Risk Assessment and Management Decisions

www.ramd.reapress.com

Risk Assess. Manage. Decis.Vol. 1, No. 1 (2024) 62–74.

Paper Type: Original Article

Optimizing GARCH Models for Financial Volatility

EbereChukwu Q. Chinedu [1] and Okechukwu J. Obulezi[2,*]

[1]Department of Statistics, Faculty of Physical Sciences, Nnamdi Azikiwe University, P.O. Box 5025, Awka, Nigeria email: eq.chinedu@stu.unizik.edu.ng;, [2]Department of Statistics, Faculty of Physical Sciences, Nnamdi Azikiwe University, P.O. Box 5025, Awka, Nigeria email: oj.obulezi@unizik.edu.ng;

Citation:

Received: 14 March 2024 Revised: 21 May 2024 Accepted: 02/08/2024	Chinedu, E. Q., & Obulezi, O. J. (2024). Optimizing GARCH Models for Financial Volatility. Risk Assessment and Management Decisions, 1(1), 62-74

Abstract

This paper delves into the intricate process of refining GARCH (Generalized Autoregressive Conditional Heteroskedasticity) model parameters for precise financial volatility forecasting. Leveraging data from yfinance, traditional approaches using autocorrelation function (ACF) and partial autocorrelation function (PACF) plots were employed. Optimal values derived from visual diagnostics did not yield significant parameters. So we proceeded to set both autoregressive order (p) and moving average order (q) to 1 produced the most favorable AIC and BIC metrics. Furthermore, the model, refined through this process, was seamlessly transitioned into a user-friendly web application for enhanced accessibility and practical implementation by financial analysts.

Keywords: Cryptocurrency, Volatility Forecasting, GARCH Model, Financial Modeling, Time Series Analysis, Financial Markets, Statistical Modelling.

1|Introduction

Financial markets are inherently dynamic, characterized by fluctuations and volatility that pose challenges for risk management and decision-making. Financial volatility forecasting plays a crucial role in various financial activities, including risk management, portfolio optimization, and investment decisions. Numerous models have

Corresponding Author: Okechukwu J. Obulezi

been developed to capture and predict the time-varying nature of financial asset returns, with the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model being one of the most widely used [1].

The GARCH model addresses the issue of heteroskedasticity, where the variance of the error term in a time series is not constant, but rather conditional on past variances and shocks [2]. This characteristic makes it suitable for modeling financial data, where volatility often exhibits clustering and persistence [3].

Determining the optimal parameters for a GARCH model is crucial for achieving accurate and reliable forecasts. Traditional approaches often rely on visual inspection of autocorrelation function (ACF) and partial autocorrelation function (PACF) plots [4]. However, this method can be subjective and prone to bias [5].

Several studies have explored alternative methods for parameter estimation. For instance, [6] proposes a grid search algorithm to efficiently identify optimal GARCH parameters, while [7] utilizes Bayesian techniques to incorporate prior information about the model parameters.

Additionally, recent research has focused on the development of user-friendly interfaces and web applications that facilitate the application of GARCH models for financial analysts and practitioners [8]. These tools aim to increase accessibility and reduce the technical expertise required for utilizing sophisticated forecasting models.

This paper builds upon existing research [10, 11, 12, 13, 14, 15, 16, 17, 18, 19] by exploring an alternative approach for GARCH parameter estimation and its subsequent integration into a web application. The effectiveness of this approach will be evaluated through empirical analysis and compared to traditional methods.

2|Methodology

In this paper, we focused on an index from the cryptocurrency market: Bitcoin(BTC-USD). This focus, coupled with a 10-year lookback period aims to capture and predict the volatility of the most popular digital asset.

2.1|GARCH

The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model is a statistical model used to analyze and forecast financial volatility over time. Developed by Tim Bollerslev in the 1980s, the GARCH model is an extension of the ARCH (Autoregressive Conditional Heteroskedasticity) model and is particularly valuable in capturing the time-varying nature of volatility observed in financial time series data.

The key idea behind the GARCH model is that financial asset returns exhibit volatility clustering, meaning periods of high volatility tend to follow each other, and periods of low volatility are clustered together as well. The model captures this clustering by incorporating past squared returns (volatility) as a predictor of future volatility. The autoregressive component accounts for the persistence of volatility patterns.

The GARCH (p, q) model, where "p" represents the autoregressive order and "q" represents the moving average order, can be defined as follows:

The GARCH (p, q) model equation is expressed as:

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

where:

σ_t^2 is the conditional variance of the financial asset's return at time t.

ω is the constant term, representing the long-term average of the conditional variance.

α_i are the autoregressive parameters for the squared returns at different lags (ε_{t-i}^2), capturing the persistence of volatility.

β_j are the moving average parameters for the conditional variances at different lags (σ_{t-j}^2), representing the impact of past conditional variances on the current conditional variance.

ε_{t-i} is the standardized residual at time, t, based on the observed return and the estimated conditional variance.

2.2|Parameter Selection

In the context of a GARCH model, the autoregressive order (p) and moving average order (q) are crucial parameters influencing the model's ability to capture and predict volatility dynamics. The determination of these orders often involves analyzing autocorrelation and partial autocorrelation functions.

(1) Autoregressive Order (p):

- The autoregressive order (p) represents the number of past squared returns that significantly contribute to the current volatility.
- A common criterion is to observe the lag at which the partial autocorrelation function (PACF) sharply drops or becomes negligible after initially being significant. This suggests the optimal p value.

(2) Moving Average Order (q):

- The moving average order (q) indicates the number of past conditional variances that influence the current volatility.
- A standard approach involves examining the autocorrelation function (ACF) plot. The q value is often chosen based on the lag where the autocorrelation drops sharply or becomes negligible after an initial significant value.

These criteria aim to strike a balance between capturing relevant information from past observations and avoiding overfitting the model to noise. The goal is to select values of p and q that provide a parsimonious yet effective representation of the underlying volatility structure in the financial time series data

We utilized the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots to inform the selection of optimal values for the autoregressive order (p) and moving average order (q) parameters in our GARCH model.

2.3|Model Selection/Assessment

In the process of model selection, we considered various combinations of autoregressive order (p) and moving average order (q) for our GARCH model. The optimal model will be determined based on the criteria of minimizing both the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) as well as the significance of the parameters of the model, with significance of parameters given higher importance.

2.4|Web Application Development

The plotly dash library was used to deploy this model to the web.

3|Analysis

Before running the model on the data obtained from yfinance, we ran some some Exploratory Data Analysis (EDA) on the dataset in order to understand the nature of the data.

3.1|Exploratory Data Analysis (EDA)

- Distribution (1): This is the shape this digital asset has taken on over the course of 10 years.s
- Conditional volatility (2): This is, in essence, looking at the volatility of an asset with respect to time.
- Unonditional volatility(3): This is looking at the volatility of an asset divorced from time.

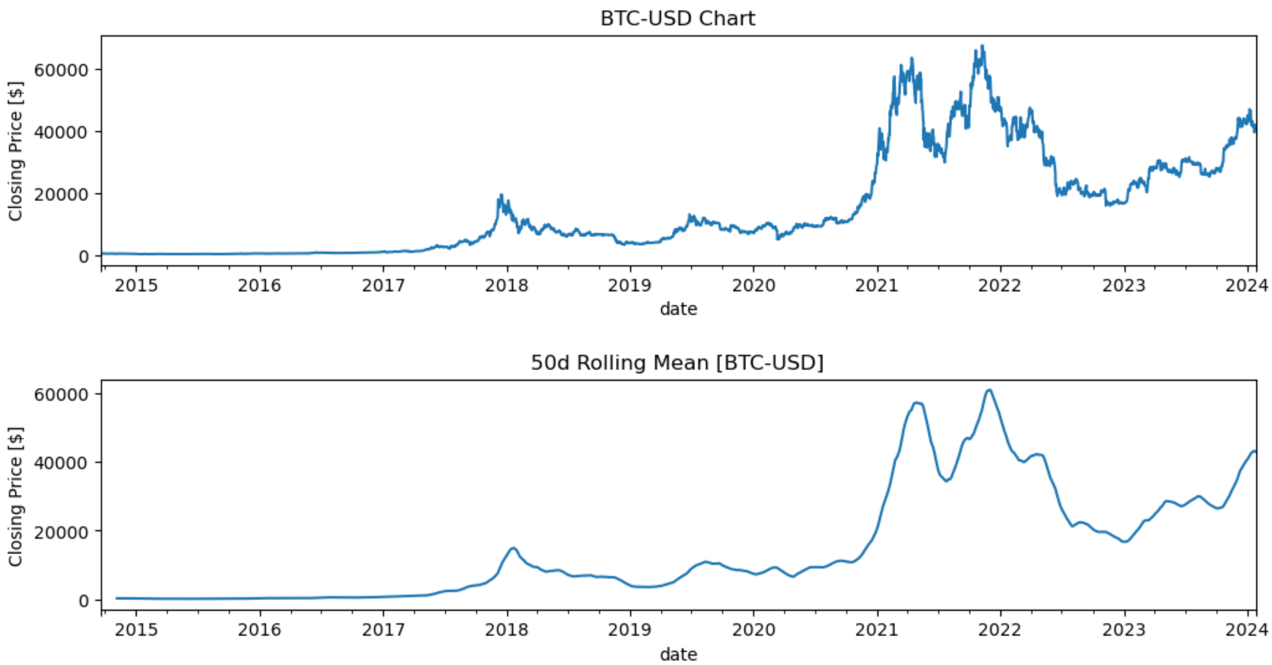

Figure 1. Bitcoin Chart

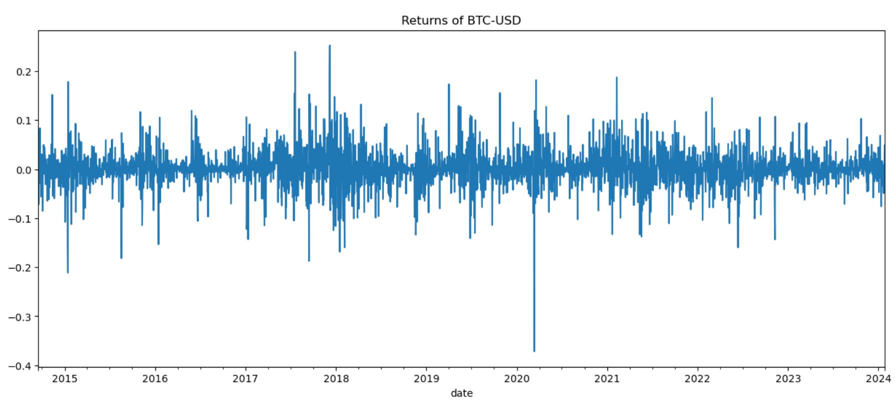

Figure 2. Conditional Volatility

- Rolling volatility (4): A 50-day volatility, which was used in this plot, captures the volatility of the asset at rolling windows of 50-days.

3.2|Determination of lags

- Squared returns (5): This enables us to see the overall volatility.
- ACF plot (6): A graphical representation of the correlation of a time series with itself at different lag
- PACF plot (7): A graphical representation of the correlation of a time series with itself at different lags, after removing the effects of the previous lags

From the ACF and PACF plot, it can be seen that the top two significant lags are at time step 1 and 7. Hence, we'll proceed to build a two GARCH model with the following parameters for p and q:

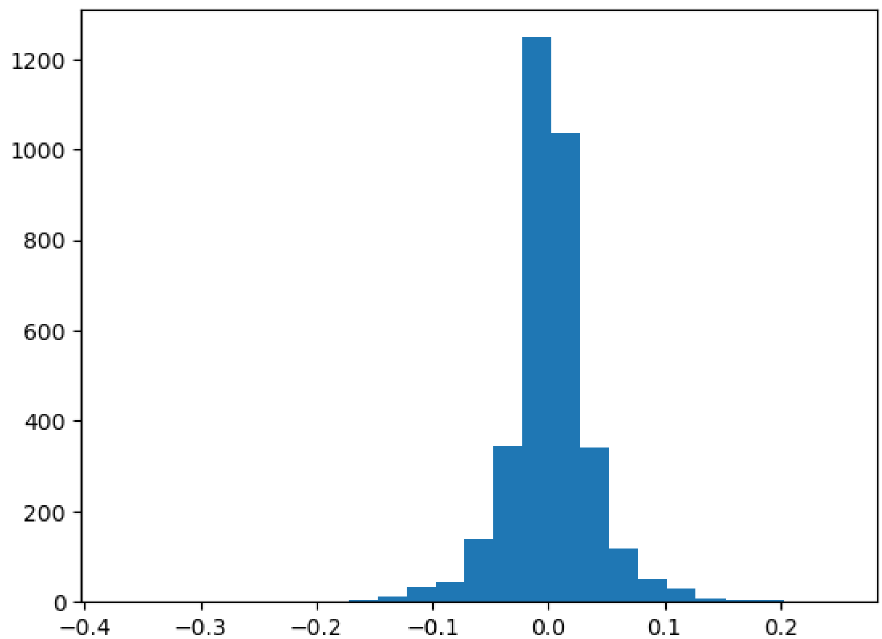

FIGURE 3. Unconditional Volatility

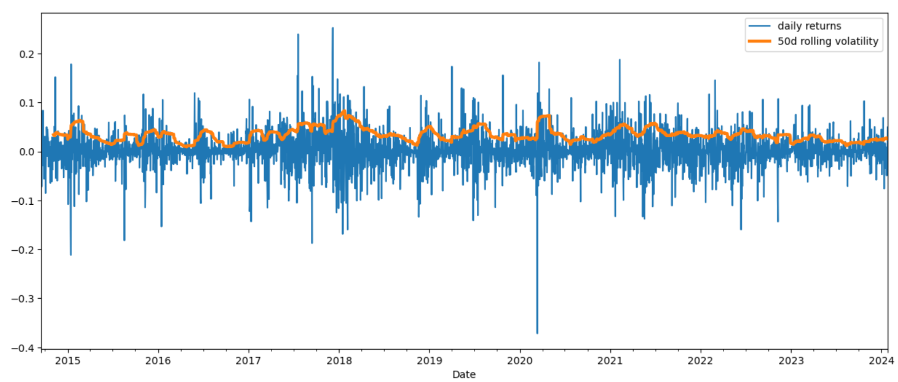

FIGURE 4. Rolling Volatility

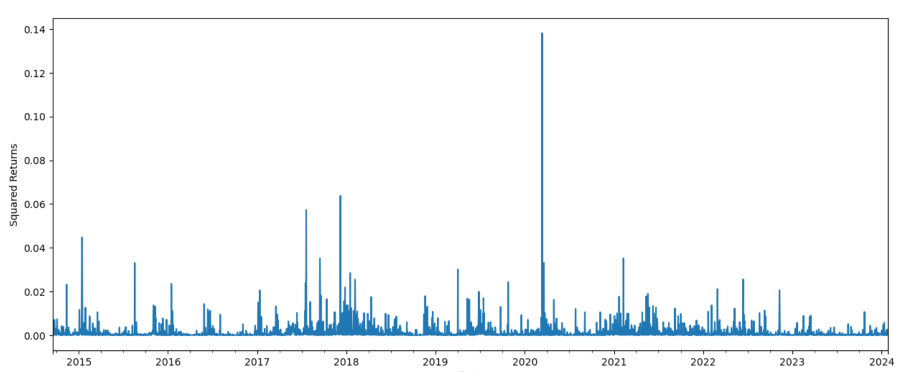

FIGURE 5. Squared Returns

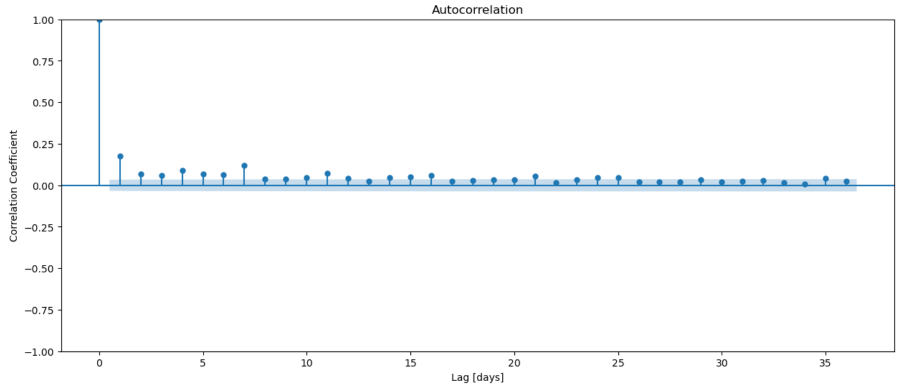

FIGURE 6. ACF

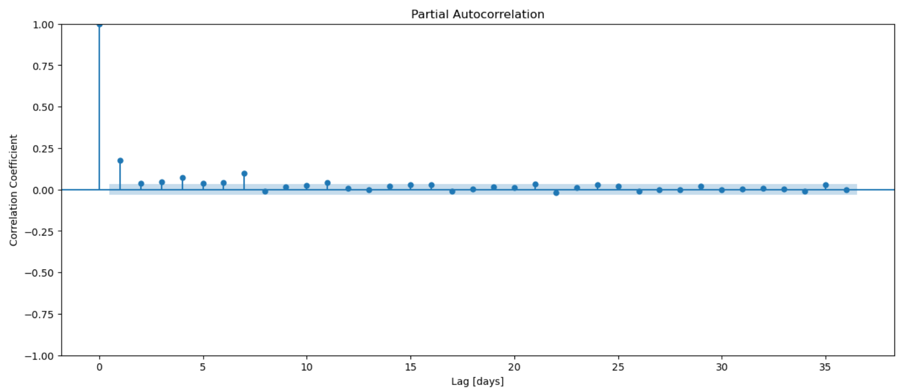

FIGURE 7. PACF

(1) p = 7, q = 7
(2) p = 1, q = 1

3.3|Model Building

The dataset was first split to obtain the training set which is 80% of the dataset. No test set was created because this is a time series data. Instead, walk-forward validation will be performed on the remaining 20% of the data.

The code below shows how the splitting was done

```
# Determine cutoff point
cutoff_test = int(0.8 * len(returns))
# Split data at cutoff mark
y_train = returns[:cutoff_test]

print("y_train type:", type(y_train))
print("y_train shape:", y_train.shape)
y_train.tail()
```

Then we built the first GARCH model setting $p = 7$ and $q = 7$ using the code below:

```
# Build and train model
model = arch_model(
```

```
    y_train ,
    p = 7,
    q = 7,
    rescale = False
).fit(disp = 0)
print("model type:", type(model))

# Show model summary
model.summary()
```

This was the result obtained from this model:

Constant Mean - GARCH Model Results

Dep. Variable:	close	**R-squared:**	0.000
Mean Model:	Constant Mean	**Adj. R-squared:**	0.000
Vol Model:	GARCH	**Log-Likelihood:**	3868.08
Distribution:	Normal	**AIC:**	-7704.15
Method:	Maximum Likelihood	**BIC:**	-7614.54
		No. Observations:	2000
Date:	Thu, Jan 25 2024	**Df Residuals:**	1999
Time:	08:10:31	**Df Model:**	1

FIGURE 8. Result showing AIC and BIC

- An AIC value of -7704.15 and a BIC score of -7614
- From 9, it can be clearly seen that of all 15 parameters of the model none of them are significant. Hence, this model isn't good for making volatility prediction.

Then next model was built setting $p = 1$ and $q = 1$ using the code below:

```
# Build and train model
model = arch_model(
    y_train ,
    p = 1,
    q = 1,
    rescale = False
).fit(disp = 0)
print("model type:", type(model))

# Show model summary
model.summary()
```

The result from this model is shown below:

Here [10], the AIC and BIC scores are higher than in the previous model but unlike the previous model, all the parameters of this model are all significant. Hence, this model will be used to conduct the walk-forward validation.

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	1.4635e-04	2.423e-04	0.604	0.546	[-3.285e-04,6.212e-04]
alpha[1]	0.0286	3.346e-02	0.854	0.393	[-3.701e-02,9.415e-02]
alpha[2]	0.0286	0.112	0.255	0.799	[-0.191, 0.248]
alpha[3]	0.0286	0.152	0.187	0.851	[-0.270, 0.327]
alpha[4]	0.0286	0.119	0.241	0.810	[-0.204, 0.261]
alpha[5]	0.0286	9.451e-02	0.302	0.762	[-0.157, 0.214]
alpha[6]	0.0286	0.161	0.177	0.859	[-0.288, 0.345]
alpha[7]	0.0286	0.231	0.124	0.902	[-0.424, 0.481]
beta[1]	0.1000	1.502	6.658e-02	0.947	[-2.844, 3.044]
beta[2]	0.1000	2.250	4.445e-02	0.965	[-4.309, 4.509]
beta[3]	0.1000	3.193	3.132e-02	0.975	[-6.158, 6.358]
beta[4]	0.1000	2.078	4.812e-02	0.962	[-3.973, 4.173]
beta[5]	0.1000	2.594	3.855e-02	0.969	[-4.984, 5.184]
beta[6]	0.1000	1.159	8.630e-02	0.931	[-2.171, 2.371]
beta[7]	0.1000	4.576	2.185e-02	0.983	[-8.868, 9.068]

FIGURE 9. Result showing the significance of the model's parameters

3.4|Model Evaluation and Walk-forward Validation

3.4.1|Model Evaluation

We first evaluate the model by creating a time series plot with the returns and the conditional volatility obtained from the model.

The code below was used to generate the plot displayed above

```
# Plot `y_ambuja_train`
y_train.plot(ax = ax, label = f"{ticker} Daily Returns")

# Plot conditional volatility * 2
(model.conditional_volatility * 2).plot(ax = ax, label = "2SD Conditional Volatility", colo

# Plot conditional volatility * -2
(model.conditional_volatility.rename("") * -2).plot(ax = ax, color = "C1")


# Add axis labels
```

Constant Mean - GARCH Model Results

Dep. Variable:	close	**R-squared:**	0.000
Mean Model:	Constant Mean	**Adj. R-squared:**	0.000
Vol Model:	GARCH	**Log-Likelihood:**	5206.86
Distribution:	Normal	**AIC:**	-10405.7
Method:	Maximum Likelihood	**BIC:**	-10382.1
		No. Observations:	2737
Date:	Tue, Jan 30 2024	**Df Residuals:**	2736
Time:	02:36:52	**Df Model:**	1

Mean Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	2.2666e-03	7.229e-04	3.136	1.715e-03	[8.498e-04,3.683e-03]

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	3.0262e-05	3.621e-06	8.358	6.365e-17	[2.317e-05,3.736e-05]
alpha[1]	0.1000	2.962e-02	3.376	7.354e-04	[4.194e-02, 0.158]
beta[1]	0.8800	2.620e-02	33.582	3.038e-247	[0.829, 0.931]

Figure 10. Result showing the significance of the model's parameters

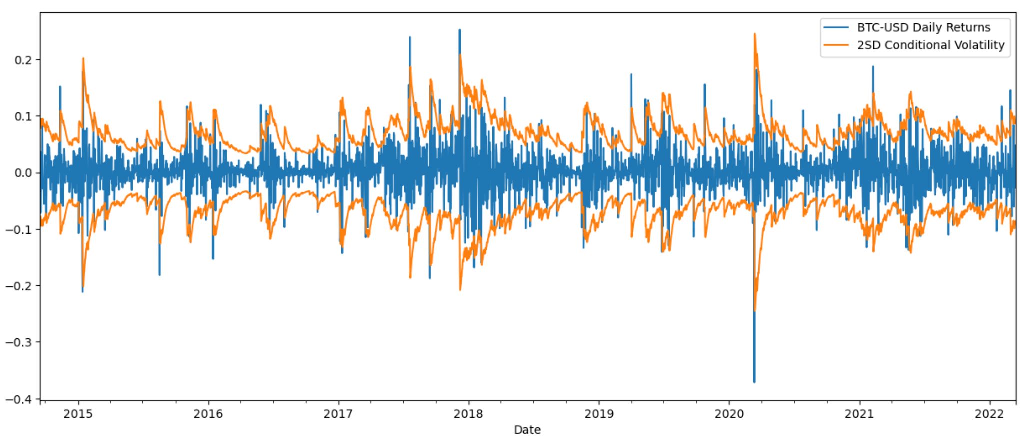

Figure 11. Conditional volatility of model fit to daily returns of the asset

```
plt.xlabel("Date")


# Add legend
```

```
plt.legend();
```

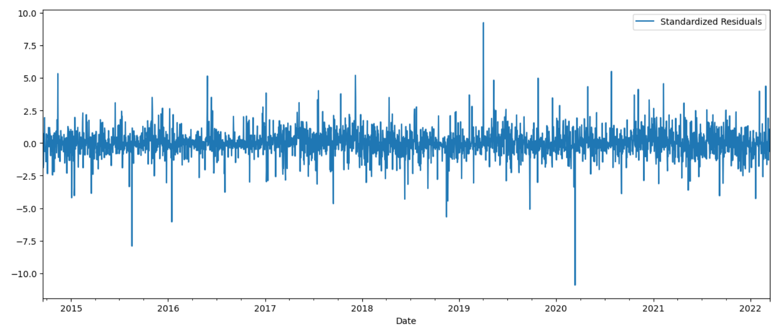

FIGURE 12. Time series plot of model's residuals

A histogram of the standardized residuals of the model demonstrates a bell-shaped distribution centered around zero, indicating that the residuals exhibit a pattern consistent with a normal distribution. This suggests that the model adequately captures the underlying patterns in the data, and the residuals display the expected behavior under the assumption of normality. Such normality in the distribution of residuals is a desirable characteristic in statistical modeling as it validates the model's assumptions and enhances the reliability of its results.

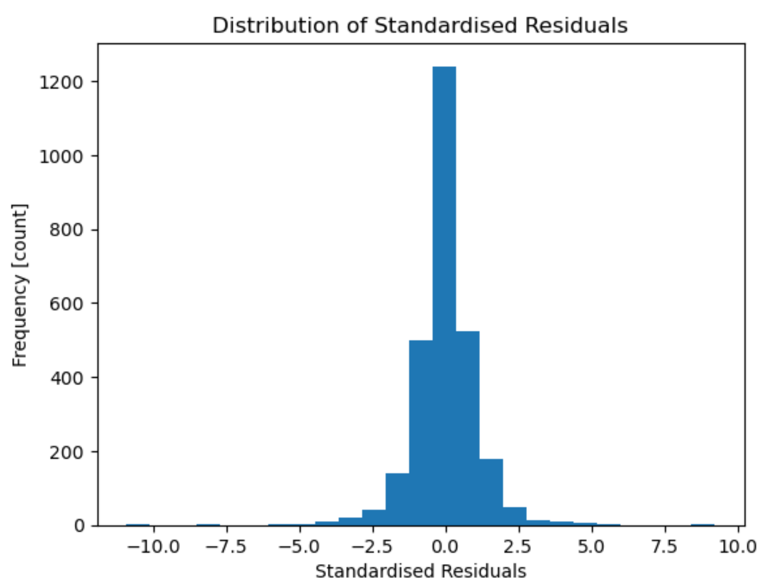

FIGURE 13. Distribution of model's residuals (Histogram)

A look at the ACF plot of the model's residual shows that there is no correlation between the current time step and previous time steps, which is what we want to see.

3.4.2|Walk-forward Validation

The code for the walk-forward validation is shown below:

```
# Create empty list to hold predictions
predictions = []

# Calculate size of test data (20%)
```

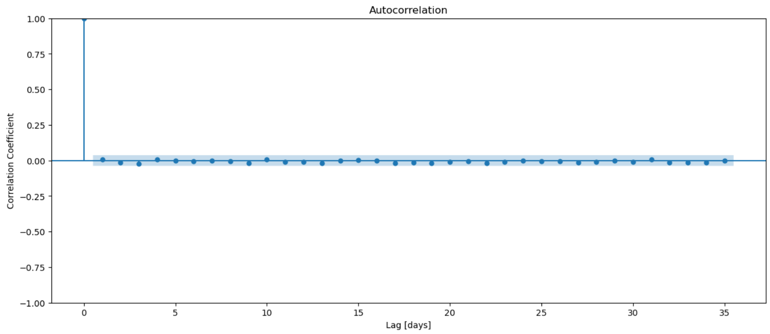

FIGURE 14. Distribution of model's residuals (ACF)

```
test_size = int(len(returns) * 0.2)

# Walk forward
for i in range(test_size):
    # Create test data
    y_train = returns.iloc[: -(test_size - i)]

    # Train model
    model = arch_model(y_train, p=1,q=1, rescale=False).fit(disp =0)

    # Generate next prediction (volatility, not variance)
    next_pred = model.forecast(horizon= 1, reindex = False).variance.iloc[0,0] ** 0.5

    # Append prediction to list
    predictions.append(next_pred)

# Create Series from predictions list
y_test_wfv = pd.Series(predictions, index=returns.tail(test_size).index)

print("y_test_wfv type:", type(y_test_wfv))
print("y_test_wfv shape:", y_test_wfv.shape)
y_test_wfv.head()
```

The code below produces an output that shows how well the model captures the volatility of unseen data.

```
# Plot returns for test data
returns.tail(test_size).plot(ax=ax, label=f"{ticker} Return")

# Plot volatility predictions * 2
(2 * y_test_wfv).plot(ax=ax, c="C1", label="2 SD Predicted Volatility")

# Plot volatility predictions * -2
(-2 * y_test_wfv).plot(ax=ax, c="C1")

# Label axes
plt.xlabel("Date")
plt.ylabel("Return")
```

```
# Add legend
plt.legend();
```

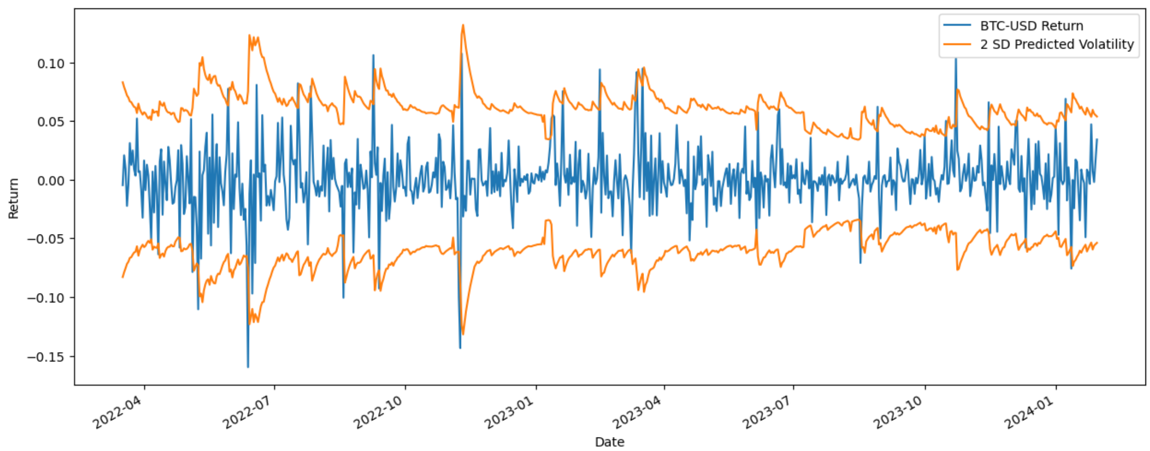

FIGURE 15. Model fits unseen data

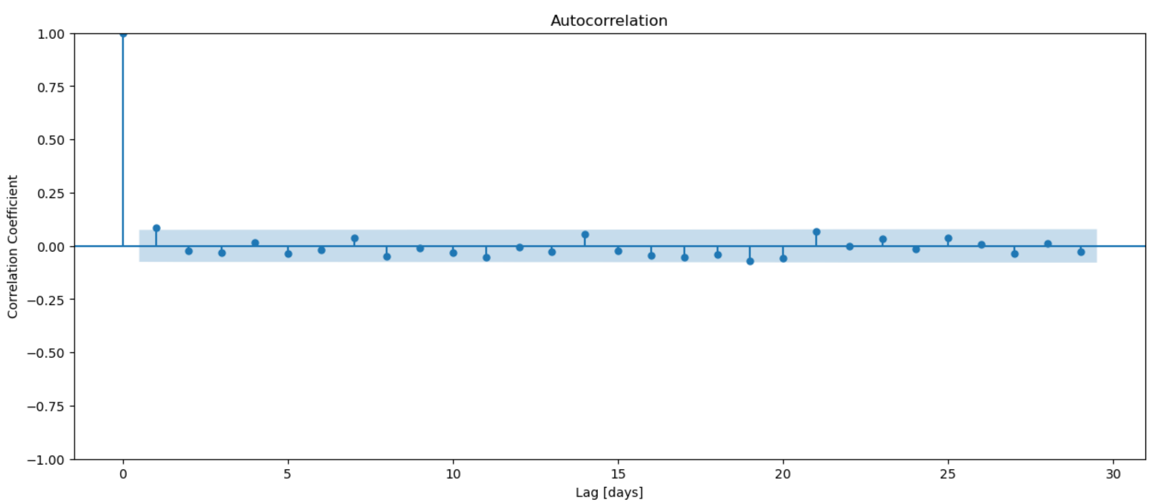

FIGURE 16. ACF plot of model fit on unseen data

An ACF plot (16) showing that the nature in which the model fits the data exhibits no "echo". That is, there is no correlation between the current timestamp and previous timestamps. That is exactly what we would expect to see in a good model. Hence, the model fits the unseen data appropriately.

5|Conclusion

To summarise, the GARCH model was used to predict the volatility of a popular digital asset—Bitcoin (BTC-USD)—, over the course of 10 years. Several visuals were created to gain a proper and indepth understanding of the distribution and volatility of the asset. A PACF and ACF were plotted to determine the most appropriate values for the parameters of the GARCH model. The model with the most significant parameters, while taking the AIC and BIC into consideration, was selected and used for predicing the volatility of the asset.

This model picked was converted to a user-friendly website for easy interaction.

Acknowledgments

The authors would like to express their sincere gratitude to the editors and anonymous reviewers for their invaluable comments and constructive feedback, which significantly contributed to the enhancement of this paper.

Author Contribution

Chinedu, E. Q., Obulezi, O. J.: methodology, software, and editing. Chinedu, E. Q.: conceptualization. Obulezi, O .J.: writing and editing. All authors have read and agreed to the published version of the manuscript.

Funding

The authors declare that no external funding or support was received for the research presented in this paper, including administrative, technical, or in-kind contributions. Data Availability All data supporting the reported findings in this research paper are provided within the manuscript.

Conflicts of Interest

The authors declare that there is no conflict of interest concerning the reported research findings. Funders played no role in the study's design, in the collection, analysis, or interpretation of the data, in the writing of the manuscript, or in the decision to publish the results.

References

[1] Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica: Journal of the Econometric Society*, 50(4), 987–1007.

[2] Bollerslev, T. (2008). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 144(1), 316–333.

[3] McNeely, A. J., & Engle, R. F. (2005). The GARCH model for financial time series forecasts. *Journal of Economic Education*, 36(4), 384–411.

[4] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time series analysis, forecasting and control. John Wiley & Sons.

[5] Hyndman, R. J., & Athanasopoulos, G. (2013). Forecasting: principles and practice. OTexts.

[6] Doornik, J. A., & Hansen, E. (2000). Efficient estimation of GARCH models with dynamic conditional variances: A user's guide. Retrieved from https://arxiv.org/pdf/2109.01044

[7] Koop, G. (2003). Bayesian econometrics. John Wiley & Sons.

[8] Pfaff, M. (2020). Financial econometrics using Python: A practical guide. Apress.

[9] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877–1901.

[10] Bauwens, L., Laurent, S., & Rombouts, J. V. K. (2006). Multivariate GARCH models: a survey. *Journal of Applied Econometrics*, 21(1), 79–109.

[11] Lundbergh, S., & Teräsvirta, T. (2002). Evaluating GARCH models. *Journal of Econometrics*, 110(2), 417–435.

[12] Christoffersen, P., & Jacobs, K. (2004). Which GARCH model for option valuation? *Management Science*, 50(9), 1204–1221.

[13] Francq, C., & Zakoian, J. M. (2019). GARCH models: structure, statistical inference and financial applications. John Wiley & Sons.

[14] Van der Weide, R. (2002). GO-GARCH: a multivariate generalized orthogonal GARCH model. *Journal of Applied Econometrics*, 17(5), 549–564.

[15] Van der Weide, R. (2002). GO-GARCH: a multivariate generalized orthogonal GARCH model. *Journal of Applied Econometrics*, 17(5), 549–564.

[16] Van der Weide, R. (2002). GO-GARCH: a multivariate generalized orthogonal GARCH model. *Journal of Applied Econometrics*, 17(5), 549–564.

[17] Vrontos, I. D., Dellaportas, P., & Politis, D. N. (2003). A full-factor multivariate GARCH model. *The Econometrics Journal*, 6(2), 312–334.

[18] Ling, S., & McAleer, M. (2003). Asymptotic theory for a vector ARMA-GARCH model. *Econometric Theory*, 19(2), 280–310.

[19] Ghahramani, M., & Thavaneswaran, A. (2008). A note on GARCH model identification. *Computers & Mathematics with Applications*, 55(11), 2469–2475.