LUNDS UNIVERSITET

Comparative Analysis of Econometric and Machine Learning Approaches for Forecasting Bitcoin Return Volatility:

GARCH versus LSTM Models.

May 2024

Master Thesis | Faculty of Economics.

Author: Filip Podvorec

Supervisor: Andreas Johansson

Lund University School of Economics and Management

Abstract

The fundamental principle of investing, applicable at both systematic and individual levels, is the art of balancing the portfolio between expected returns and acceptable risk exposure. Risk management involves predicting the direction of the underlying asset's movement and taking appropriate actions accordingly. Traditionally, statistical models such as GARCH have been used to achieve this objective. However, with advances in machine learning, new algorithms may prove more effective by identifying more complex patterns. This paper compares the accuracy of GARCH and LSTM models in predicting future Bitcoin return volatility using the root mean squared error (RMSE) on out-of-sample data. The results indicate that the LSTM model outperforms the GARCH model, suggesting significant potential for machine learning applications in enhancing risk management and forecasting practices.

Keywords: Bitcoin, Forecast, LSTM, GARCH, Risk Management, Machine Learning

Table of Contents

1.0 Introduction

Bitcoin was the first cryptocurrency launched in 2009 to serve as a direct electronic cash system between two parties. The rationale behind creating this decentralized currency stemmed from Satoshi Nakamoto (2008), the pseudonymous creator of Bitcoin, who argued that society would benefit from a direct transactional system that operates without financial intermediaries. This application of Bitcoin challenges the traditional financial system by aiming to replace fiat currency as the primary medium of exchange (Chen, Li, & Sun, 2020).

The main difference between Bitcoin and more traditional assets lies in the basis of their underlying value. The price of traditional stocks is based on the financial performance of the companies they represent, with ownership of a stock signifying a partial stake in the business. Conversely, Bitcoin is considered a speculative asset without intrinsic value; its price is purely determined by market confidence (Royal, 2024). This unconventional characteristic contributes to its pronounced volatility, driven by internal and external factors (Bouri, Molnár, Azzi, Roubaud, & Hagfors, 2017). Since its establishment, Bitcoin has made significant advancements, commanding a market dominance of 47.8% within a market valued at $1.77 trillion as of late 2023 (York, 2024). Over 300 hedge funds specialize in digital assets, and approximately 8,985 cryptocurrencies are circulating in the market (Howarth, 2024; PwC, 2022). This indicates a wider recognition and acceptance in recent years. Consequently, Bitcoin has become valuable to portfolio diversification strategies (Galaxy Digital Research, 2023). As a result, researchers have begun exploring various forecasting methods to predict Bitcoin's volatility for effective risk management (Oprea, Georgescu, & Bâra, 2024).

Traditional forecasting methods includes autoregressive (AR) models, moving averages (MA), and autoregressive moving averages (ARMA) models. However, these models fail to account for heteroscedasticity in time series. To address this limitation, Bollerslev (1986) extended them to form generalized autoregressive conditional heteroskedastic (GARCH) models. Recent studies have applied these econometric models to Bitcoin forecasting, such as Yildirim and Bekun (2023), who examine the forecast accuracy of Bitcoin using ARCH and GARCH models, and Li, Chen, Xu, and Li (2022), who aim to build an investment trading model for Bitcoin and gold using ARMA models. Additionally, Wirawan, Widiyaningtyas and Hasan

(2019) utilize ARIMA models to predict Bitcoin price changes. Conversely, machine learning models used for forecasting Bitcoin include, but are not limited to, decision trees (Rathan, Sai, & Manikanta, 2019), random forests (Basher & Sadorsky, 2022), gradient boosting machines (Heo, Kwon, Kim, Han, & An, 2018), and recurrent neural networks (Shen, Wan, & Leatham, 2021). D'Ameto, Levantesi, and Piscopo (2022) report promising results using the Jordan Neural Network and suggest further research into Long Short-Term Memory (LSTM) Networks. Based on their recommendation, this paper evaluates the predictive power of LSTM models and compares them to traditional GARCH models.

While traditional methods are effective for traditional assets, they fail to adequately capture the non-linear components inherent in cryptocurrencies. However, by incorporating external factors and having the capability to handle multiple data sources, machine learning algorithms may prove more efficient (Ryll & Seidens, 2019). Consequently, machine learning methods have been introduced as an alternative because they effectively capture non-linearity. This observation forms the basis of my hypothesis that LSTM will outperform GARCH models, shaping the research question of this paper: "*How do LSTM models compare to GARCH models in terms of forecasting Bitcoin return volatility?*".

The possibility of incorporating external factors in predicting Bitcoin volatility has led to further research into the potential value drivers of the cryptocurrency. One of the first papers on the subject is by Kristoufek (2013), who hypothesizes that Bitcoin is solely speculative with no fundamental value. This hypothesis contributes to the common perception that Bitcoin is largely influenced by market sentiment. Estrada (2017) studies this relationship and finds a bidirectional relationship between Bitcoin volatility and the VIX index, which represents the market expectation of future volatility in the S&P 500. This finding is further supported by Gaies, Nakhli, Sahut, and Guesmi (2021), who structure their research around this observation to determine the extent to which Bitcoin is influenced by the macroeconomic climate. An asymmetric positive correlation is observed: positive shocks had a more significant impact in the short term, while adverse shocks were more dominant over the long term. Additionally, several papers find evidence of a correlation between Bitcoin and systematic risk, using S&P 500 as a proxy (Wang, Liu, & Wu, 2022; Baur, Hong, & Lee, 2018). However, Garcia, Tessone,

Mavrodiev, and Perony (2014) depart from Kristoufek's (2013) original viewpoint, arguing that Bitcoin's fundamental value is linked to production costs, using energy prices as a proxy. Finally, Koutmos (2018) indicates a pattern of volatility spillover among cryptocurrencies.

The common denominator among these papers is the focus on individual factors influencing Bitcoin prices and how these elements subsequently affect its volatility. However, to my knowledge, there is currently no research testing the collective predictive power of these variables for forecast accuracy. This paper aims to fill this informational gap by comparing the forecast accuracy of various time-series models, including both econometric models and machine learning algorithms. Building on the previous studies mentioned above that identify key value drivers of Bitcoin, this paper distinguishes itself by analyzing the collective predictive power of fourteen external and internal variables during a sample period between January 1, 2018, and January 1, 2024.

The primary objective of this research is to determine if new algorithms can compete with traditional methods, thereby identifying the most effective approach for forecasting Bitcoin return volatility. Specifically, this study compares the forecast performance of GARCH and LSTM models using the root mean squared error (RMSE) as an evaluation metric. According to the forecast evaluation, the GJR-GARCH (2,2,0) model performed the best among the GARCH models on the validation data. When compared to the LSTM and benchmark models on the test data, the LSTM configuration exhibited the lowest modelling error, establishing it as the superior forecasting method.

2.0 Data Collection and Methodology

2.1 Data Collection

In this paper, most of the models implemented are one-dimensional, requiring only the daily realized volatility of Bitcoin. However, for the multivariate LSTM model, I use additional features as input data. These features include Ethereum, XRP, Bitcoin, Energy Prices, the VIX index, and S&P 500 futures (SPY). The decision to include other cryptocurrencies as features is based on the findings of Ghorbel and Jeribi (2021), which show higher volatility spillover among cryptocurrencies. The selection is narrowed down to Ethereum and XRP due to their comparability in terms of market capitalization. Furthermore, the mining process of Bitcoin requires significant energy levels, which is found to be correlated with Bitcoin prices (Maiti, 2022). Based on Estrada's (2017) findings and the assumption that elevated fear levels in the traditional stock market correlate with increased fear in the cryptocurrency market due to Bitcoin's speculative nature, the VIX index is an effective price driver of Bitcoin. Finally, following Kein, Pham Thu, and Walther's (2018) findings on the significant correlation between Bitcoin and the S&P 500, the time series of S&P 500 futures (SPY) is utilized as a proxy.

I collect the daily Open, High, Low, and Close (OHLC) data and volume of Bitcoin, Ethereum, XRP, VIX index, and SPY from Yahoo Finance, while the global price of energy index is collected from Fred (2024). The sample period is between January 1, 2018, and January 1, 2024. The cryptocurrencies in the dataset are traded seven days a week, while the other features only trade on trading days, which averages 252 days per year. I restrict the time series only to include active trading days to synchronize the dataset. As a result, the time series comprises 1,567 observations per feature.

2.2 Dimension Reduction

However, by involving various external features, the computational resources required by machine learning algorithms increase significantly (Al-Jarrah, Yoo, Muhadat, Karagiannidis, & Taha, 2015). Thus, to facilitate the training of input data and decrease computational costs, I reduce the number of features while trying to retain the most meaningful properties of the

dataset. This process is known as dimension reduction (Barla, 2023). Given that involving all OHLC columns can be redundant due to their similarity, I use these to create two new features: the High-Low spread and the Open-Close spread. These two new features effectively capture the daily movements of the time series while minimizing computational expenses. The mathematical expressions are illustrated in Equations 1 and 2, while the summary statistics of the final selected features are shown in Table 1.

$$\mathrm{HL}_{\mathrm{sprd}} = \log\left(\frac{\mathrm{High} - \mathrm{Low}}{\mathrm{Close}}\right) \tag{1}$$

$$\mathrm{CO}_{\mathrm{sprd}} = \left(\frac{\mathrm{Close} - \mathrm{Open}}{\mathrm{Open}}\right) \tag{2}$$

Feature	**Count**	**Mean**	**STD**	**Min**	**Max**
ETH_HL_sprd	1567	-3,06	0,68	-5,64	-0,29
ETH_CO_sprd	1567	0,001	0,05	-0,42	0,26
ETH_volume	1567	22,7	0,86	20,7	25,2
XRP_HL_sprd	1567	-3,04	0,72	-5,18	0,03
XRP_CO_sprd	1567	0,001	0,057	-0,43	0,73
XRP_volume	1567	21,1	0,94	18,7	24,3
BTC_HL_sprd	1567	-3,39	0,7	-5,63	-0,5
BTC_CO_sprd	1567	0,001	0,04	-0,37	0,19
BTC_volume	1567	23,7	0,8	21,8	26,6
VIX_HL_sprd	1567	-2,4	0,51	-3,97	-0,07
VIX_CO_sprd	1567	-0,006	0,07	-0,3	1,02
Energy Price	1567	175,7	71	55,9	127,6
S&P_Future_HL_Sprd	1567	0,0137	0,011	0,00	0,1259
S&P_Future_CO_Sprd	1567	0,001	0,01	-0,098	0,1096

Table 1: Summary statistics of chosen drivers of Bitcoin volatility. HL_sprd stands for the spread between the series' daily high and low prices, while CO_sprd stands for the spread between daily Close and Open. ETH is short for Ethereum, BTC for Bitcoin, and S&P Future stands for the S&P 500 Future (SPY).

2.3 Stationarity

Time series often exhibit non-stationary characteristics (Cheng, Sa-Ngasoongsong, Beyca, Le, Yang, Kong, & Bukkapatnam, 2015). While neural network models effectively capture non-linearity, econometric models, such as those in the GARCH family, often perform less efficiently. I use the return rather than the price series of Bitcoin to level the playing field, as it has more suitable statistical properties for econometric forecasting (Tsay, 2005). Following the methodology of Shen, Wan, and Leatham (2021), this study utilizes the logarithm of returns, as mathematically expressed in Equation 3:

$$r_{t,t+i} = \log\left(\frac{P_{t+i}}{P_t}\right) \tag{3}$$

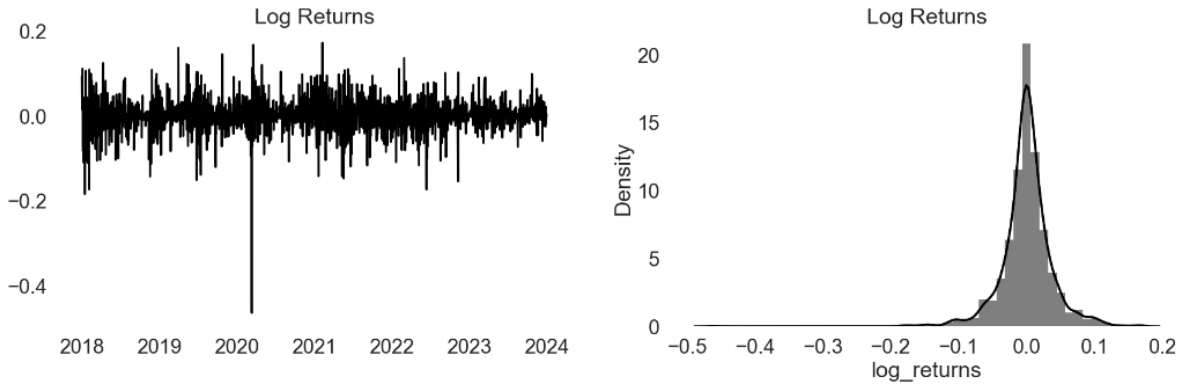

Figure 3: The distribution of returns and the logarithm of the returns of Bitcoin.

Figure 3 indicates a slight negative skewness combined with a positive kurtosis, which is characterized by a higher peak and thicker tails than a normal distribution. The left-hand side of the figure also suggests a stationary series. This is confirmed by an Augmented Dickey-Fuller (ADF) test, with a p-value of 2.82 x 10^{-29}, indicating that the time series is indeed stationary. Given that this paper aims to predict future volatility, an unobservable variable, I use the daily realized volatility as a proxy. The daily returns are transformed into daily realized volatility as illustrated in Equation 4:

$$\sigma_{\text{daily}} = \sqrt{\frac{\sum_{i=1}^{n}(r_i)^2}{n-1}} \tag{4}$$

Where n refers to the lookback window, which is explained in detail in section 2.4.

2.4 Lookback Window and Prediction Horizon

Forecasting, as a technique, involves using historical data to predict future values. The purpose of employing a lookback window is to provide a defined time frame of historical data as a reference point for forecasting future trends (Woo, Liu, Sahoo, Kumar, & Hoi, 2023). Typical scaling frequencies include daily, weekly, monthly, or annually. The frequency selection is based on a trade-off between mitigating potential overfitting issues in shorter time frames and addressing the reduced responsiveness to recent changes in longer intervals. This paper follows the recommendation of Dutta, Kumar, and Basu (2020) to utilize a lookback period of 30 days for both the benchmark and LSTM models while an expanding window forecast is employed for the GARCH models. Since the paper aims to evaluate short-term Bitcoin forecasting, these windowing techniques are used to forecast the upcoming 30 days.

2.5 Normalizing the Dataset

When constructing the portfolio of models, the inputs vary significantly. Therefore, it is advantageous to normalize the features of the dataset. The objective is to transform the numerical data to a min-max scale ranging from 0 to 1, as defined in Equation 5.

$$x' = \frac{x - min}{max - min} \tag{5}$$

The MinMaxScaler is initialized and fitted exclusively to the volatility data from the training set, which is critical to prevent the introduction of a forward-looking bias. Forward-looking bias occurs when a model learns from the distributions of the validation and test data (Harris, 2022). By restricting the scaler fitting to the training data only, I effectively confine the influence of any future data, thus preventing data leakage. The validation and test sets are similarly transformed to ensure consistency in the data presented to the model.

Utilizing a min-max scaler maintains the order of the dataset while achieving a normalized range, thereby facilitating the training process (Patro & Sahu, 2015). This is particularly important as the dataset includes a volume column for the three cryptocurrencies, which is substantially larger than the other features. If left unadjusted, these features would bias the model towards them. Moreover, in neural network applications, employing a scaler mitigates the risk of internal covariate shift, a phenomenon where the weights in the hidden layers are influenced by the distribution of the dataset (Gogia, 2019).

2.6 Spitting the Dataset

The dataset, comprising 1,567 data points, is divided into three groups: the training, the validation, and the test sets. A 365-day period is allocated for both the test and validation sets, and the remaining 837 data points are assigned to the training set. Figure 5 illustrates these three splits. The model is trained using the training set and subsequently validated on the validation set. Upon completion of the training, the test set is applied to evaluate the model's performance on out-of-sample data and assess the practicality of the model (Gillis, n.d.).

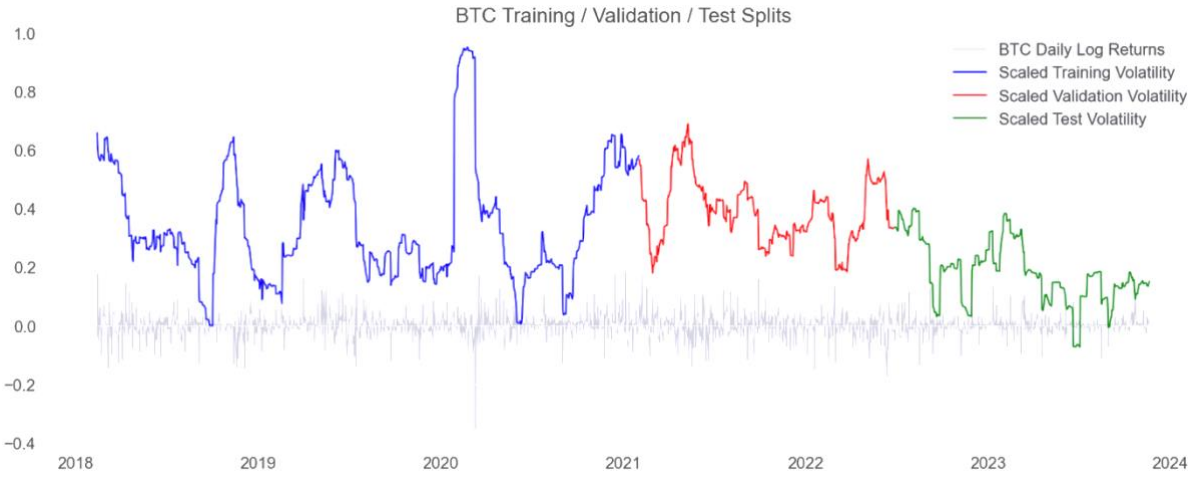

Figure 5: Data split between training-, validation- and test sets.

2.7 Performance Metric

In this paper, I employ RMSE to evaluate the performance of the models. This metric measures the prediction error by squaring the difference between the predicted and actual values. Thus, the lower the score, the more accurate the model. The rationale behind using this metric is that it gives significant weight to large errors (Chai & Draxler, 2014). This is particularly useful for forecasting financial assets, as significant errors imply substantial financial losses.

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

Where y_i are the observed values, $\hat{y}_i$ are the predicted values, and n is the number of observations.

2.8 Benchmark models

It is essential to establish benchmarks when comparing more sophisticated models. In this study, the objective of these benchmark models is to serve as a standard for evaluating the performance of the GARCH and LSTM models (Pappenberger, Ramos, Wetterhall, Alfieri, Bogner, Mueller, & Salamon, 2015). They do so by contextualizing the results, indicating whether significant improvement has been achieved. This helps justify the increased complexity inherent in both GARCH and especially LSTM models.

For this purpose, I select two simple models: the Mean Baseline and a model based on a martingale. The Mean Baseline model is chosen based on the research of Fouque, Papanicolaou, and Sircar (2000), who observe that volatility tends to revert to its mean over time. Conversely, the martingale model is selected following the findings of Cont (2007), who demonstrates that volatility is often autocorrelated and clustered in the short term. Specifically, the Mean Baseline model is designed by averaging all historical data and using this indicator to forecast future values. In contrast, the martingale model is based on a stochastic process where the best predictor of tomorrow's volatility, given the current and historical values, is the volatility of today.

2.9 Econometric Models

The rationale behind choosing GARCH to represent the econometric models circles back to the characteristics of the Bitcoin time series. As depicted in Figure 3 and confirmed by Engle's ARCH test, with a p-value of 4.24×10^{-7} rejecting the null hypothesis, the time series exhibits significant evidence of volatility clustering, a hallmark of heteroscedasticity. Standard ARIMA models are not designed to account for this since they assume constant variance in the residuals. GARCH, developed by Bollerslev (1986) as an extension of ARIMA to purposely handle heteroscedasticity, is therefore very suitable for this study.

Four different variations of GARCH are employed, where the first model is the GARCH (p, q) model. However, a limitation of this model is its inability to account for asymmetries or leverage effects, which are prevalent in time series data (Engle & Ng, 1993). Leverage effects refer to the phenomenon where adverse shocks have a more significant impact on volatility than positive shocks (Corsi & Renó, 2012). Three extensions of the GARCH model designed to capture asymmetry using different approaches are employed to address this.

Firstly, the GJR-GARCH model, developed by Glosten and Runkle (1993), is utilized. This model incorporates a skew-t distribution to account for potential 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 by repeatedly sampling from the observed data. Finally, the Simulation TGARCH model is introduced to determine whether simulation-based approaches are more efficient at capturing the dynamics of the time series compared to the other approaches. The mathematical expressions for the GARCH-type models are shown in Table 2:

Model	**Mathematical expressions**	
GARCH	$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$	(7)
GJR-GARCH & TGARCH	$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{o} \gamma j r_{t-j}^2 \, I[\varepsilon_{t-j} < 0] + \sum_{k=1}^{q} \beta_k \sigma_{t-k}^2$	(8)

Table 2: Mathematical expressions for GARCH model and GJR-GARCH and TGARCH model

The models describe the variance of a time series process as dependent on the entire history of the series. In these models, σ_t^2 is the conditional variance at time t, ω refer to the constant term, and α_i are the coefficients for the lagged squared residuals (ε_{t-i}^2). Moreover, β_j represents the coefficients for the lagged conditional variances (σ_{t-j}^2). What separates the models is the asymmetry term, represented by $\gamma j r_{t-j}^2 I[\varepsilon_{t-j} < 0]$, where γj are the coefficients for the leverage effect terms ($r_{t-j}^2 I[\varepsilon_{t-j} < 0]$), capturing asymmetry.

This paper employs an expanding window forecast in the learning process to ensure that the GARCH models utilize the entire series' history. Unlike a rolling window forecast, which maintains a fixed size, an expanding window forecast gradually increases with each time step, always including all previous observations (Feng, Zhang, & Wang, 2023). Furthermore, the model configuration of each GARCH variant is determined through an iterative process. This process involves testing all combinations of parameters and selecting those that generate the lowest RMSE score. To identify the most effective model structure, the parameters for GARCH range from 0 to 4. For the construction of GJR-GARCH, Bootstrap TGARCH, and Simulation TGARCH models, the number of lagged conditional variances and squared residuals ranges from 0 to 4, and the asymmetry term can be either 0 or 1. The results reveal that the GARCH (2,1) model, GJR-GARCH (2,2,0), Bootstrap TGARCH (3,0,2), and Simulation TGARCH (1,0,2) perform the best. Hence, these model configurations are chosen for further analysis in the paper.

2.10 LSTM Models

The second set of models selected for comparison stems from machine learning. A broad range of algorithms are suitable for time series forecasting practices. Recurrent neural networks (RNN) have shown very promising results (Sako, Mpinda, & Rodrigues, 2022). RNNs are machine-learning processes designed to recognize complex patterns from raw data inputs (Mittal, 2019). Through internal memory, these deep learning algorithms can handle sequential data by processing both current inputs and previously learned information (Shrestha & Mahmood, 2019). However, a shortcoming of RNNs is that they suffer from vanishing and exploding gradients (Mittal, 2019). During the backpropagation process, the gradient is

assigned to apply updates to the weights. A vanishing gradient assigns increasingly smaller updates, hindering the RNN from learning relationships between events that occur far from each other. Conversely, an exploding gradient assigns progressively larger updates, making the training process unstable (Sharma, 2023). Hochreiter and Schmidhuber (1997) introduced LSTM networks to address these issues. Consequently, since this paper deals with long sequences in time series, the LSTM is selected to represent machine learning approach.

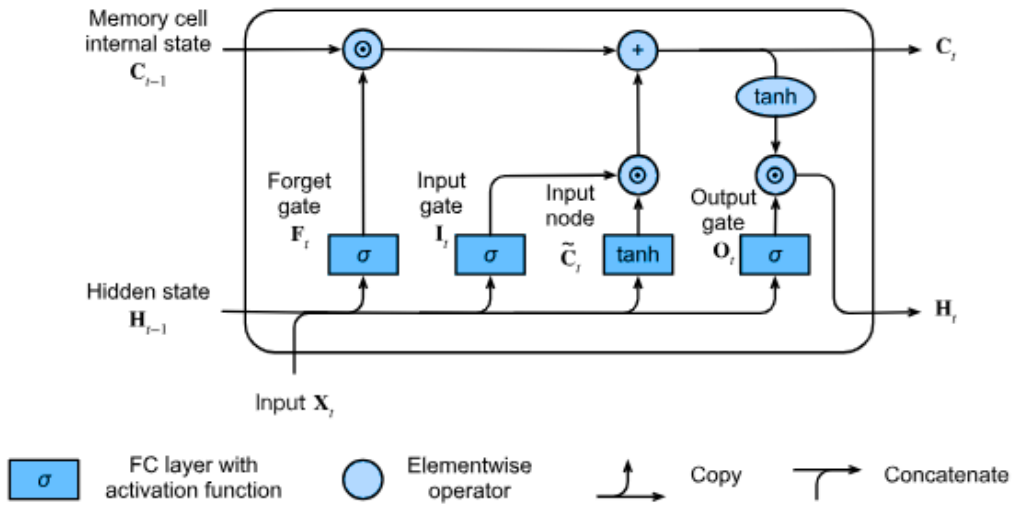

Figure 6: Architecture of Long-Term Short Memory illustrating the flow of information between the three gates (Zhang, Lipton, Li, & Smola, 2021)

Mathematical expression:

$$X = \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} \tag{9}$$

$$f_t = \sigma\left(W_f \cdot X + \beta_f\right) \tag{10}$$

$$i_t = \sigma\left(W_i \cdot X + \beta_i\right) \tag{11}$$

$$o_t = \sigma\left(W_\circ \cdot X + \beta_\circ\right) \tag{12}$$

$$\tilde{C}_t = \tanh\left(W_C \cdot [\, h_{t-1}, x_t \,] + b_C\right) \tag{13}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{14}$$

$$h_t = o_t \odot \tanh\left(C_t\right) \tag{15}$$

A visual illustration of how an LSTM cell operates is shown in Figure 6. The first step of the process involves the algorithm utilizing the previously learned information stored in the cell state (C_{t-1}) and the hidden state (H_{t-1}). By combining these with the current input (X_t), the cell computes the outputs of the forget gate (f_t), input gate (i_t), and output gate (o_t). The forget gate determines which portions of the cell state to retain or discard, while the input gate identifies which new information from the current input is relevant to store in the cell state. In the second step, a new value ($\tilde{c}_t$) for the cell state is created from the current input and the previous hidden state, resulting in an updated cell state. Thirdly, the hidden state is updated based on the new cell state, filtered through the output gate. Finally, the updated states are passed to the subsequent time step in the sequence (Zhang et al., 2021).

2.11 Model Implementation

Several factors are considered when constructing the LSTM model. Due to the complexity and large sequences inherent in LSTM models, all the models are trained using the Adam optimization, which is well-suited for navigating their complex landscapes (Kingma & Ba, 2015). Another factor to consider is overfitting. Overfitting occurs when a model captures the noise in the original dataset, resulting in high efficiency during the training phase but a significant reduction in predictive power when applied to new data (Twin, 2021).

The most important aspect of preventing overfitting is accurately splitting the data into a training and test set, which is addressed in Chapter 2.6. In addition, several other techniques are tested, including early stopping and dropout layers. The early stopping function operates by stopping the training process once the performance on the validation set ceases to improve (Brownlee, 2019). Dropout layers function by omitting nodes at a pre-set probability to avoid co-adaptations in the units, making them more independent (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). This process is illustrated in Figure 7.

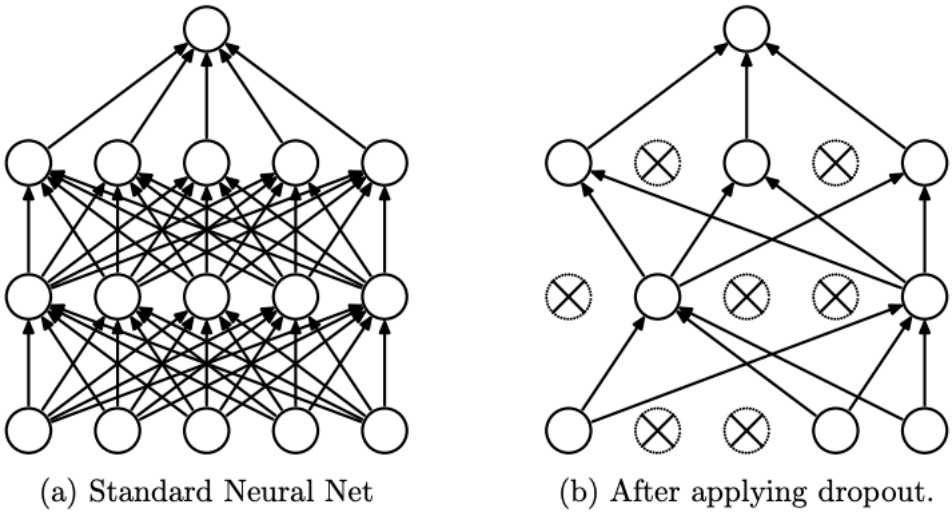

Figure 7: Visual representation of before and after adding dropout layers (Srivastava et al., 2014)

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 (Navas, 2022). According to Dutta, Kumar, and Basu (2020), 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 Dutta, Kumar, and Basu (2020) by using the default value in the Keras package (Chollet, 2015). The hyperparameters and their respective options for this paper are illustrated in Table 3.

Parameter	**Options**
Dropout	0.0, 0.1, 0.2
Activation function	Relu, Tanh, Sigmoid
LSTM Layers	2,3,4
Batch Size	32, 64

Table 3: Variations of LSTM models involved in hyperparameter tuning.

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.

3.0 Results

3.1 Results of the Validation Data

The sample period is divided into three parts: the training set, validation set, and test set. Table 4 demonstrates the RMSE scores for all models on the validation set. The results illustrate that surpassing the baseline models' performance was challenging. Among the baseline models, the Mean Baseline model exhibited significantly higher predictive accuracy with a score of 0.114418, whereas the martingale model performed the worst with a score of 0.173328. The GARCH (2,1) model attained a score of 0.196515, while the GJR-GARCH, Bootstrap TGARCH, and Simulation TGARCH models achieved scores of 0.154005, 0.159141, and 0.162033, respectively. The hyperparameter-tuned LSTM configuration achieved the best result of 0.001259.

Although the results of the validation set are important, they primarily serve to identify the two final models for the test set evaluation. As demonstrated in Table 4, the GJR-GARCH (2,2,0) model achieved the lowest RMSE among the GARCH variants. Therefore, this model is selected to represent the GARCH family and will be compared with the LSTM model on the test set.

	Models	**RMSE on validation data**
1	Mean Baseline	0.114418
2	Martingale	0.173328
3	GARCH (2,1)	0.196515
4	GJR-GARCH (2,2,0)	0.154005
5	Bootstrap TGARCH (3,0,2)	0.159141
6	Simulation TGARCH (3,0,2)	0.162033
7	LSTM	0.001259

Table 4: Volatility Forecasting Models and their respective RMSE score on validation data.

3.2 Results of the Test Data

Table 5 consists of the final models and their respective RMSE scores on the out-of-sample data. The models included are the two benchmark models, GJR-GARCH (2,2,0) and the LSTM model. Both the GJR-GARCH (2,2,0) and the LSTM model outperformed the benchmark models. The Mean Baseline obtained the highest RMSE score with 0.325274, followed by the Martingale, which achieved a score of 0.218204. The GJR-GARCH model achieved a score of 0.131036, while the LSTM model obtained a score of 0.001445. Figure 8 visually illustrates how the two final models performed compared to the actual data. As indicated by the numerical analysis in Table 5 and the graphical depiction in Figure 8, the LSTM model demonstrates the highest prediction accuracy among all the models evaluated.

	Final models	**RMSE on test data**
1	Mean Baseline	0.325274
2	Martingale	0.218204
3	GJR-GARCH (2,2,0)	0.131036
4	LSTM	0.001445

Table 5: Volatility Forecasting Models and their respective RMSE score on test data.

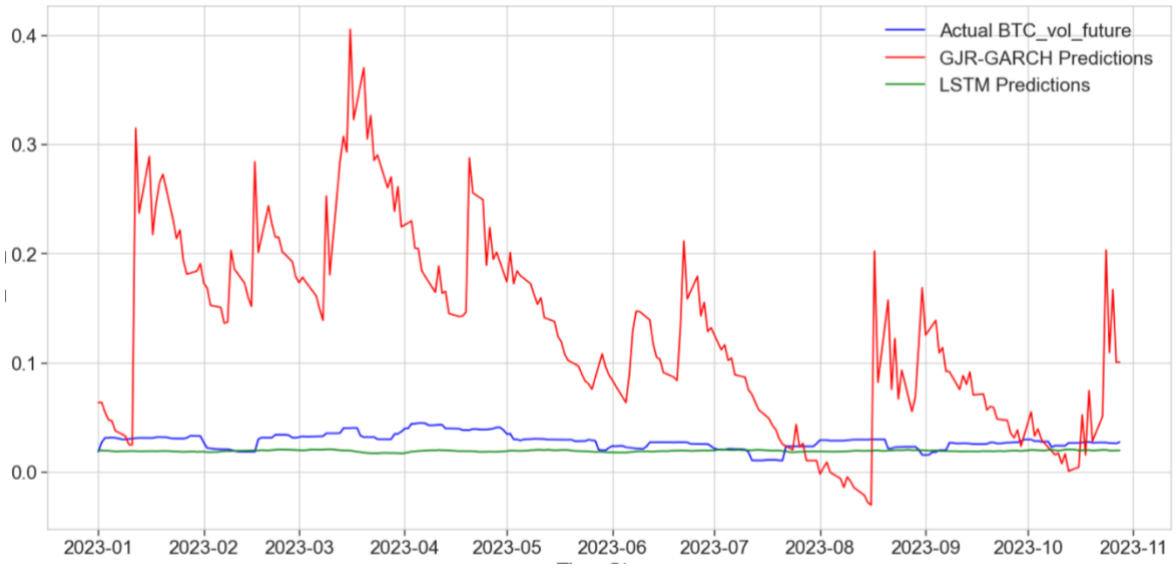

Figure 8: Actual vs. Predicted Bitcoin Values Using GJR_GARCH and LSTM Models.

4.0 Discussion

As for the result of the validation data, the GARCH (2,1) model performs well, as expected. Interestingly, GJR-GARCH, Bootstrap, and Simulation TGARCH perform similarly, indicating no significant difference between the approaches. However, another notable finding is that neither GJR-GARCH nor the two variations of TGARCH performed better with the leverage parameter; all iterations suggested that the modelling error is lower when excluding the leverage effect. These results imply that the time series of Bitcoin does not exhibit significant leverage effects.

In Table 5, both models achieve lower prediction errors than the benchmark models. This indicates that increasing complexity and applying more sophisticated models is worthwhile for improving forecast accuracy. However, the trade-off between computational cost and increased efficiency remains subjective. Furthermore, the Mean Baseline model exhibits the highest RMSE among all models, signifying that using the averages of historical data is a poor measure for future predictions. This may be because Bitcoin is heavily influenced by market sentiment, and without an intrinsic value, using the average of past information will not accurately reflect future value.

The first important finding from the final model selection is that all models, except the GJR-GARCH model, perform worse on the out-of-sample data. This is expected, as the models are trained on the in-sample data and tend to perform better here. In contrast, the out-of-sample data is untouched, leading to more significant prediction errors. However, one possible reason for the improved performance of the GJR-GARCH model is that the patterns captured during the training phase are more representative of the test set than the validation set. Furthermore, the LSTM model significantly outperformed the GARCH model with RMSE scores of 0.001445 and 0.131036, respectively. This suggests that the potential implementation of deep learning algorithms with high-dimensional feature sets in risk management may soon be feasible.

The main reason behind the significant difference between the LSTM model and the remaining models is that the LSTM configuration is trained on multiple features while the other models are univariate. This advantage in the training phase allows the model to learn more complex relationships and ultimately achieve a much lower RMSE score on both the validation and out-of-sample data. Surprisingly, the final configuration of the LSTM excludes the dropout ratio. This suggests that in an attempt to avoid overfitting, the model could potentially underfit instead. The key is to find a balance between these two extremes, which can be more easily identified by allowing multiple options in the hyperparameter tuning process.

However, while LSTM offers lower modelling error, the model has other drawbacks. Firstly, the LSTM model is associated with significantly higher computational costs than the GARCH model. Additionally, the framework of the LSTM model has a 'black box' nature, which complicates the interpretation of each feature's contribution to predicting the return volatility of Bitcoin. As a result, there is a trade-off between the two approaches to forecasting Bitcoin volatility, and the preferred approach will ultimately depend on the prediction objective.

5.0 Limitations and Potential Further Avenues

5.1 Model-related limitations

The first limitation of the paper is the model selection bias, where the GARCH models are uniformly univariate and do not consider external features that the LSTM leverages. Additionally, the hyperparameter options for the LSTM framework are significantly restricted due to the exponentially longer training times associated with each added option. Finally, the feature set is limited to fourteen variables, potentially constraining the LSTM model's ability to predict the return volatility of Bitcoin compared to using a broader set of features. However, this constraint is implemented to facilitate the training process and mitigate the risk of overfitting.

5.2 Data-related limitations

The sample period used in this study is restricted from January 1, 2018, to January 1, 2024. Most existing research uses the entire time series, resulting in a more extensive training set. However, this study starts the analysis from 2018 to exclude data from earlier years, which may be less suitable for predicting future volatility due to significant historical events. These include the launch of Bitcoin options in 2016 (Zhang, Ardern, & Hu, 2022) and the announcement of Bitcoin futures in 2017 (CME Group, 2017), which have caused structural changes in Bitcoin trading dynamics. These factors, combined with the pronounced volatility of the earlier period, make the pre-2018 data less suitable for predicting future volatility. It is worth noting that structural changes in Bitcoin have occurred post-2018; however, there is a trade-off between capturing more recent market behavior and maintaining a sufficiently large dataset for reliable model training.

5.3 Further Avenues

Potential further avenues for this study include addressing these limitations. The result is more insightful if a multivariate GARCH model is introduced so that the training phase between the GARCH and LSTM models is comparable. Secondly, expanding the number of hyperparameters and their respective options would indicate whether the predictive power increases proportionally with the complexity of the LSTM model' architecture. Thirdly,

incorporating a more extensive set of features might further reduce the modelling error. However, this could also induce overfitting. Finally, one could replicate this study using the entire available dataset since the establishment of Bitcoin to evaluate whether it will increase or decrease the modelling error of the models.

6.0 Conclusion

There is a comprehensive strand of literature on forecasting Bitcoin return volatility using econometric and machine learning models (Yildirim & Bekun, 2023; Wirawan, Widiyaningtyas, & Hasan, 2019; Shen, Wan, & Leatham, 2021). However, this paper distinguishes itself by collecting a unique set of features based on previous research evidencing their correlation to Bitcoin (Kristoufek, 2013; Estrada, 2017; Gaies et al., 2021; Wang et al., 2022; Baur et al., 2018; Garcia et al., 2014; Koutmos, 2018). Specifically, this paper analyzes 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. This is subsequently evaluated on the out-of-sample data, where the final LSTM model achieves an RMSE score of 0.001445. In contrast, the final GARCH model obtains a score of 0.131036, indicating that the LSTM model not only compares to GARCH models but also outperforms them.

7.0 References

Al-Jarrah, O. Y., Yoo, P. D., Muhadat, S., Karagiannidis, G. K., & Taha, K. (2015). Efficient machine learning for big data: A review. *Big Data Research, 2*(3), 87-93.

Barla, N. (2023). Dimensionality Reduction for Machine Learning. *MLOps Blog*. Available online: https://neptune.ai/blog/dimensionality-reduction [Accessed April 25, 2024]

Basher, S. A., & Sadorsky, P. (2022). Forecasting Bitcoin price direction with random forests: How important are interest rates, inflation, and market volatility? *Machine Learning with Applications, p. 9*, 100355.

Baur, D. G., Hong, K., & Lee, A. D. (2018). Bitcoin: Medium of exchange or speculative assets? *Journal of International Financial Markets, Institutions and Money, 54*, 177-189.

Bollerslev, T. (1986). A conditionally heteroskedastic time series model for speculative prices and rates of return. *The Review of Economics and Statistics*, 542-547.

Bouri, E., Molnár, P., Azzi, G., Roubaud, D., & Hagfors, L. I. (2017). On the hedge and haven properties of Bitcoin: Is it really more than a diversifier? *Finance Research Letters, 20*, 192-198.

Brownlee, J. (2019). A gentle introduction to early stopping to avoid overtraining neural networks. *Machine Learning Mastery*. Available online: https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/ [Accessed April 23, 2024]

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE). *Geoscientific Model Development Discussions, 7*(1), 1525-1534.

Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics*, *365*, 112395.

Cheng, C., Sa-Ngasoongsong, A., Beyca, O., Le, T., Yang, H., Kong, Z., & Bukkapatnam, S. T. (2015). Time series forecasting for nonlinear and non-stationary processes: a review and comparative study. *Iie Transactions*, *47*(10), 1053-1071.

Chollet, F. (2015). Keras: Deep Learning for humans. Available online: https://github.com/keras-team/keras [Accessed May 8, 2024]

CME Group. (2017). CME Group announces launch of Bitcoin futures [Press release]. CME Group. Available online: https://www.cmegroup.com/media-room/press-releases/2017/10/31/cme_group_announceslaunchofbitcoinfutures.html [Accessed April 14, 2024]

Cont, R. (2007). Volatility clustering in financial markets: empirical facts and agent-based models. In *Long memory in economics* (pp. 289-309). Berlin, Heidelberg: Springer Berlin Heidelberg.

Corsi, F., & Renó, R. (2012). Discrete-time volatility forecasting with persistent leverage effect and the link with continuous-time volatility modeling. *Journal of Business & Economic Statistics, 30*(3), 368-380.

D'Amato, V., Levantesi, S., & Piscopo, G. (2022). Deep learning in predicting cryptocurrency volatility. *Physica A: Statistical Mechanics and its Applications*, *596*, 127158.

Dutta, A., Kumar, S., & Basu, M. (2020). A gated recurrent unit approach to bitcoin price prediction. *Journal of risk and financial management*, *13*(2), 23.

Engle, R. F., & Ng, V. K. (1993). Measuring and Testing the Impact of New on Volatility. *The Journal of Finance, 48*(5), 1749-1778.

Feng, Y., Zhang, Y., & Wang, Y. (2023). Out-of-sample volatility prediction: Rolling window, expanding window, or both? *Journal of Forecasting,* 43(3), 567-582.

Fouque, J. P., Papanicolaou, G., & Sircar, K. R. (2000). Mean-reverting stochastic volatility. *International Journal of Theoretical and Applied Finance, 3*(01), 101-142.

FRED. (2024). Global price of Energy index (PNRGINDEXM) [Data set]. Available online: https://fred.stlouisfed.org/series/PNRGINDEXM[Accessed April 4, 2024]

Gaies, B., Nakhli, M. S., Sahut, J. M., & Guesmi, K. (2021). Is Bitcoin rooted in confidence?–Unraveling the determinants of globalized digital currencies. *Technological Forecasting and Social Change*, *172*, 121038.

Galaxy Digital Research. (2023). Bitcoin in a Portfolio: The Impact and Opportunity. Available online: https://www.galaxy.com/insights/research/bitcoin-in-a-portfolio-impact-and-opportunity/[Accessed April 2, 2024]

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.

Ghorbel, A., & Jeribi, A. (2021). Investigating the relationship between volatilities of cryptocurrencies and other financial assets. *Decisions in Economics and Finance*, *44*(2), 817-843.

Gillis, A. S. (n.d.). Data splitting. In *TechTarget*. Available online: https://www.techtarget.com/searchenterpriseai/definition/data-splitting[Accessed April 17, 2024]

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.

Gogia, N. (2019). Why scaling is important in machine learning? *Analytics Vidhya*. Available online: https://medium.com/analytics-vidhya/why-scaling-is-important-in-machine-learning-aee5781d161a[Accessed April 7, 2024]

Harris, M. (2022). Look-Ahead Bias In Backtests And How To Detect It. *Medium*. Available online: https://mikeharrisny.medium.com/look-ahead-bias-in-backtests-and-how-to-detect-it-ad5e42d97879[Accessed May 7, 2024]

Heo, J. S., Kwon, D. H., Kim, J. B., Han, Y. H., & An, C. H. (2018). Prediction of cryptocurrency price trend using gradient boosting. *KIPS Transactions on Software and Data Engineering, 7*(10), 387-396.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780.

Howarth, J. (2024). How many Cryptocurrencies are There in 2024? Available online: https://explodingtopics.com/blog/number-of-cryptocurrencies [Accessed April 2, 2024]

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint* arXiv:1412.6980.

Koutmos, D. (2018). Return and volatility spillovers among cryptocurrencies. *Economics Letters, 173*, 122-127.

Kristoufek, L. (2013). BitCoin meets Google Trends and Wikipedia: Quantifying the relationship between phenomena of the Internet era. *Scientific reports*, *3*(1), 3415.

Li, R., Chen, W., Xu, W., & Li, C. (2022). Prediction on the Value Trends of Bitcoin and Gold on Account of ARMA Time Series Forecasting Model. *Academic Journal of Computing & Information Science, 5*(7), 79-84.

Maiti, M. (2022). Dynamics of Bitcoin prices and energy consumptions. *Chaos, Solitons & Fractals: X, 9*, 100086.

Mittal, A. (2019). Understanding RNN and LSTM. *Medium*. Available online: https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e[Accessed May 6, 2024]

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Navas, J. (2022). What is hyperparameter tuning? *Anyscale*. Available online: https://www.anyscale.com/blog/what-is-hyperparameter-tuning [Accessed May 16, 2024]

Oprea, S. V., Georgescu, I. A., & Bâra, A. (2024). Is Bitcoin ready to be a widespread payment method? Using price volatility and setting strategies for merchants. *Electronic Commerce Research*, 1-39.

Pappenberger, F., Ramos, M. H., Wetterhall, F., Alfieri, L., Bogner, K., Mueller, A., & Salamon, P. (2015). How do I know if my forecasts are better? Using benchmarks in hydrological ensemble prediction. *Journal of Hydrology, 522*, 697-713.

Patro, S. G. O. P. A. L., & Sahu, K. K. (2015). Normalization: A preprocessing stage. *arXiv preprint* arXiv:1503.06462.

PwC. (2022). PwC Global Crypto Hedge Fund Report 2022. Available online: https://www.pwc.com/id/en/media-centre/press-release/2022/english/pwc-global-crypto-hedge-fund-report-2022.html [Accessed April 3, 2024]

Rathan, K., Sai, S. V., & Manikanta, T. S. (2019). Crypto-Currency price prediction using Decision Tree and Regression techniques. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 190-194). Tirunelveli, India: IEEE.

Royal, J. (2024). Crypto vs. stocks. *Bankrate*. Available online: https://www.bankrate.com/investing/crypto-vs-stocks/[Accessed April 2, 2024]

Sako, K., Mpinda, B. N., & Rodrigues, P. C. (2022). Neural networks for financial time series forecasting. *Entropy, 24*(5), 657.

Sharma, K. (2023). Vanishing/Exploding Gradients Problem. *Medium.* Available online: https://medium.com/@kushansharma1/vanishing-exploding-gradients-problem-1901bb2db2b2 [Accessed May 8, 2024]

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.

Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE access*, *7*, 53040-53065.

Estrada, J. C. S. (2017). Analyzing bitcoin price volatility. *University of California, Berkeley*.

Srivastata, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research,* 15(1), 1929-1958.

Ryll, L., & Seidens, S. (2019). Evaluating the performance of machine learning algorithms in financial market forecasting: A comprehensive survey. *arXiv preprint arXiv:1906.07786*.

Tsay, R. S. (2005). *Analysis of Financial Time Series*. John Wiley & Sons.

Twin, A. (2021). Understanding Overfitting and How to Prevent it. *Investopedia*. Available online: https://www.investopedia.com/terms/o/overfitting.asp [Accessed May 6, 2024]

Wang, P., Liu, X., & Wu, S. (2022). Dynamic Linkage between Bitcoin and Traditional Financial Assets: A Comparative Analysis of Different Time Frequencies. *Entropy*, 24(11), 1565.

Wirawan, I. M., Widiyaningtyas, T., & Hasan, M. M. (2019). 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.

Woo, G., Liu, C., Sahoo, D., Kumar, A., & Hoi, S. (2023, July). Learning deep time-index models for time series forecasting. In *International Conference on Machine Learning* (pp. 37217-37237). PMLR.

Yildirim, H., & Bekun, F. V. (2023). Predicting volatility of bitcoin return with ARCH, GARCH and EGARCH models. *Future Business Journal*, 9(1), 75.

York, B. (2024). 2024 Institutional Crypto Hedge Fund & Venture Report. *VisionTrack*. Available online: https://assets.ctfassets.net/yksdf0mjii3y/2kHBAm1V6MA1NRAHzG2xws/e8b626b82d27824ad83df6076e3393b2/GLXY_2024_Whitepaper_VisionTrack-2024InstitutionalCrypto_final.pdf[Accessed April 5, 2024]

Zakoian, J. M. (1994). Threshold heteroskedastic models. *Journal of Economic Dynamics and Control,* 18(5), 931-955.

Zhang, A., Ardern, G., & Hu, M. (2022). Crypto Options Market: History, Present, and Future. March 2022.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.