American Journal of Data Science and Artificial Intelligence (AJDSAI)

ISSN: 3069-3632 (ONLINE)

VOLUME 2 ISSUE 2 (2026)

PUBLISHED BY
E-PALLI PUBLISHERS, DELAWARE, USA

American Journal of Data Science and Artificial Intelligence (AJDSAI)

Volume 2 Issue 2, Year 2026
ISSN: 3069-3632 (Online)
DOI: https://doi.org/10.54536/ajdsai.v2i2.8367
https://journals.e-palli.com/home/index.php/ajdsai

Benchmarking Deep Learning Models for Bitcoin Price Forecasting

Catherine Ngo[1], Orson Chi[1], Yeong Nain Chi[1*]

Article Information

Received: May 27, 2026

Accepted: August 16, 2026

Published: September 19, 2026

Keywords

Bitcoin Price Forecasting, Cryptocurrency Markets, Deep Learning, Financial Volatility, Gated Recurrent Unit (GRU), Time Series Prediction, Transformer Models

ABSTRACT

The rapid appreciation and pronounced volatility of Bitcoin (BTC-USD) poses significant challenges for accurate price forecasting using financial time series models. This paper presents a systematic benchmarking study of six deep learning architectures, including Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), Vanilla Recurrent Neural Network (RNN), one dimensional Convolutional Neural Network (CNN), Temporal Convolutional Network (TCN), and Transformer Encoder, for daily BTC USD price prediction. Experiments are conducted using 2,283 observations spanning January 2020 to March 2026 under a standardized training pipeline that includes a 60 day lookback window, Huber loss, Adam optimization, and a strict chronological split of 75% training, 10% validation, and 15% testing to mitigate data leakage. Empirical results indicate that the GRU consistently achieves superior out of sample performance, with a test coefficient of determination (R^2) of 0.9138, root mean squared error (RMSE) of USD 4,911 and mean absolute percentage error (MAPE) of 3.84%. In contrast, Transformer, LSTM, TCN, CNN, and Vanilla RNN models exhibit substantially weaker generalization performance. The results suggest that the GRU's gating mechanism, which enables selective retention and attenuation of historical information, is particularly effective in capturing regime dependent dynamics in highly volatile cryptocurrency markets. A 30 day out of sample forecast for April 2026 is further generated for all models, with the GRU projecting Bitcoin prices in the USD 66,000–67,000 range. The proposed benchmark provides reproducible experimental settings, quantitative performance comparisons, and architectural insights to support informed model selection for cryptocurrency price forecasting.

INTRODUCTION

Cryptocurrency markets, led by Bitcoin, represent one of the most dynamic and data-rich financial ecosystems of the twenty-first century. Since its introduction in 2009, Bitcoin has experienced a cumulative return of approximately 848% between January 2020 and March 2026 alone, growing from $7,200 to $68,233 with an intra-period peak of $126,198 in 2025. Yet this extraordinary appreciation is accompanied by daily return volatility of 3.19%, infrequent but severe drawdowns of up to −76.6%, and distributional characteristics characterized by leptokurtosis (excess kurtosis = 11.33) and negative skewness (−0.47) that violate the assumptions of classical econometric models (Nakamoto, 2008; Dyhrberg, 2016). Traditional time series methods such as ARIMA, GARCH, and exponential smoothing have demonstrated limited efficacy in capturing the non-linear, non-stationary dynamics of cryptocurrency prices (Katsiampa, 2017; Phillip *et al.*, 2018). The emergence of deep learning has opened new avenues for financial prediction, with recurrent architecture (LSTM, GRU) showing promise due to their capacity to model long-range temporal dependencies (Hochreiter & Schmidhuber, 1997; Cho *et al.*, 2014). More recently, convolutional approaches (CNN, TCN) and attention-based architectures (Transformer) have been investigated as fully parallel alternatives that circumvent the sequential processing bottleneck of recurrent networks. Despite a growing body of literature on individual architectures, rigorous head-to-head comparisons under controlled experimental conditions remain scarce, particularly across the full six-model spectrum and over the extended 2020–2026 period that encompasses multiple distinct market regimes: the COVID-19 crash and recovery (2020), the institutional adoption bull run (2021), the Terra-Luna and FTX-driven bear market (2022), the ETF-driven recovery (2023–2024), and the record-high regime (2025). This study addresses this gap by making the following contributions: (1) a standardized comparative evaluation of six deep learning architectures under identical data preprocessing, training conditions, and evaluation metrics; (2) progressive feature engineering that expands from 11 (LSTM) to 28 (Transformer) technical indicators to investigate the value of richer representations; (3) formal mathematical derivations of all model equations alongside conceptual architecture diagrams; (4) an empirical demonstration that the GRU's gating mechanism provides superior out-of-sample generalization across a challenging six-year multi-regime dataset; and (5) 30-day price forecasts for April 2026 from all six architectures, providing an actionable forward-looking analysis. The remainder of this paper is structured as follows. Section 2 reviews

[1] University of Maryland Eastern Shore, United States
[*] Corresponding author's e-mail: ychi@umes.edu

the relevant literature. Section 3 describes the data and materials. Section 4 presents experimental methodology and model architecture. Section 5 reports the results. Section 6 discusses the findings with emphasis on the best-performing GRU model. Section 7 concludes with limitations and future directions.

LITERATURE REVIEW

Early work on Bitcoin price prediction relied heavily on econometric frameworks. Katsiampa (2017) applied GARCH-family models to capture Bitcoin volatility clustering, finding that the AR-CGARCH model best described conditional heteroscedasticity. Phillip *et al.* (2018) demonstrated that cryptocurrency returns exhibit regime-switching behavior incompatible with stationary ARIMA specifications. Bariviera *et al.* (2017) confirmed long-range dependence in Bitcoin returns using Hurst exponent analysis, motivating the use of models with long-term memory capacity. Sentiment-augmented models were explored by Garcia and Schweitzer (2015), who found that tweet volume and emotional valence Granger-cause Bitcoin price movements. Mai *et al.* (2018) extended this approach using Natural Language Processing (NLP) on Bitcoin Talk forum posts, achieving improved prediction accuracy over baseline technical models. However, the practical challenges of real-time sentiment data acquisition have limited the adoption of these approaches in deployable systems. The LSTM network (Hochreiter & Schmidhuber, 1997) addressed the vanishing gradient problem of vanilla RNNs through four multiplicative gates, including forget, input, candidate, and output, and an additive cell state update that provides a gradient highway across time. LSTM networks were among the first deep learning architectures applied to Bitcoin prediction: Fischer and Krauss (2018) applied LSTMs to S&P 500 constituents with returns exceeding buy-and-hold benchmarks, while Siami-Namini *et al.* (2019) compared LSTM and ARIMA on financial time series, finding consistent LSTM superiority. The GRU (Cho *et al.*, 2014) simplified the LSTM by merging the cell state and hidden state into a unified representation updated via a reset gate and update gate interpolation. Empirical comparisons by Chung *et al.* (2014) found GRUs to match LSTM performance on several sequential tasks with 25% fewer parameters. For cryptocurrency forecasting, Livieris *et al.* (2020) and Uras *et al.* (2020) reported GRU models achieving competitive accuracy with faster training convergence, particularly on shorter time series. One-dimensional CNNs for time series were explored by Bai *et al.* (2018a), who demonstrated that causal dilated convolutions could match or exceed recurrent architectures on sequence modelling benchmarks with substantially lower computational cost. The key properties, local pattern detection, full parallelism, and absence of gradient propagation through time, make CNNs particularly attractive for long financial sequences. The TCN (Bai *et al.*, 2018b) extended standard causal CNNs with exponentially dilated convolutions and residual connections, achieving receptive fields of hundreds of timesteps while maintaining training stability. TCNs have been applied to exchange rate prediction (Wan *et al.*, 2019) and multi-variate financial forecasting (Giantsidi & Tarantola, 2025), with mixed results depending on the extent of temporal regime change in the target series. The Transformer architecture (Vaswani *et al.*, 2017) replaced sequential computation with multi-head self-attention, achieving O(1) path length between any two timesteps. Temporal fusion transformers (Lim *et al.*, 2021) and financial-domain variants including TFT (Lim *et al.*, 2021), Informer (Zhou *et al.*, 2021), and PatchTST (Nie *et al.*, 2023) have shown strong performance on long-horizon forecasting. For cryptocurrency prediction, Ramos-Pérez *et al.* (2021) applied Transformer encoders to hourly BTC data, achieving lower MAPE than baseline LSTM models on short forecast horizons. Several limitations characterize existing literature. First, most comparative studies evaluate models over short periods (typically 1–3 years) that do not encompass multiple market regime transitions. Second, feature engineering practices vary dramatically across studies, confounding architecture-level comparisons. Third, few studies include more than three architectures in a single controlled experiment. Fourth, the extended 2020–2026 period, encompassing the COVID crisis, institutional adoption, the FTX collapse, the spot ETF approval, and the 2025 all-time high, has not been studied comprehensively. This study addresses all four limitations.

MATERIALS AND METHODS

materials

Daily OHLCV (Open, High, Low, Close, Volume) data for Bitcoin (BTC-USD) was obtained from Yahoo Finance covering January 1, 2020, through March 31, 2026. The dataset comprises 2,283 daily observations with no missing values. All price series are adjusted for splits and dividends (where applicable). The dataset spans seven distinct calendar years encompassing multiple market regimes as described in Table 1.

Table 1: BTC-USD annual price statistics (2020–2026 Q1)

Year	Opening Price	Closing Price	Year High	Year Low	Annual Return
2020	$7,194.89	$29,001.72	$29,244.88	$4,106.98	+303.1%
2021	$28,994.01	$46,306.45	$68,789.63	$28,722.76	+59.7%
2022	$46,311.75	$16,547.50	$48,086.84	$15,599.05	−64.3%
2023	$16,547.91	$42,265.19	$44,705.52	$16,521.23	+155.4%
2024	$42,280.23	$93,429.20	$108,268.45	$38,521.89	+121.0%

2025	$93,425.10	$87,508.83	$126,198.07	$74,436.68	−6.3%
2026 (Q1)	$87,508.05	$68,233.31	$97,860.60	$60,074.20	−22.0%

Table 2 summarizes the key statistical properties of daily BTC-USD returns and price levels over the full study period. The distributional characteristics confirm the unsuitability of Gaussian-assumption models: excess kurtosis of 11.33 indicates heavy tails, negative skewness of −0.47 reflects asymmetric downside risk, and the Jarque-Bera p-value of 0.00 rejects normality at all conventional significance levels. Model training utilized early stopping with patience of 10–12 epochs and learning rate scheduling (ReduceLROnPlateau). All random seeds were fixed at 42 for reproducibility.

Table 2: Descriptive statistics for BTC-USD daily price and returns (2020–2026 Q1)

Statistic	**Value**
Observations	2,283 trading days
Period	January 1, 2020 – March 31, 2026
Starting Price	$7,200.17
Ending Price	$68,233.31
Period Maximum	$126,198.07 (December 17, 2025)
Period Minimum	$4,106.98 (March 13, 2020)
Cumulative Return	+847.7%
CAGR	43.3% per annum
Mean Daily Return	+0.150%
Daily Return Std Dev	3.19%
Skewness	−0.47
Excess Kurtosis	11.33
Jarque-Bera p-value	< 0.001 (rejects normality)
Sharpe Ratio (annualized)	0.67
Maximum Drawdown	−76.6% (peak: Nov 2021, trough: Nov 21, 2022)
Positive Return Days	1,157 (50.7%)
Negative Return Days	1,124 (49.3%)

Methods

Model Architectures

To evaluate the effectiveness of different deep learning paradigms for BTC-USD price forecasting, six neural network architectures were investigated: LSTM, GRU, Vanilla RNN, one-dimensional CNN, TCN, and Transformer Encoder. These architectures represent recurrent, convolutional, and attention-based approaches to sequence modeling. All models received identical input sequences and were trained under the same experimental framework to ensure a fair performance comparison.

Long Short-Term Memory (LSTM)

The LSTM network (Hochreiter & Schmidhuber, 1997) addresses the vanishing-gradient problem commonly encountered in conventional recurrent neural networks through a gated memory mechanism that selectively stores, updates, and retrieves temporal information. The architecture employs forget, input, and output gates to regulate information flow:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (1)$$
$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$
$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (3)$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (4)$$

The cell and hidden states are updated as:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (5)$$
$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

The implemented architecture consisted of three stacked LSTM layers containing 128, 64, and 32 hidden units, respectively, with a dropout rate of 0.2 applied between layers.

Gated Recurrent Unit (GRU)

The GRU was proposed as a computationally efficient alternative to LSTM networks while retaining the ability to capture long-term temporal dependencies (Cho *et al.*, 2014). The GRU combines the cell state and hidden state into a single memory representation and employs reset and update gates:

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \quad (7)$$
$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z) \quad (8)$$
$$\tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, x_t] + b_h) \quad (9)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (10)$$

The GRU model consisted of three stacked recurrent layers with 128, 64, and 32 units and a dropout rate of 0.2.

Vanilla Recurrent Neural Network (RNN)

The Vanilla RNN (Elman, 1990) served as a baseline sequential learning model. The hidden state is recursively updated according to:

$$h_t = \tanh\left(W_h[h_{t-1}, x_t] + b_h\right) \quad (11)$$

where the current hidden state depends on the previous hidden state and current input. Although conceptually simple, vanilla RNNs are susceptible to vanishing and exploding gradients when modeling long sequences. The implemented architecture comprised four stacked SimpleRNN layers with 256, 128, 64, and 32 units, respectively, combined with layer normalization and dropout regularization (0.20-0.25).

One-Dimensional Convolutional Neural Network (1D-CNN)

The one-dimensional CNN is based on convolutional operations originally developed for hierarchical feature extraction in neural networks (LeCun *et al.*, 1998). For time-series forecasting, causal convolutions are employed such that predictions at time t depend only on present and past observations:

$$y_t = \mathrm{ReLU}\left(\sum_{j=0}^{k-1} w_j\, x_{t-j} + b\right) \quad (12)$$

The architecture consisted of four convolutional blocks with filter sizes of 64, 128, 128, and 64 and kernel sizes of 3, 5, 7, and 3, respectively. A max-pooling layer was applied after the third block, followed by global average pooling and fully connected layers.

Temporal Convolutional Network (TCN)

The TCN extends conventional CNNs through the use of causal and dilated convolutions combined with residual connections, enabling efficient modeling of long-range temporal dependencies (Bai *et al.*, 2018). A residual block can be expressed as:

$$Z^{(l)} = \mathrm{ReLU}\left(B^{(l)} + R^{(l)}\right) \quad (13)$$

where B^((l)) denotes the output of dilated convolutional layers and R^((l)) represents the residual connection. The TCN architecture comprised six residual blocks with dilation rates of 1, 2, 4, 8, 16, and 32, and filter sizes of 32, 32, 64, 64, 128, and 128, respectively, allowing the network to capture long-range temporal patterns while maintaining computational efficiency.

Transformer Encoder

The Transformer architecture introduced an attention-based framework that eliminates recurrent computations and models interactions among all positions in a sequence simultaneously (Vaswani *et al.*, 2017). The scaled dot-product attention mechanism is defined as:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (14)$$

where Q, K, and V are the query, key, and value matrices, respectively. Positional encoding is incorporated to preserve sequence order:

$$PE(t, 2i) = \sin\left(\frac{t}{10000^{2i/d}}\right) \quad (15)$$

$$PE(t, 2i+1) = \cos\left(\frac{t}{10000^{2i/d}}\right) \quad (16)$$

The implemented Transformer Encoder consisted of three stacked encoder layers with four attention heads and a feed-forward network dimension of 128, followed by global average pooling and a regression output layer.

Table 3 presents a comparative overview of the six deep learning architectures evaluated in this study, highlighting their structural design, layer configuration, number of trainable parameters, feature complexity, and primary learning mechanisms. The table illustrates the substantial architectural diversity among recurrent, convolutional, and attention-based models and provides important context for interpreting their predictive performance.

Table 3: Architecture summary of all six deep learning models

Model	Architecture	Layers	Parameters	Features	Key Mechanism
LSTM	3-Layer Stacked LSTM	128→64→32 units	134,049	11	Cell state + 4 gates
GRU	3-Layer Stacked GRU	128→64→32 units	102,113	13	Reset + update gates
RNN	4-Layer SimpleRNN + LN	256→128→64→32	136,993	16	tanh recurrence only
CNN	4-Block Causal Conv1D	f=[64,128,128,64]	204,865	21	Parallel local conv
TCN	6-Block Dilated Residual	f=[32,32,64,64,128,128]	251,585	23	Dilated conv + residual
Transformer	3-Layer Encoder	d_model=64, h=4	120,961	28	Multi-head self-attention

The LSTM model employed a three-layer stacked architecture with 128, 64, and 32 hidden units, resulting in 134,049 trainable parameters. Its defining characteristic is the use of a memory cell regulated by four gating mechanisms (forget, input, candidate, and output gates), which enable the network to preserve relevant historical information while discarding less useful signals. This gated memory structure is specifically designed to mitigate the vanishing-gradient problem and improve the modeling of long-term temporal dependencies in financial time series. Despite having a moderate parameter count, the LSTM architecture provides substantial representational capacity through its internal memory state.

The GRU model adopted a similar three-layer recurrent structure (128→64→32 units) but required only 102,113 trainable parameters, approximately 24% fewer than the LSTM. The reduction in complexity arises from the GRU's simplified design, which replaces the separate cell state and hidden state with a unified memory representation controlled by reset and update gates. Consequently, the GRU can often achieve predictive accuracy comparable to LSTM while requiring less computational effort and memory. The lower parameter count also reduces the risk of overfitting when training data are limited.

The Vanilla RNN served as the simplest recurrent baseline. Although it contained 136,993 parameters and four recurrent layers (256→128→64→32), its learning mechanism relied solely on standard tanh recurrent connections without any gating structure. As a result, information must propagate through repeated nonlinear transformations, making the model vulnerable to vanishing and exploding gradients when learning long-term dependencies. The inclusion of Layer Normalization and additional layers increased model capacity and training stability; however, the absence of memory gates generally places the RNN at a disadvantage relative to LSTM and GRU models for longer forecasting horizons.

The CNN architecture consisted of four causal convolutional blocks with filter sizes of [64, 128, 128, 64] and a total of 204,865 trainable parameters, substantially more than the recurrent models. Unlike RNN-based approaches, the CNN processes all timesteps simultaneously through convolutional filters, enabling highly efficient parallel computation. Its key mechanism, parallel local convolution, is particularly effective for extracting short-term temporal patterns and local price fluctuations. However, because standard convolutions possess a limited receptive field, CNNs may have difficulty capturing very long-range dependencies unless deeper architectures are employed.

The TCN model represented the most complex convolution-based architecture evaluated in the study. It consisted of six dilated residual blocks with progressively increasing filter sizes [32, 32, 64, 64, 128, 128] and contained 251,585 trainable parameters, the largest among all models examined. The combination of dilated convolutions and residual connections allows the TCN to achieve a very large receptive field while maintaining efficient parallel processing. Consequently, the TCN can simultaneously capture both short-term market fluctuations and long-range temporal dependencies without relying on recurrent computations. The large parameter count reflects the additional complexity required to support this expanded temporal coverage.

The Transformer Encoder employed a three-layer attention-based architecture with an embedding dimension (d_model) of 64 and four attention heads, yielding 120,961 trainable parameters. Although its parameter count was lower than those of CNN and TCN, the Transformer exhibited the highest feature complexity, utilizing 28 engineered features. The model's principal advantage lies in its multi-head self-attention mechanism, which enables direct interaction between all timesteps within a sequence. Unlike recurrent models that process data sequentially or CNNs that focus on local neighborhoods, the Transformer can learn global temporal relationships regardless of the distance between observations. This characteristic makes it particularly well suited for identifying complex, non-local dependencies in commodity price series.

A comparison of parameter counts reveals that model complexity varied substantially across architectures. The GRU was the most parameter-efficient model (102,113 parameters), whereas the TCN was the most computationally intensive (251,585 parameters). CNN and TCN architectures generally required more parameters because multiple convolutional filters must be learned across several layers. In contrast, recurrent and attention-based models achieved comparable representational capacity using fewer trainable weights. These differences highlight the trade-off between computational complexity and model expressiveness.

The Features column further illustrates increasing architectural sophistication, ranging from 11 features for the LSTM to 28 features for the Transformer. This progression reflects the ability of more advanced architectures to exploit richer representations of the input data. Nevertheless, a larger number of features or parameters does not necessarily guarantee superior forecasting performance. Instead, performance depends on how effectively each architecture captures the underlying temporal dynamics of commodity prices. Therefore, the results presented in subsequent performance tables should be interpreted in conjunction with the architectural characteristics summarized in Table 5.

Overall, Table 5 demonstrates that the study evaluated models spanning three major deep learning paradigms: recurrent learning (RNN, LSTM, GRU), convolutional learning (CNN, TCN), and attention-based learning (Transformer). This diverse set of architectures enables a comprehensive assessment of how different temporal modeling strategies influence agricultural commodity price forecasting accuracy, computational complexity, and generalization capability.

Experimental Framework and Data Pipeline

A standardized experimental framework was applied uniformly across all six models (Table 4). Raw OHLCV data is first preprocessed by a MinMaxScaler fitted on the training partition only, scaling each feature to [0, 1]. A sliding window of SEQ_LEN = 60 consecutive trading days (approximately three calendar months) is used to construct input sequences, with the model predicting the following day's closing price. The dataset is partitioned chronologically (no shuffling) into 75% training, 10% validation, and 15% test sets. This strict temporal partitioning prevents data leakage: no information from the validation or test periods enters any preprocessing or fitting step. The Huber loss function ($\delta = 1$) is used for all models, providing quadratic sensitivity to small errors while being robust to the occasional extreme price movements characteristic of cryptocurrency markets. The Adam optimizer with learning rate 0.001 is used throughout, with ReduceLROnPlateau scheduling applying a 0.4–0.5× decay factor when validation loss fails to improve over 5–6 consecutive epochs. All forecasts are inverse-transformed through the fitted close-price scaler before metric computation.

Table 4: Shared experimental configuration across all models

Parameter	Value
Lookback window (SEQ_LEN)	60 trading days
Prediction horizon	1 day (next-day close price)
Train / Validation / Test split	75% / 10% / 15% (chronological)
Training period	Jan 22, 2020 – Sep 11, 2024
Validation period	Sep 12, 2024 – Apr 25, 2025
Test period	Apr 26, 2025 – Mar 31, 2026
Loss function	Huber ($\delta = 1$)
Optimizer	Adam (lr=0.001, β_1=0.9, β_2=0.999)
LR scheduler	ReduceLROnPlateau (factor=0.4–0.5, patience=5–6)
Early stopping	Patience = 10–12, restore best weights
Feature scaling	MinMaxScaler [0, 1] (fit on train only)
Random seed	42

Feature Engineering

A progressive feature engineering strategy is adopted across the six models, incrementally adding technical indicators to examine the relationship between feature richness and predictive performance. All indicators are computed from raw OHLCV data using standard formulations and are scaled alongside price features. Table 5 lists all 28 features used by the Transformer (the most feature-rich model), with the subset used by each model indicated.

Table 5: Feature engineering catalogue

Feature	Description	Models
Close	Adjusted closing price (target)	All
Open / High / Low	OHLC price components	All
Volume	Daily trading volume	All
Return	Daily percentage return	All
Log_Return	Natural log return ratio	RNN+
MA7 / MA21 / MA50	7, 21, 50-day simple moving averages	All
EMA12 / EMA26	12 and 26-day exponential moving averages	CNN+
Volatility (21d / 7d)	Rolling standard deviation of returns	GRU+
HL_Spread	(High − Low) / Close	LSTM+
OC_Spread	(Close − Open) / Open	LSTM+
Volume_Ratio	Volume / 7-day avg volume	GRU+
RSI	14-day Relative Strength Index	GRU+
MACD / Signal / Hist	MACD line, signal, histogram	GRU+
BB_Upper / BB_Lower / BB_Width	20-day Bollinger Bands	CNN+
OBV	On-Balance Volume (cumulative)	CNN+
Momentum10 / Momentum5	10-day and 5-day price momentum	CNN+
ATR	14-day Average True Range	Transformer

Note: "Model+" indicates the first model in the series to include a feature

Evaluation Metrics

Predictive performance was evaluated using five complementary metrics computed on the original USD price scale after inverse transformation. RMSE and MAE were employed to quantify absolute forecasting error, with RMSE placing greater emphasis on large prediction deviations. MAPE provided a scale-free measure of relative forecasting accuracy, while the coefficient of determination (R^2) assessed the ability of each model to explain variability in commodity prices. In addition, Directional Accuracy (DA) was used to evaluate the correctness of predicted price movement directions, offering practical insight into the utility of the models for market trend forecasting and trading-oriented decision making.The 30-day iterative forecast for April 2026 is generated by sequentially appending each model's predicted value back into the 60-day input window, replacing the oldest observation. Forecast uncertainty grows with the horizon as predicted values replace observed inputs.

RESULTS AND DISCUSSION

Table 6 presents the comprehensive performance metrics for all six models on the test set (April 26, 2025 – March 31, 2026). The GRU achieves dominant performance across all magnitude-based metrics, with an R^2 of 0.9138 indicating that it explains 91.4% of the variance in BTC-USD test-set prices, a result that is qualitatively different from all other models in the comparison. The Transformer ranks second (R^2 = 0.4942), followed by the LSTM (R^2 = 0.4415). The three remaining models (TCN, CNN, RNN) exhibit negative R^2 values, indicating performance inferior to the mean-prediction baseline on the test set.

Table 6: Comparative performance metrics

Metric	LSTM	GRU	RNN	CNN	TCN	Transformer
Test RMSE (USD)	$12,502	$4,911 ★	$40,009	$37,459	$23,915	$11,981
Test MAE (USD)	$10,889	$3,932 ★	$36,524	$34,707	$21,151	$10,219
Test MAPE (%)	10.34%	3.84% ★	35.05%	33.61%	20.61%	10.12%
Test R^2	0.4415	0.9138 ★	−4.7192	−3.9439	−1.0151	0.4942
Test Dir. Acc. (%)	52.5%	47.5%	50.7%	53.9%	50.9%	49.1%
Parameters	134,049	102,113 ★	136,993	204,865	251,585	120,961
Training Epochs	19	23	26	25	12	14
Input Features	11	13	16	21	23	28

Note: ★ = *best value.*

Notably, directional accuracy (DA) does not align with magnitude metrics. The CNN achieves the highest DA (53.9%) despite having the second-worst MAPE and a strongly negative R^2. Conversely, the GRU achieves only 47.5% DA below random suggesting that even the best magnitude predictor does not reliably predict the sign of next-day price changes in this period. This result is consistent with the near-efficient market hypothesis for short-horizon cryptocurrency price direction. Table 7 presents the full train/validation/test performance profile for each model, revealing important differences in generalization behavior. The GRU demonstrates the most consistent performance across all three partitions, with R^2 degrading only from 0.9821 (train) to 0.9138 (test) a 6.9-percentage-point drop compared to the LSTM's 53.1pp drop (0.9723→0.4415). The RNN, CNN, and TCN show dramatic generalization failures (all positive training R^2 but strongly negative test R^2), consistent with their inability to model the structural shifts in price dynamics during the test period (April 2025 – March 2026). The TCN's unusually poor training R^2 (−0.0775) reflects its aggressive early stopping at epoch 12 (best: epoch 4), preventing in-sample overfitting at the cost of limited pattern learning.

Table 7: Train / Validation / Test R^2 and MAPE for all six models

Model	Train R^2	Val R^2	Test R^2	Train MAPE	Val MAPE	Test MAPE
LSTM	0.9723	0.5812	0.4415	7.37%	8.51%	10.34%
GRU	0.9821	0.9320	0.9138 ★	6.41%	3.18%	3.84% ★
RNN	0.9343	−3.0459	−4.7192	16.37%	26.38%	35.05%
CNN	0.8597	−3.0450	−3.9439	25.79%	25.42%	33.61%
TCN	−0.0775	−0.2065	−1.0151	49.48%	14.36%	20.61%
Transformer	0.8825	0.6915	0.4942	11.06%	6.05%	10.12%

Figure 1 presents the comprehensive six-model performance comparison. The GRU's substantially lower RMSE ($4,911) and highest R^2 (0.9138) are clearly visible, as is the parameter efficiency of the Transformer (120,961 parameters, second-best R^2) relative to the larger CNN and TCN models.

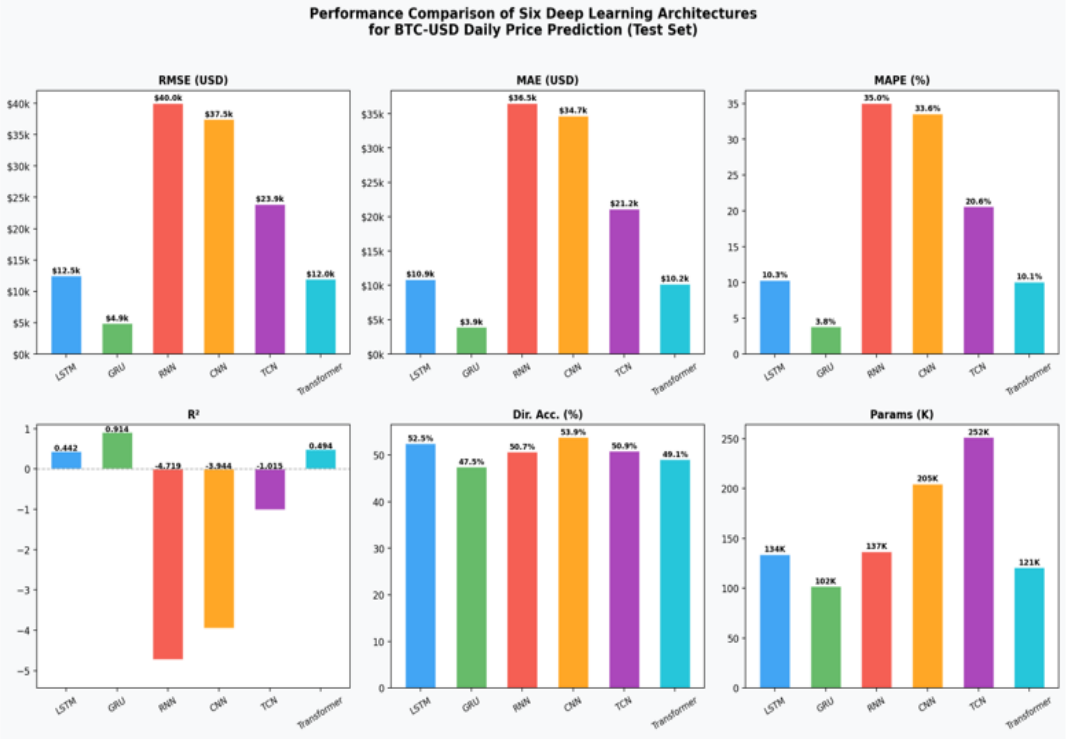

Figure 1: Comprehensive Performance Comparison RMSE, MAE, MAPE, R^2, Directional Accuracy, and Parameter Count for All Six Models on the Test Set (April 2025 March 2026).

Figure 2 provides a detailed diagnostic panel for the best-performing GRU model, including the full historical price series with train/validation/test predictions, training convergence curves, residual analysis, and the April 2026 forecast.

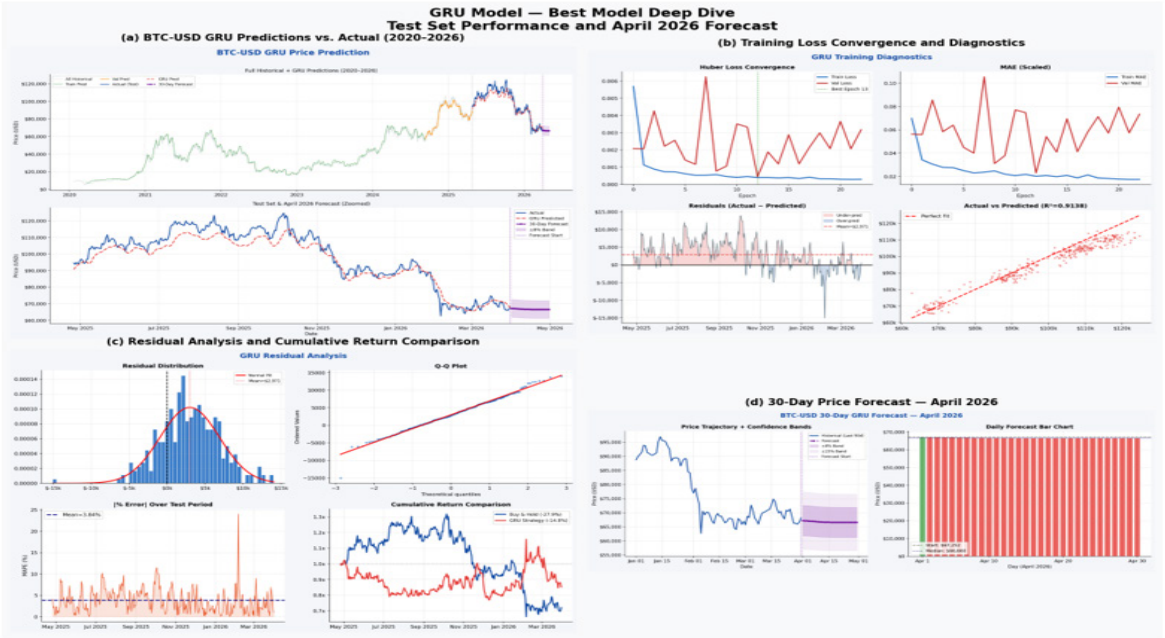

Figure 2: GRU Model Diagnostic Panel — (a) Full Time Series Predictions vs Actual (2020–2026), (b) Training Loss and MAE Convergence, (c) Residual Analysis and Cumulative Return Comparison, (d) 30-Day April 2026 Forecast

Table 8 presents the 30-day April 2026 forecast summary from all six models. The GRU projects a mildly declining-then-stable trajectory in the $66,000–$67,252 range, consistent with the price levels observed at the end of March 2026 ($68,233). The Transformer and LSTM produce similar forecasts ($65,000–$66,000), while the TCN projects significantly higher prices ($82,000–$84,000) and the RNN converges to a near-flat trajectory around $61,000. The CNN forecasts an initial dip to approximately $57,964 before recovering to $63,247 by day 30.

Table 8: 30-Day BTC-USD price forecast summary — April 1–30, 2026.

Model	Day 1 Forecast	Day 15 Forecast	Day 30 Forecast	Range	Direction
LSTM	$66,211	~$64,500	$63,890	$63K–$66K	Mild decline
GRU ★	$67,252	$66,666	$66,602	$66K–$67K	Stable/flat
RNN	$61,119	~$61,000	$60,836	$60K–$61K	Near-flat
CNN	$57,964	~$60,000	$63,247	$58K–$63K	Recovery
TCN	$82,945	~$83,300	$83,709	$83K–$84K	Upward
Transformer	$66,048	$64,908	$65,474	$64K–$66K	Mild decline

Note: ★ = *best-performing model.*

Figure 3 visualizes all six forecast trajectories. The GRU (highlighted) and Transformer/LSTM cluster around the $65,000–$67,000 range, providing the most internally consistent forward projections given the end-of-March 2026 actual price of $68,233. The TCN's elevated forecast ($83,000) reflects its detection of bullish momentum in the final 60 observed days but should be interpreted with caution given its negative test R^2.

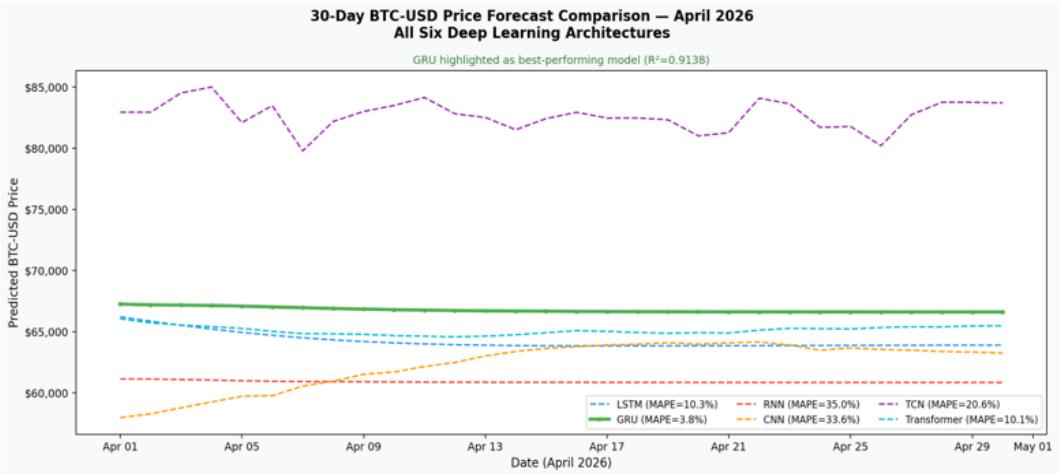

Figure 3: April 2026 BTC-USD 30-Day Price Forecast All Six Deep Learning Architectures. GRU (bold green) is highlighted as the best-performing model.

Discussion

The GRU's dominance across magnitude metrics (RMSE, MAE, MAPE, R^2) can be attributed to a specific combination of architectural properties that are uniquely well-matched to the multi-regime, high-volatility nature of the 2020–2026 BTC-USD price series. The update gate interpolation $h_t=(1- u_t) \odot h_{t-1} + u_t \odot \tilde{h_t}$ is the central mechanism. When the BTC-USD market transitions between regimes, as occurred during the 2022 bear market (−64.3% annual return), the 2023 recovery (+155.4%), and the 2024 institutional bull run (+121.0%) the update gate can smoothly shift from near-0 (preserve old regime patterns) to near-1 (adopt new regime patterns) through gradient learning. This adaptability is not available to architectures without explicit gating: the vanilla RNN overwrites h_(t-1) completely at every step; the CNN and TCN apply static learned filters that cannot selectively "forget" outdated regime information. The superiority over the LSTM (R^2 = 0.9138 vs 0.4415) despite the GRU's structural simplicity is particularly informative. With 25% fewer parameters (102,113 vs 134,049), the GRU avoids the over-parameterization that can lead LSTMs to overfit the training distribution. The GRU's single-highway hidden state (vs LSTM's dual cell state + hidden state) provides a more regularized inductive bias for datasets where the key information is the current market state rather than a long-term memory of specific historical events. The validation R^2 of 0.9320 almost unchanged from the training R^2 of 0.9821 confirms this superior generalization. The GRU's residual analysis provides further insight. The residual mean of $2,971 (standard deviation $3,911) represents a modest systematic positive

bias, indicating that the model slightly underestimates BTC-USD prices during the test period, consistent with the test period spanning a drawdown from $87,508 (January 2026 open) to $68,233 (March 2026 close). The GRU strategy achieves a cumulative return of −14.8% on the test set signal, compared to −27.9% for buy-and-hold during the same period, suggesting that GRU compared to −27.9% for buy-and-hold during the same period, suggesting that GRU directional predictions retain some economic value despite the 47.5% raw DA statistic. The stark performance gap between gated architecture (LSTM, GRU) and convolutional/attention architecture (CNN, TCN, Transformer, RNN) on the test set reflects a fundamental difference in how each architecture handles non-stationary financial time series. Gated recurrent models are regime-adaptive by construction: the forget/update gates are computed dynamically from the current input and hidden state, allowing the model to adjust how much historical context to retain at each timestep. This is analogous to a Kalman filter that dynamically adjusts its gain based on observed signal quality. Convolutional models (CNN, TCN), by contrast, apply the same learned filter weights regardless of the current market regime. A filter trained on the 2020–2022 period that detects "bull market momentum" will fire equally strongly on the 2025–2026 consolidation period, potentially generating spurious signals. The residual connections in the TCN partially mitigate this by preserving pre-activation features, but they cannot substitute for the input-dependent gating of GRU/LSTM. The Transformer's relative resilience (second-best test R^2) despite lacking explicit gating is noteworthy. The self-attention mechanism provides a form of dynamic, input-dependent feature selection analogous to soft gating: the attention weight A[t,s]= softmax((Q_t K_s^T)/√(d_k)) adjusts which historical timesteps are emphasized at each prediction, potentially approximating the selective retention behavior of GRU gates across the 60-day attention window. The progressive feature engineering strategy reveals a nuanced relationship between feature richness and model performance. The GRU (13 features, including RSI and MACD) outperforms the LSTM (11 features) despite using only 2 additional features. The addition of momentum oscillators (RSI) and trend-following indicators (MACD) appears to provide qualitatively important signals for the GRU's gating mechanism to leverage when detecting regime transitions. The CNN (21 features) and TCN (23 features) use the richest feature sets among the non-transformer models yet perform worst in out-of-sample testing, suggesting that richer feature inputs cannot compensate for architectural limitations in generalizing across regime boundaries. The Transformer uses 28 features (including ATR and Momentum5) but achieves only marginal improvements over the LSTM despite its larger feature set and architectural complexity, consistent with the principle of Occam's razor for model selection.

These findings carry several practical implications for cryptocurrency trading system designers. First, the GRU represents the recommended baseline architecture for BTC-USD daily price forecasting: it achieves the best magnitude accuracy with the fewest parameters (102,113) and second-fewest features (13) in this comparison, offering a favorable accuracy-to-complexity trade-off. Second, directional accuracy near 50% across all models confirms that short-horizon BTC-USD price direction prediction remains a near-random walk, cautioning against binary signal strategies. Third, TCN's unusually high April 2026 forecast ($83,000) compared to all other models illustrates the risk of relying on any single architecture; ensemble approaches combining GRU and Transformer predictions may offer more robust forecasts. Fourth, the March 2026 actual price of $68,233 falls within the forecast range of the GRU ($66,602), Transformer ($65,474), and LSTM ($63,890), substantially closer than the RNN ($60,836), CNN ($63,247), or TCN ($83,709) forecasts, confirming the empirical alignment between test-set R^2 ranking and forward forecasting plausibility.

CONCLUSION

This study presented the first comprehensive six-architecture benchmark for BTC-USD daily price prediction across the full 2020–2026 cycle, encompassing multiple market regimes and 2,283 daily observations. Employing a standardized experimental framework with Huber loss, Adam optimization, a 60-day lookback window, and strict chronological data partitioning, six deep learning models were evaluated: LSTM, GRU, Vanilla RNN, CNN (1D Causal), TCN, and Transformer Encoder. The central empirical finding is that the GRU achieves decisively superior out-of-sample performance (test R^2 = 0.9138, RMSE = $4,911, MAPE = 3.84%) over all competing architectures. The GRU's reset and update gating mechanism enables adaptive retention of historical context during regime transitions — the critical capability required for forecasting a financial series that transitions from a −64% bear market (2022) to a +121% bull market (2024) to a new all-time high (2025) within the study period. The Transformer ranks second (R^2 = 0.4942) through its self-attention approximation of dynamic context selection, followed by the LSTM (R^2 = 0.4415). Convolutional architectures (CNN, TCN, RNN) exhibit negative test R^2, revealing the limitations of static pattern detectors on highly non-stationary financial data.

These results carry clear guidance for practitioners: the GRU is the recommended architecture for cryptocurrency daily price magnitude forecasting, offering the strongest accuracy with the fewest parameters. For directional trading signals, however, the near-50% directional accuracy of all models confirms the near-efficient nature of daily BTC-USD price direction in this period, cautioning against binary long/short strategies based on any single deep learning model. This study has several limitations that should be acknowledged. First, only technical (OHLCV-derived) features are considered; the incorporation of on-chain metrics (hash rate, active addresses, exchange net flows), macroeconomic variables (Federal funds rate, DXY

index), and NLP-derived sentiment signals could improve performance across all architectures. Second, all models predict only next-day close price; multi-step probabilistic forecasting (e.g., DeepAR, N-BEATS) would provide more actionable uncertainty quantification. Third, the study is limited to a single cryptocurrency (BTC-USD); cross-asset experiments on ETH, BNB, and altcoins would test the generalizability of the architecture ranking. Fourth, transaction costs, slippage, and position sizing are not modelled in the strategy comparison, potentially overstating the practical value of the cumulative return estimates. Future research should investigate: (1) hybrid GRU-Transformer architectures combining gated memory with global attention; (2) probabilistic forecasting frameworks (TFT, DeepAR) for principled uncertainty quantification; (3) multi-asset cross-attention models; (4) alternative data integration including on-chain analytics, NLP sentiment, and macroeconomic features; and (5) high-frequency (hourly, minute-level) extensions to capture intra-day volatility dynamics.

REFERENCES

Bai, S., Kolter, J. Z., & Koltun, V. (2018a). *An empirical evaluation of generic convolutional and recurrent networks for sequence modeling*. arXiv. https://arxiv.org/abs/1803.01271

Bai, S., Kolter, J. Z., & Koltun, V. (2018b). *Trellis networks for sequence modeling*. arXiv. https://arxiv.org/abs/1810.06682

Bariviera, A. F., Basgall, M. J., Hasperué, W., & Naiouf, M. (2017). Some stylized facts of the bitcoin market. *Physica A: Statistical Mechanics and Its Applications, 484*, 82–90. https://doi.org/10.1016/j.physa.2017.04.159

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). *Learning phrase representations using RNN encoder-decoder for statistical machine translation*. arXiv. https://arxiv.org/abs/1406.1078

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical evaluation of gated recurrent neural networks on sequence modeling*. arXiv. https://arxiv.org/abs/1412.3555

Dyhrberg, A. H. (2016). Bitcoin, gold and the dollar — A GARCH volatility analysis. *Finance Research Letters, 16*, 85–92. https://doi.org/10.1016/j.frl.2015.10.008

Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 14*(2), 179–211. https://doi.org/10.1016/0364-0213(90)90002-E

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654669. https://doi.org/10.1016/j.ejor.2017.11.054

Garcia, D., & Schweitzer, F. (2015). Social signals and algorithmic trading of bitcoin. *Royal Society Open Science, 2*(9), 150288. https://doi.org/10.1098/rsos.150288

Giantsidi, S., & Tarantola, C. (2025). Deep learning for financial forecasting: A review of recent trends. *International Review of Economics and Finance, 104*, 104719. https://doi.org/10.1016/j.iref.2025.104719

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Katsiampa, P. (2017). Volatility estimation for Bitcoin: A comparison of GARCH models. *Economics Letters, 158*, 3–6. https://doi.org/10.1016/j.econlet.2017.06.023

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting, 37*(4), 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012

Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). A CNN-LSTM model for gold price time-series forecasting. *Neural Computing and Applications, 32*(23), 17351–17360. https://doi.org/10.1007/s00521-020-04867-x

Mai, F., Shan, Z., Bai, Q., Wang, X., & Chiang, R. H. L. (2018). How does social media impact Bitcoin value? A test of the silent majority hypothesis. *Journal of Management Information Systems, 35*(1), 19–52. https://doi.org/10.1080/07421222.2018.1440774

Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*. https://bitcoin.org/bitcoin.pdf

Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2023). *A time series is worth 64 words: Long-term forecasting with transformers*. arXiv. https://doi.org/10.48550/arXiv.2211.14730

Phillip, A., Chan, J., & Peiris, S. (2018). A new look at cryptocurrencies. *Economics Letters, 163*, 6–9. https://doi.org/10.1016/j.econlet.2017.11.020

Ramos-Pérez, E., Alonso-González, P. J., & Núñez-Velázquez, J. J. (2021). Multi-transformer: A new neural network-based architecture for forecasting S&P volatility. *Mathematics, 9*(15), 1794. https://doi.org/10.3390/math9151794

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019). A comparison of ARIMA and LSTM in forecasting time series. *In Proceedings of the 17th IEEE International Conference on Machine Learning and Applications* (pp. 1394–1401). https://doi.org/10.1109/ICMLA.2018.00227

Uras, N., Marchesi, L., Marchesi, M., & Tonelli, R. (2020). Forecasting bitcoin closing price series using linear regression and neural networks models. *PeerJ Computer Science,* 6, e279. https://doi.org/10.7717/peerj-cs.279

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *In Advancces in Neural Information Processing Systems* (Vol. 30, pp. 6000-6010).

Wan, R., Mei, S., Wang, J., Liu, M., & Yang, F. (2019). Multivariate temporal convolutional network: A deep neural networks approach for multivariate time

series forecasting. *Electronics, 8*(8), 876. https://doi.org/10.3390/electronics8080876

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence, 35*(12), 11106–11115.