EUROPEAN JOURNAL OF ACCOUNTING, FINANCE & BUSINESS
Volume **13** / **2025** ISSN 2344-102X
Issue **2** ISSN-L 2344-102X

CRYPTOCURRENCY MARKET FORECASTING BASED ON GARCH-LSTM NEURAL NETWORKS: A CASE STUDY OF BITCOIN AND ETHEREUM

Habib ZOUAOUI
Faculty of economics, Department of Finance and accounting, University of Relizane, Algeria
habib.zouaoui@univ-relizane.dz
ORCID:0000-0001-7694-2473
Meryem-Nadjat NAAS
Faculty of economic sciences, Department of Management; University of Relizane, Algeria
meryemnadjat.naas@univ-relizane.dz
ORCID:0009-0004-4018-1261

Abstract
This study investigates the effectiveness of a hybrid forecasting model that combines Generalized Autoregressive Conditional Heteroskedasticity (GARCH) with Long Short-Term Memory (LSTM) neural networks, specifically applied to the cryptocurrency market, focusing on Bitcoin and Ethereum. The inherent volatility of cryptocurrencies presents substantial challenges for accurate price prediction, necessitating advanced methodologies that can adapt to fluctuating market conditions.
We first utilize GARCH models to analyze and capture the time-varying volatility in the returns of Bitcoin and Ethereum, enabling a comprehensive understanding of the underlying market dynamics. Following this, we implement LSTM networks to exploit their capability to model complex, non-linear relationships in sequential data, enhancing the predictive power of the model. The performance of the GARCH-LSTM framework is rigorously evaluated using historical price data for Bitcoin and Ethereum, employing key metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to assess forecasting accuracy. The results demonstrate that the hybrid approach significantly outperforms traditional forecasting methods, providing more reliable predictions and insights into market trends. This study contributes to the growing body of literature on cryptocurrency forecasting by illustrating the potential of combining econometric techniques with advanced machine learning methods, offering valuable implications for traders and investors in the cryptocurrency ecosystem. However, the experimental results revealed that the LSTM model outperformed the other eight methods in terms of forecasting performance measures, the RMSPE validation is 0.112561, and the RMSE validation is 0.011456.

Keywords: *Cryptocurrency; Forecasting; Deep learning; Blockchain; GARCH; LSTM.*

JEL Classification:*C4, C45, C55, C65, G11*

INTRODUCTION

Forecasting cryptocurrency volatility is essential for market participants looking to mitigate risks and capitalize on opportunities. Given the inherent unpredictability of these assets, a combination of traditional statistical models, machine learning, and real-time market data can help improve the accuracy of volatility forecasts. By understanding the drivers of volatility and leveraging appropriate forecasting methods, participants can make more informed trading and investment decisions in the rapidly evolving cryptocurrency landscape.
In this virtual world, Cryptocurrencies can no longer be ignored. “Cryptocurrencies are digital, straightforward and simple to use in comparison with the traditional currencies”, The individual investor, the big cooperation and even governments (who seek ways to control their influence in their financial markets) are interested in them. Moreover, to provide a better understanding of the topic under investigation, we present a review of recent studies related to our research area. There is a comprehensive strand of literature on forecasting Bitcoin return volatility using econometric and machine learning models (Cevik et al., 2023; Wirawan et al., 2019; Shen et al., 2021). However, this paper distinguishes itself by collecting a unique set of features based on previous research evidencing their correlation to Bitcoin (Estrada, 2017; Gaies et al., 2023; Wang et al., 2022; Garcia et al., 2014; Koutmos, 2018). Specifically, this paper analyses the GARCH and LSTM models’ effectiveness in predicting Bitcoin return volatility, using RMSE as the performance metric. The modelling error of each model is assessed on the validation set, and the final GARCH model is selected based on the result.

Furthermore, originally developed in computer science, DL has since permeated diverse domains, including medicine, neuroscience, physics, astronomy, and operations management (Chai & Ngai, 2020).

DOI: 10.4316/EJAFB.2025.13214

Volume **13** / **2025** ISSN 2344-102X
Issue **2** ISSN-L 2344-102X

However, despite its growing adoption, there remains a lack of comprehensive literature reviewing DL applications in the finance and banking (F&B) sector (Huang et al., 2020). This gap highlights the need for further research to systematically examine DL's potential in financial modelling, risk assessment, fraud detection, algorithmic trading, and other F&B applications

Briefly, this study investigates whether machine learning (ML) models outperform traditional econometric models in predictive accuracy, as measured by lower forecasting errors and higher precision. Specifically, we conduct a comparative analysis between conventional econometric approaches - such as the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model - and deep learning (DL) techniques, including Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks.

Therefore, the research question is: Which approach - conventional regression-based modelling or deep learning model - performs better in forecasting the future volatility of cryptocurrencies? Broadly, this study seeks to test the following hypotheses:

H1: The DL models provide the forecasting future values of cryptocurrencies volatility with higher accuracy than GARCH models.

H2: The GARCH models provides the forecasting future values of cryptocurrencies volatility with higher accuracy than DL models.

I. LITERATURE REVIEW

In order to better understand, we should present some recent studies on our research, we count the number of articles that use various DL models in cryptocurrencies volatility, as shown in Table 1:

Table 1. Reviewed Previous Studies

Authors/Year	Methodology	Key Findings
Patel et al. (2022)	LSTM, GRU vs. GARCH	LSTM outperformed GARCH in capturing long-term volatility patterns.
Zhang et al. (2021)	Transformer-based model vs. GARCH	Transformers achieved higher accuracy than GARCH in predicting intra day volatility spikes.
Wang et al.(2023)	Attention-based RNN vs. GARCH	Attention mechanisms improved accuracy over GARCH, especially in high-volatility periods.
Boongasame & Songram (2023)	LSTM, SMA, WMA and EMA	LSTM showed a MAPE of 0.0927%.
Fleischer et al. (2022)	LSTM, ARIMA	LSTM showed superior performance than ARIMA using RMSE
Patel et al. (2020)	LSTM with GRU	LSTM with GRU forecasts the prices with high accuracy compared to exiting models.
Tripathy, N (2023)	ARIMA, LSTM, FB-prophet	FB-prophet as the most efficient model.
Kanaparthi (2024)	LSTM	LSTM demonstrate robustness in predicting Bitcoin prices, outperforming traditional ARIMA models
Khaniki et al. (2024)	Indicators-Performer-BiLSTM	The hybrid model boosts accuracy and efficiency, outperforming others in cryptocurrency prediction.
Li & Dai (2020)	CNN-LSTM hybrid	Outperformed in prediction accuracy, showcasing the strength of hybrid models
Rachid et al. (2024)	XGBoost, ANN, LSTM, SVR.	The use of advanced deep learning techniques effectively manages the complexities of the cryptocurrency market, offering significant improvements over traditional methods and guiding investors in the cryptocurrency markets
AlMadany et al. (2024)	ARMA -GARCH-EGARCH- LSTM	The hybrid EGARCH-LSTM or GARCH-LSTM models demonstrate slightly better accuracy compared with the other models
Jirou et al. (2025)	Discrete Wavelet Transform (DWT) and (LSTM)	The findings show that the proposed DWT-LSTM model outperforms a large set of benchmark models in terms of forecasting accuracy

DOI: 10.4316/EJAFB.2025.13214

Volume **13** / **2025** ISSN 2344-102X
Issue **2** ISSN-L 2344-102X

Boozary et al. (2025)	LSTM, data-driven marketing strategies	The study's importance is derived from its systematic examination of various machine learning (ML) techniques employed to predict the price of Bitcoin, with a particular emphasis on their integration into data-driven marketing strategies

Source: Authors' analysis from literature review

Table above is a summarized comparison of results from recent studies (2022–2024) on forecasting cryptocurrency volatility using deep learning models versus GARCH (Generalized Autoregressive Conditional Heteroskedasticity). These results highlight the key findings and performance metrics from the studies.
Therefore, the deep learning models ou performed GARCH in almost all aspects, including accuracy, adaptability, and handling of non-linear and high-frequency data. They are particularly effective for extreme event prediction and scenario generation. However, the GARCH models remained computationally efficient and interpretable but wireless accurate in capturing the complex, dynamic nature of cryptocurrency volatility (Bouteska et al., 2024).

II. MATERIALS AND METHODS

- **Generalized Autoregressive Conditional Heteroscedasticity (GARCH) Model**

The GARCH model is a class of models used to model time series data where the variance (volatility) changes over time and exhibits clustering. It was introduced by Robert Englein 1982 and later generalized by Tim Bollerslev in 1986. However, The GARCH model of Bollerslev (Bollerslev, 1986) is the most popular univariate volatility model. itis formulated on returns. The GARCH (p, q) model can be presented as (Lahmiri & Bekiros, 2019):

$$\varepsilon_t/\psi_{t-1} \backsim N(0, h_t)$$
$$h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i . \varepsilon_{t=i}^2 + \sum_{j=1}^{p} \beta_j \, h_{t-j} \tag{1}$$

where: ε_t is the innovation process from the conditional mean equation of returns, ψ_{t-1} is the set of all information available at time t-1, N is the conditional normal distribution and h_t is the conditional variance.

The standard restrictions for non-negativity of the conditional variance are $\alpha_0 > 0$, $\alpha_i \geq 0$, $\beta_j \geq 0$ (for i = 1,2,…,q; j = 1,2,…,p), however, weaker conditions can also be assumed (Nelson & Cao,1992). For covariance stationarity, the following condition has to be satisfied $\alpha_1 + \ldots + \alpha_q + \beta_1 + \ldots + \beta_p < 1$. We use the maximum likelihood method for estimation of parameters. We apply the GARCH (1, 1) model with lags one whichis the most frequently used in empirical studies.

Firstly, the GJR-GARCH model is an extension of the GARCH model developed by Glosten and Runkle (1993) that accounts for asymmetric effects of positive and negative shocks on volatility. In financial markets, negative news (bad news) often causes higher volatility than positive news of the same magnitude. The GJR-GARCH model adds a term to capture this asymmetry. Secondly, the Threshold GARCH model, developed by Zakoian (1994), is applied using a bootstrap technique. The bootstrap process involves resampling to estimate the distribution of statistics.

Finally, The Threshold GARCH (TGARCH) model, also known as the GJR-GARCH model, is an extension of the standard GARCH model that incorporates asymmetric effects in volatility. Specifically, it allows for different responses to positive and negative shocks (e.g., good news vs. bad news). The mathematical expressions for the GARCH - type models are shown in Mathematical expressions (2) for GARCH and GJR-GARCH and TGARCH model (García-Medina & Aguayo-Moreno, 2024).

GARCH:
$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i . \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j . \sigma_{t-j}^2$$

GJR-GARCH & TGARCH:
$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i . \varepsilon_{t-i}^2 + \sum_{j=1}^{0} \gamma_j . r_{t-j}^2 I\left[\varepsilon_{t-j} < 0\right] + \sum_{k=1}^{q} \beta_k . \sigma_{t-k}^2 \quad . \tag{2}$$

III. DEEP LEARNING (DL) MODELS

This section is devoted to briefly describe the basic principle of two *Non-linear* deep learning models that will be used later for cryptocurrencies volatility forecasting such as RNN, LSTM, (Zouaoui & Naas, 2023).

DOI: 10.4316/EJAFB.2025.13214

Volume **13** / **2025** ISSN 2344-102X
Issue **2** ISSN-L 2344-102X

3.1. Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNNs) are a class of deep learning models designed for sequential data, making them particularly effective for time series forecasting. Unlike feed forward neural networks, RNNs process inputs sequentially while maintaining a "memory" of previous time steps through hidden states, enabling them to capture temporal dependencies in data. Below is a step-by-step guide to implementing RNNs for forecasting task:

1.1 Define the Forecasting Problem: Identify the variable you want to forecast (e.g., sales, stock prices).

- Determine the time frame for your predictions (e.g., daily, weekly).

1.2. Collect and Prepare Data:

- Data Collection: Gather historical data relevant to the forecasting task.
- Data Preprocessing: Clean the data to handle missing values and normalize or standardize the data to improve model performance.
- Feature Engineering: Create features that may enhance the model, such as lagged values (previous observations) and Moving averages or other aggregations.

1.3. Transform Data into Sequences: Convert the time series data into sequences suitable for RNN input. This usually involves creating input-output pairs based on a specified look-back period.

1.4. Split Data into Training and Testing Sets: Split the dataset while maintaining the chronological order (e.g., 80% for training, 20% for testing).

1.5. Choose a deep learning framework: Select a framework such as TensorFlow/Keras or PyTorch.

1.6. Build the RNN Model: Define the architecture, including input layers, RNN layers (LSTM or GRU), and output layers.

1.7. Compile the model: Choose an optimizer (e.g., Adam) and a loss function (e.g., Mean Squared Error).

1.8. Train the model: Fit the model on the training data, specifying the number of epochs and batch size.

1.9. Evaluate the model: Assess model performance on the test set using appropriate metrics (e.g., RMSE, MAE).

1. 10. Make predictions: Use the trained model to make forecasts.

1. 11. Visualize results: Plot actual vs. predicted values to assess the model's performance visually.

1. 12. Fine-tuning and optimization: Experiment with hyperparameters (e.g., number of layers, units, learning rate) and use techniques like early stopping and regularization (e.g., dropout) to improve model robustness.

3.2. Long Short-Term Memory (LSTM) Model

Long Short-Term Memory (LSTM) is a specialized type of Recurrent Neural Network (RNN) designed to recognize patterns and make predictions based on sequential data. Its key innovation is an internal architecture that allows it to effectively learn long-range dependencies and remember information for extended periods, overcoming the primary limitation of standard RNNs: the vanishing gradient problem. In essence, an LSTM is an algorithm that can remember and forget information selectively, making it exceptionally powerful for tasks where context and order are crucial.

Moreover, LSTMs provide a powerful tool for time series forecasting, capable of handling complex patterns inherent in sequential data. Their unique architecture allows them to maintain context over time, making them a popular choice in many applications (Persson, 2024). Furthermore, Figure1 shows a schematic diagram of LSTM model:

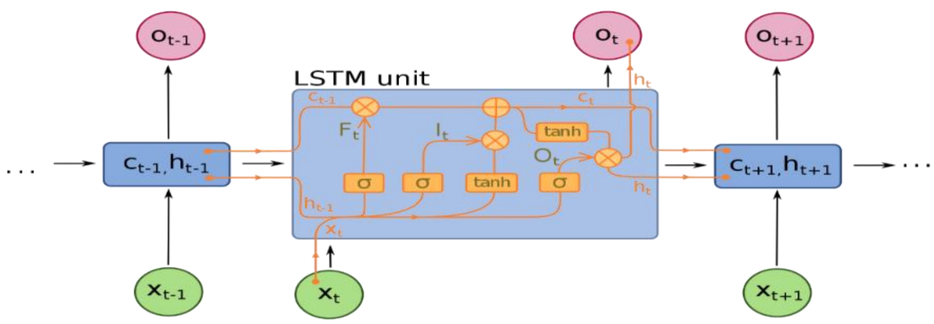


Figure 1. Schematic diagram of LSTM model
Source: authors based on python code GitHub.

- *Input Gate* (r_t) & *Candidate Cell State* (d_t): The input gate decides which new information to store in the cell state:

DOI: 10.4316/EJAFB.2025.13214

Volume **13** / **2025** ISSN 2344-102X
Issue **2** ISSN-L 2344-102X

$$r_t = \sigma\left(W_f.\,[h_{t-1}, x_t]\right) + b_f \tag{3}$$

The candidate cell state represents the new information that could be added to the cell state.

$$\mathrm{d_t} = \tanh(W_d.\,[h_{t-1}, x_t]) + b_d \tag{4}$$

- *Forget Gate:* The forget gate decides how much of the previous memory should be discarded from the cell state:

$$\mathrm{f_t} = \sigma(W_i.\,[h_{t-1}, x_t]) + b_i \tag{5}$$

when: σ = Sigmoid function that outputs values between 0 and 1 (0 means “forget” and 1 means “retain”).

- *Cell State Update:* The new cell state is updated based on the forget gate and input gate decisions:

$$C_\mathrm{t} = \mathrm{f_t}.\,C_{t-1} + r_t.\,d_t \tag{6}$$

The forget gate C_t scales the previous cell state C_{t-1}, and the input gate r_t scales the candidate cell state d_t.

- *Output gate:* The output gate controls what part of the cell state to output as the next hidden state h_t:

$$o_\mathrm{t} = \sigma(W_0.\,[h_{t-1}, x_t]) + b_0 \tag{7}$$

$$h_\mathrm{t} = o_\mathrm{t}\tanh C_\mathrm{t} \tag{8}$$

The output gate decides how much of the cell state should be passed to the next time step.

IV. RESULTS AND ANALYSIS

4.1. Data Description

The study investigates the daily prices of a financial portfolio consisting of nine prominent cryptocurrencies: Bitcoin (BTC), Ethereum (ETH), XRP, Bitcoin Cash (BCH), Tether (USDT), Litecoin (LTC), EOS, Binance Coin (BNB), and Stellar (XLM). This analysis spans from January 1, 2020, to May 14, 2024, comprising a total of 1,598 observations. the Figure 2 shows the development of the cumulative returns:

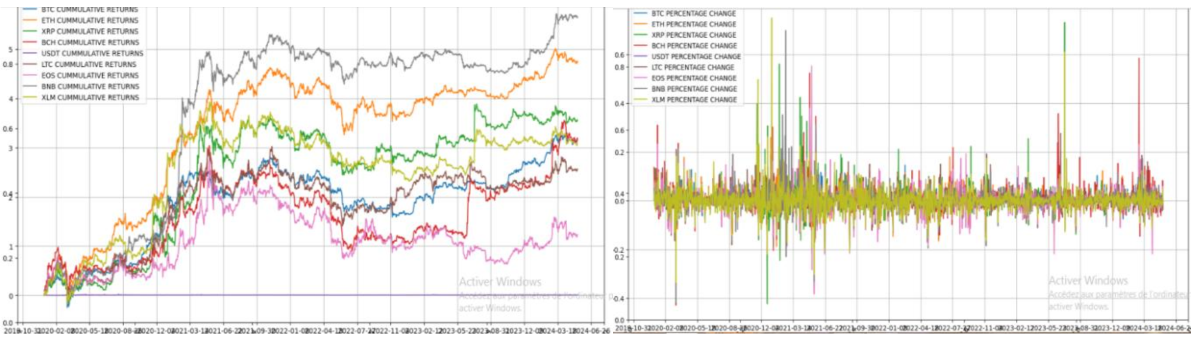


Figure 2. The cumulative returns of the cryptocurrencies
Source: based on python code GitHub

The cryptocurrency market from 2020 to 2025 has undergone dramatic fluctuations influenced by a convergence of economic, technological, and regulatory factors. Beginning with a sharp recovery from the initial pandemic-induced downturn in 2020, the market saw unprecedented interest from institutional investors and a surge in decentralized finance (DeFi) applications. In 2021, Bitcoin and other cryptocurrencies reached new all-time highs, but subsequent corrections highlighted the market's volatility.

The bear market of 2022 was marked by significant declines, driven by macroeconomic pressures, high-profile collapses, and increased regulatory scrutiny. However, 2023 witnessed a tentative recovery, buoyed by easing inflation concerns and ongoing institutional interest, alongside advancements in blockchain technologies.

As we move into 2025, expectations centre around potential bullish trends driven by Bitcoin's halving event and evolving regulatory frameworks that may foster greater market stability and adoption. This period underscores

DOI: 10.4316/EJAFB.2025.13214

Volume **13** / **2025** ISSN 2344-102X
Issue **2** ISSN-L 2344-102X

the cryptocurrency market's resilience and its ongoing evolution as a significant component of the global financial landscape. these fluctuations are linked to several reasons, including:

- Fraud and Security Breaches:

- High-profile fraud cases have under mined trust in digital assets, eroding their perceived legitimacy.
- Repeated hacking incidents targeting cryptocurrency platforms (e.g., exchange breaches, wallet exploits) have exposed systemic vulnerabilities in security practices.

- Market Instability and Platform Failures:

- Several major platforms collapsed due to mismanagement or insolvency, most notably FTX, the industry's largest exchange before its abrupt bankruptcy in 2022.
- Regulatory crackdowns in key markets (e.g., the U.S., EU) forced some platforms to exit restrictive jurisdictions and relocate to less regulated regions, further destabilizing the ecosystem.

- Regulatory Shifts:

- Governments worldwide have adopted stricter oversight, reversing the laissez-faire approach of earlier years. The U.S. has emerged as a leader in this trend, pursuing aggressive enforcement actions against non-compliant entities.
- Such measures have driven fragmentation, with some platforms withdrawing from major economies entirely.

- Geopolitical and Macroeconomic Influences:

From 2020 to 2025, the cryptocurrency market has been profoundly influenced by a range of geopolitical and macroeconomic factors. The COVID-19 pandemic initiated a period of economic uncertainty, prompting increased interest in cryptocurrencies as alternative assets, especially amid expansive government stimulus measures. As institutional adoption surged, major corporations began investing in digital currencies, further entrenching them in mainstream finance.

However, this period also witnessed heightened regulatory scrutiny, with governments implementing stricter frameworks that affected market dynamics and investor confidence. Geopolitical tensions and economic sanctions led to a rise in cryptocurrency usage in conflict zones, showcasing their utility in unstable environments.

Furthermore, the macroeconomic conditions, including rising inflation and interest rates, contributed to significant market volatility, impacting investor sentiment and trading behaviours. Looking ahead to 2025, anticipated regulatory clarity and technological advancements, including the development of central bank digital currencies (CBDCs), are poised to reshape the landscape of digital assets, influencing both stability and adoption. This interplay of geopolitical and macroeconomic factors underscores the evolving role of cryptocurrencies with in the global financial system.

2. Cointegration Test

Cointegration is a statistical property used to identify long-term equilibrium relationships between non-stationary time series. Unlike correlation, which measures short-term linear dependence, cointegration reveals whether two or more series move together over time, even if they exhibit short-term deviations. However, From the results obtained we saw some very low Cointegration for our desired Bitcoin Pair.

```python
# obtaining the P-value for every possible pair within our selected (9- Crypto currency Universe)

for a1 in dt.columns:
    for a2 in dt.columns:
        if a1 != a2:
            test_result = ts.coint(dt[a1], dt[a2])
            print(a1 + ' and ' + a2 + ': p-value = ' + str(test_result[1]))
```

```
BTC_RET and ETH_RET: p-value = 1.5311977888300337e-24
BTC_RET and XRP_RET: p-value = 0.0
BTC_RET and BCH_RET: p-value = 2.645070809908137e-13
BTC_RET and USDT_RET: p-value = 1.338906581366561e-29
BTC_RET and LTC_RET: p-value = 0.0
BTC_RET and EOS_RET: p-value = 1.1537839063997665e-25
BTC_RET and BNB_RET: p-value = 2.2614703511807234e-14
BTC_RET and XLM_RET: p-value = 0.0
```

Figure 3. Cointegration model
Source: authors based on python code GitHub.

BTC/XRP Pair, the BTC/LTC Pair and the BTC/XLM Pair tend to give the lowest P-Values, however since the desired P-value is anything less than 0.05, all of the pairs would be considered to have passed the Cointegration test. Hence, we decided to do the correlation test. Correlation is the degree to which the pair tend to move together.

3. Defining Correlation

From the analysis and the values gotten we can see that although pairs all seen to have very low Cointegration value, but we observed that the Ethereum (ETH) had the best correlation value (81.97%) (Table 2

DOI: 10.4316/EJAFB.2025.13214

Volume 13 / 2025 ISSN 2344-102X
Issue 2 ISSN-L 2344-102X

and Figure 4). Thus, we will be proceeding with the Bitcoin/Ethereum pair for the next two phases; Price forecasting and developing trade strategy:

Table 2. Correlation Matrix

```
# Defining Correlation Matrix
corrMatrix = dt.corr()
dt.corr()
```

	BTC_RET	ETH_RET	XRP_RET	BCH_RET	USDT_RET	LTC_RET	EOS_RET	BNB_RET	XLM_RET
BTC_RET	1.000000	0.819727	0.556156	0.690805	-0.147977	0.761016	0.679108	0.659034	0.610452
ETH_RET	0.819727	1.000000	0.594234	0.703566	-0.157795	0.787622	0.717390	0.678911	0.644230
XRP_RET	0.556156	0.594234	1.000000	0.552475	-0.094448	0.614780	0.642634	0.505741	0.745004
BCH_RET	0.690805	0.703566	0.552475	1.000000	-0.119270	0.774201	0.775969	0.562846	0.605454
USDT_RET	-0.147977	-0.157795	-0.094448	-0.119270	1.000000	-0.134814	-0.115667	-0.125746	-0.115556
LTC_RET	0.761016	0.787622	0.614780	0.774201	-0.134814	1.000000	0.765212	0.629907	0.638396
EOS_RET	0.679108	0.717390	0.642634	0.775969	-0.115667	0.765212	1.000000	0.613144	0.684819
BNB_RET	0.659034	0.678911	0.505741	0.562846	-0.125746	0.629907	0.613144	1.000000	0.530834
XLM_RET	0.610452	0.644230	0.745004	0.605454	-0.115556	0.638396	0.684819	0.530834	1.000000

Source: based on python code GitHub

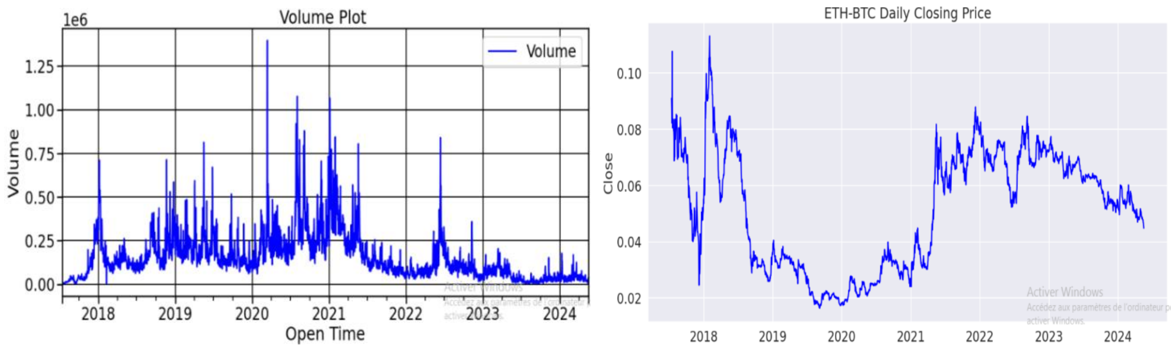


Figure 4. ETH-BTC Daily closing price
Source: based on python code GitHub

From graph above, we can conclude that there is no trend & seasonality. In March 2020 there was highest no of traders trading in ETH-BTC and after that there was significant decay in no. of traders.

Therefore, our study gives live forecasting that is whenever we run the model it well get updated and added new data points up to current time frame. So, some conclusions may change due that. So, best practice is update conclusions also, whenever you run the model.

From Heatmap we select best pair of cryptocurrency ETH-BTC (Base currency-Quote Currency) for further analysis because both are top 2 Cryptocurrencies of Crypto market. We fit MA on close price of ETH-BTC when MA is too large it not capable to capturing some spike and dips and when MA is small its again there is again noisy observation problem so there is compromise and we need to looking for some other methods like AR, ARMA, ARIMA, ACF, EWMA.

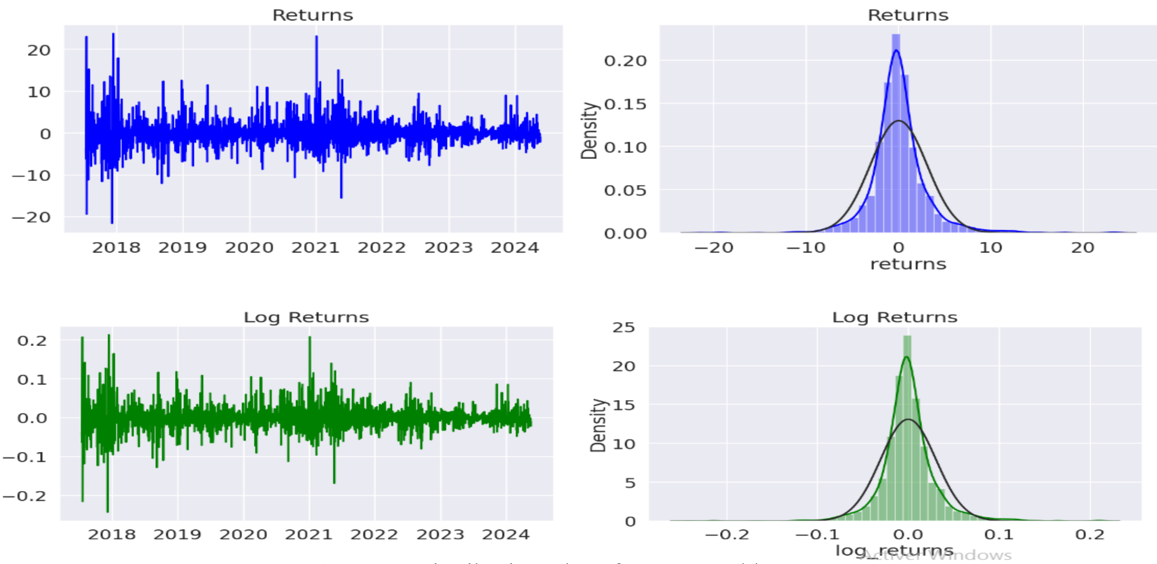


Figure 5. Distribution Plot of returns and log returns

DOI: 10.4316/EJAFB.2025.13214

Volume **13** / **2025**
Issue **2**
ISSN 2344-102X
ISSN-L 2344-102X

Source: based on python code GitHub

Both Returns & Log Returns show some: slight positive skewness, positive kurtosis (leptokurtic) - higher peak with thicker tails than the standard normal distribution (Figure 5 and Figure 6).

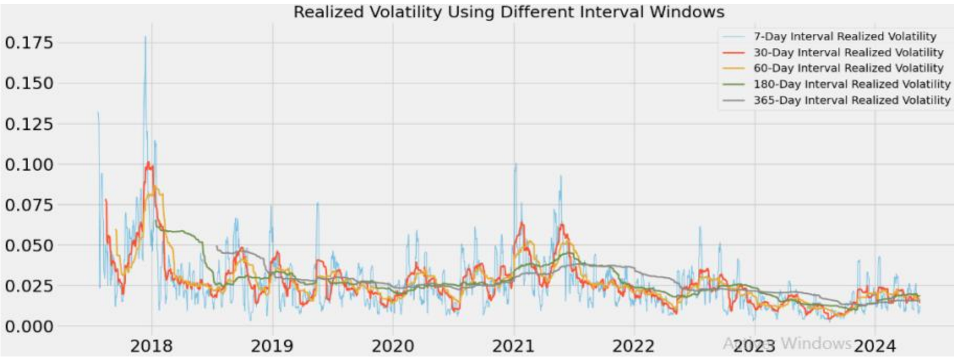


Figure 6. Realized volatility using diffrent interval windows
Source: based on python code GitHub

The choice of a 30-day rolling window for volatility analysis balances two critical trade-offs:

- Noise Reduction: Shorter intervals (e.g., 7 days) capture excessive short-term fluctuations, obscuring meaningful trends.
- Volatility Preservation: Longer intervals (e.g., 90 days) over-smooth the data, diluting volatility signals and artificially reverting to the mean.
- Data Efficiency: A 30-day window mitigates the "warm-up" data loss problem (e.g., fewer wasted initial data points compared to longer windows).

4. Forecasting Performance of Benchmark Models

The dataset, comprising 1598 data points, is divided into three groups: the training, the validation, and the test sets. Therefore, Hyperparameter tuning is conducted to account for these factors and to identify the optimal LSTM configuration. The tuning process involves a grid search of all possible combinations of the pre-set hyperparameters to find the optimal structure based on a loss function, which in this case is RMSE (Dudek et al., 2024). According to the primary hyperparameters for subjective input are dropout ratio, activation function, number of hidden layers, batch size, and learning rate alpha. Their findings further suggest that altering the learning rate alpha does not result in significant outcome changes. Hence, this paper follows (Seabe et al., 2023) by using the default value in the Keras package. The process is illustrated below (see Figure 7):

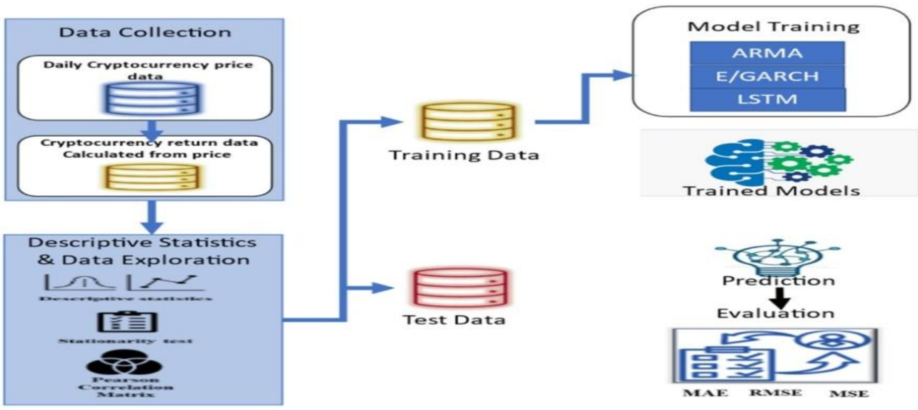


Figure 7. Flowchart of forecasting process
Source: Author`s elaboration based on Mohammed (2024)

The hyperparameters and their respective options for this paper are illustrated in Table 3:

DOI: 10.4316/EJAFB.2025.13214

EUROPEAN JOURNAL OF ACCOUNTING, FINANCE & BUSINESS
Volume **13** / **2025** ISSN 2344-102X
Issue **2** ISSN-L 2344-102X

Table 3. Variations of LSTM models involved in hyper-parameter tuning

Parameter	Options
Activation function	RELU,Tanh, Sigmoid
Loss function	RMSE, RMSPE
Neurons	[100,100,100,100,100,1]
Learning rate	0.001
Optimiser	Adam
LSTM Layers	2,3,4
BtachSize	32,64

Source**: authors based on python code GitHub.**

Here, [100, 100, 100, 100, 100, 1] represents the number of neurons from the first to the last network layer. The architecture of the final model includes a dropout rate of 0.0, a tanh activation function, three bidirectional LSTM layers with 32, 16, and 16 units, respectively, a lookback window of 30, a batch size of 64, and four additional layers. The optimizer used is Adam, and the metric RMSE. A key advantage of deep learning algorithms is their ability to perform feature selection and data scaling automatically (Abraham, 2020). In this study, we present the actual versus predicted cryptocurrency values over time, using automatically scaled data generated by our proposed deep learning approach (see Appendix A, B). Table 4 shows the RMSE and RMSPE results for all techniques evaluated (Khan, 2023).

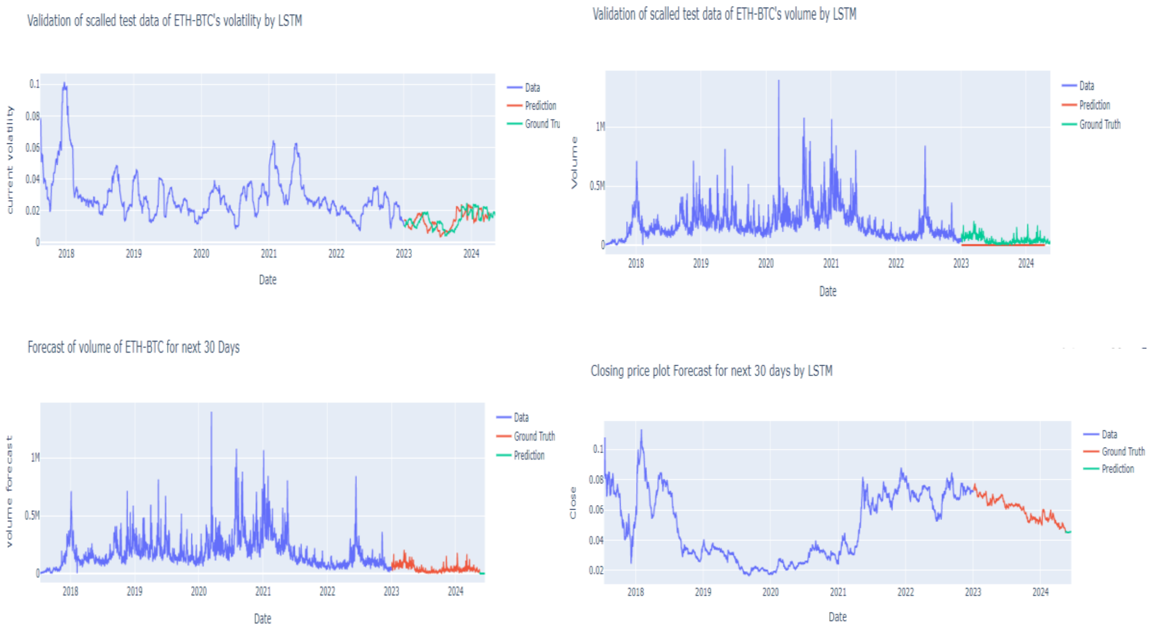


Figure 8. ETH-BTC Volatility Prediction based on LSTM
Source: authors based on python code github.

Table 4. Benchmark Methods & Forecast Accuracy

Model	Validation RMSPE	Validation RMSE
LSTM	**0.112561**	**0.011456**
Mean Baseline	35.992981	NAN
Random Walk Naive Forecasting	4.876007	NAN
GARCH(1,1), Constant Mean, Normal Dist	1.966938	0.187648
Analytical GJR-GARCH (1,1,1), Constant Mean, skewt Dist	2.478508	0.193394
Simulation TARCH (1,1), Constant Mean, skewt Dist	4.729480	0.094409
Boostrap TARCH (1,1), Constant Mean, skewt Dist	4.875756	0.088403
Boostrap TARCH (1,2,2), Constant Mean, skewt Dist	4.624266	0.097845

Source: authors based on python code github.
Note: the values of RMSE are multiplied by 10^3, the lowest values of RMSPE and RMSE are in bold. The evaluation period is 01/01/2020-14/05/2024.

The study tested eight different forecasting models, and among them, the LSTM (Long Short-Term Memory) model performed the best in predicting the target variable (likely the ETH/BTC price). Therefore, RMSPE and RMSE are two metrics used to measure the accuracy of the model's predictions, when RMSPE (Root

Volume **13** / **2025** ISSN 2344-102X
Issue **2** ISSN-L 2344-102X

Mean Squared Percentage Error): This value represents how far off predictions are in percentage terms. A lower RMSPE (0.112561 in this case) suggests better model accuracy. and RMSE (Root Mean Squared Error): This measures how much the predictions deviate from actual values in the original scale. A lower RMSE value (0.011456 here) also suggests the model is highly accurate. However, The Mean Baseline model (a simple model that likely predicts future values of ETH-BTC based on the historical average) is claimed to be "superior" with a value of 35.992981. These values being the lowest among the models indicate that the LSTM model outperformed the others. However, Preliminary analysis reveals deep learning algorithms show significant promise for cryptocurrency price prediction, exhibiting improved performance in modelling the market's characteristic volatility in cryptocurrencies markets. The findings contribute to the growing body of literature on the application of artificial intelligence in finance and offer insights into the potential improvements in the accuracy of cryptocurrencies forecasting achievable through deep learning methodologies. As financial markets continue to evolve, embracing innovative technologies like deep learning becomes imperative for refining risk assessment and decision-making processes in the realm of cryptocurrency trading.

Finally, we check stationarity of close price, so our close prices are not stationary hence instead of calculating volatility of close price we choose to formulate returns and then volatility of returns for further analysis. But before that we forecast close prices by LSTM, and they are slightly decreasing. we plot volume plot and then forecast it by LSTM, so our forecast shows rate of change of decreasing volume is reduced, so its good indicator. We calculate returns and then log returns for practicality purposes, it's generally preferable to use the log returns especially in mathematic modelling, because it helps eliminate non-stationary properties of time series data and makes it more stable. There's another advantage to log returns, which is that they're additive across time: Then we compare the distribution plots and stationarity plot of Returns and Log Returns, and their distribution is normal, and stationarity plots are also same. Then we fit auto ARIMA model to log returns but there is high heteroskedasticity so instead of going to traditional time series model we have to choose time series models for High Volatility such as Baseline models and GARCH model We fit Baseline models, GARCH model and LSTM machine learning model and compare them based on RMSPE & RMSE so our LSTM model have lowest RMSPE & RMSE values. So, we forecast volatility by LSTM model Our LSTM forecast for Volatility are decreasing in future for next one month. Overall conclusion is that its good time to tread in BTC Instead of ETH. Moreover, close price forecast by LSTM shows that trend of future close price is decreasing hence Price of Bitcoin is increasing or price of Etherium is decreasing. So, we can transfer our Etherium portfolio into Bitcoin.

IV. CONCLUSION

Based on the above results, we hypothesized that deep learning models have demonstrated superior performance over GARCH in forecasting cryptocurrency volatility, particularly in complex and high-frequency scenarios. However, GARCH remains relevant for simpler tasks and scenarios where interpretability is critical. The future of volatility forecasting lies in hybrid approaches and the integration of advanced deep learning techniques with traditional econometric models.As briefly mentioned above, we think there's potential application of deep learning approach in the forecasting of volatility and would like to explore that option in the future.

The relative importance of the GARCH component versus the LSTM component likely shifted over the 14-year period. because in the early years (pre-2017), when markets were less mature and more driven by technical trading, the GARCH component may have been more dominant. In later years (post-2020), with the influence of institutional flows, DeFi, NFTs, and macro correlations, the LSTM's ability to capture these complex, novel dependencies became increasingly critical.

Therefore, The GARCH-LSTM model implicitly captured volatility asymmetry (the "leverage effect") without being explicitly programmed for it, a task where standard GARCH often fails. However, during sharp downturns (e.g., the 2018 crash, the 2022 LUNA/FTX collapse), the LSTM component likely contributed significantly to forecasts by recognizing the unique "signature" of panic selling and cascading liquidations, which differ from the volatility patterns of bull markets. Further, the model acts as an intelligent regime-detection tool, understanding that "bad news" increases volatility more than "good news" and that certain market states have distinct volatility dynamics.

Moreover, it is widely recognized that economic events can influence marketdynamics. However, cryptocurrencies possess unique characteristics that distinguish them from traditional stocks and commodities, making the inclusion of standard economic calendar events potentially less relevant. I amconducting further research to identify significant events that may have driven Bitcoin's price movements and to incorporate these into a future Multivariate LSTM model to enhance predictive accuracy. Additionally, we plan to explore higher-frequency data (e.g., intra-day) and experiment with different time intervals for data aggregation to further refine our analysis.

Finally, based on the current results, future research can predict a large performance improvement by optimizing the parameters of these algorithms for application in more common forecasting scenarios.

DOI: 10.4316/EJAFB.2025.13214

Volume **13** / **2025** ISSN 2344-102X
Issue **2** ISSN-L 2344-102X

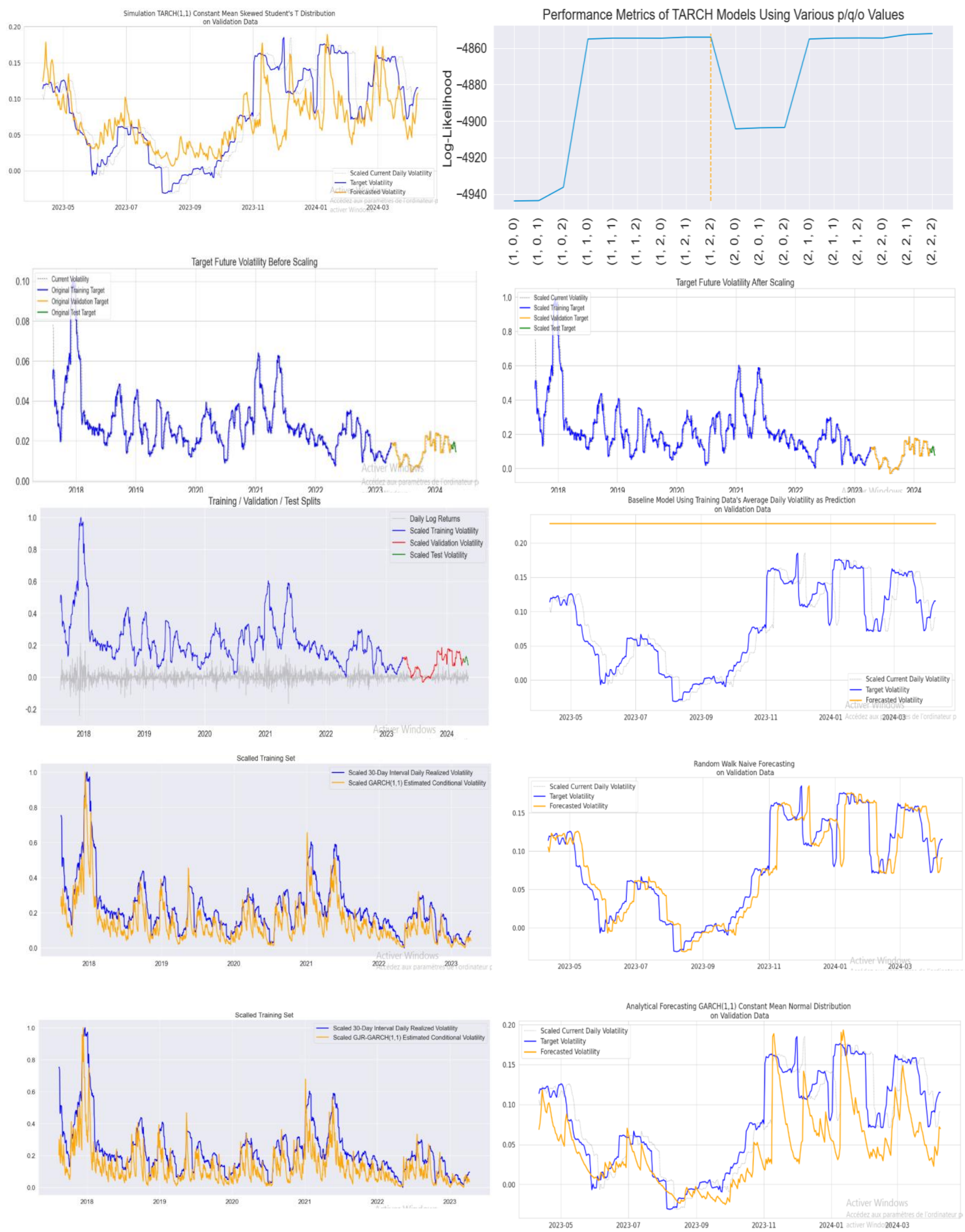
REFERENCES

1. Abraham, M. (2020). Studying the patterns and long-run dynamics in cryptocurrency prices. Journal of Corporate Accounting & Finance, 31(3), 98-113.
2. AlMadany, N. N., Hujran, O., Al Naymat, G., & Maghyereh, A. (2024). Forecasting cryptocurrency returns using classical statistical and deep learning techniques. *International Journal of Information Management Data Insights*, *4*(2), 100251, https://doi.org/10.1016/j.jjimei.2024.100251
3. Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, *31*(3), 307-327.
4. Boongasame, L., & Songram, P. (2023). Cryptocurrency price forecasting method using long short-term memory with time-varying parameters. *Indonesian Journal of Electrical Engineering and Computer Science*, *30*(1), 435-443. doi:10.11591/ijeecs.v30.i1.pp435-443.
5. Boozary, P., Sheykhan, S., & GhorbanTanhaei, H. (2025). Forecasting the Bitcoin price using the various Machine Learning: A systematic review in data-driven marketing. *Systems and Soft Computing*, 200209. https://doi.org/10.1016/j.sasc.2025.200209
6. Bouteska, A., Abedin, M. Z., Hajek, P., & Yuan, K. (2024). Cryptocurrency price forecasting–a comparative analysis of ensemble learning and deep learning methods. *International Review of Financial Analysis*, *92*, 103055.
7. Cevik, E. I., Gunay, S., Dibooglu, S., & Yıldırım, D. Ç. (2023). The impact of expected and unexpected events on Bitcoin price development: Introduction of futures market and COVID-19. Finance Research Letters, 54, 103768.
8. Chai, J., & Ngai, E. W. (2020). Decision-making techniques in supplier selection: Recent accomplishments and what lies ahead. *Expert Systems with Applications*, *140*, 112903., https://doi.org/10.1016/j.eswa.2019.112903
9. Dudek, G., Fiszeder, P., Kobus, P., & Orzeszko, W. (2024). Forecasting cryptocurrencies volatility using statistical and machine learning methods: A comparative study. *Applied soft computing*, *151*, 111132. https://doi.org/10.1016/j.asoc.2023.111132
10. Estrada, J. C. S. (2017). Analyzing bitcoin price volatility. *University of California, Berkeley*.
11. Fleischer, J. P., von Laszewski, G., Theran, C., & Parra Bautista, Y. J. (2022). Time series analysis of cryptocurrency prices using long short-term memory. *Algorithms*, *15*(7), 230. doi:10.3390/a15070230.
12. Gaies, B., Nakhli, M. S., Sahut, J. M., & Schweizer, D. (2023). Interactions between investors' fear and greed sentiment and Bitcoin prices. *The North American Journal of Economics and Finance*, *67*, 101924. https://doi.org/10.1016/j.najef.2023.101924
13. Garcia, D., Tessone, C. J., Mavrodiev, P., & Perony, N. (2014). The digital traces of bubbles: feedback cycles between socio-economic signals in the Bitcoin economy. *Journal of the royal society interface*, *11*(99), 20140623.
14. García-Medina, A., & Aguayo-Moreno, E. (2024). LSTM–GARCH hybrid model for the prediction of volatility in cryptocurrency portfolios. *Computational Economics*, *63*(4), 1511-1542.
15. Glosten, L. R., Jagannathan, R., & Runkle, D. E. (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. The journal of finance, 48(5), 1779-1801. https://algotrading101.com/learn/yfinance-guide/
16. Huang, J., Chai, J., & Cho, S. (2020). Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China*, *14*(1), 13.. https://doi.org/10.1186/s11782-020-00082-6
17. Jirou, I., Jebabli, I., & Lahiani, A. (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*, 102575. https://doi.org/10.1016/j.ribaf.2024.102575
18. Kanaparthi, V. (2024). Robustness Evaluation of LSTM-based Deep Learning Models for Bitcoin Price Prediction in the Presence of Random Disturbances. *International Journal of Innovative Science and Modern Engineering,* 12(2), 14–23. doi:10.35940/ijisme.b1313.12020224
19. Khan, F.U., Khan, F. & Shaikh, P.A. (2023).Forecasting returns volatility of cryptocurrency by applying various deep learning algorithms. *Future Business Journal*, 9, 25 .https://doi.org/10.1186/s43093-023-00200-9
20. Khaniki, M. A. L., Manthouri, M. (2024). Enhancing Price Prediction in Cryptocurrency Using Transformer Neural Network and Technical Indicators. *ar Xivpreprint*, arXiv:2403.03606.doi:10.48550/arXiv.2403.03606.
21. Koutmos, D. (2018). Bitcoin returns and transaction activity. Economics Letters, 167, 81-85.
22. Lahmiri, S., & Bekiros, S. (2019). Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solitons & Fractals*, *118*, 35-40. https://doi.org/10.1016/j.chaos.2018.11.014
23. Li, Y., & Dai, W. (2020). Bitcoin price forecasting method based on CNN‐LSTM hybrid neural network model. *The journal of engineering*, *2020*(13), 344-347. doi:10.1049/joe.2019.120
24. Mohammed, A. N. (2024). Hybrid cryptocurrency price prediction: Integrating EGARCH and LSTM with explainable AI. International Journal of Innovative Science and Research Technology, 9(12), 731-738. https://ijisrt.com/assets/upload/files/IJISRT24DEC1137.pdf
25. Nelson, D. B., & Cao, C. Q. (1992). Inequality constraints in the univariate GARCH model. Journal of Business & Economic Statistics, 10(2), 229-235.
26. Patel, M. M., Tanwar, S., Gupta, R., & Kumar, N. (2020). A deep learning-based cryptocurrency price prediction scheme for financial institutions. *Journal of information security and applications*, *55*, 102583. doi:10.1016/j.jisa.2020.102583
27. Patel, N. P., Parekh, R., Thakkar, N., Gupta, R., Tanwar, S., Sharma, G., ... & Sharma, R. (2022). Fusion in cryptocurrency price prediction: A decade survey on recent advancements, architecture, and potential future directions. *IEEE Access*, *10*, 34511-34538. https://doiorg/10.1109/ACCESS.2022.3163023
28. Persson P. F. (2024). Comparative Analysis of Econometric and Machine Learning Approaches for Forecasting Bitcoin Return Volatility: GARCH versus LSTM Models, Master Thesis, *Lund University School of Economics and Management,* Lund, Suedia
29. Rachid , B., Issam, A., Mounir , A.K., & A, Zaaouat. (2024), Cryptocurrency Forecasting Using Deep Learning Models: A Comparative Analysis, *HighTech and Innovation Journal,* V.5(4), PP:1055-1067. http://dx.doi.org/10.28991/HIJ-2024-05-04-013
30. Seabe, P. L., Moutsinga, C. R. B., & Pindza, E. (2023). Forecasting cryptocurrency prices using LSTM, GRU, and bi-directional LSTM: a deep learning approach. *Fractal and Fractional*, *7*(2), 203.
31. Shen, Z., Wan, Q., & Leatham, D. J. (2021). Bitcoin return volatility forecasting: A comparative study between GARCH and RNN. Journal of Risk and Financial Management, 14(7), 337. https://doi.org/10.3390/jrfm14070337 .
32. Tripathy, N., Hota, S., & Mishra, D. (2023). Performance analysis of bitcoin forecasting using deep learning techniques. Indonesian Journal of Electrical Engineering and Computer Science, 31(3), 1515-1522.
33. Wang, Y., Andreeva, G., & Martin-Barragan, B. (2023). Machine learning approaches to forecasting cryptocurrency volatility: Considering internal and external determinants. *International Review of Financial Analysis*, *90*, 102914..https://doi.org/10.1016/j.irfa.2023.102914
34. Wang, Y., Wang, C., Sensoy, A., Yao, S., & Cheng, F. (2022). Can investors' informed trading predict cryptocurrency returns? Evidence from machine learning. *Research in International Business and Finance*, *62*, 101683.

Volume **13** / **2025** ISSN 2344-102X
Issue **2** ISSN-L 2344-102X

35. Wirawan, I. M., Widiyaningtyas, T., & Hasan, M. M. (2019, September). Short term prediction on bitcoin price using ARIMA method. In 2019 International Seminar on Application for Technology of Information and Communication (iSemantic) (pp. 260-265). IEEE. doi: 10.1109/ISEMANTIC.2019.8884257
36. Zakoian, J. M. (1994). Threshold heteroskedastic models. Journal of Economic Dynamics and control, 18(5), 931-955.
37. Zhang, Z., Dai, H. N., Zhou, J., Mondal, S. K., García, M. M., & Wang, H. (2021). Forecasting cryptocurrency price using convolutional neural networks with weighted and attentive memory channels. *Expert Systems with Applications*, *183*, 115378.
38. Zouaoui, H., & Naas, M.N. (2023). Option pricing using deep learning approach based on LSTM-GRU neural networks: Case of London stock exchange. *Data Science in Finance and Economics*, 3(3), 267–284. https://doi.org/10.3934/DSFE.2023016

- **APPENDIX A:**



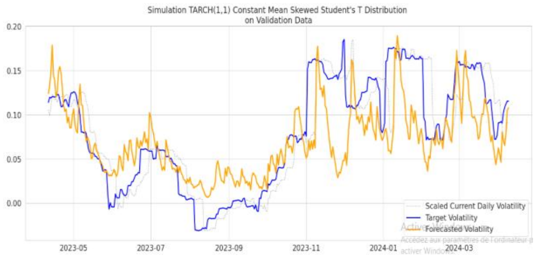
Source: authors based on python code GitHub.

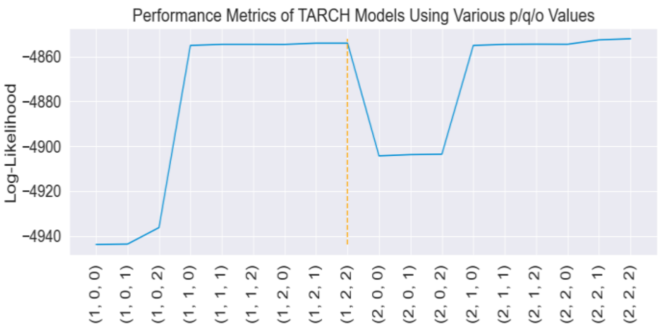
DOI: 10.4316/EJAFB.2025.13214

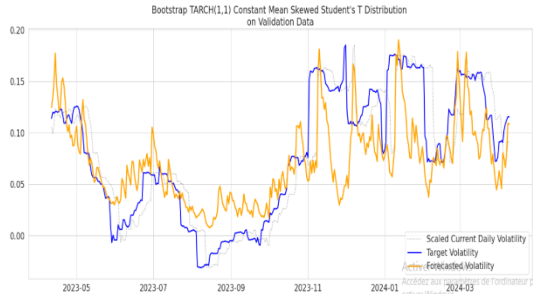
EUROPEAN JOURNAL OF ACCOUNTING, FINANCE & BUSINESS

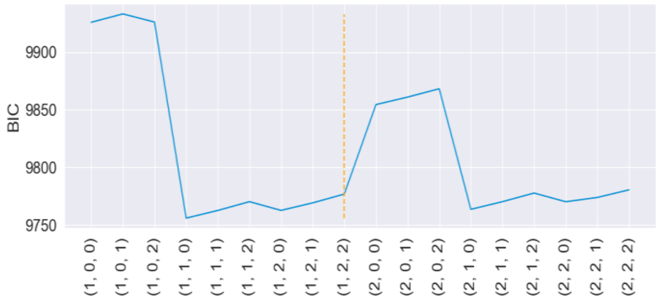
Volume 13 / 2025 ISSN 2344-102X
Issue 2 ISSN-L 2344-102X

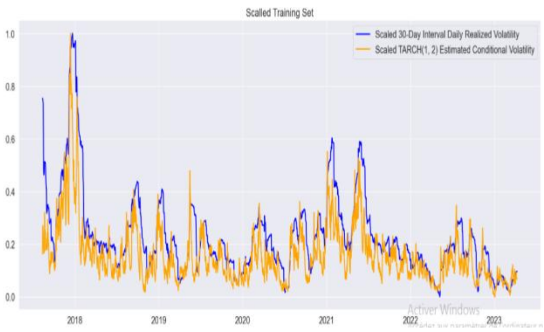
- **APPENDIX B:**

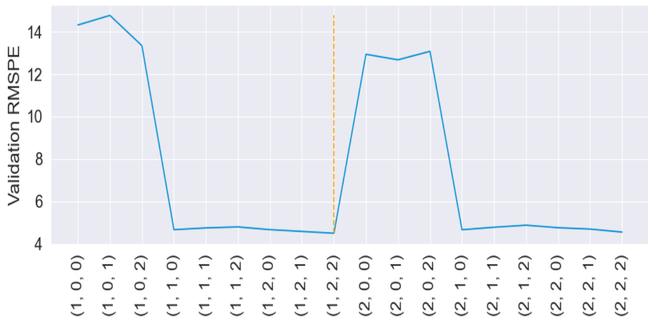


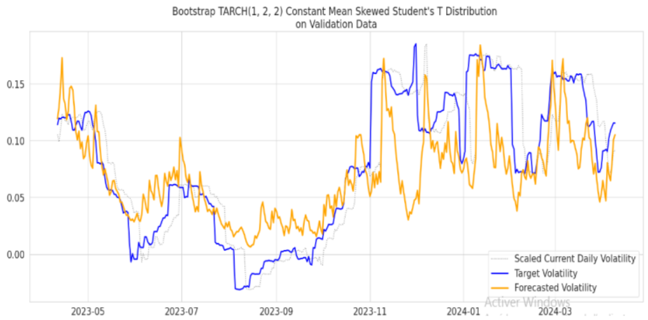












Source: authors based on python code GitHub.

DOI: 10.4316/EJAFB.2025.13214