Journal of Mathematics and Modeling in Finance (JMMF)
Vol. 6, No. 1, Winter & Spring 2026
Research paper

Ethereum Price Prediction with a GRU–Transformer Encoder Hybrid Model

Yones Esmaeelzade Aghdam[1], **Hamid Mesgarani**[2], **Ali Heidarvand**[3]

[1] Department of Mathematics, Faculty of Statistics, Mathematics and Computer Science, Allameh Tabataba'i University, Tehran, Iran
yesmaeelzade@atu.ac.ir

[2] Department of Mathematics, Shahid Rajaee Teacher Training University, Tehran, Iran
hmesgarani@sru.ac.ir

[3] Department of Mathematics, Shahid Rajaee Teacher Training University, Tehran, Iran
heidarvand.ali@gmail.com

Abstract:
Predicting the price of Ethereum remains a significant challenge due to the extreme volatility and nonlinear dynamics inherent in the cryptocurrency market. This study proposes a novel hybrid model that integrates a Gated Recurrent Unit (GRU) with a Transformer Encoder to effectively capture both short-term and long-term temporal dependencies for enhanced Ethereum price forecasting. The model was trained on daily historical data from 2017 to 2023. The dataset, sourced from Yahoo Finance, includes Ethereums open, high, and low prices, along with its trading volume. Additionally, Bitcoins closing price and two technical indicators, On-Balance Volume (OBV) and Average True Range (ATR), were incorporated. Pearson and Spearman correlation analyses confirmed strong interdependencies among the selected features. The model underwent training for 90 epochs, utilizing the Mean Squared Error (MSE) as the loss function and the Adam optimizer. Under identical experimental conditions, the proposed hybrid model significantly outperformed several baseline architectures, including standalone GRU, Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), Transformer Encoder, and CNN–GRU hybrid models. Specifically, the model achieved a Mean Absolute Error (MAE) of 0.007199 (equivalent to $34.03), which is considerably lower than Ethereums average daily price fluctuation of $74.73. These findings demonstrate that the GRU–Transformer Encoder hybrid model is highly effective in extracting intricate patterns from volatile financial time series. Consequently, it can serve as a practical and robust tool for market trend analysis and risk management.

Keywords: Ethereum price prediction; cryptocurrency volatility; gated recurrent unit; Transformer encoder; financial time series; machine learning.
Classification: 97M30; 35R11; 35R02; 92B20.

1 Introduction

Ethereum stands as the worlds second-largest cryptocurrency by market capitalization, trailing only Bitcoin. However, unlike Bitcoin, which primarily functions as a

[1]Corresponding author
Received: 12/08/2025 Accepted: 22/11/2025
https://doi.org/10.22054/JMMF.2025.87542.1204

store of value, Ethereum offers a robust infrastructure for executing smart contracts and hosting decentralized applications. These unique capabilities have propelled Ethereums widespread adoption across diverse domains, including decentralized finance, blockchain-based gaming, and the non-fungible token market.

The cryptocurrency market, and Ethereum in particular, is notoriously volatile. These significant price fluctuations are influenced by a multitude of factors, such as international economic and geopolitical news, shifting market sentiment, and network-specific changes like system upgrades. Consequently, accurate prediction of Ethereums price is of paramount importance, not only for individual traders but also for major corporations and exchanges seeking effective risk management strategies.

The inherent volatility of Ethereums price often renders traditional statistical models, such as ARIMA and GARCH, inadequate for accurate forecasting. Consequently, deep learning models, particularly GRU and LSTM, have gained widespread application in this field. Nevertheless, these models are not without their limitations, struggling to effectively capture long-term dependencies and possessing a constrained capacity for learning complex nonlinear patterns.

In recent years, numerous studies have focused on enhancing the accuracy of cryptocurrency price forecasting, with a specific emphasis on Ethereum. For instance, Saputra et al. in [12] found that their GRU model surpassed LSTM in performance, yielding a Root Mean Squared Error (RMSE) of 0.0234 and an MAE of 0.0168. Although the study by Saputra et al. (2025) compared basic recurrent models (LSTM and GRU) and concluded that GRU achieves superior performance, their research is limited by the use of standalone architectures. Standalone models, even with carefully tuned hyperparameters, struggle to simultaneously capture complex long-term dependencies and the highly nonlinear volatility characteristic of cryptocurrency markets, leaving room for improvement in prediction accuracy. Furthermore, their study relied solely on closing prices as the input feature for forecasting—an approach that overlooks critical market information, such as trading volume or correlations with Bitcoin, which strongly influence price dynamics. Similarly, Esam Mahdi et al. (2025) [8] demonstrated that a hybrid Transformer + GRU model significantly outperformed both BiLSTM (MAE = 675.427) and Bi-GRU (MAE = 608.416), achieving an MAE of 78.809. While an MAE of 78.809 might not be considered optimal, it underscores the considerable potential of hybrid architectures in substantially reducing forecasting errors. Their approach, however, restricts inputs largely to historical price, trading volume, and the Fear and Greed Index, and leaves open the possibility of using more informative technical indicators and a more carefully optimized hybrid architecture.

Auliyah et al. in [1] employed a hybrid model integrating LSTM and GRU for Ethereum price prediction, reporting an RMSE of 0.1922. These findings collectively suggest that GRU models are effective in forecasting highly volatile time series, such as that of Ethereum. Moreover, several studies have confirmed that

combining GRU with other architectures can lead to a significant reduction in forecasting errors. However, the architecture in [1] is limited to recurrent units (LSTM and GRU) and does not incorporate attention-based mechanisms or a Transformer encoder, which can better capture long-range dependencies and complex global patterns in highly volatile time series.

In a related study, Ming Che Lee (2025) introduced the Temporal Fusion Transformer (TFT), a Transformer-based model optimized for time series, which demonstrated superior performance compared to both LSTM and GRU [7]. Their study reported MAE values of 242.8 for TFT, 255.4 for LSTM, and 258.6 for GRU. Distinct from recurrent neural networks, the Transformer architecture can directly model long-term dependencies without relying on sequential order. However, directly applying Transformer models to high-volatility financial data like Ethereums might impede the effective learning of short-term temporal dynamics.

To address this challenge, the present study proposes a hybrid model that synergistically leverages two complementary architectures: GRU, to capture short-term temporal dependencies via its recurrent structure, and the Transformer Encoder, to effectively extract long-term dependencies.

Regarding feature design, the proposed model integrates key features that capture the multifaceted dynamics of Ethereums price movements. These include Ethereums open, high, low, and close prices, along with its trading volume. To account for external influences, Bitcoins closing price is incorporated as a significant external indicator. Additionally, two widely recognized technical indicators, OBV and ATR, are included. This comprehensive feature set allows the model to thoroughly consider the factors affecting Ethereum price fluctuations.

For the training and evaluation phases, historical Ethereum price data were employed. Following data normalization, input sequences were meticulously generated using a sliding window technique. The model was trained utilizing the MSE loss function and optimized via the Adam optimizer. Performance was rigorously evaluated on test data using RMSE, MAE, and normalized MAE. Experimental results showcase the proposed models efficacy, achieving an MAE of 0.007199 (equivalent to \$34.03), which signifies robust predictive performance and superiority over numerous baseline models.

The paper proceeds as follows. Section 2, Related Work and Motivation, reviews the technical background and identifies existing literature gaps. Section 3, Data, details the dataset and preprocessing pipeline, including the primary and auxiliary features of the task (Bitcoin price, OBV, and ATR indicators), scaling procedures, and the train/test partitioning. Section 4, Proposed Model, describes the hybrid GRU–Transformer Encoder architecture and its design rationale, along with training specifics such as the sliding window setup, MSE loss, and the Adam optimizer. Section 5, Model Evaluation, reports the experimental protocol, evaluation metrics (MAE, MSE, RMSE), and comparisons against baseline models. Section 6, Discussion and Interpretation of Results, analyzes the empirical evidence, notes data

limitations, and outlines avenues for future research. Finally, the last section lists the bibliographic sources.

2 Related Work and Motivation

In a study by Saputra et al. (2025), a comparative analysis of LSTM and GRU models for Ethereum price prediction revealed that the GRU model achieved superior performance, with an MAE of 0.0168 and an RMSE of 0.0234. This demonstrated effectiveness of GRU in forecasting Ethereum prices is a primary motivation for its integration into the hybrid model proposed herein.

Similarly, Esam Mahdi et al. (2025) introduced a hybrid Transformer–GRU model that significantly outperformed BiLSTM and BiGRU in Ethereum price prediction. Their comparative analysis reported an MAE of 78.809 for their hybrid Transformer–GRU model, whereas BiLSTM and BiGRU yielded MAEs of 675.427 and 608.416, respectively. While Mahdi et al.'s model achieved a substantial error reduction compared to baselines, the observed MAE of 78.809 still presents room for improvement. Consequently, the present research was inspired to adopt a more optimized hybrid architecture, combining GRU with a Transformer Encoder, which achieves a considerably lower MAE of 34.03, thereby demonstrating a substantial advancement over Mahdi et al.'s approach.

Kaur et al. (2025) [5] also compared LSTM and GRU models for Ethereum price prediction. Their findings revealed that the GRU model achieved an MAE of 0.02131, outperforming the LSTM model, which recorded an MAE of 0.02471. This further underscores the relative advantage of GRU in Ethereum forecasting tasks.

In another study published in *Fractal and Fractional*, Tanwar et al. in [14] compared the performance of LSTM, BiLSTM, and GRU models in predicting the prices of three cryptocurrencies. For Ethereum prediction, the results based on RMSE were 148.52 for LSTM, 98.31 for GRU, and 83.95 for BiLSTM. Although GRU outperformed LSTM, the BiLSTM model achieved the lowest error among the three.

Collectively, these findings suggest that baseline models such as GRU and LSTM, even in their bidirectional forms (BiLSTM and BiGRU), struggle to adequately address the complexities of Ethereum price prediction and tend to exhibit relatively high error rates. Given the highly volatile nature of Ethereum and its propensity for sudden, sharp price movements, these fundamental models often fail to perform reliably under unstable market conditions. Consequently, this research focuses on hybrid model architectures to enhance prediction performance.

Murray et al. in [11] conducted a comparative analysis of traditional statistical models, machine learning methods, and deep learning techniques such as LSTM, GRU, and hybrid architectures for cryptocurrency price prediction, including Ethereum. Their study demonstrated that deep learning models consistently outperformed traditional and statistical approaches. Specifically, the LSTM model achieved the best

performance for Ethereum, with an RMSE of 0.0300, closely followed by GRU and a hybrid LSTM–GRU architecture, both yielding an RMSE of 0.0309. In contrast, conventional models like ARIMA and Random Forest exhibited notably higher error rates (RMSEs of 0.0320 and 0.0332, respectively). These findings reinforce the superiority of deep learning approaches in managing the inherent volatility of Ethereum, further motivating the adoption of advanced hybrid architectures in the present research.

Furthermore, the study by Siami-Namini and Siami Namin in [13] reported that the LSTM model achieved an average RMSE reduction of 84% to 87% compared to the ARIMA model. Although their dataset incorporated macroeconomic indicators like inflation rates and Gross Domestic Product (GDP), their findings provide compelling evidence against the use of traditional models in complex forecasting tasks such as cryptocurrency price prediction.

In conclusion, a comprehensive review of existing studies indicates that baseline models like LSTM and GRU, when used in isolation, face significant limitations under unstable market conditions. Furthermore, traditional models such as ARIMA and GARCH have proven inadequate for accurately forecasting Ethereum prices. Consequently, hybrid architectures, such as the integration of GRU with a Transformer encoder, clearly lead to higher prediction accuracy and lower error rates.

3 Data

Selecting appropriate features in time series forecasting models plays a crucial role in their performance. In this study, in addition to primary Ethereum data, other relevant data, such as the price of Bitcoin and related indicators, were also utilized. These features, used as input variables, have enabled effective modeling of both the short-term and long-term behavior of Ethereum.

1) Primary Ethereum features. Five features—*open* (the price at which the Ethereum market opened at the beginning of the trading day), *close* (the price at which the market closed at the end of the trading day), *high* (the highest price of Ethereum during the trading day), *low* (the lowest price of Ethereum during the trading day), and *volume* (the total value of assets traded within a specified time frame, i.e. one day)—were extracted from Ethereums daily trading data. In total, the dataset comprises 2,243 daily records, encompassing historical information on Ethereum from November 9, 2017, to December 31, 2023. These data were retrieved from the Yahoo Finance database.

2) Auxiliary features. In addition to the primary Ethereum features, three other auxiliary features—Bitcoins closing price, the OBV technical indicator, and the ATR technical indicator—were selected for training the model. Among the eight aforementioned features, seven were used as input features, and one of them (ETH close) was selected as the target feature (label).

OBV indicator

This indicator was introduced in 1963 by Joseph E. Granville in his book [4], *Granvilles New Key to Stock Market Profits*. It posits that if trading volume increases significantly without a corresponding price change, it is likely that the price will soon move in the direction of the volume. It is also assumed that informed or intelligent investors enter the market before others, and this action is first reflected in trading volume rather than price.
The general formula for its calculation is as follows:

$$\mathrm{OBV}_t = \mathrm{OBV}_{t-1} + \begin{cases} V_t, & \text{if } C_t > C_{t-1}, \\ 0, & \text{if } C_t = C_{t-1}, \\ -V_t, & \text{if } C_t < C_{t-1}, \end{cases}$$

where

- OBV_t is the current On–Balance Volume level at period t,
- OBV_{t-1} is the previous On–Balance Volume level at period $t-1$,
- V_t is the trading volume during period t,
- C_t is the closing price at period t,
- C_{t-1} is the closing price at period $t-1$.

In the present study, the OBV value for each day was calculated using Ethereums daily close price and volume and was added to the model as a feature. This indicator is useful for identifying the direction of capital accumulation in the market and for determining whether the market is in an upward or downward trend.

ATR indicator

This indicator was first introduced by J. Welles Wilder Jr. in his book [16], *New Concepts in Technical Trading Systems*. The purpose of this indicator is to measure market volatility regardless of trend direction. Its focus is on the magnitude of actual price movements, rather than their trend.
To compute this indicator, the True Range (TR) is first calculated using the following formula:

$$\begin{aligned} \mathrm{TR}_t &= \max\big(\mathrm{High}_t, \mathrm{Close}_{t-1}\big) - \min\big(\mathrm{Low}_t, \mathrm{Close}_{t-1}\big), \\ \mathrm{ATR} &= \frac{1}{n}\sum_{i=1}^{n} \mathrm{TR}_i, \\ \mathrm{ATR}_t &= \frac{\mathrm{ATR}_{t-1}\,(n-1) + \mathrm{TR}_t}{n}, \end{aligned}$$

where

- TR_i is the True Range in period i (see Wilders definition),
- ATR is the n-period simple average of the True Range,
- ATR_t is the smoothed ATR at the current period t,
- ATR_{t-1} is the smoothed ATR from the previous period $(t-1)$,
- TR_t is the True Range for the current period t,
- n is the look-back length (e.g. 14 by Wilders original work).

In this study, the ATR indicator for each day was extracted using Ethereums daily low, high, and close data. This indicator assists the model in learning different behaviors during periods of high or low volatility.
After discarding the first 13 days required to initialize the ATR indicator, the final dataset used for modeling comprises 2,230 daily records of Ethereum (ETH) prices, spanning from 2017 to the end of 2023. Within this dataset, the average closing price for Ethereum is approximately $1,243. The average highest daily price (High) stands at $1,278, while the average lowest daily price (Low) is $1,203.
Among these data, the maximum closing price is $4,812, and the minimum is around $84, indicating high volatility in this cryptocurrency during the examined period. Additionally, the daily trading volume (Volume) averaged around $12 billion, with the highest recorded daily volume reaching approximately $84 billion.
In terms of calculated technical indicators, the OBV indicator has a wide range, from approximately -40 billion to over 1.7 trillion, with an average of around 719 billion. The ATR indicator, which measures price volatility, has an average value of $74.8 and a maximum of $599, reflecting significant volatility in the Ethereum market.
Collectively, these statistics reveal that Ethereums price data are characterized by high variance, fluctuating behavior, and instability. These attributes underscore the necessity for precise modeling and justify the application of hybrid deep learning methods. Furthermore, an examination of the correlation between the input features and the target feature (ETH close) indicates a significant impact of the selected features on Ethereums price dynamics.

4 Proposed Model

The proposed hybrid model is architected to sequentially process data: initially employing a Gated Recurrent Unit (GRU) to capture short-term temporal dependencies, followed by feeding its output into a Transformer encoder to identify long-term dependencies and nonlinear relationships within the data. This study utilizes an encoder-only structure, omitting a decoder, to mitigate computational complexity and accelerate training speed.

Table 1: Spearman vs. Pearson correlation coefficients.

Feature	Spearman	Pearson
High	0.999174	0.998841
Low	0.999196	0.998843
Open	0.998298	0.997747
Volume	0.475014	0.469278
BTC Close	0.928829	0.923916
OBV	0.892801	0.898152
ATR	0.913860	0.809232

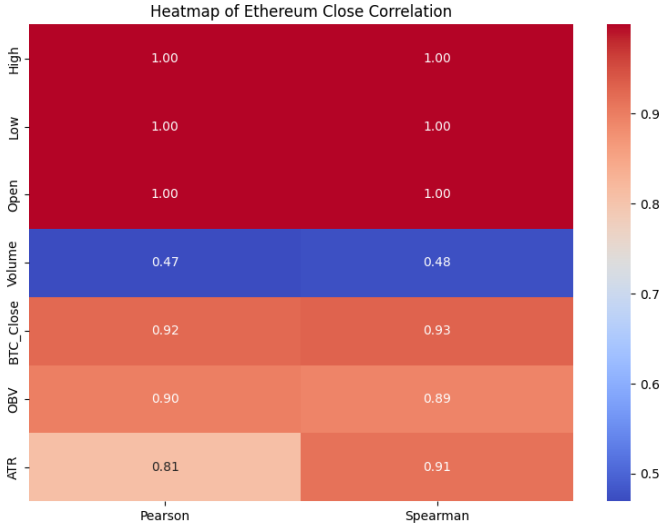


Figure 1: Heatmap of Pearson and Spearman correlation coefficients for Ethereum close price.

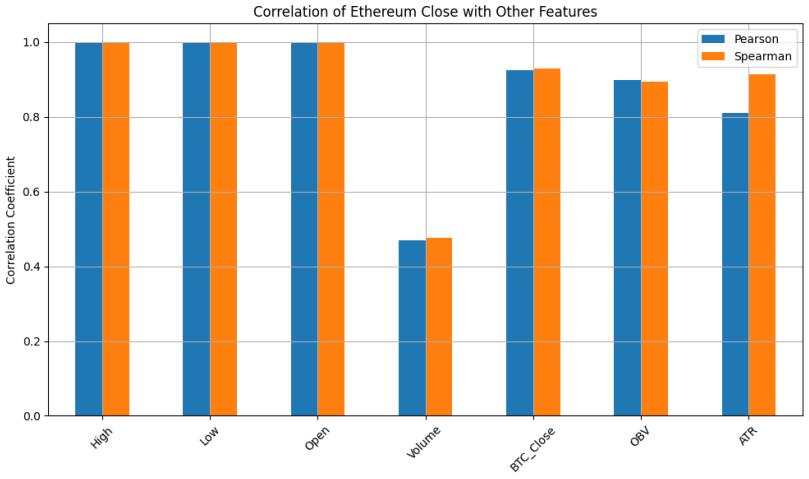


Figure 2: Bar chart comparing Pearson vs. Spearman correlations across features.

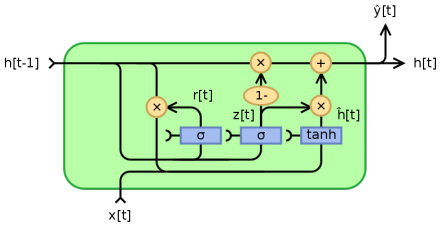
While the GRU possesses a simpler architecture than LSTM networks, it has been evaluated as performing comparably for analyzing highly volatile financial markets [3]. Furthermore, Mohammadjafari (2024) demonstrated in a recent study that the GRU achieves higher accuracy than LSTM in predicting cryptocurrency values, using Bitcoin as a case study [10]. Conversely, Vaswani et al. (2017) established that the Transformer encoder, equipped with a multi-head attention mechanism, excels at modeling long-term dependencies between data points—a task that recurrent models often find challenging [15].

This hybrid model is engineered to capitalize on the complementary strengths of both the GRU and Transformer architectures, thereby mitigating their individual limitations. Specifically, the integration of a GRU with a Transformer encoder not only preserves predictive accuracy but also offers reduced computational complexity and accelerated training times relative to a full Transformer model.

Introduced by Cho et al. in 2014 [2], the Gated Recurrent Unit (GRU) represents a streamlined variant of the LSTM network. Its architecture is characterized by two principal gates: the reset gate and the update gate. Despite its simpler structure and the consequent reduction in computational demands, the GRU demonstrates competitive performance and has found widespread application in natural language processing and time series analysis tasks.

In Figure 3, $\sigma(\cdot)$ represents the logistic sigmoid function, $\tanh(\cdot)$ denotes the hyperbolic tangent function, $\odot$ signifies element-wise multiplication, and $[\cdot, \cdot]$ indicates vector concatenation.

The Transformer model traditionally comprises two primary components: an encoder and a decoder. While originally designed for machine translation tasks, its application in time series forecasting, such as cryptocurrency price prediction, ne-



$$z_t = \sigma\big(W_z\,[h_{t-1},\,x_t]\big), \tag{1}$$
$$r_t = \sigma\big(W_r\,[h_{t-1},\,x_t]\big), \tag{2}$$
$$\tilde{h}_t = \tanh\big(W\,[r_t \odot h_{t-1},\,x_t]\big), \tag{3}$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \tag{4}$$

Figure 3: GRU cell: schematic (left) and governing update equations (right).

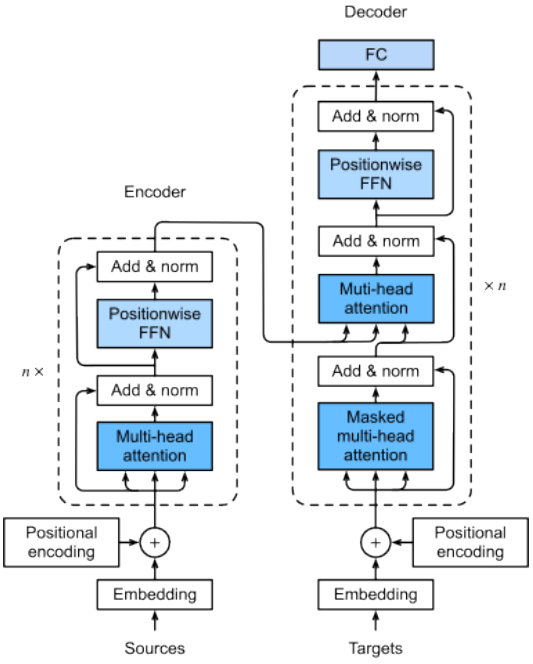


Figure 4: Encoder–decoder architecture of the Transformer model (Vaswani *et al.*, 2017).

cessitates a different approach. In these forecasting scenarios, the objective is to interpret historical sequences and generate a singular numerical output, rather than producing a new sequence from a source to a target as in translation. Consequently, the decoder, which is responsible for generating output sequences, becomes superfluous.

An encoder-only architecture provides a simplified structure that enables the model to focus on extracting long-term and nonlinear patterns directly from the input sequence. Thus, our proposed model uses a Transformer encoder. Typically, Transformer models begin with an embedding layer to convert input tokens into numerical vectors. However, as the input data in this study consist of pre-existing numerical features, an embedding layer is not required and has therefore been omitted.

Within our proposed architecture, the Transformer encoder is strategically positioned subsequent to the GRU layer. The encoder itself is composed of multiple stacked layers, each featuring two fundamental modules: multi-head self-attention and a position-wise feedforward neural network.

To effectively incorporate positional information for each time step, learnable positional encoding is employed. In [9], the authors have demonstrated that this technique, which utilizes trainable vectors instead of fixed sinusoidal patterns, offers superior adaptability to the complexities inherent in financial data.

Furthermore, the inherent simplicity of the GRU architecture translates to a reduced number of parameters and, consequently, faster training times.

The process commences with the addition of the input sequence $\mathbf{X} \in \mathbb{R}^{T \times d_{\text{model}}}$ to the positional encoding matrix:

$$\mathbf{X}' = \mathbf{X} + \mathbf{P}.$$

Subsequently, this combined input is processed by the multi-head attention layer. Within this layer, three distinct linear projections—Key (K), Query (Q), and Value (V)—are computed from the input data:

$$\mathbf{V} = \mathbf{X}W^V, \quad \mathbf{K} = \mathbf{X}W^K, \quad \mathbf{Q} = \mathbf{X}W^Q,$$

where W^Q, W^K, $W^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$ represent learnable weight matrices. The attention mechanism, which computes the relationships between each time step and all other time steps within the sequence, is calculated as follows:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\Big(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\Big)\,\mathbf{V}.$$

To increase the models learning capacity, the self-attention mechanism is implemented in parallel across multiple attention heads. Each head focuses on different aspects of temporal relationships in the data. The concatenated outputs of all attention heads are then projected through a linear transformation using the learnable weight matrix W^O, resulting in the final output of the multi-head attention module.

In the next stage, the attention output is passed through a two-layer fully connected neural network:

$$\mathrm{FFN}(x) = \max\big(0,\ xW_1 + b_1\big)\, W_2 + b_2.$$

For enhanced stability and to preserve the integrity of original information throughout the network, residual connections and layer normalization are applied within each layer:

$$x_{\text{out}} = \mathrm{LayerNorm}\big(x + \mathrm{SubLayer}(x)\big).$$

This process is iterated across all encoder layers; in our proposed model, this repetition occurs three times. Upon completion, the final sequence output is condensed into a compact vector via global average pooling:

$$\mathbf{z} = \frac{1}{T}\sum_{t=1}^{T} h_t,$$

where T is the number of time steps, h_t is the encoder output at time step t, and $\mathbf{z}$ is the final aggregated vector representing the entire sequence. Finally, $\mathbf{z}$ is passed through a fully connected layer to produce the final price prediction.

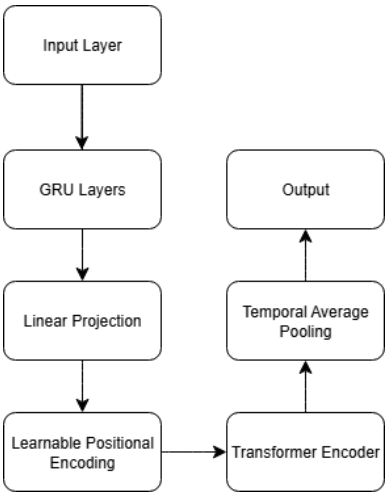


Figure 5: Overall architecture of the proposed hybrid GRU–Transformer network.

4.1 Scaling

The presence of input variables with varying magnitudes, such as close price and volume, can lead to models overemphasizing features with larger scales. To mitigate this issue, all data are normalized before training. In this study, Min–Max normalization was employed, transforming each data point into a value within the range $[0, 1]$. This scaling method was applied to both input features and the target

variable (Ethereum close price). Following the prediction, the output was rescaled to its original range:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}},$$

where x is the original (raw) input value, x' is the normalized (scaled) value, and $x_{\min}$ and $x_{\max}$ denote the minimum and maximum values of the feature, respectively.

4.2 Input sequences

To predict the price for a given day, the proposed model requires a sequence of preceding prices. For this purpose, the sliding window technique was employed. The window size was configured to 30, signifying that the model predicts the price of day 31 based on the data from the preceding 30 consecutive days. This approach is particularly effective in enabling the model to capture short-term patterns.
The training dataset comprises 80% of the data, spanning from November 2017 to September 2022, while the test dataset includes the remaining 20%, from September 2022 to December 2023. To prevent data leakage, the data split was executed in chronological order. Initially, all input sequences were constructed using a 30-day sliding window, where each input sample consists of data from the last 30 days, and the corresponding target (label) is the actual price on day 31. It was ensured that each sequence exclusively utilizes past data to predict the subsequent days price.
Following the generation of sequences, the dataset was split into training and testing sets with an 80/20 ratio, respectively. Meticulous care was taken to ensure that no test sequence contained information from the training period.
Since normalization can potentially lead to information leakage from future data, Min–Max scaling was performed exclusively on the training set. Subsequently, the derived scaling parameters were applied to the test set. This procedure guarantees that the model has no access to future data during training or inference, thereby ensuring an unbiased and realistic evaluation of model performance.
For data input into the model, PyTorchs `Dataset` and `DataLoader` structures were utilized. The batch size employed in this study was 60. To preclude variability introduced by random weight initialization, a random seed was applied to ensure reproducibility.

4.3 Loss function

To train the proposed model, the MSE loss function was employed. This function quantifies the deviation of the models prediction from the actual value:

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

where n is the total number of samples, y_i is the true (ground truth) value of the i-th sample, and $\hat{y}_i$ is the predicted value of the i-th sample. The MSE is a widely adopted loss function in neural network modeling. Its squared term inherently penalizes larger errors more significantly, incentivizing the model to reduce substantial deviations more aggressively and thereby promoting greater accuracy in predictions.

4.4 Optimizer

For this study, the Adam optimizer (Adaptive Moment Estimation) was employed. This optimization method is designed to interpret gradient directions more effectively and execute more adaptive steps compared to traditional algorithms such as standard gradient descent.

Adam was initially introduced in 2014 by Diederik P. Kingma and Jimmy Ba in their seminal paper, "Adam: A Method for Stochastic Optimization" [6]. It has since gained widespread adoption across numerous deep learning applications due to its efficiency and robustness.

Algorithm 1 Adam optimization scheme (schematic form)

Step 1: **while** w_t does not converge
do {
Step 2: Calculate gradient $g_t = \partial f(x, w)/\partial w$
Step 3: Calculate $p_t = m_1 \cdot p_{t-1} + (1 - m_1) \cdot g_t$
Step 4: Calculate $q_t = m_2 \cdot q_{t-1} + (1 - m_2) \cdot g_t^2$
Step 5: Calculate $\hat{p}_t = p_t/(1 - m_1^t)$
Step 6: Calculate $\hat{q}_t = q_t/(1 - m_2^t)$
Step 7: Update the parameter $w_t = w_{t-1} - \alpha \cdot \hat{p}_t/(\sqrt{\hat{q}_t} + \epsilon)$
}
Step 8: return w_t

5 Model Evaluation

To comprehensively evaluate model performance, this study utilized three standard error metrics: MAE, MSE, and RMSE. Each metric was calculated in both its normalized form and its raw, non-normalized value.

The table below summarizes the performance of the proposed hybrid GRU–Transformer Encoder model in forecasting Ethereum prices, presenting results for all three evaluation metrics in their normalized form:

It is worth noting that the average daily price fluctuation of Ethereum during the study period was approximately $74.73. To assess the advantages of the proposed hybrid GRU–Transformer Encoder model, its performance was benchmarked against four baseline models—GRU, LSTM, CNN, and Transformer Encoder—as

Table 2: Evaluation metrics for the proposed GRU–Transformer Encoder model (normalized values).

Model	RMSE	MSE	MAE
GRU–Transformer Encoder	0.010544	0.000111	0.007199

well as another hybrid model, CNN–GRU. All models were evaluated under identical configurations and datasets.

Table 3: Comparison of models by error metrics (normalized and non-normalized values).

Model	RMSE	MSE	MAE	RMSE ($)	MSE (^{2})	MAE ($)
GRU	0.0256	0.000657	0.0203	121.21	14691.62	95.92
LSTM	0.0296	0.000875	0.0234	139.87	19562.60	110.85
CNN	0.0224	0.000503	0.0168	105.98	11232.40	79.45
Transformer Encoder	0.0243	0.000592	0.0221	114.99	13221.70	104.32
CNN–GRU	0.0294	0.000862	0.0234	138.77	19257.75	110.44
GRU–Transformer Encoder	0.010544	0.000111	0.007199	49.85	2484.82	34.03

To statistically evaluate the models improvement over a naive benchmark method, the Clark–West test ($h = 1$) was performed. Using MSE, the mean adjusted difference d_{CW} was 327.4063, with its positive value indicating that the model outperforms the benchmark prediction. The CW statistic was 2.459 with a p-value ≈ 0.0139, demonstrating statistical significance at the 5% level. The test was also conducted using MAE, yielding a mean $d_{\mathrm{CW(MAE)}}$ of 15.3655 and a statistic of 12.031 with a p-value ≈ 0.0000. These results indicate that the model significantly reduces prediction errors compared to the naive method, both in squared and absolute terms.

Regarding price movement direction, the model achieved a directional accuracy of 59.8%, compared to 47.1% for the naive approach. The Pesaran–Timmermann test for direction independence resulted in $z = 4.029$, $p \approx 0.0001$, indicating a significant correlation between the model's predicted directions and the actual directions. The Diebold–Mariano test (directional 0.1 loss) further showed a negative mean difference $d_{\mathrm{dir}} = -0.1268$ and statistic $= -3.534$ with $p \approx 0.0004$, confirming that the model predicts movement direction better than the naive method. Additionally, the 95% confidence intervals for directional prediction success rates were estimated using both binomial and block bootstrap methods: for the model, 54.9%–64.5%; for the naive benchmark, 42.3%–52.0%, further emphasizing the model's superiority.

To evaluate the predictive performance of the proposed model and compare it with classical approaches, an experiment was conducted including GRU+Transformer

Encoder, ARIMA(5,1,2), and GARCH(1,1). The results, summarized in terms of RMSE, MAE, and directional accuracy, are presented in Table 4.

Table 4: Comparison with classical forecasting models: error metrics and directional accuracy (non-normalized values).

Model	RMSE ($)	MAE ($)	Directional Accuracy
GRU+Transformer Encoder	49.85	34.03	62.00%
ARIMA(5,1,2)	291.79	231.27	15.04%
GARCH(1,1)	289.09	230.57	49.88%

As shown, the GRU+Transformer model significantly outperforms the benchmark models, demonstrating superiority in both prediction error (RMSE and MAE) and market directional accuracy. These results indicate that the combined GRU and Transformer architecture not only provides higher numerical precision in short-term price forecasting but also better captures the direction of market movements.

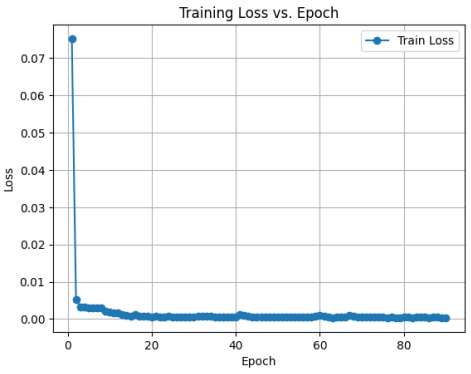


Figure 6: Training loss progression.

6 Discussion and Interpretation of Results

The integration of the GRU and Transformer Encoder models resulted in a hybrid architecture that significantly reduces prediction error. The GRU component effectively captures short-term temporal dependencies, while the Transformer Encoder excels at modeling long-term relationships within the time-series data. Numerical results underscore the superiority of this hybrid configuration when compared against individual baseline models.

The model was rigorously evaluated on an entirely unseen portion of the dataset, specifically the final 20% of the data chronologically following the training period. The figures comparing the predicted and actual Ethereum prices demonstrate a

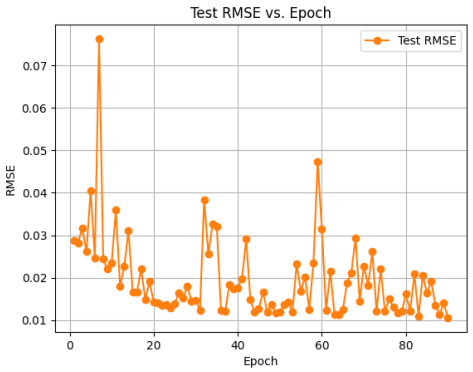


Figure 7: Test RMSE vs. epoch.

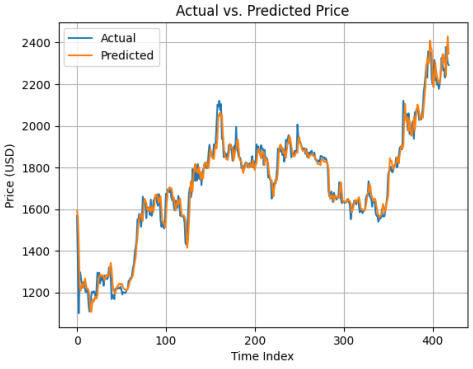


Figure 8: Predicted vs. actual Ethereum price.

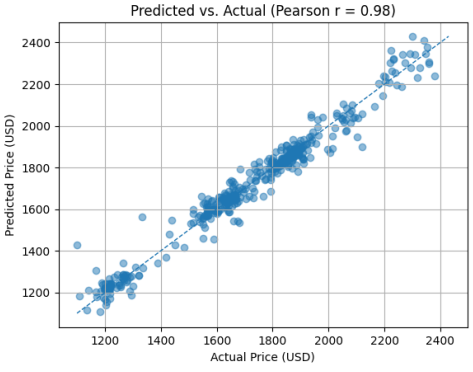


Figure 9: Predicted vs. actual Ethereum price: Pearson correlation ($r = 0.98$).

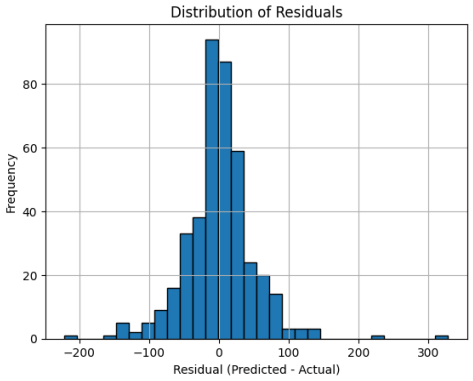


Figure 10: Analysis of model errors.

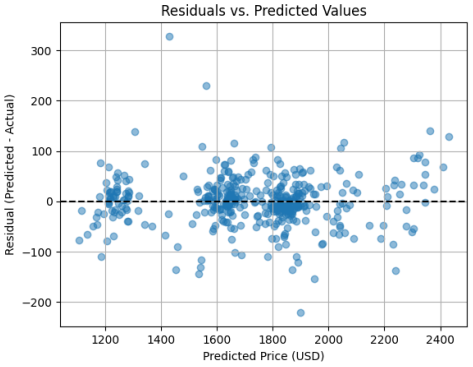


Figure 11: Residuals vs. predicted values.

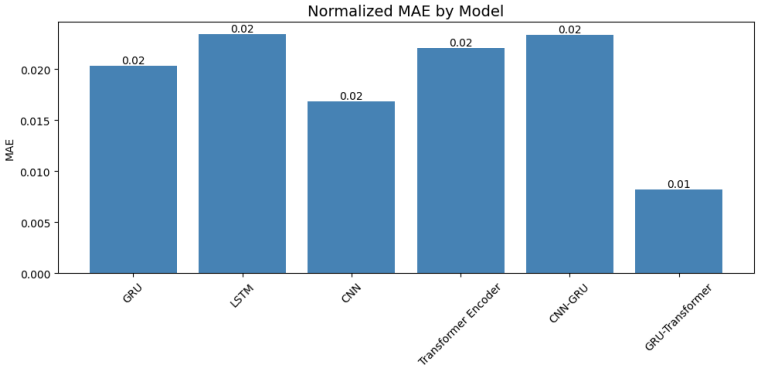


Figure 12: Normalized mean absolute error by model.

strong alignment with the real market trend. This alignment is maintained even during periods of heightened volatility within the study timeframe, including significant real-world events such as the COVID-19 pandemic, the Russia–Ukraine war, and cryptocurrency transaction bans.

Notably, the model exhibited robust performance on both the training and test datasets, a characteristic indicative of strong generalization capability and resilience to overfitting.

A key distinction of the proposed hybrid model, compared to many previous studies, is the incorporation of two significant technical indicators: OBV and ATR. Both indicators exhibit a strong correlation with Ethereums closing price, providing valuable supplementary information for the prediction task.

One of the primary objectives of this research was to develop a model capable of simultaneously processing short-term fluctuations and long-term trends—a capability that most single-architecture models either lack or struggle to achieve effectively. Ultimately, the GRU–Transformer Encoder combination successfully fulfilled this objective.

In this study, a hybrid model combining GRU and Transformer Encoder architectures was developed and rigorously evaluated for predicting the price of Ethereum cryptocurrency. The models architecture was specifically designed to overcome the limitations of traditional and standalone deep learning models in highly volatile markets, such as that of Ethereum.

The proposed hybrid model demonstrated significantly higher prediction accuracy compared to the baseline models. Specifically, the MAE reached 0.007199, a notably low value considering the inherent daily price volatility of Ethereum.

Comparative results against CNN, GRU, Transformer Encoder, LSTM, and CNN–GRU models showed that the proposed hybrid model consistently achieved the lowest error values across all evaluation metrics, including MAE, MSE, and RMSE. The incorporation of auxiliary features, specifically Bitcoins price alongside the technical indicators ATR and OBV, played a significant role in enhancing the models predictive accuracy.

Among the main advantages and innovations of this research is the selective utilization of the Transformers encoder block, rather than the full Transformer architecture, in combination with the GRU. This approach not only reduces model complexity but also accelerates the training process. Furthermore, the strategic inclusion of effective technical indicators represents a meaningful contribution to the models design, enabling a more nuanced understanding of market dynamics.

A primary limitation of this study stems from the data source utilized. Yahoo Finance does not provide Ethereum data at frequencies shorter than daily intervals, which may limit the capture of finer-grained market movements.

To analyze feature importance, a feature ablation approach was applied, in which each feature was individually replaced with a reference value (mean), and the resulting change in the models prediction error was measured. The baseline RMSE

of the model without any feature removal was 49.85 USD. The results indicate that the most significant impact comes from the ETH-high and ETH-low features, where replacing them with the mean increased the RMSE to 237.99 and 244.14 USD, corresponding to ΔRMSE of +188.14 and +194.29 USD, respectively. This demonstrates that Ethereums daily high and low prices are the most critical for final price prediction. Other features, such as trading volume (ETH-volume) and ATR, had smaller effects, with ΔRMSE around +4 USD. OBV, BTC-close, and open exhibited moderate effects, with ΔRMSE of +44.02, +21.19, and +39.70 USD, respectively. Overall, this analysis clearly shows that the model is most sensitive to ETH daily prices, while volume and other indicators provide complementary information.

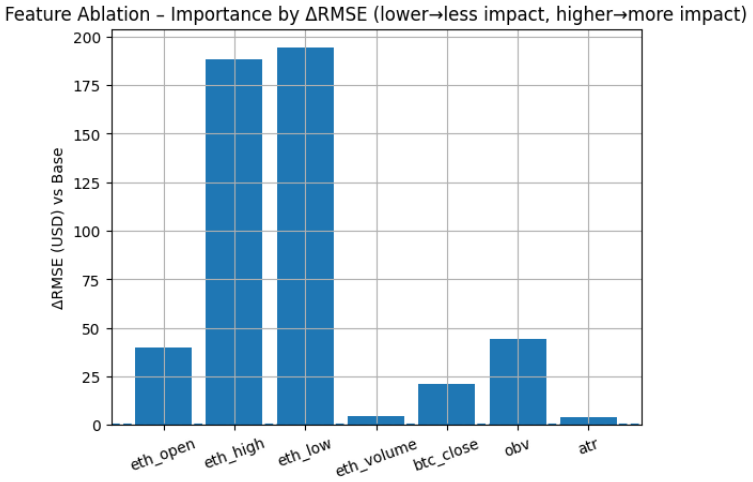


Figure 13: Feature ablation analysis.

To evaluate the model's practical applicability in trading, a simple trading strategy based on the model's buy/sell signals was tested over a 419-day period. The model achieved an average daily return of 0.249%, compared to 0.131% for the Buy & Hold strategy. Cumulative returns were 157.18% for the model versus 45.79% for Buy & Hold, highlighting a substantial improvement in performance. The annual Sharpe ratio was 2.19 for the model and 0.88 for Buy & Hold, indicating higher risk-adjusted returns for the prediction-based strategy. These results suggest that the GRU+Transformer model can be practically useful for trading decisions, providing higher returns than a simple market strategy.

To assess the influence of past information on the model's predictions, temporal importance was analyzed using the Integrated Gradients method. The results show that the importance of time steps increases as they approach the prediction moment. This finding indicates that the model assigns significantly greater weight to more recent data, with information from the latest time steps playing a crucial role in

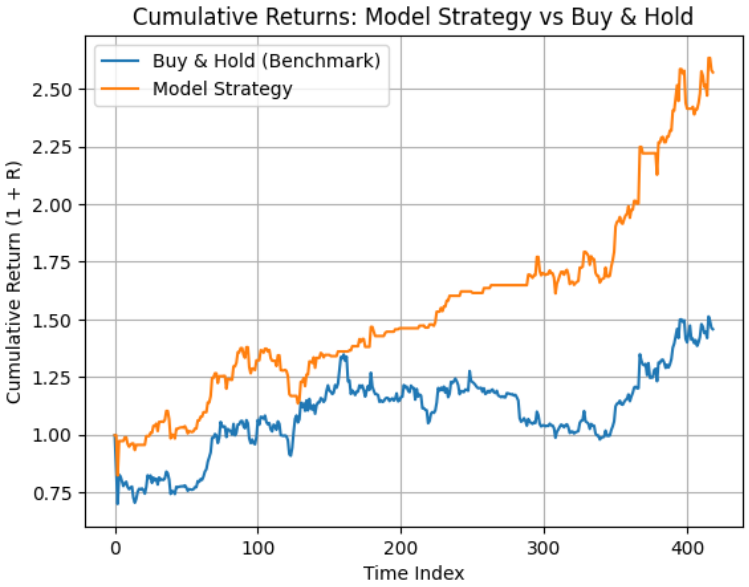


Figure 14: Practical utility: trading strategy backtesting.

its final decisions. Such behavior aligns well with the dynamic nature of financial markets, where recent information typically has the strongest impact on future trends.

It is recommended that future research explore architectures with parallel processing capabilities to more effectively capture both short-term fluctuations and long-term trends concurrently.

The presented model was developed solely for research and analytical purposes, and its use for real trading, market intervention, or any direct financial decision-making is not recommended. To prevent potential misuse or market disruption, appropriate limitations and security considerations have been incorporated into its implementation. All evaluations and analyses reported in this study were conducted in a controlled and simulated environment, and the authors do not endorse deploying the model in actual trading platforms without proper expert supervision and necessary safeguards. All code and data used in this study have been made openly available in a GitHub repository to ensure transparency and reproducibility of the results. The purpose of releasing the project is to enable researchers and practitioners to examine the methods, implement improvements, and pursue further developments.

https://github.com/aliheidarvand/Ethereum-Predictor

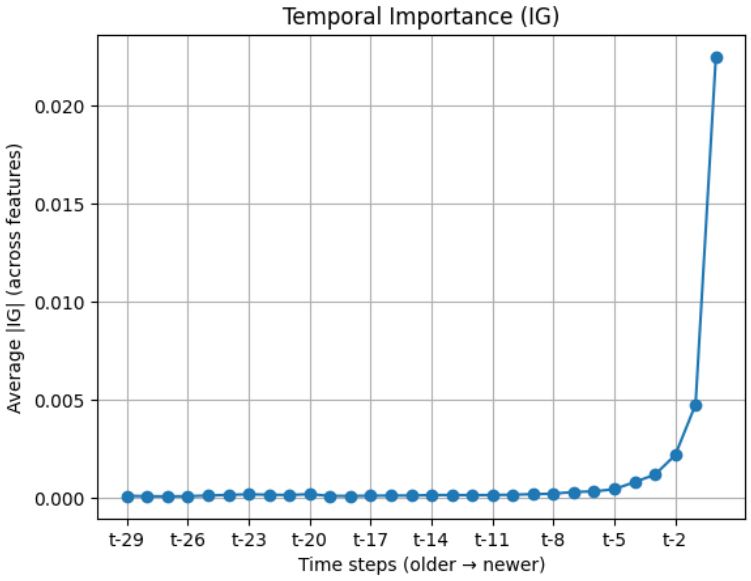


Figure 15: Temporal importance analysis.

Bibliography

[1] U. A. AULIYAH, *Cryptocurrencies price estimation using deep learning hybride model of LSTM–GRU*, The Indonesian Journal of Computer Science, **13** (2024), no. 4.

[2] K. CHO, B. VAN MERRIËNBOER, C. GULCEHRE, D. BAHDANAU, F. BOUGARES, H. SCHWENK AND Y. BENGIO, *Learning phrase representations using RNN encoder–decoder for statistical machine translation*, arXiv preprint arXiv:1406.1078, (2014).

[3] J. CHUNG, C. GULCEHRE, K. CHO AND Y. BENGIO, *Empirical evaluation of gated recurrent neural networks on sequence modeling*, arXiv preprint arXiv:1412.3555, (2014).

[4] J. E. GRANVILLE, *Granville's New Key to Stock Market Profits*, Pickle Partners Publishing, 2018.

[5] R. KAUR, M. UPPAL, D. GUPTA, S. JUNEJA, S. Y. ARAFAT, J. RASHID, J. KIM AND R. ALROOBAEA, *Development of a cryptocurrency price prediction model: leveraging GRU and LSTM for Bitcoin, Litecoin and Ethereum*, PeerJ Computer Science, **11** (2025), e2675.

[6] D. P. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014).

[7] M. C. LEE, *Temporal Fusion Transformer-based trading strategy for multi-crypto assets using on-chain and technical indicators*, Systems, **13** (2025), no. 6, 474.

[8] E. MAHDI, C. MARTIN-BARREIRO AND X. CABEZAS, *A novel hybrid approach using an attention-based Transformer+GRU model for predicting cryptocurrency prices*, Mathematics, **13** (2025), no. 9, 1484.

[9] S. MONISH, M. MOHTA AND S. RANGASWAMY, *Ethereum price prediction using machine learning techniques – A comparative study*, International Journal of Engineering Applied Sciences and Technology, **7** (2022), 137–142.

[10] A. MOHAMMADJAFARI, *Comparative study of Bitcoin price prediction*, arXiv preprint arXiv:2405.08089, (2024).

[11] K. MURRAY, A. ROSSI, D. CARRARO AND A. VISENTIN, *On forecasting cryptocurrency prices: A comparison of machine learning, deep learning, and ensembles*, Forecasting, **5** (2023), no. 1, 196–209.

[12] A. Saputra, A. N. Hidayat and D. Siahaan, *Ethereum price prediction using LSTM and GRU*, Proc. 5th Int. Conf. on Computer and Communication Engineering Technology (CCET), (2025).

[13] S. Siami-Namini and A. Siami Namin, *Forecasting economics and financial time series: ARIMA vs. LSTM*, arXiv preprint arXiv:1803.06386, (2018).

[14] S. Tanwar, N. P. Patel, S. N. Patel, J. R. Patel, G. Sharma and I. E. Davidson, *Deep learning-based cryptocurrency price prediction scheme with inter-dependent relations*, IEEE Access, **9** (2021), 138633–138646.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, *Attention is all you need*, Advances in Neural Information Processing Systems, **30** (2017).

[16] J. W. Wilder Jr., *New Concepts in Technical Trading Systems*, Trend Research, Greensboro, NC, 1978.

How to Cite: Yones Esmaeelzade Aghdam[1], Hamid Mesgarani[2], Ali Heidarvand[3], *Ethereum Price Prediction with a GRU–Transformer Encoder Hybrid Model*, Journal of Mathematics and Modeling in Finance (JMMF), Vol. 6, No. 1, Pages:67–89, (2026).