbitrary. A window size of 30 was chosen because it represents short- to medium-term seasonal patterns, while a window size of 60 captures medium-to longer-term patterns. A window that is too short tends to capture random data, thus failing to represent long-term patterns, while a window that is too long risks making the model less responsive to recent changes and increasing the risk of overfitting. Therefore, choosing window sizes of 30 and 60 is considered a decision that balances model complexity and the ability to capture price dynamics.

Previous studies also support similar configurations. [32] used 7, 30, and 90-day time horizons to distinguish short, medium, and long-term trends in cryptocurrency price prediction. Furthermore, [28] found that the best prediction performance for Bitcoin prices with LSTM occurred at window sizes between 30 and 80, with optimal results at window sizes around 30, 50, and 60 days. Therefore, the selection of window sizes of 30 and 60 in this study is based not only on practical considerations but also on previous research practices and empirical findings.

Stage 3: Dataset Splitting

After processing the data, the dataset was split into training and testing subsets in a 70:30 ratio. The training subset was utilized for model learning, whereas the testing subset served to evaluate the model's ability to generalize to unseen observations. This split was chosen because it is a commonly used proportion in time series research, providing a balance between data availability for learning and performance evaluation.

Stage 4: Autoencoder Model

The autoencoder architecture in this study consists of several main components with layer configurations arranged symmetrically but inverted between the encoder and decoder. In the encoder section, two LSTM or GRU layers with 256 and 128 neurons are used to extract latent representations from sequential data. The choice of layer type is adjusted to the model architecture: AE-LSTM and AE-LSTM-GRU use LSTM as the encoder, while AE-GRU and AE-GRU-LSTM use GRU. The activation function used in each hidden layer is ReLU, while the output layer uses linear activation to generate continuous values of Bitcoin prices.

To improve generalizability, dropout is inserted after the encoder layers. Experiments were conducted with dropout values of 0.1, 0.2, and 0.3; however, 0.3 degraded performance and was excluded from further tests. The latent representation is then converted to a sequential format using RepeatVector, allowing the decoder to process it. In the decoder section, two GRU or LSTM layers with 128 and 256 neurons are used to rebuild the data sequence from the latent representation and generate predictions. The decoder utilizes GRU for the AE-GRU and AE-LSTM-GRU architectures, and LSTM for the AE-LSTM and AE-GRU-LSTM architectures. Dropout is also reapplied on the decoder side to maintain the model's generalizability to new data. Finally, a dense output layer with a single neuron generates the final predicted value, which is the Bitcoin closing price at the next step.

The hyperparameter tuning process was performed manually. The learning rate values tested were 0.001 and 0.0005, while the number of epochs tested was 50, 70, and 100. For the batch size, the author used 32 as the

Corresponding author: Yuni Yamasari, yuniyamasari@unesa.ac.id, Department of Informatics, Surabaya State University, Surabaya, Indonesia.
Digital Object Idenfier (DOI): https://doi.org/10.35882/ijeeemi.v7i4.132
Copyright © 2025 by the authors. Published by Jurusan Teknik Elektromedik, Politeknik Kesehatan Kemenkes Surabaya Indonesia. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

main configuration. An experiment with a batch size of 64 was attempted, but model performance declined, so it was discontinued. The optimizer used was Adam. The two-layer configuration was chosen because it can capture complex patterns while reducing the computational load. Adding a third layer was tested, but it did not significantly improve performance and increased the computational load.

Stage 5: Model Evaluation

The performance of each model was evaluated using four key quantitative metrics: RMSE, MAE, R², and MAPE. These four metrics were chosen because they each provides a different perspective on predictive accuracy. RMSE emphasizes the root mean squared error [2], MAE calculates the average absolute difference between actual and predicted values [27], R² indicates how much variation in the actual data the model can explain, while MAPE expresses the prediction error as a percentage of the actual value [11]. The formula for each metric is defined in Eq. (6)–(9) [27]:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (6)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}[y_i - \hat{y}_i] \quad (7)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad (8)$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \quad (9)$$

Where y_i represents the actual value at observation i, $\hat{y}_i$ denotes the predicted value at observation i, $\bar{y}_i$ indicates the average actual values, and n refers to the total number of data points that have been tested. These four metrics objectively compare the performance of the models. The results of this evaluation serve as the basis for determining which autoencoder architecture has the optimal and most stable performance in predicting Bitcoin prices.

III. Results

This section presents the experimental results of a Bitcoin price prediction model using four different autoencoder architectures: AE-GRU, AE-LSTM, AE-LSTM-GRU, and AE-GRU-LSTM. The evaluation was conducted using a univariate approach, with the input being the closing price from historical data. Two variations of window sizes, 30 and 60, were used to evaluate the model's performance in predicting Bitcoin prices. The experimental results are presented in two subsections to facilitate analysis according to the window sizes used.

A. Model Evaluation with Widow Size 30

This subsection presents the experimental results of the four autoencoder models using a window size of 30 and a neuron configuration of 256-128. All experimental results are presented in Table 2.

Table 2. Results of the window 30 experiment

Model	Dropout	Learning_rate	RMSE	MAE	R²	MAPE
			Epoch 50			
AE-GRU	0.1	0.001	3429.264	2661.295	0.978914	4.93%
AE-LSTM	0.1	0.001	2921.892	1903.128	0.984692	3.36%
AE-LSTM-GRU	0.1	0.001	3865.553	2862.885	0.973208	4.93%
AE-GRU-LSTM	0.1	0.001	7115.663	3443.263	0.909214	4.6%
AE-GRU	0.1	0.0005	3834.451	2773.095	0.973637	4.89%
AE-LSTM	0.1	0.0005	3911.795	2270.943	0.972563	3.63%
AE-LSTM-GRU	0.1	0.0005	2812.408	2369.49	0.985818	5.32%
AE-GRU-LSTM	0.1	0.0005	4321.566	2251.743	0.966513	3.25%
AE-GRU	0.2	0.001	3796.226	2739.313	0.97416	5.06%
AE-LSTM	0.2	0.001	6183.586	3146.323	0.93144	4.63%
AE-LSTM-GRU	0.2	0.001	3310.733	2345.792	0.980347	4.25%
AE-GRU-LSTM	0.2	0.001	8689.223	4339.375	0.864621	5.95%
AE-GRU	0.2	0.0005	5173.035	3839.033	0.952018	6.41%
AE-LSTM	0.2	0.0005	8432.326	5733.039	0.872508	8.82%
AE-LSTM-GRU	0.2	0.0005	2338.418	1880.577	0.990195	4.45%
AE-GRU-LSTM	0.2	0.0005	6752.872	3970.351	0.918235	5.75%
			Epoch 70			

Corresponding author: Yuni Yamasari, yuniyamasari@unesa.ac.id, Department of Informatics, Surabaya State University, Surabaya, Indonesia.
Digital Object Idenfier (DOI): https://doi.org/10.35882/ijeeemi.v7i4.132
Copyright © 2025 by the authors. Published by Jurusan Teknik Elektromedik, Politeknik Kesehatan Kemenkes Surabaya Indonesia. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

AE-GRU	0.1	0.001	2766.065	1721.56	0.986281	2.76%
AE-LSTM	0.1	0.001	5438.275	2933.777	0.946971	4.22%
AE-LSTM-GRU	*0.1*	*0.001*	*1530.667*	*1006.686*	*0.995799*	*1.98%*
AE-GRU-LSTM	0.1	0.001	7477.452	3544.485	0.899747	4.91%
AE-GRU	0.1	0.0005	2119.795	1540.924	0.991943	3.24%
AE-LSTM	0.1	0.0005	4492.502	2610.526	0.963812	4.02%
AE-LSTM-GRU	0.1	0.0005	2768.796	1915.315	0.986254	3.33%
AE-GRU-LSTM	0.1	0.0005	5177.568	2789.49	0.951934	4.56%
AE-GRU	0.2	0.001	6629.937	5084.804	0.921185	8.82%
AE-LSTM	0.2	0.001	6322.317	3750.106	0.928329	6.89%
AE-LSTM-GRU	0.2	0.001	3806.43	3028.175	0.974021	5.9%
AE-GRU-LSTM	0.2	0.001	7906.629	3973.997	0.887909	5.83%
AE-GRU	0.2	0.0005	5409.829	3714.107	0.947525	5.74%
AE-LSTM	0.2	0.0005	6310.743	3830.583	0.928592	5.76%
AE-LSTM-GRU	0.2	0.0005	4719.558	3425.214	0.960062	5.64%
AE-GRU-LSTM	0.2	0.0005	7923.133	4921.846	0.88744	7.96%
Epoch 100						
AE-GRU	0.1	0.001	2727.399	1824.317	0.986662	3.17%
AE-LSTM	0.1	0.001	7004.223	3671.098	0.912035	5.35%
AE-LSTM-GRU	0.1	0.001	5763.608	4489.412	0.940437	8.03%
AE-GRU-LSTM	0.1	0.001	6731.428	3905.364	0.918754	7.06%
AE-GRU	0.1	0.0005	2929.858	1997.478	0.984608	3.47%
AE-LSTM	0.1	0.0005	4663.41	2689.134	0.961006	4.57%
AE-LSTM-GRU	0.1	0.0005	2203.797	1540.018	0.991292	2.96%
AE-GRU-LSTM	0.1	0.0005	6722.281	3893.984	0.918974	6.07%
AE-GRU	0.2	0.001	4441.779	3586.755	0.964625	6.61%
AE-LSTM	0.2	0.001	7902.306	4292.717	0.888031	6.23%
AE-LSTM-GRU	0.2	0.001	6476.914	4353.62	0.924782	6.67%
AE-GRU-LSTM	0.2	0.001	8923.275	4189.424	0.85723	5.48%
AE-GRU	0.2	0.0005	4721.029	3331.331	0.960037	5.27%
AE-LSTM	0.2	0.0005	9091.267	6553.849	0.851804	10.63%
AE-LSTM-GRU	0.2	0.0005	3923.718	2646.496	0.972395	4.32%
AE-GRU-LSTM	0.2	0.0005	7470.937	3749.004	0.899922	5.32%

The experimental results indicate that the AE-LSTM-GRU model outperforms the other models, consistently providing the best performance across several configurations. At epoch 70, dropout 0.1, and learning rate 0.001, this model yields an RMSE of 1530.67, MAE of 1006.69, R^2 of 0.995799, and MAPE of 1.98%, achieving the best performance among all experiments with a window size of 30. Other models also show competitive performance; for example, the AE-GRU model shows quite good results, especially at the configuration of epoch 70, dropout 0.1, and learning rate 0.001, with RMSE 2119.795, MAE 1540.924, R^2 0.991943, and MAPE 3.24%. Although these results are promising, their performance is generally inferior to that of AE-LSTM-GRU. The AE-LSTM model shows varying performance. Although some configurations, such as epoch 50, dropout 0.1, and learning rate 0.001, recorded a low MAPE of 3.36%. However, in other configurations, such as epoch 100, dropout 0.2, and learning rate 0.0005, the performance decreased drastically, with a MAPE of 10.63%. Meanwhile, the AE-GRU-LSTM model showed fluctuating performance. In some configurations, such as epoch 100, dropout 0.2, and learning rate 0.001, the performance decreased, with an RMSE of 8,923.275 and

Corresponding author: Yuni Yamasari, yuniyamasari@unesa.ac.id, Department of Informatics, Surabaya State University, Surabaya, Indonesia.
Digital Object Idenfier (DOI): https://doi.org/10.35882/ijeeemi.v7i4.132
Copyright © 2025 by the authors. Published by Jurusan Teknik Elektromedik, Politeknik Kesehatan Kemenkes Surabaya Indonesia. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

an R^2 of only 0.857230. Still, this model showed an advantage in MAPE, at 5.48%, which is lower than that of other models. Interestingly, in the configuration of epoch 50, dropout 0.1, and learning rate 0.0005, this model recorded the lowest MAPE value of 3.25% with R^2 of 0.966513. This indicates that despite the model's inconsistent performance, AE-GRU-LSTM still has the potential for relatively accurate predictions in specific scenarios.

B. Model Evaluation with a Widow Size of 60

Evaluating the model's performance with a window size of 60 provides further insight into the model's sensitivity to the length of the historical data sequence. The experimental results in Table 3 reveal a consistent pattern, where the AE-LSTM-GRU architecture consistently produces the best performance across various configurations.

Table 3. Experimental results with window size 60

Model	Dropout	Learning_rate	RMSE	MAE	R^2	MAPE
			Epoch 50			
AE-GRU	0.1	0.001	2552.326	1818.75	0.988202	3.36%
AE-LSTM	0.1	0.001	4798.185	2529.67	0.958305	3.72%
AE-LSTM-GRU	0.1	0.001	2981.476	2253.453	0.983901	4.15%
AE-GRU-LSTM	0.1	0.001	6342.235	3432.905	0.927152	5.73%
AE-GRU	0.1	0.0005	2715.524	1899.173	0.986645	3.27%
AE-LSTM	0.1	0.0005	5371.558	3754.327	0.947744	6.4%
AE-LSTM-GRU	0.1	0.0005	1920.459	1310.813	0.993321	2.51%
AE-GRU-LSTM	0.1	0.0005	4515.061	2451.906	0.96308	3.66%
AE-GRU	0.2	0.001	8648.432	6881.709	0.864541	11.8%
AE-LSTM	0.2	0.001	6293.11	3856.629	0.928276	6.05%
AE-LSTM-GRU	0.2	0.001	3116.238	2245.646	0.982413	4.34%
AE-GRU-LSTM	0.2	0.001	7008.327	3348.797	0.911047	4.79%
AE-GRU	0.2	0.0005	6907.701	5278.14	0.913583	8.97%
AE-LSTM	0.2	0.0005	7387.127	4880.533	0.901171	7.56%
AE-LSTM-GRU	0.2	0.0005	2972.20	2055.315	0.984001	3.52%
AE-GRU-LSTM	0.2	0.0005	6631.129	4228.911	0.920365	6.81%
			Epoch 70			
AE-GRU	0.1	0.001	3893.781	2798.962	0.972542	4.87%
AE-LSTM	0.1	0.001	4043.486	2368.699	0.97039	4.24%
AE-LSTM-GRU	0.1	0.001	1741.668	1344.763	0.994506	3.01%
AE-GRU-LSTM	0.1	0.001	6770.345	3604.569	0.916986	5.97%
AE-GRU	0.1	0.0005	2704.693	1793.415	0.986751	3.08%
AE-LSTM	0.1	0.0005	4664.707	3237.689	0.960592	5.87%
AE-LSTM-GRU	0.1	0.0005	2318.164	1602.202	0.990268	2.89%
AE-GRU-LSTM	0.1	0.0005	5529.233	2686.246	0.944632	3.76%
AE-GRU	0.2	0.001	4118.333	2709.775	0.969283	4.2%
AE-LSTM	0.2	0.001	7228.332	3866.783	0.905375	5.54%
AE-LSTM-GRU	0.2	0.001	4877.379	3278.133	0.956917	5.16%
AE-GRU-LSTM	0.2	0.001	7983.805	4481.742	0.884561	7.97%
AE-GRU	0.2	0.0005	5172.442	4001.398	0.951547	7.25%
AE-LSTM	0.2	0.0005	5989.762	3503.878	0.935024	5.15%
AE-LSTM-GRU	0.2	0.0005	3751.287	2811.668	0.974515	5.28%
AE-GRU-LSTM	0.2	0.0005	7785.48	4875.349	0.890225	7.57%

Corresponding author: Yuni Yamasari, yuniyamasari@unesa.ac.id, Department of Informatics, Surabaya State University, Surabaya, Indonesia.
Digital Object Idenfier (DOI): https://doi.org/10.35882/ijeeemi.v7i4.132
Copyright © 2025 by the authors. Published by Jurusan Teknik Elektromedik, Politeknik Kesehatan Kemenkes Surabaya Indonesia. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Epoch 100						
AE-GRU	0.1	0.001	3846.806	2905.041	0.9732	5.54%
AE-LSTM	0.1	0.001	5025.757	2539.64	0.954256	3.86%
AE-LSTM-GRU	*0.1*	*0.001*	*1552.172*	*1057.777*	*0.995637*	*2.09%*
AE-GRU-LSTM	0.1	0.001	8029.3	3938.915	0.883242	5.33%
AE-GRU	0.1	0.0005	5340.529	4323.243	0.948346	8.01%
AE-LSTM	0.1	0.0005	4009.478	2256.43	0.970886	3.72%
AE-LSTM-GRU	0.1	0.0005	4232.755	3452.015	0.967553	6.38%
AE-GRU-LSTM	0.1	0.0005	6325.473	3227.5	0.927537	4.77%
AE-GRU	0.2	0.001	6051.985	4278.215	0.933667	6.79%
AE-LSTM	0.2	0.001	6881.514	3520.857	0.914237	5.51%
AE-LSTM-GRU	0.2	0.001	3406.909	2072.748	0.978979	3.35%
AE-GRU-LSTM	0.2	0.001	9002.479	4715.743	0.853224	6.58%
AE-GRU	0.2	0.0005	4978.72	3384.558	0.955108	5.25%
AE-LSTM	0.2	0.0005	5792.09	3018.927	0.939242	4.47%
AE-LSTM-GRU	0.2	0.0005	5426.413	4114.773	0.946672	6.8%
AE-GRU-LSTM	0.2	0.0005	7005.188	3678.629	0.911127	5.17%

The experimental results demonstrate that the AE-LSTM-GRU model outperforms other models and consistently yields the best performance in several configurations. In the configuration of epoch 100, dropout 0.1, and learning rate 0.001, RMSE 1552.17, MAE 1057.78, R^2 0.995637, and MAPE 2.09%. This MAPE value is the lowest among all configurations, indicating high prediction accuracy. The stability of this model is also seen from its performance in other configurations, such as epoch 70 with dropout 0.1 and learning rate 0.001 with RMSE 1741.67 and MAPE 3.01%, which is still better than AE-GRU with RMSE 3,893.78; MAPE 4.87% or AE-GRU-LSTM with RMSE 6,770.35; MAPE of 5.97% with the same configuration.

The dropout rate also significantly impacts model performance. A dropout rate of 0.1 generally produces more stable and better results than a rate of 0.2. For example, in AE-GRU with a learning rate of 0.001 and 50 epochs, a dropout of 0.1 yielded an RMSE of 2552.33, significantly better than the 8648.43 achieved with a dropout of 0.2. Regarding the learning rate, 0.001 is more optimal than 0.0005. For example, in AE-LSTM-GRU with 100 epochs, a dropout of 0.1, and a learning rate of 0.001 yielded an RMSE of 1552.17, while 0.0005 yielded an RMSE of 4232.76.

A comparison between the two window sizes (30 and 60) reveals that window size 30 yields the best prediction performance in its optimal configuration, where the AE-LSTM-GRU model achieves an RMSE of 1,530.67 and a MAPE of 1.98%. Meanwhile, at window size 60, the same model recorded an RMSE of 1,552.17 and a MAPE of 2.09%, which, although slightly higher, still indicates excellent accuracy. While window size 30 excels in accuracy in the best configuration, window size 60 tends to provide more stable overall performance, especially for specific models, and is more adaptable to fluctuations in historical data over a longer period.

C. Model Comparison Summary

Table 4. Summary of the best results for each model on windows 30 and 60

Models	Window Size	RMSE	MAE	R^2	MAPE
AE-GRU	30	2119.80	1540.92	0.9919	3.24%
AE-LSTM	30	2921.89	1903.13	0.9847	3.36%
AE-LSTM-GRU	30	1530.67	1006.69	0.9958	1.98%
AE-GRU-LSTM	30	4321.57	2251.74	0.9665	3.25%
AE-GRU	60	2552.33	1818.75	0.9882	3.36%
AE-LSTM	60	4009.48	2256.43	0.9709	3.72%
AE-LSTM-GRU	60	1552.17	1057.78	0.9956	2.09%
AE-GRU-LSTM	60	4515.06	2451.91	0.9631	3.66%

Corresponding author: Yuni Yamasari, yuniyamasari@unesa.ac.id, Department of Informatics, Surabaya State University, Surabaya, Indonesia.
Digital Object Idenfier (DOI): https://doi.org/10.35882/ijeeemi.v7i4.132
Copyright © 2025 by the authors. Published by Jurusan Teknik Elektromedik, Politeknik Kesehatan Kemenkes Surabaya Indonesia. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics e-ISSN: 2656-8624
Homepage: https://ijeeemi.org/; Vol. 7, No. 4, pp. 687-700, November 2025

To facilitate reader's interpretation of the experimental results, Table 4 summarizes each model's best performance at two window sizes, 30 and 60. This table displays the RMSE, MAE, R^2, and MAPE values for each model's optimal configuration, enabling a more comprehensive comparison. This summary shows that AE-LSTM-GRU is the superior model across both window sizes. A window of 30 yields the best accuracy with a MAPE of 1.98%, while a window of 60 yields nearly equivalent results (MAPE of 2.09%) but with greater stability. The AE-GRU model comes in second, while AE-LSTM and AE-GRU-LSTM exhibit more variable and less consistent performance.

D. Visualization of Prediction and Actual Price Comparison

Fig. 3. compares the actual price and the prediction results of the AE-LSTM-GRU model at a window size of 30. It can be seen that the prediction line consistently follows the trend of the actual price line, both during periods of rising and falling prices. The closeness between the two lines indicates that the model can accurately capture price movement patterns. This aligns with the numerical evaluation results, where the model recorded an RMSE of 1530.667, MAE of 1006.686, R^2 of 0.995799, and MAPE of 1.98%. Fig.4. presents the prediction results of the proposed AE-LSTM-GRU model with the best configuration at a window size of 60. Although the RMSE value of 1552.172, MAE 1057.777, R^2 0.995637, and MAPE 2.09% show a slight decrease in performance compared to the configuration with a window size of 30, the graph shows that the model is still able to maintain accuracy in capturing the main trends of price movements and fluctuations that occur throughout the data period.

IV. Discussion

A. Model Performance

The results show that the AE-LSTM-GRU model consistently provides the best performance compared to the other three autoencoder architectures, both at window sizes of 30 and 60. This superiority can be explained by the combination of LSTM's strength in capturing long-term dependency patterns in time-series data and GRU's efficiency in the decoding process, which reduces computational complexity. Thus, this model can maintain high accuracy while avoiding overfitting to highly volatile historical Bitcoin price data. Furthermore, analysis of window sizes shows that a configuration with a window size of 30 tends to provide the best accuracy. In contrast, a window size of 60 produces more stable performance across various configurations. A shorter window (30 days) allows the model to focus more on local, medium-term patterns, making it more adaptable to rapidly changing market dynamics. This is evident in the AE-LSTM-GRU model, which achieved an RMSE of 1,530.67 and a MAPE as low as 1.98% with a window of 30, indicating high sensitivity in capturing short- to medium-term price trends. Conversely, a longer window (60 days) provides a broader historical context, allowing the model to maintain predictive stability under extreme price fluctuations. Although the best accuracy is slightly lower (RMSE = 1,552.17; MAPE 2.09%), these results are more consistent across various parameter combinations. In other words, a window of 30 is superior for optimal accuracy, while a window of 60 is superior for long-term performance consistency.

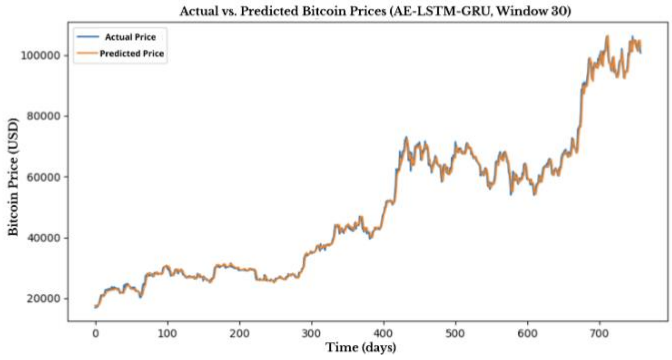

Fig. 3. Prediction vs Actual (AE-LSTM-GRU, Window 30)

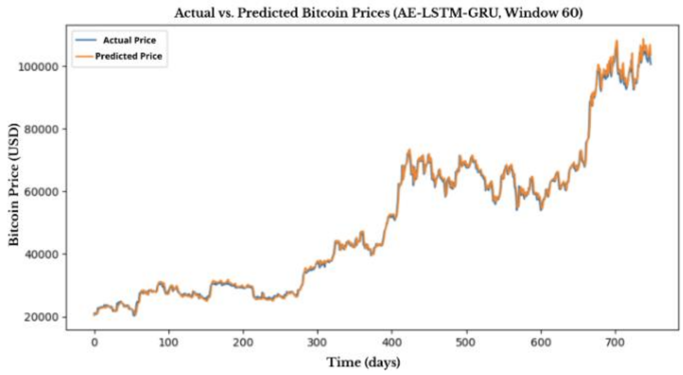

Fig. 4. Prediction vs Actual (AE-LSTM-GRU, Window 60)

Besides the widely discussed RMSE and MAPE as key indicators, this evaluation also considers MAE and R^2 to provide a more comprehensive picture of model performance. The MAE value indicates the average error in actual price units, thus complementing information from RMSE, which is more sensitive to extreme errors. In the best configuration with a window of 30, the AE-LSTM-GRU model achieved an MAE as low as 1006.69, indicating that the average deviation of predictions from actual prices is relatively small. Meanwhile, the R^2 value assesses how well the model explains Bitcoin price variations. The results show that in the optimal configuration, AE-LSTM-GRU recorded an R^2 value of 0.9958 in the 30-window and 0.9956 in the 60-window, indicating that the model can predict more than 99% of price variations. These high R^2 values confirm the model's reliability and demonstrate that, while RMSE and MAPE are the focus, MAE and R^2 also support the claim that AE-LSTM-GRU is superior to other architectures.

Corresponding author: Yuni Yamasari, yuniyamasari@unesa.ac.id, Department of Informatics, Surabaya State University, Surabaya, Indonesia.
Digital Object Idenfier (DOI): https://doi.org/10.35882/ijeeemi.v7i4.132
Copyright © 2025 by the authors. Published by Jurusan Teknik Elektromedik, Politeknik Kesehatan Kemenkes Surabaya Indonesia. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

To ensure model robustness, this study also examined the potential for overfitting and underfitting by comparing the training loss and validation loss of AE-LSTM-GRU (window 30) using an 80:10:10 split. The results show that both curves decline steadily until around the 30th epoch, then converge, with a final loss value of ≈ $2.38×10^{-4}$ and val_loss ≈ $2.82×10^{-4}$. The slight difference between the two indicates no signs of overfitting or underfitting. This consistent loss trend is reflected in the final evaluation results (RMSE = 2,221.76; MAPE = 2.15%), which are relatively balanced across the training, validation, and testing datasets. Thus, AE-LSTM-GRU excels in accuracy and proves robust and generalizable to historical Bitcoin data.

B. Statistical Significance Test

This study also conducted statistical tests using ANOVA and Tukey's HSD post hoc test to strengthen the analysis of performance differences between models.

Table 5. ANOVA Test Results for RMSE and MAPE

Factor	df	F	p-value	Significance
		RMSE		
Model	3	29.52	<0.001	Significant
Window	1	0.07	0.790	Not significant
Epoch	2	2.14	0.124	Not significant
		MAPE		
Model	3	2.17	0.097	Not significant
Window	1	0.003	0.957	Not significant
Epoch	2	0.34	0.711	Not significant

Table 6. Tukey HSD Test Results (RMSE between Models)

Comparison	Mean Diff	p-value	Significance
AE-GRU vs AE-GRU-LSTM	2634.59	<0.001	Yes
AE-GRU vs AE-LSTM	1468.74	0.004	Yes
AE-GRU vs AE-LSTM-GRU	-962.18	0.103	No
AE-GRU-LSTM vs AE-LSTM	-1165.85	0.031	Yes
AE-GRU-LSTM vs AE-LSTM-GRU	-3596.77	<0.001	Yes
AE-LSTM vs AE-LSTM-GRU	-2430.92	<0.001	Yes

ANOVA results on the RMSE metric (Table 5) indicate that the Model factor has a significant impact on prediction performance ($F = 29.52$, $p < 0.001$). Conversely, the Window size and number of epochs factors did not show a significant effect ($p>0.05$). This suggests that the model architecture makes a significant contribution to prediction quality. For the MAPE metric, the ANOVA revealed no significant differences between models, window sizes, or epochs ($p > 0.05$). In other words, despite variations in relative error values, the percentage error rates between configurations are statistically comparable. Further analysis using the Tukey HSD test (Table 6) reveals significant differences between several models in terms of RMSE. Specifically, AE-LSTM-GRU proved significantly superior to AE-LSTM, AE-GRU, and AE-GRU-LSTM

C. In-depth Analysis and Comparison with Previous Studies

In addition to comparing performance between models, it is essential to understand the nature of prediction errors. The results show that although AE-LSTM-GRU performed best, errors occurred in specific periods. This is primarily due to several factors. First, Bitcoin's high price volatility often results in sudden price spikes or drops that are difficult to capture using historical patterns. This analysis confirms that prediction errors not only reflect model limitations but also reflect the highly volatile and complex nature of Bitcoin data. Second, the limited univariate input data (closing prices only) exclude essential information from other variables, such as trading volume, high and low prices, and non-technical data, such as news and social media. This results in the model lacking relevant context in predicting price movements. Third, in terms of model capacity, the autoencoder focuses on reconstructing the main pattern, so short-term fluctuations with small amplitudes may not be captured well. This explains why deviations at extreme points still occur, despite the low mean absolute percentage error (MAPE).

It should be noted that comparisons with previous studies are not always equitable due to differences in metrics and evaluation scales. Some studies report errors in normalized data, while this study uses an inverse transform to scale the results to the original price. Therefore, the comparison was conducted using MAPE, a scale-independent metric that is more relevant for cross-study evaluation. [24] reported MAPEs of 2.58% for LSTM and 3.02% for GRU in Bitcoin prediction. In contrast, the AE-LSTM-GRU model in this study achieved a MAPE of 1.98%, surpassing the accuracy achieved by previous studies. This finding is consistent with studies [24] that confirmed the effectiveness of LSTM in capturing complex temporal patterns, and studies [30], [33] that highlighted the efficiency of GRU in short-term prediction stability. This study contributes to the introduction of a hybrid autoencoder architecture by combining LSTM and GRU within an autoencoder framework, a rarely explored approach that has yielded superior results.

D. Practical Implications and Limitations

The results of this study indicate that the choice of window size should be tailored to the application's objectives. The

Corresponding author: Yuni Yamasari, yuniyamasari@unesa.ac.id, Department of Informatics, Surabaya State University, Surabaya, Indonesia.
Digital Object Idenfier (DOI): https://doi.org/10.35882/ijeeemi.v7i4.132
Copyright © 2025 by the authors. Published by Jurusan Teknik Elektromedik, Politeknik Kesehatan Kemenkes Surabaya Indonesia. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

AE-LSTM-GRU model with a window size of 30 is more suitable for short- to medium-term predictions, such as daily or weekly trading, because it can accurately capture the most recent price changes. Conversely, using a window size of 60 is more appropriate for long-term predictions, emphasizing stability, for example, in investment planning or monitoring price trends over a broader time horizon.

However, this study has limitations. First, the model employs a univariate approach using only the closing price as input, thereby not considering the influence of other variables, such as opening price, high price, low price, trading volume, or external factors like news or market sentiment. Second, the data only covers one asset type, Bitcoin, so the model's generalizability to other cryptocurrencies has not been verified. Future studies could expand the input to multivariate by adding technical features or sentiment indices based on text or news analysis. This approach is expected to improve accuracy while strengthening the model's generalizability to more complex market conditions. These findings confirm that combining the advantages of two types of RNNs within an autoencoder framework can produce more accurate, stable, and practical crypto predictions. This opens opportunities for developing hybrid RNN-based prediction models for various assets and market conditions.

V. Conclusion

This study aims to evaluate the performance of four autoencoder architectures —namely, AE-GRU, AE-LSTM, AE-LSTM-GRU, and AE-GRU-LSTM —in predicting Bitcoin prices using a univariate approach based on historical closing price data (Close). Experiments were conducted with two variations of window sizes, 30 and 60, and tested several combinations of essential parameters, including dropout, learning rate, and number of epochs. The experimental results show that the AE-LSTM-GRU model consistently outperforms other models in various configurations at window sizes 30 and 60. At a window size of 30, the best performance was achieved with an RMSE of 1,530.67 and a MAPE of 1.98%. At a window size of 60, the best values were an RMSE of 1,552.17 and a MAPE of 2.09%. The combination of LSTM as an encoder and GRU as a decoder proved to strike a balance between capturing long-term dependencies and maintaining sequential processing efficiency, resulting in a stable model. However, this study still employs a univariate approach, using only the closing price as input. In future research, the model can be developed using a multivariate approach by incorporating additional variables such as the opening price, daily low, daily high, trading volume, or other relevant external factors. Furthermore, integrating sentiment analysis from news, social media, or cryptocurrency-related discussion forums can provide a more comprehensive perspective on price movements.

REFERENCES

[1] V. Akila, M. V. S. Nitin, I. Prasanth, S. Reddy M, and G. Akash Kumar, “A Cryptocurrency Price Prediction Model using Deep Learning,” *E3S Web of Conferences*, vol. 391, 2023, doi: 10.1051/e3sconf/202339101112.

[2] 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.

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

[4] M. Melina *et al.*, “Cryptocurrency Time Series Forecasting Model Using Gru Algorithm Based on Machine Learning,” *Barekeng*, vol. 19, no. 2, pp. 1317–1328, 2025, doi: 10.30598/barekengvol19iss2pp1317-1328.

[5] 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. August 2024, p. 200209, 2025, doi: 10.1016/j.sasc.2025.200209.

[6] 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 Transactions on Internet of Things*, vol. 10, pp. 1–8, 2024, doi: 10.4108/eetiot.5379.

[7] R. Kaur *et al.*, “Development of a cryptocurrency price prediction model: leveraging GRU and LSTM for Bitcoin, Litecoin and Ethereum,” *PeerJ Comput Sci*, vol. 11, pp. 1–30, 2025, doi: 10.7717/peerj-cs.2675.

[8] I. A. K. Shaikh, P. V. Krishna, S. G. Biswal, A. S. Kumar, S. Baranidharan, and K. Singh, “Bayesian Optimization with Stacked Sparse Autoencoder based Cryptocurrency Price Prediction Model,” *Proceedings - 5th International Conference on Smart Systems and Inventive Technology, ICSSIT 2023*, no. March, pp. 653–658, 2023, doi: 10.1109/ICSSIT55814.2023.10061153.

[9] A. Viéitez, M. Santos, and R. Naranjo, “Machine learning Ethereum cryptocurrency prediction and knowledge-based investment strategies,” *Knowl Based Syst*, vol. 299, no. March, 2024, doi: 10.1016/j.knosys.2024.112088.

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

[11] A. Ladhari and H. Boubaker, “Deep Learning Models for Bitcoin Prediction Using Hybrid Approaches with Gradient-Specific Optimization,” *Forecasting*, vol. 6,

Corresponding author: Yuni Yamasari, yuniyamasari@unesa.ac.id, Department of Informatics, Surabaya State University, Surabaya, Indonesia.
Digital Object Idenfier (DOI): https://doi.org/10.35882/ijeeemi.v7i4.132
Copyright © 2025 by the authors. Published by Jurusan Teknik Elektromedik, Politeknik Kesehatan Kemenkes Surabaya Indonesia. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

no. 2, pp. 279–295, 2024, doi: 10.3390/forecast6020016.

[12] P. L. Seabe, C. R. B. Moutsinga, and E. Pindza, “Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach,” pp. 1–18, 2023, doi: 10.3390/fractalfract7020203.

[13] 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.

[14] P. Satria and T. Putra Purnama, “Bitcoin Price Prediction Using LSTM Model: April 2024 Halving Case Study,” vol. 6424, no. April, pp. 2580–6424, 2024, doi: 10.21831/elinvo.v9i1.72518.

[15] N. Tripathy *et al.*, “Bitcoin volatility forecasting: a comparative analysis of conventional econometric models with deep learning models,” *International Journal of Electrical and Computer Engineering*, vol. 15, no. 1, pp. 614–623, 2025, doi: 10.11591/ijece.v15i1.pp614-623.

[16] S. Ranjan, P. Kayal, and M. Saraf, “Bitcoin Price Prediction: A Machine Learning Sample Dimension Approach,” *Comput Econ*, vol. 61, no. 4, pp. 1617–1636, 2023, doi: 10.1007/s10614-022-10262-6.

[17] I. Akouaouch and A. Bouayad, “A new deep learning approach for predicting high-frequency short-term cryptocurrency price,” *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 1, pp. 513–523, 2025, doi: 10.11591/eei.v14i1.7377.

[18] S. Sujana and B. G. Jairam, “Machine Learning based Framework to Predict the Bitcoin Price,” *15th International Conference on Advances in Computing, Control, and Telecommunication Technologies, ACT 2024*, vol. 1, pp. 1228–1235, 2024.

[19] S. E. Parameswaran, V. Ramachandran, and S. Shukla, “Crypto Trend Prediction Based on Wavelet Transform and Deep Learning Algorithm,” *Procedia Comput Sci*, vol. 235, pp. 1179–1189, 2024, doi: 10.1016/j.procs.2024.04.112.

[20] 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.

[21] M. YURTSEVER, “Gold Price Forecasting Using LSTM, Bi-LSTM and GRU,” *European Journal of Science and Technology*, no. January, 2021, doi: 10.31590/ejosat.959405.

[22] K. Ateeq, A. A. Al Zarooni, A. Rehman, and M. A. Khan, “A Mechanism for Bitcoin Price Forecasting using Deep Learning,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 8, pp. 441–448, 2023, doi: 10.14569/IJACSA.2023.0140849.

[23] D. M. Gunarto, S. Sa’adah, and D. Q. Utama, “Predicting Cryptocurrency Price Using RNN and LSTM Method,” *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 12, no. 1, pp. 1–8, 2023, doi: 10.32736/sisfokom.v12i1.1554.

[24] A. Yanimaharta and H. A. Santoso, “Performance Evaluation of Deep Learning Models for Cryptocurrency Price Prediction using LSTM, GRU, and Bi-LSTM,” *International Journal of Systematic Innovation*, vol. 8, no. 1, pp. 49–62, 2025, doi: 10.6977/IJoSI.202403_8(1).0005.

[25] C. Raj and M. Chattopadhyay, “Bitcoin Price Prediction using LSTM Autoencoder Regularized by False Nearest Neighbor Loss,” *Chahatraj.Github.Io*, no. October 2020, pp. 12827–12834, 2024, doi: 10.1007/s00500-024-10301-4.

[26] N. P. N. P. Dewi, L. Yungho, K. Mustofa, and M. Riasetiawan, “ENHANCING DIESEL BACKUP POWER FORECASTING WITH LSTM , GRU , AND AUTOENCODER-BASED INPUT ENCODING,” vol. 14, no. 1, pp. 183–199, 2025, doi: 10.23887/janapati.v14i1.92079.

[27] J. Dip Das, R. K. Thulasiram, C. Henry, and A. Thavaneswaran, “Encoder–Decoder Based LSTM and GRU Architectures for Stocks and Cryptocurrency Prediction,” *Journal of Risk and Financial Management*, vol. 17, no. 5, 2024, doi: 10.3390/jrfm17050200.

[28] L. Li, A. Arab, J. Liu, J. Liu, and Z. Han, “Bitcoin options pricing using LSTM-Based prediction model and blockchain statistics,” *Proceedings - 2019 2nd IEEE International Conference on Blockchain, Blockchain 2019*, pp. 67–74, 2019, doi: 10.1109/Blockchain.2019.00018.

[29] M. P. A. Saputra, R. A. Ibrahim, and R. S. Saputra, “Comparative Analysis of LSTM and GRU Models for Ethereum (ETH) Price Prediction,” *Lecture Notes in Electrical Engineering*, vol. 968, no. 1, pp. 399–410, 2025, doi: 10.1007/978-981-19-7346-8_34.

[30] 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 Analytics and Applied Mathematics (DAAM)*, no. September 2023, pp. 8–17, 2023, doi: 10.15282/daam.v4i2.10195.

[31] R. M. L. G, N. Selby, and A. Taparia, “An Efficient Stock Price Prediction Mechanism Using Multivariate Sequential LSTM Autoencoder,” pp. 0–19, 2023, doi: 10.21203/rs.3.rs-2599921/v1.

[32] 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.

Corresponding author: Yuni Yamasari, yuniyamasari@unesa.ac.id, Department of Informatics, Surabaya State University, Surabaya, Indonesia.
Digital Object Idenfier (DOI): https://doi.org/10.35882/ijeeemi.v7i4.132
Copyright © 2025 by the authors. Published by Jurusan Teknik Elektromedik, Politeknik Kesehatan Kemenkes Surabaya Indonesia. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

[33] 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.

AUTHOR BIOGRAPHY

Nurun Nafisah earned her bachelor's degree in Computer Science (S.Kom) from the State University of Surabaya (UNESA) in 2024, graduating as one of the top students and receiving the Adiwisudawan award for graduation period 110. She is currently pursuing a master's degree in Informatics at the same university while also working as an IT staff member at MAXART INC., a Japanese-based company. Her professional and academic interests center around programming and data mining, where she is passionate about developing efficient, data-driven solutions. She continuously hones her skills in software development, data analysis, and emerging technologies to contribute to innovative projects in academia and industry.

Yuni Yamasari is a lecturer at the State University of Surabaya (UNESA) with an extensive academic and professional background. She holds a bachelor's degree (S1), a master's degree (S2), and a doctoral degree (S3) from Institut Teknologi Sepuluh Nopember (ITS), with research interests in Software Engineering (RPL) and Data Mining. Throughout her career, she has actively contributed to the academic community through community service programs, scientific publications, and authorship of academic books. She has also presented her research in various scientific seminars and obtained Intellectual Property Rights (HKI) for her works. With her strong expertise and dedication, she continues to advance research and innovation in the fields of software development, data analysis, and applied informatics.

Ervin Yohannes is a lecturer at the State University of Surabaya (UNESA) with extensive academic expertise in computer science. He earned his doctoral degree (Ph.D.) in Computer Science from National Central University, Taiwan, in 2023, after completing his master's and bachelor's degrees in Informatics. Throughout his academic career, he has published numerous scientific articles in reputable journals and conferences, authored academic books, and obtained Intellectual Property Rights (HKI) for his works. Dr. Ervin is actively engaged in teaching, research, and academic mentorship, fostering innovation and contributing to the advancement of knowledge in the field of computer science.

Corresponding author: Yuni Yamasari, yuniyamasari@unesa.ac.id, Department of Informatics, Surabaya State University, Surabaya, Indonesia.
Digital Object Idenfier (DOI): https://doi.org/10.35882/ijeeemi.v7i4.132
Copyright © 2025 by the authors. Published by Jurusan Teknik Elektromedik, Politeknik Kesehatan Kemenkes Surabaya Indonesia. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).