Cryptocurrency Price Prediction using a Hybrid Deep Learning Approach with Explainable AI Integration

Chaolu Liu
School of Computing, National College of Ireland
Dublin, Leinster, Ireland
cl5248@hotmail.com

Furqan Rustam*
School of Computing, National College of Ireland
Dublin, Leinster, Ireland
School of Computer Science, University College Dublin
Dublin, Leinster, Ireland
furqan.rustam1@gmail.com

Anca Delia Jurcut†
School of Computer Science, University College Dublin
Dublin, Leinster, Ireland
anca.jurcut@ucd.ie

Abstract

Cryptocurrency markets exhibit pronounced nonlinearity and abrupt regime shifts, making accurate price forecasting a formidable challenge. Conventional econometric models such as ARIMA and GARCH often fail to capture these complex dynamics under high-volatility conditions. In response, we propose a hybrid deep learning framework, a ConvLSTM–GRU pipeline, that combines one-dimensional convolutions for localized pattern extraction. LSTM layers for long-term dependency modelling, and GRU layers to selectively refine salient temporal features. Technical indicators (e.g., moving averages, Bollinger Bands, and RSI) and trading volume are integrated as input channels to enrich the feature space with signals that capture volatility and momentum. We then employ Keras Tuner's Hyperband algorithm to search over convolutional filter counts, recurrent unit sizes, dense layer widths, dropout rates, and learning rates, optimizing the network for minimal validation mean squared error. On a chronological split of BTCUSDT daily closing prices from August 2017 to July 2025, the tuned ConvLSTM–GRU achieves an R^2 of 0.9922 on the held-out test set, outperforming standalone LSTM, GRU, CNN, and traditional machine learning baselines. Furthermore, we applied SHAP with ConvLSTM–GRU to improve decision-making transparency and trustworthiness.

CCS Concepts

• **Applied computing** → *Forecasting*.

Keywords

Cryptocurrency Forecasting, Deep Learning, ConvLSTM-GRU, Technical Analysis, Explainable AI

ACM Reference Format:
Chaolu Liu, Furqan Rustam, and Anca Delia Jurcut. 2025. Cryptocurrency Price Prediction using a Hybrid Deep Learning Approach with Explainable AI Integration. In *2025 The 9th International Conference on Computer Science and Artificial Intelligence (CSAI 2025), December 12–15, 2025, Beijing, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3788149.3788168

*Corresponding author.
†Corresponding author.

ACM ISBN 979-8-4007-1962-2/25/12
https://doi.org/10.1145/3788149.3788168

1 Introduction

Cryptocurrencies, decentralised digital assets underpinned by blockchain technology, have emerged as transformative instruments in global financial markets. Bitcoin (BTC), the foremost cryptocurrency by market capitalisation, exemplifies this shift, with its price dynamics influencing investment strategies, regulatory frameworks, and economic policies [18]. However, the extreme volatility of cryptocurrencies, driven by speculative trading, macroeconomic factors, and technological advancements, poses significant risks to investors and financial institutions. Accurate price prediction is thus critical for mitigating these risks, optimizing trading strategies, and enhancing market stability.

Traditional approaches, like ARIMA and GARCH, generally do not represent the erratic, non-linear behavior and the non-stationarity of cryptocurrency prices very well. Machine learning (ML) and deep learning (DL) techniques of AI have been favored. Studies on temporal dependency and volatility regime modeling have presented better performances with Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), or hybrid-based models [1, 4, 5]. These include attention-based convolutional recurrent models that have done the best in forecasting Ethereum prices [14]. As well as ensemble models like XGBoost that perform exceptionally well in short-term forecasts [10]. Although recurrent-based architectures, such as GRU, demonstrate superior performance in stable markets, their effectiveness diminishes during high-volatility periods characterised by abrupt regime shifts. Furthermore, many existing hybrid architectures lack a systematic mechanism for simultaneously capturing the local, high-frequency patterns critical during volatility spikes, and the long-term dependencies required for trend forecasting [18]. This limitation, along with underexplored hyperparameter tuning for reproducibility, motivates a robust, optimized hybrid architecture that adapts to dynamic markets and enhances model trustworthiness.

The proposed research deals with these gaps using a ConvLSTM–GRU hybrid model for BTC price prediction fine-tuning by Bayesian hyperparameter tuning. The model synthesises convolutional layers for local feature extraction, LSTM networks for sequential dependency modelling, and GRU layers to prioritise salient temporal patterns. This study systematically evaluates the comparative efficacy of hybrid architectures versus traditional machine learning methods in predicting BTC price movements under high-volatility regimes, with a specific focus on model robustness to abrupt market regime shifts and non-linear price dynamics. It makes the following contributions:

- Propose a novel tri-component sequential feature refinement pipeline (Conv1D-LSTM-GRU). This architecture is theoretically designed to systematically extract high-frequency local patterns via Conv1D, establish long-term memory and dependency representation via LSTM, and provide gated refinement of salient features via GRU. This specific, non-trivial ordering is demonstrated to be theoretically superior for volatile financial time series through an ablation study.
- Enhanced predictive performance by incorporating technical indicators, multiscale statistics, Bollinger Bands (BB), Moving Average Convergence/Divergence (MACD), Relative Strength Index (RSI), and volume features.
- The model's theoretical integrity and robustness are ensured through systematic hyperparameter optimization using Keras Tuner's Hyperband algorithm, addressing the common limitation of underexplored tuning in existing works, and using a chronological data split to avoid lookahead bias.
- Achieved superior R^2 of 0.9922 on BTCUSDT data (Aug 2017–July 2025), outperforming deep learning and ML baselines; utilized SHAP for explainability.

The report is organized as follows: Section 2 reviews related work, Section 3 outlines the methodology, Section 4 presents experiments and analysis, and Section 5 concludes with future directions.

2 Literature Review

Cryptocurrency price prediction has been extensively studied using machine learning, deep learning, and hybrid methods [18]. Table 1 summarizes key related works.

Table 1: Previous Literature Summary

Ref	Year	Method	Crypto	Hybrid	Key Contribution
[12]	2020	LSTM-GRU	Litecoin, Monero	Yes	First hybrid LSTM-GRU for financial institutions
[20]	2020	CNN, RNN variants	Ethereum	No	Comparative study of DL architectures
[4]	2021	GRU, LSTM, bi-LSTM	BTC + 2 Others	No	GRU outperformed LSTM and bi-LSTM
[19]	2021	CNN + WAMC	BTC + 3 Others	Yes	Memory channels for inter-crypto dependencies
[13]	2022	Fuzzy TS + GA	BTC + 8 Others	Yes	First FTS algorithm using entire historical data
[11]	2022	Walk-forward ensemble	BTC + 14 Others	Yes	Handles missing values and bias in datasets
[9]	2022	RNN-LSTM	BTC + 3 Others	Yes	Superior performance on multiple error metrics
[15]	2022	Deep state-space	BTC + 9 Others	Yes	Combines probabilistic SSM with DNN capabilities
[3]	2023	RNN (GRU, LSTM)	BTC	No	Technical analysis enhances DL performance
[10]	2023	CNN, DFNN, GRU, GBM	BTC + 5 Others	No	CNN most reliable with limited training data
[1]	2024	LightGBM, GRU, RNN, etc.	BTC + 3 Others	Yes	LightGBM yields highest profitability
[8]	2024	LSTM + SVM/SVR	BTC + 3 Others	Yes	Combines technical indicators with random trading
[14]	2024	CNN-BiGRU + attention	Ethereum	Yes	Attention mechanism improves interpretability
[16]	2024	GRU, LSTM, SVM	Ethereum	No	Knowledge-based strategies with profit factor 5.16
[2]	2024	LSTM + VAR	BTC + 2 Others	Yes	Error pattern incorporation improves accuracy
[5]	2024	NRBO-CNN-BiLSTM-Att	BTC	Yes	50% MAPE reduction compared to the LSTM
[6]	2025	DWT-LSTM	BTC + another	Yes	Incorporates Twitter uncertainty, commodities variables
[17]	2025	GMCN(1,N) + PSO	BTC + 2 Others	Yes	Grey model for small samples with information priority
[7]	2025	DL + technical indicators	BTC + 2 Others	Yes	Optimal integration at 4-hour intervals

Hybrid models combining recurrent and convolutional neural networks are prominent in cryptocurrency prediction research. For instance, an LSTM-GRU framework demonstrated high accuracy for Litecoin and Monero [12]. While a GRU-based model outperformed LSTM and BiLSTM for BTC, Litecoin, and Ethereum with MAPE as low as 0.2116% for Litecoin [4]. The Weighted & Attentive Memory Channels model, integrating GRU with self-attention and convolutional layers, modeled interdependencies among cryptocurrencies [19]. A hybrid walk-forward ensemble optimized predictions for 15 cryptocurrencies, outperforming classical and deep learning models [11]. Meta-heuristic algorithms optimized a CNN-BiLSTM-Attention model, reducing MAPE by over 50% for BTC [5]. Standalone LSTM models also outperformed traditional methods in RMSE and R^2 for price prediction [9], with LightGBM excelling in BTC and Ethereum tasks [1].

Technical analysis is pivotal in cryptocurrency prediction. The TechSupportLSTM model, using 24 indicators (e.g., BB, RSI) with LSTM and SVM/SVR, enabled profitable trading [8]. Integrating BB with TimesNet improved Sharpe ratios for Ethereum [7]. Grey multivariate convolution models achieved high accuracy in volatile BTC and Ethereum intervals [17]. Candlestick patterns and technical indicators in RNNs generated actionable trading signals for BTC hourly data [3].

Model optimization is critical. Genetic algorithms optimized CNN architectures, achieving a MAPE of 0.08 across six cryptocurrencies [10]. A deep state-space model provided probabilistic day-ahead price predictions, outperforming classical models [15]. A hybrid LSTM-VAR framework modeled residual structures, improving accuracy for BTC and Binance Coin [2]. Innovative approaches address specific challenges. A hybrid fuzzy time series genetic algorithm outperformed standalone models for nine cryptocurrencies [13]. An attention-based CNN-RNN model enhanced Ethereum prediction interpretability [14]. LSTM excelled in Ethereum closing price prediction [20]. Knowledge-based strategies using GRU and SVM achieved a 5.16 profit factor in Ethereum trading, though sentiment analysis had minimal impact [16].

Research Gaps: Hybrid architectures are a prominent area of research [1, 2, 5–9, 11–15, 17, 19], but a significant gap exists in benchmarking them against traditional ML models (e.g. Random Forest, XGBoost, LightGBM) within a unified framework. Furthermore, reproducibility is limited due to the underexplored nature of systematic hyperparameter tuning in studies [9, 13, 14, 17]. Some works focus solely on technical indicators or lagged variables [3, 8], with limited integration of advanced preprocessing techniques, potentially compromising model robustness. A common methodological flaw in many works is the failure to explain how train-test splits were created in time-series data, which could violate temporal dependencies and lead to unreliable results, as in [8, 9, 14, 17]. Additionally, practical implementations of walk-forward validation fail to address leakage risks during scaling, such as look-ahead bias.

3 Proposed Methodology

Our proposed system introduces an approach to cryptocurrency price prediction, as illustrated by the workflow in Figure 1. We gather daily BTCUSDT trading data[1] spanning from August 17,

[1] https://www.binance.com/trade/BTC_USDT

2017, to July 26, 2025, which includes open, high, low, and close prices, trading volumes in BTC and USDT, and daily trade counts (Table 2). We construct a rich feature set comprising lagged closing prices, multi-scale rolling statistics, adaptive BB measures, and widely used technical indicators such as MACD, RSI, and BB width. Each feature is computed in a strictly chronological manner to prevent lookahead bias, and we apply forward-fill imputation for any missing values at the initial time steps. The data were then divided into training, validation, and test sets by time. During training and validation, normalization is performed using the *min-max scaling* technique on only the training set; the same scaling parameters are applied to validation and test sets.

Figure 1: Proposed Methodology Overview

Table 2: Historical Daily Trading Data for BTCUSDT

Date	Open	High	Low	Close	Volume BTC	Volume USDT	Trade count
26/07/2025	117 614	118 297	117 138	117 920	6 991.672060	8.236 291e+08	790 372
25/07/2025	118 341	118 452	114 723	117 614	38 406.348730	4.456 779e+09	3 282 355
...							
18/08/2017	4 285	4 372	3 939	4 108	1 199.888264	5.086 958e+06	5 233
17/08/2017	4 261	4 485	4 201	4 285	795.150377	3.454 770e+06	3 427

We developed a robust forecasting backbone built upon a hybrid ConvLSTM-GRU neural network (Figure 3). This architecture is designed to identify both short-term patterns and long-term dependencies within the data. Through systematic hyperparameter tuning with Keras Tuner, we identified the optimal model configuration and evaluated it on an independent test set.

Data Augmentation and Preprocessing: We use daily BTCUSDT closing prices from 2017-08-17 to 2025-07-26 ($T = 2900$ days), with a strict chronological split to prevent lookahead bias. The training set spans 2017-08-17 to 2023-12-23, the validation set covers 2023-12-24 to 2024-10-08, and the test set runs from 2024-10-09 to 2025-07-26.

To augment the dataset, we define lagged features. The lag operator $\mathcal{L}^k(C_t) = C_{t-k}$ represents the closing price k days ago. We include lags $k \in \{1, 2, 3, 7\}$, yielding the four-dimensional feature vector $\mathbf{X}_{\text{lag}} = \left[\mathcal{L}^1(C_t),\ \mathcal{L}^2(C_t),\ \mathcal{L}^3(C_t),\ \mathcal{L}^7(C_t)\right]_{t=1}^{T}$ at each time step t. This allows the model to leverage temporal autocorrelation at both short and slightly longer horizons. We also compute multi-scale rolling statistics for window sizes $w \in \{7, 14, 30\}$. These include the rolling mean $\mu_w(t) = \frac{1}{w}\sum_{k=0}^{w-1}\mathcal{L}^k(C_t)$, which represents the average closing price over the past w days and encapsulates the local price level, and the rolling standard deviation $\sigma_w(t) = \sqrt{\frac{1}{w}\sum_{k=0}^{w-1}\left(\mathcal{L}^k(C_t) - \mu_w(t)\right)^2}$, which measures the variability (volatility) within that same window. In practice, we interpret these as follows (Table 3):

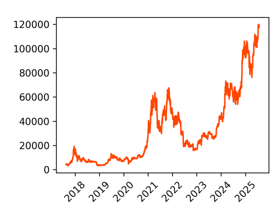
Adaptive BB are computed to capture dynamic upper and lower volatility envelopes. The upper band is defined as $BB_{\text{high}}(t) = \mu_{30}(t) + 2\,\sigma_{30}(t)$ and the lower band as $BB_{\text{low}}(t) = \mu_{30}(t) - 2\,\sigma_{30}(t)$. Here, $\mu_{30}(t)$ and $\sigma_{30}(t)$ are the 30-day moving average and standard deviation, respectively. The width ratio $BB_{\text{width}}(t) = \frac{BB_{\text{high}}(t) - BB_{\text{low}}(t)}{\mu_{30}(t)} = \frac{4\,\sigma_{30}(t)}{\mu_{30}(t)}$ offers a dimensionless measure of relative volatility. Additional technical indicators are also integrated. The MACD is calculated as $\text{MACD}(t) = \text{EMA}_{12}(C_t) - \text{EMA}_{26}(C_t)$, where EMA_{12} and EMA_{26} are 12-day and 26-day exponential moving averages, respectively. This captures momentum by comparing short-term and long-term trends. The 14-day RSI is computed as $\text{RSI}(t) = 100 - \frac{100}{1+\text{RS}}$, where RS is the ratio of average gains to average losses over the past 14 days. The BB Width is also included, defined as $\text{BBWidth}(t) = \frac{2\,\sigma_{20}(C_t)}{\mu_{20}(C_t)} \times 100\%$, where $\sigma_{20}(C_t)$ and $\mu_{20}(C_t)$ are the 20-day rolling standard deviation and mean, respectively. This indicator measures relative volatility [7, 8].

Table 3: Interpretation of Rolling Window Features

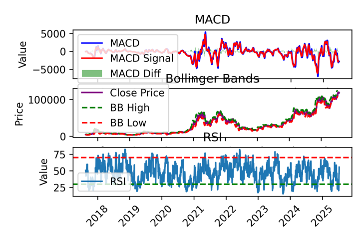
Days	μ_w Usage	σ_w Usage
7	Short-term trend	High-frequency volatility
14	Trend alignment	Medium-term stability
30	Market regime detection	Crash risk indicator

To ensure no information from the future enters the training pipeline, we follow specific protocols regarding data leakage prevention. First, we use chronological splitting to strictly partition data, so that the validation and test sets contain only data points that occur after the end of the training period. Second, we perform separate normalization where each feature X is min–max scaled using statistics computed solely on the training set: $\tilde{X}_t^{(\text{test})} = \frac{X_t^{(\text{test})} - \min\{X^{(\text{train})}\}}{\max\{X^{(\text{train})}\} - \min\{X^{(\text{train})}\}}$. This prevents information leakage by ensuring that scaling parameters are derived exclusively from the training set. Finally, we use forward-fill imputation for any NaN values arising from initial rolling-window computations ($t < w$), computing the average of observed prices up to time t with $C_t = \frac{1}{t}\sum_{k=1}^{t} C_k$ for $t < w$. This ensures that the first few days, which lack a full w-day history, are filled with an unbiased average of available data.

Figure 2a and 2b visualise the BTC-USD price trajectory (2017–2025) and the overlaid technical indicators (MACD, BB, RSI), respectively. These plots illustrate prolonged bull and bear cycles, punctuated by volatility spikes that coincide with major macroeconomic or regulatory events, underscoring the need for a hybrid model that can adapt to both regime shifts and local fluctuations.



(a) BTC-USD price trajectory



(b) MACD, Bollinger Bands, and RSI

Figure 2: BTC price trends and indicators (2017–2025)

Hybrid ConvLSTM–GRU Architecture: Our hybrid ConvLSTM–GRU model, shown in Figure 3, is designed to capture both localized patterns and long-term dependencies in cryptocurrency prices. A Conv1D layer with 32 filters (kernel size 3) first extracts local temporal features. The filter count was chosen to balance expressivity and prevent overfitting. This is followed by a ReLU activation.

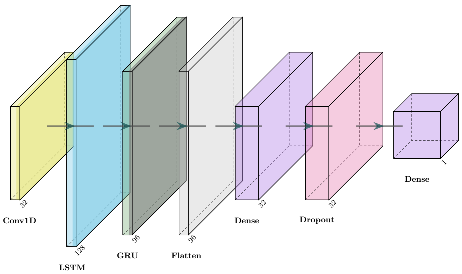


Figure 3: Hybrid ConvLSTM–GRU Model Architecture

Next, an LSTM layer with 128 hidden units captures long-term trends from the convolutional block's output. Hyperparameter tuning determined 128 units as optimal, avoiding underfitting or overfitting. A GRU layer with 96 hidden units then refines the LSTM's output into a final hidden state. The GRU offers a computationally efficient way to capture salient patterns. The final hidden state is flattened and passed to a Dense layer of width 32 with ReLU activation, which narrows the feature representation.

Surprisingly, a Dropout layer with a rate of 0.0 was found to be optimal, as the preceding layers provide sufficient implicit regularization. Finally, an Output Dense layer with 1 unit produces the next-day price prediction. The model is trained using Mean Squared Error (MSE) to handle large price moves. We also experimented with an attention module, but it led to severe overfitting and was omitted from the final model.

Proposed Approach Optimization: We employ Keras Tuner's Hyperband algorithm to search the following hyperparameter domains as shown in Table 4:

Table 4: Hyperparameter Search Space

	Conv filters	LSTM units	GRU units	Dense units	Dropout rate	Learning rate
Domain	{32,64,96,128}	{32,64,96,128}	{32,64,96,128}	{32,64,96,128}	[0.0,0.5]	$[10^{-4}, 10^{-2}]$
Type	Discrete	Discrete	Discrete	Discrete	Continuous	Log-uniform

For Conv filters, we start at 32 filters since prior CNN-based crypto studies [10] found 32–64 to be optimal. We allow up to 128 to explore richer local pattern extraction. For recurrent units, we search across {32, 64, 96, 128} for both LSTM and GRU. This range covers lightweight (32) to moderately large (128) hidden dimensions, acknowledging that too few units fail to capture volatility dynamics, whereas too many induce overfitting on limited data. Post-recurrent dense width is searched within the same set {32, 64, 96, 128}, balancing representational capacity with risk of overfitting. We explore a Dropout rate of [0.0, 0.5] to allow Hyperband to turn on regularization if needed. A log-uniform search for the Learning rate between 10^{-4} and 10^{-2} is standard for Adam optimizers. Lower rates ($< 10^{-4}$) led to slow convergence, while higher rates ($> 10^{-2}$) caused divergence.

Our optimization strategy involves initializing Hyperband with maximum epochs $T = 100$, reduction factor $\eta = 3$, and a maximum of 25 total trials. For each bracket $b = 1, \ldots, \lfloor \log_\eta(T) \rfloor$, Hyperband samples $n = \lfloor T/\eta^{b-1} \rfloor$ configurations. Each configuration is trained for $r = \eta^{b-1}$ epochs initially. After the first round of training, configurations are ranked by validation MSE, and only the top $\lfloor n/\eta^s \rfloor$ continue to the next round, each now trained for $r \cdot \eta$ more epochs. This successive halving continues until the bracket is completed. The best performer across all brackets is selected. Early stopping with a patience of 10 epochs halts training of any configuration whose validation loss does not improve, preventing wasteful computation. To ensure reproducibility, we fixed all random seeds to 42 (NumPy, TensorFlow, Python's `random`). Training used a batch size of 32, the Adam optimizer with a cosine decay schedule, and He normal initialization for all convolutional and recurrent kernels. Each configuration ran for up to 100 epochs, with early stopping (patience = 10). After Hyperband identified the best hyperparameter set, we retrained the ConvLSTM–GRU on the combined training and validation sets for up to 100 epochs, again with early stopping, before evaluating on the test set. After running Hyperband for 25 trials, which took approximately 18 minutes on Google Colab GPU, the best ConvLSTM–GRU configuration achieved a minimum validation loss of $\text{val\_loss}_{\text{min}} = 1.1284 \times 10^{-5}$. Table 5 presents the optimal hyperparameters found from this process.

Table 5: Optimal ConvLSTM–GRU Hyperparameters

Conv filters	LSTM units	GRU units	Dense units	Dropout rate	Learning rate
32	128	96	32	0.0	2.15×10^{-3}

4 Experiments

We compare nine models across traditional machine learning and deep learning categories: XGBoost, LightGBM, Random Forest (Traditional ML); RNN, LSTM, GRU, CNN (Deep Learning); and CNN–LSTM, ConvLSTM–GRU (Hybrid Architectures). For evaluation on the test set, we use Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-squared (R^2).

Table 6 presents the final model performance on the test set, ranked by R^2. The **ConvLSTM–GRU** model achieved the best performance across all metrics, with an RMSE of 1061.29, MAE of 792.70, MAPE of 0.0082, and an R^2 of 0.9922. Following closely were other deep learning models like LSTM, GRU, CNN–LSTM, CNN, and RNN. Traditional machine learning models such as Random Forest, XGBoost, and LightGBM performed significantly worse, indicating their limited ability to capture the complex temporal dependencies in cryptocurrency price data.

Table 6: Test Set Performance (Post-Optimization)

Metric	ConvLSTM–GRU	LSTM	GRU	CNN–LSTM	CNN	RNN	Random Forest	XGBoost	LightGBM
RMSE	1061.29	7163.83	7338.86	7619.20	7993.95	8829.45	31935.43	33567.76	34168.52
MAE	792.70	6030.22	6566.68	6841.24	6877.93	7591.72	29214.83	31128.23	31689.93
MAPE	0.0082	0.0593	0.0656	0.0685	0.0677	0.0752	0.2938	0.3150	0.3207
R^2	0.9922	0.6441	0.6264	0.5974	0.5568	0.4593	-4.9400	-5.5628	-5.7998

Figure 4 visually illustrates the predictive performance. The ConvLSTM–GRU architecture demonstrates the highest fidelity to the actual price trajectory, accurately reflecting both uptrends and downtrends. Conversely, alternative models exhibit more pronounced deviations, especially during periods of elevated volatility, confirming the hybrid's robustness in volatile regimes.

To isolate the contribution of each major component in our ConvLSTM–GRU architecture, we conducted an ablation study on the test set by systematically removing or altering one module at a time. Table 7 summarises the performance impact of each variant. Removing the initial convolutional layer (LSTM–GRU variant) increased

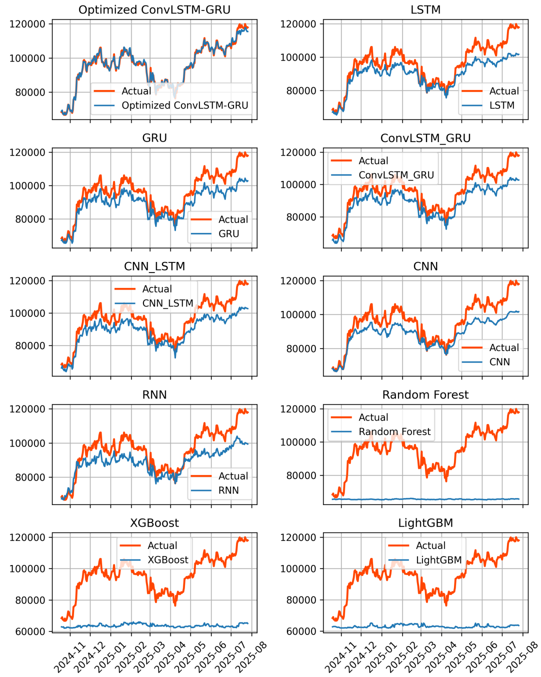


Figure 4: ConvLSTM–GRU predictions closely follow actual BTC prices, demonstrating strong model fit

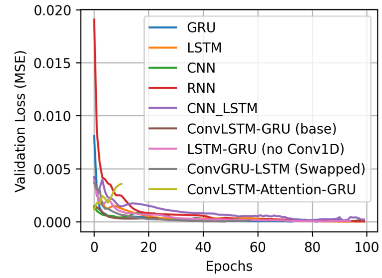


Figure 5: MSE across epochs showing ConvLSTM–GRU achieves fastest and stable convergence among all models

RMSE by over 600% and lowered R^2 by 38.24 percentage points, confirming that local pattern extraction is critical for capturing short-term price dynamics. Swapping the GRU and LSTM layers (ConvGRU–LSTM) further degraded performance, suggesting that placing the convolution–LSTM block before the GRU yields better sequential feature refinement.

Table 7: Model Variants Performance (Ablation Study)

Variant	RMSE	MAE	MAPE	R^2
ConvLSTM–GRU	1061.29	792.70	0.0082	0.9922
LSTM–GRU (no Conv1D)	7500.86	6576.85	0.0655	0.6098
ConvGRU–LSTM (Swapped)	8124.35	7120.25	0.0708	0.5422
ConvLSTM–Attention–GRU	10199.23	9178.74	0.0919	0.2785

Naively adding an unregularized attention layer caused severe overfitting; test set R^2 dropped from 0.9922 to 0.2785, highlighting the need for careful integration and regularization when employing attention in volatile time-series forecasting. These results validate the design choices of our hybrid architecture: the Conv1D front end and the specific LSTM→GRU ordering are both essential, and attention must be incorporated with tailored regularization to be beneficial. Tree-based model tuning with grid search did not improve R^2, suggesting prior overfitting. In contrast, ConvLSTM–GRU significantly benefited from Hyperband scheduling, converging faster and reaching a lower validation loss than other deep models. Figure 5 shows validation loss (MSE) versus epochs for the top deep-learning contenders. These results show that our Hyperband-tuned ConvLSTM–GRU achieves the best test performance and learns more steadily with lower variance across epochs.

The ConvLSTM–GRU achieved the lowest RMSE, MAE, and MAPE, and the highest R^2 on the test set. While tree-based models performed well on training data, they failed to generalize. Notably, hybrid architectures like CNN–LSTM and ConvLSTM–GRU outperformed standalone RNNs or CNNs. Furthermore, the inclusion of technical indicators and a volatility-aware loss function significantly contributed to the model's robustness.

4.1 Discussion

This section highlights that the ConvLSTM–GRU model outperforms both traditional ML and deep learning baselines, with notably lower RMSE and higher R^2, demonstrating its strength in capturing complex temporal patterns. While traditional models such as XGBoost and Random Forest performed well during training, their poor generalization on the test set highlights their sensitivity to overfitting and inability to capture long-range dependencies. Deep learning models like LSTM and GRU improved robustness but still underperformed compared to hybrid models. CNN–LSTM showed intermediate performance, validating the advantage of combining spatial (temporal-local) and sequential learning. ConvLSTM–GRU further extended this by integrating convolutional filters with gated recurrence and (optionally) attention mechanisms.

Technical indicators (MACD, RSI, BB) and rolling statistics provided critical signals for price movement. Feature attribution analysis using SHAP (Figure 6) indicates that lagged closing prices (Close_Lag_3, Close_Lag_7, Close_Lag_2) and moving averages (Close_MA_30, Close_MA_14) contributed most significantly to prediction accuracy. While Bollinger Bands (BB_Low, BB_High) also showed considerable importance, BB_Width had a notably lower contribution. MACD-related features (MACD, MACD_Signal, MACD_Diff) showed moderate importance, and volume (Volume USDT, Volume BTC) contributed least, reinforcing the economic interpretability of the engineered features. Without these engineered features, models' R^2 dropped by 8–12 percentage points.

The integration of SHAP explainability not only enhances interpretability for fund managers by quantifying feature influences (e.g., Volume, technical indicators), but also provides regulators with tools for market surveillance. By highlighting shifts in predictive feature importance, the model can flag unusual trading activity, support due diligence, and contribute to systemic risk assessment, thereby improving trust, transparency, and stability in cryptocurrency markets.

Limitations: Despite its strong performance, our model still struggles during extreme market volatility events, such as flash crashes or exchange outages, where price discontinuities exceed

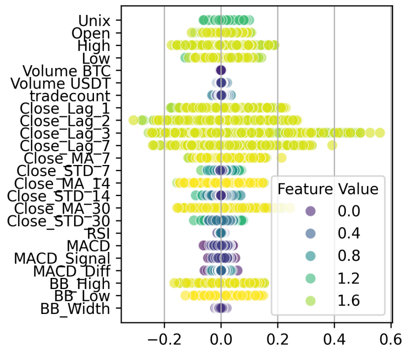


Figure 6: SHAP plot for the ConvLSTM-GRU Model

historical norms. The architecture also requires significant computation for tuning and training (18 minutes for Hyperband, plus ∼2 hours for final training), which may limit applicability for real-time deployment. Furthermore, while our engineered features improve performance, they remain handcrafted; future work could incorporate unsupervised or self-supervised feature learning to capture latent macroeconomic or sentiment signals not encoded by standard technical indicators.

5 Conclusion

To predict the BTC price, our methodology included dynamic lag features, multi-scale rolling statistics, and adaptive indicators like MACD, RSI, and BB. We prevented data leakage via proper normalization, forward-fill imputation, and strict chronological splitting. The inclusion of temporal attention was tested but ultimately omitted due to overfitting, highlighting directions for future refinements. Hyperparameter tuning with Keras Tuner's Hyperband proved critical, with the best ConvLSTM–GRU achieving a test-set R^2 of 0.9922. We concluded that the hybrid architecture, which includes convolutional layers to extract important features and recurrent architectures like LSTM and GRU that are effective at capturing long-term dependencies, helps to identify patterns. Furthermore, SHAP helps to identify which features play an important role in decision-making, thereby enhancing trust in the process.

Future Directions: Future work will incorporate macroeconomic and sentiment signals, explore Transformer ensembles, and evaluate real-time inference. Deployment requires latency optimization (e.g., ONNX/TensorFlow Lite), streaming pipelines for tick-level data, and continuous retraining to address concept drift. We also plan to test on multiple cryptocurrencies, broader benchmarks, high-frequency data, and derivatives to assess robustness.

Acknowledgments

This work was supported by the School of Computer Science, University College Dublin, and Beijing-Dublin International College.

References

[1] Ahmed Bouteska, Mohammad Zoynul Abedin, Petr Hajek, and Kunpeng Yuan. 2024. Cryptocurrency price forecasting – A comparative analysis of ensemble learning and deep learning methods. *International Review of Financial Analysis* 92 (March 2024), 103055. doi:10.1016/j.irfa.2023.103055

[2] Mir Feiz Fallah, Rezvan Pourmansouri, and Bahador Ahmadpour. 2024. Presenting a new deep learning-based method with the incorporation of error effects to predict certain cryptocurrencies. *International Review of Financial Analysis* 95 (Oct. 2024), 103466. doi:10.1016/j.irfa.2024.103466

[3] Stéphane Goutte, Hoang-Viet Le, Fei Liu, and Hans-Jörg Von Mettenheim. 2023. Deep learning and technical analysis in cryptocurrency market. *Finance Research Letters* 54 (June 2023), 103809. doi:10.1016/j.frl.2023.103809

[4] Mohammad J. Hamayel and Amani Yousef Owda. 2021. A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms. *AI* 2, 4 (Oct. 2021), 477–496. doi:10.3390/ai2040030

[5] Xiangyi He, Yiwei Li, and Houjian Li. 2024. Revolutionizing Bitcoin price forecasts: A comparative study of advanced hybrid deep learning architectures. *Finance Research Letters* 69 (Nov. 2024), 106136. doi:10.1016/j.frl.2024.106136

[6] Ismail Jirou, Ikram Jebabli, and Amine Lahiani. 2025. A hybrid deep learning model for cryptocurrency returns forecasting: Comparison of the performance of financial markets and impact of external variables. *Research in International Business and Finance* 73 (Jan. 2025), 102575. doi:10.1016/j.ribaf.2024.102575

[7] Mingu Kang, Joongi Hong, and Suntae Kim. 2025. Harnessing technical indicators with deep learning based price forecasting for cryptocurrency trading. *Physica A: Statistical Mechanics and its Applications* 660 (Feb. 2025), 130359. doi:10.1016/j.physa.2025.130359

[8] Ying-Ho Liu and Jian-Kai Huang. 2024. Cryptocurrency trend forecast using technical analysis and trading with randomness-preserving. *Computers and Electrical Engineering* 118 (Aug. 2024), 109368. doi:10.1016/j.compeleceng.2024.109368

[9] I. Nasirtafreshi. 2022. Forecasting cryptocurrency prices using Recurrent Neural Network and Long Short-term Memory. *Data & Knowledge Engineering* 139 (May 2022), 102009. doi:10.1016/j.datak.2022.102009

[10] Azeez A. Oyedele, Anuoluwapo O. Ajayi, Lukumon O. Oyedele, Sururah A. Bello, and Kudirat O. Jimoh. 2023. Performance evaluation of deep learning and boosted trees for cryptocurrency closing price prediction. *Expert Systems with Applications* 213 (March 2023), 119233. doi:10.1016/j.eswa.2022.119233

[11] David Opeoluwa Oyewola, Emmanuel Gbenga Dada, and Juliana Ngozi Ndunagu. 2022. A novel hybrid walk-forward ensemble optimization for time series cryptocurrency prediction. *Heliyon* 8, 11 (Nov. 2022), e11862. doi:10.1016/j.heliyon.2022.e11862

[12] Mohil Maheshkumar Patel, Sudeep Tanwar, Rajesh Gupta, and Neeraj Kumar. 2020. A Deep Learning-based Cryptocurrency Price Prediction Scheme for Financial Institutions. *Journal of Information Security and Applications* 55 (Dec. 2020), 102583. doi:10.1016/j.jisa.2020.102583

[13] Shio Gai Quek, Ganeshsree Selvachandran, Jun Hao Tan, Hao Yang Adam Thiang, Nguyen Trung Tuan, and Le Hoang Son. 2022. A New Hybrid Model of Fuzzy Time Series and Genetic Algorithm Based Machine Learning Algorithm: A Case Study of Forecasting Prices of Nine Types of Major Cryptocurrencies. *Big Data Research* 28 (May 2022), 100315. doi:10.1016/j.bdr.2022.100315

[14] Dawei Shang, Ziyu Guo, and Hui Wang. 2024. Enhancing digital cryptocurrency trading price prediction with an attention-based convolutional and recurrent neural network approach: The case of Ethereum. *Finance Research Letters* 67 (Sept. 2024), 105846. doi:10.1016/j.frl.2024.105846

[15] Shalini Sharma and Angshul Majumdar. 2022. Deep state space model for predicting cryptocurrency price. *Information Sciences* 618 (Dec. 2022), 417–433. doi:10.1016/j.ins.2022.10.129

[16] Adrián Viéitez, Matilde Santos, and Rodrigo Naranjo. 2024. Machine learning Ethereum cryptocurrency prediction and knowledge-based investment strategies. *Knowledge-Based Systems* 299 (Sept. 2024), 112088. doi:10.1016/j.knosys.2024.112088

[17] Yunfei Yang, Xiaomei Wang, Jiamei Xiong, Lifeng Wu, and Yifang Zhang. 2025. An innovative method for short-term forecasting of blockchain cryptocurrency price. *Applied Mathematical Modelling* 138 (Feb. 2025), 115795. doi:10.1016/j.apm.2024.115795

[18] Junhuan Zhang, Kewei Cai, and Jiaqi Wen. 2024. A survey of deep learning applications in cryptocurrency. *iScience* 27, 1 (2024), 108509. doi:10.1016/j.isci.2023.108509

[19] Zhuorui Zhang, Hong-Ning Dai, Junhao Zhou, Subrota Kumar Mondal, Miguel Martínez García, and Hao Wang. 2021. Forecasting cryptocurrency price using convolutional neural networks with weighted and attentive memory channels. *Expert Systems with Applications* 183 (Nov. 2021), 115378. doi:10.1016/j.eswa.2021.115378

[20] Thanasis Zoumpekas, Elias Houstis, and Manolis Vavalis. 2020. ETH analysis and predictions utilizing deep learning. *Expert Systems with Applications* 162 (Dec. 2020), 113866. doi:10.1016/j.eswa.2020.113866