

How to forecast volatility of crypto derivatives when data history (implied vol.) is limited? Would the traditional econometric models work?

Ki-Hyeon Kim

Finance, Illinois Institute of Technology

December 1, 2024

Abstract

This study examines the effectiveness of traditional econometric models, such as ARMA, GARCH, and EGARCH, in forecasting the implied volatility of crypto derivatives when data history is limited. We compare these models with machine learning approaches, including LASSO and Gradient Boosting, which utilize additional features like returns and rolling volatility. The results show that simpler models like ARMA and LASSO outperform GARCH-based models, which struggle with limited data. Machine learning models demonstrate competitive performance, highlighting their adaptability to the unique characteristics of cryptocurrency markets. These findings provide valuable insights into selecting appropriate forecasting methods for crypto derivatives under data constraints.

1 Introduction

In recent years, cryptocurrency has rapidly grown from a niche digital asset into a major area of interest within the financial markets. Originally valued for its decentralized nature and potential as a new form of currency, cryptocurrency now attracts investors, institutions, and regulatory bodies worldwide. This shift has positioned cryptocurrency as a unique and important asset class, raising new questions about its impact on traditional financial markets and global investment strategies.

As interest in cryptocurrency has expanded, so too has the development of financial products based on it. One key area of growth is crypto derivatives—contracts such as futures and options that are based on the underlying value of a cryptocurrency like Bitcoin. These products allow investors to manage risks and speculate on the price of cryptocurrency in the future. However,

while crypto derivatives offer a powerful way to engage with the cryptocurrency market, they also present unique challenges. Unlike traditional assets, which have long historical records of price and volatility data, cryptocurrencies are relatively new and lack extensive historical data. This limited data history, particularly for implied volatility, makes it challenging to accurately price crypto derivatives using traditional methods.

The shortage of reliable data isn't the only issue. There is also a lack of well-established, universally accepted pricing models tailored specifically to cryptocurrency derivatives. Traditional econometric models, such as GARCH-based models commonly used in pricing options for traditional assets, may not fully capture the unique dynamics of cryptocurrency markets. Volatility in cryptocurrency is often highly irregular, marked by sudden price spikes and significant market swings, which may be hard to model accurately with approaches originally designed for more stable assets.

Despite these challenges, recent studies and research papers have explored various methods and theories for forecasting cryptocurrency volatility. Scholars have examined everything from adaptations of traditional models to entirely new approaches in an effort to better understand and predict crypto derivative prices. These studies highlight the need for further exploration, especially as the demand for crypto derivatives continues to grow.

Given these limitations and ongoing research, this paper addresses a fundamental question: With the limited data history available, can traditional econometric models like GARCH effectively forecast the volatility of crypto derivatives? Or do we need specialized models to handle the unique and highly volatile nature of cryptocurrency markets? This study seeks to analyze the potential and limitations of traditional models in this context, providing insights into whether they can serve as reliable tools for crypto volatility forecasting or if alternative approaches are necessary. To explore the question of whether traditional econometric models like GARCH can be effective in forecasting the volatility of crypto derivatives, we can look at prior studies that have examined similar models in both traditional and cryptocurrency markets.

Zhang and Zhang (2020) suggested improvements to GARCH option pricing models specifically to better capture the Volatility Risk Premium (VRP). They introduced a modified Local Risk-Neutral Valuation Relationship (mLRNVR), which allows the model to account for differences between real and risk-neutral volatility measurements. Their results showed that this adjusted GARCH model aligns well with the CBOE VIX index, providing more accurate prices for options and volatility swaps than traditional GARCH models by better capturing VRP. This approach to enhancing GARCH models highlights how modifying these models can improve accuracy in option pricing.

In a related study, Wang et al. (2016) focused on the Heston–Nandi GARCH model to price CBOE VIX futures accurately. Their main goal was to see if this model could effectively capture VIX futures prices and VRP. They tested different data sources and found that using both VIX and VIX futures data together gave the best results, as this approach improved the model's prediction accuracy both within the sample and with new data. When they only used VIX

futures data, parameter distortion occurred, making it difficult to capture VIX’s full movement. This study emphasizes the importance of using a well-rounded data set to improve model predictions.

In the cryptocurrency space, Venter and Maré (2020) studied how to calculate volatility indices for Bitcoin and CRIX (a cryptocurrency index) using GARCH models. They explored whether GARCH models could produce reliable volatility indices for different time periods (30-day, 60-day, and 90-day) and whether these indices would respond similarly to market shocks. Their findings revealed that both Bitcoin and CRIX volatility indices reacted in similar ways to market shocks, with short-term volatility (30 days) generally being lower than long-term volatility (60 and 90 days). However, during major market changes, 30-day volatility rose sharply, reflecting immediate market conditions. This study shows how GARCH models can be adapted to capture the unique characteristics of cryptocurrency volatility.

These studies suggest that GARCH models can be effective in capturing volatility under certain conditions. However, they also highlight the need for adjustments and the importance of data selection, particularly when applying these models in unique markets like cryptocurrency, where historical data is limited. This research will build on these findings by exploring how well traditional GARCH models work in forecasting the volatility of crypto derivatives when data history is limited and whether additional adjustments or new approaches might be needed.

Our study explores whether traditional econometric models, such as ARMA, GARCH, and EGARCH, can forecast implied volatility in crypto derivatives when data history is limited. Additionally, we compare these models with machine learning approaches, which can incorporate a broader range of features and adapt to complex, nonlinear relationships.

The results of our analysis reveal three important findings. First, simple models like ARMA perform surprisingly well with limited data. In fact, ARMA outperformed both GARCH and EGARCH models, demonstrating that simpler methods may be better suited for short-term volatility predictions in data-constrained environments. Second, feature-based machine learning models, such as LASSO and Linear Regression, performed as well as or better than ARMA. These models leverage additional features, such as returns and rolling volatility, to improve prediction accuracy. Lastly, GARCH and EGARCH, which are designed to capture long-term volatility clustering, struggled in this setting. This suggests that these models require more extensive historical data to deliver reliable forecasts.

These findings highlight the potential of simpler models and machine learning approaches for predicting crypto derivative volatility when data is scarce. As the demand for accurate pricing and risk management tools in cryptocurrency markets grows, understanding the limitations and strengths of different forecasting methods becomes increasingly important. This study contributes to the ongoing conversation by providing insights into the effectiveness of traditional and modern models in this challenging context.

2 Material and Methods

2.1 Data

For this study, we used data from the ProShares Bitcoin Strategy ETF (BITO), obtained from Yahoo Finance. The dataset includes daily closing prices, which we used to calculate returns and rolling volatility as features. To simulate implied volatility for analysis, we generated synthetic data within a realistic range, representing the challenges of limited historical data for crypto derivatives. The dataset covers a two-year period, providing enough data for training and testing while reflecting the short history of cryptocurrency markets. These features allowed us to compare traditional econometric models and machine learning approaches in predicting implied volatility.

Traditional Model

2.2 Conditional Mean

The ARMA (p, q) model is used to describe the conditional mean as:

$$\mu_t = \phi_0 + \sum_{i=1}^p \phi_i r_{t-i} + \sum_{j=1}^q \theta_j Z_{t-j}$$

After fitting the ARMA (p, q) model, we calculate the parameters and the leftover errors (called residuals). Bitcoin daily returns often show groups of high and low volatility, which suggests that the volatility changes over time (this is called "conditional heteroscedasticity"). To check this, we test if the residuals have patterns like ARCH (Autoregressive Conditional Heteroscedasticity). If ARCH effects are found, we will explain the models for conditional variance.

2.3 GARCH Model

Bollerslev developed the generalized autoregressive conditional heteroscedasticity (GARCH) model in 1986. Both the ARCH and GARCH models describe how financial asset volatility changes over time. The GARCH model improves upon ARCH by allowing the conditional variance to behave like an ARMA process. The GARCH process is defined as follows:

$$Z_t = h_t \epsilon_t, \quad \{\epsilon_t\} \sim \text{IID}(0, 1) \quad (6)$$

$$h_t^2 = \alpha_0 + \sum_{i=1}^m \alpha_i Z_{t-i}^2 + \sum_{j=1}^n \beta_j h_{t-j}^2 \quad (7)$$

where $\{Z_t\}$ is the residual series of the best-fitting ARMA (p, q) model. The conditional variance of the residual series behaves like an ARMA process.

It is expected that the standardized squared residuals from the fitted ARMA-GARCH model are not autocorrelated and show no remaining ARCH effects. The ARCH LM test checks if this condition holds true.

2.4 Model Selection

All GARCH-type models were estimated using the maximum likelihood method. Since the fitted models are not nested, different criteria were used to compare and select the best model:

- The Akaike Information Criterion (AIC), introduced by Akaike(1974), is defined as:

$$\text{AIC} = 2k - 2 \ln L(\hat{\Theta}), \quad (1)$$

where k is the number of unknown parameters, Θ represents the vector of parameters, and $L(\hat{\Theta})$ is the maximum likelihood estimate.

- The Bayesian Information Criterion (BIC), proposed by Schwarz(1978), is defined as:

$$\text{BIC} = k \ln n - 2 \ln L(\hat{\Theta}), \quad (2)$$

where n is the number of observations.

2.5 EGARCH

The conditional mean of financial returns can be written as:

$$y_t = E(y_t | I_{t-1}) + \epsilon_t$$

where:

- $y_t = \Delta \log P_t$: Represents the log-difference in stock prices (P_t).
- I_{t-1} : Information set available at time $t - 1$.
- ϵ_t : Represents the error term, which exhibits conditional heteroscedasticity.

The EGARCH specification proposed by Nelson is given as:

$$\log h_t = \omega + \alpha |\eta_{t-1}| + \gamma \eta_{t-1} + \beta \log h_{t-1}, \quad |\beta| < 1$$

where:

- h_t : The conditional variance at time t , expressed in logarithmic form.
- $\eta_t = \epsilon_t / \sqrt{h_t}$: The standardized shocks, which follow $iid(0, \omega)$.
- $|\beta| < 1$: Ensures the stability of the model.

The inclusion of $\log h_{t-1}$ ensures the model's stability. Asymmetry in the returns is captured when $\gamma \neq 0$, while leverage effects exist when $\gamma < 0$ and $-\gamma < \alpha < \gamma$. The specification in Equation (2) is for EGARCH(1,1), but this can be extended to EGARCH(p,q).

As η_{t-1} depends on both ϵ_t and $\sqrt{h_t}$, estimating this model using quasi-maximum likelihood is challenging. The absolute value of $|\eta_{t-1}|$ is not differentiable with respect to the parameters, and the use of logarithms and absolute values makes the invertibility of the model problematic.

Machine Learning Model

2.6 Lasso Regression

Lasso regression selects variables by solving the following optimization problem:

$$\hat{\beta}_{\text{lasso}} = \arg \min_{\beta} \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

Here:

- $\|Y - X\beta\|_2^2$: Represents the sum of squared residuals.
- $\|\beta\|_1 = \sum_{j=1}^p |\beta_j|$: The ℓ_1 -norm of the coefficients, which adds a penalty for non-zero coefficients.
- $\lambda \geq 0$: A tuning parameter controlling the balance between model fit and sparsity. A larger λ results in more coefficients being shrunk to zero.

The ℓ_1 -penalty ensures sparsity by setting some coefficients to zero, which helps in variable selection. Lasso is computationally efficient and well-suited for cases where the number of predictors (p) exceeds the number of observations (n).

2.7 K-Nearest Neighbors Regressor (KNN):

KNN regression predicts the output for new data by averaging the outputs of the K -nearest neighbors in the training dataset. The neighbors are determined using a chosen distance metric, such as Euclidean distance. KNN is simple and effective for non-linear relationships in the data.

2.8 Decision Tree Regressor:

The Decision Tree Regressor is based on the CART (Classification and Regression Trees) algorithm. It splits the data into regions by creating binary splits based on input features. The final prediction is the average output within the region corresponding to the input.

2.9 Random Forest Regressor:

Random Forest is an ensemble method that builds multiple decision trees and combines their predictions. Each tree is trained on a random subset of the data, and the final prediction is the average of the outputs from all trees. This approach reduces overfitting and improves generalization.

2.10 Extra Trees Regressor (Extremely Randomized Trees):

Extra Trees is similar to Random Forest but uses the entire dataset without bootstrapping. Node splits are chosen randomly rather than being optimized. This method reduces computational cost and variance compared to Random Forest.

2.11 AdaBoost Regressor:

AdaBoost is a sequential ensemble method where each model corrects the errors of the previous model by adjusting weights for mispredicted data points. The final prediction is a weighted average of all models, focusing on improving difficult predictions.

2.12 Gradient Boosting Regressor:

Gradient Boosting minimizes residual errors by sequentially training models to predict the differences between predicted and actual values. It combines these models iteratively using gradient descent to optimize the error function, making it effective for complex datasets.

Error Metrics for Regression Models

2.13 Mean Square Error (MSE)

The Mean Square Error (MSE) is calculated as:

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^m (X_i - Y_i)^2$$

where m is the total number of data points, X_i represents the predicted values, and Y_i represents the actual values. The best value for MSE is 0, and the worst value is ∞ .

MSE is particularly useful for identifying outliers, as the ℓ_2 -norm used in its calculation assigns larger weights to larger errors. This magnification makes MSE sensitive to significant deviations, which can help detect poorly predicted data points. Additionally, MSE is closely related to the coefficient of determination (R^2) through the relationship $R^2 = 1 - \frac{\text{MSE}}{\text{MST}}$, where MST is the mean square total. This negative monotonic relationship implies that models ordered by MSE or R^2 will follow the same ranking (though in reverse order for R^2).

2.14 Root Mean Square Error (RMSE)

The Root Mean Square Error (RMSE) is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^m (X_i - Y_i)^2}$$

Like MSE, the best value for RMSE is 0, and the worst value is ∞ .

RMSE is directly related to MSE as it is the square root of MSE. The ranking of models based on RMSE will be identical to the ranking based on MSE, as they are monotonically related. RMSE provides an interpretable measure of error in the same units as the target variable, making it a preferred metric when clear interpretability is needed.

2.15 Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) is computed as:

$$\text{MAE} = \frac{1}{m} \sum_{i=1}^m |X_i - Y_i|$$

where X_i and Y_i denote the predicted and actual values, respectively. The best value for MAE is 0, and the worst value is ∞ .

MAE is less sensitive to outliers compared to MSE because it uses the ℓ_1 -norm, which treats all deviations linearly. This property makes MAE a suitable metric for datasets with corrupted or noisy data, as it avoids over-penalizing large errors. Unlike MSE, MAE provides a more robust estimate of typical model error when extreme outliers are present.

3 Results

Descriptive Statistics of Options Data

The descriptive statistics for the Call and Put options of the BITO ETF, based on the first expiration date, are presented in Tables 1 and 2. These statistics summarize key characteristics, including strike prices, last prices, implied volatility, and trading volume. The average strike price for call options is 22.58, with a maximum of 34 and a minimum of 12.50. The implied volatility for call options shows a mean of 3.35, ranging from 0.33 to 9.41, indicating significant variability. Trading volumes for call options vary widely, with a standard deviation of 2199.70, reflecting differing levels of market interest across strike prices.

Similarly, the put options exhibit an average strike price of 21.76, with implied volatility averaging 2.56 and trading volumes showing lower variability compared to calls. These statistics highlight the differences in trading activity

and price behavior between calls and puts, providing a basis for further analysis of the BITO options market.

Table 1: Statistics for Call Options

Statistic	Strike Price	Last Price	Implied Volatility	Volume
Mean	22.576923	4.305128	3.347859	846.918919
Median	22.500000	3.690000	2.734378	33.000000
Standard Deviation	6.000590	3.897683	2.677728	2199.699946
Max	34.000000	11.850000	9.406254	8766.000000
Min	12.500000	0.010000	0.328132	1.000000

Table 2: Statistics for Put Options

Statistic	Strike Price	Last Price	Implied Volatility	Volume
Mean	21.756757	0.933243	2.563454	255.029412
Median	21.500000	0.010000	2.187505	6.500000
Standard Deviation	5.861387	2.409759	1.902024	663.367693
Max	34.000000	10.100000	7.171876	3298.000000
Min	12.500000	0.010000	0.000010	1.000000

Traditional model Results and Analysis

Building on this, Table 3 presents the performance metrics for the ARMA, GARCH, and EGARCH models, which were used to forecast volatility. The metrics include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). Among the models, the ARMA model demonstrates the best performance, achieving the lowest error values across all three metrics. This suggests that ARMA is more effective in capturing the observed volatility. In comparison, the GARCH and EGARCH models exhibit similar performance, with slightly higher error values, indicating a less accurate fit to the observed data. Based on the evaluation of model selection criteria, the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) were used to compare various GARCH models. Both AIC and BIC help identify the model that balances goodness-of-fit and complexity, penalizing excessive parameters. After computing these criteria for different GARCH configurations, the GARCH(1,1) model was chosen as it provided the lowest AIC and BIC values. This indicates that GARCH(1,1) effectively captures the volatility dynamics of the data while maintaining a parsimonious structure. Hence, it was selected as the optimal model for further analysis.

Figure 1 visually compares the implied volatility from Yahoo Finance with the volatility forecasts generated by the ARMA, GARCH, and EGARCH models. The ARMA model aligns closely with the implied volatility trend, particu-

larly during periods of stable market conditions, which explains its superior error metrics in Table 3. Both the GARCH and EGARCH models capture the general volatility patterns but show slight deviations from the implied volatility during periods of extreme market movements, such as the initial spike. The EGARCH model demonstrates greater sensitivity to large market shocks, reflected in its higher peaks compared to the other models. Overall, the combination of Table 3 and Figure 1 highlights the relative strengths and weaknesses of each model in forecasting volatility under varying market conditions.

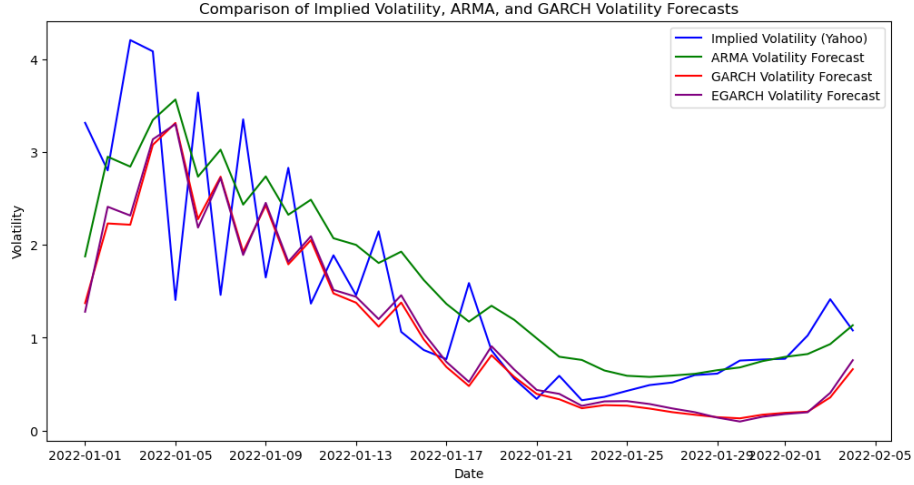


Figure 1: Comparison of implied volatility from Yahoo Finance with ARMA, GARCH, and EGARCH volatility forecasts.

Machine Learning Model Results and Analysis

To evaluate the effectiveness of machine learning models in predicting the implied volatility of crypto derivatives, we compared several algorithms, including Linear Regression (LR), LASSO, K-Nearest Neighbors (KNN), Decision Trees (CART), Support Vector Regression (SVR), Multilayer Perceptron (MLP), and ensemble methods such as AdaBoost (ABR), Gradient Boosting (GBR), Random Forest (RFR), and Extra Trees (ETR). The performance of each model was assessed using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). These metrics allow us to compare the accuracy of predictions across different models under the context of limited data history.

The results are summarized in Table 3. The best-performing model was LASSO, with the lowest MSE (0.009911), RMSE (0.099556), and MAE (0.082046). Linear Regression (LR) also showed strong performance, with slightly higher error metrics. Among nonlinear models, SVR achieved better results than KNN, CART, and the ensemble models, highlighting its ability to capture complex relationships. MLP, a neural network-based model, also provided competitive

predictions, demonstrating the potential of deep learning methods when effective feature engineering is applied. Ensemble models, including GBR, RFR, and ETR, showed moderate performance but were less effective in this context due to potential overfitting or the limited data available.

These findings highlight that simpler models, such as LASSO and LR, can outperform more complex algorithms in scenarios where data is scarce, as is often the case in cryptocurrency markets. Machine learning approaches provide a valuable alternative to traditional econometric models, offering robust predictions by leveraging engineered features effectively.

Implications of the Results

This study evaluates the performance of traditional econometric models (ARMA, GARCH, EGARCH) and machine learning models (e.g., LASSO, Linear Regression, SVR) in predicting implied volatility for crypto derivatives. Table 3 presents the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) for all models. These metrics allow us to compare the accuracy of predictions across different models under the context of limited data history.

Among traditional models, ARMA demonstrates the best performance, with the lowest MSE (0.009912), RMSE (0.099558), and MAE (0.082060). This indicates that ARMA, a simpler econometric model, is well-suited for short-term implied volatility forecasting when historical data is limited. On the other hand, GARCH and EGARCH models, which are designed to capture volatility clustering, perform worse, with MSE values around 0.049 and RMSE values exceeding 0.22. This result suggests that these models may require a larger dataset to fully capture the underlying volatility dynamics effectively.

Machine learning models demonstrate strong predictive capabilities. LASSO outperforms all models, achieving the lowest MSE (0.009911), RMSE (0.099556), and MAE (0.082046). Linear Regression (LR) and Multilayer Perceptron (MLP) also perform competitively, with MSE values around 0.010. These results highlight the ability of machine learning models to leverage feature-rich datasets, such as returns and rolling volatility, to make accurate predictions even with limited data.

Nonlinear models, such as Support Vector Regression (SVR), Gradient Boosting (GBR), Random Forest (RFR), and Extra Trees (ETR), show moderate performance, with MSE values ranging from 0.012259 to 0.015748. While these models can capture complex relationships, their slightly higher error values suggest potential overfitting or inefficiency in fully utilizing the limited dataset. Decision Tree (CART) performs the worst among all models, with the highest MSE (0.025966) and MAE (0.135282), likely due to its tendency to overfit on small datasets.

The findings presented in Table 3 offer important insights into the selection of models for forecasting implied volatility in crypto derivatives. Among the traditional models, ARMA proves to be the most effective, particularly in settings with limited data. Its simplicity and adaptability to short-term data

patterns allow it to outperform more complex GARCH-based models, which struggled due to the constraints of the small dataset. GARCH and EGARCH, while powerful for capturing volatility clustering in larger datasets, fail to deliver competitive results in this scenario, emphasizing their dependency on a longer historical record.

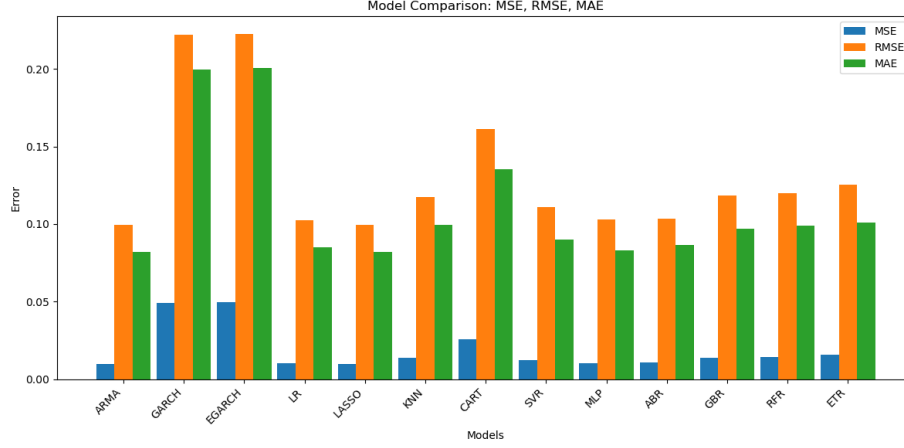
On the other hand, machine learning models demonstrate significant promise in leveraging engineered features for volatility forecasting. LASSO and Multilayer Perceptron (MLP) emerged as robust alternatives, achieving competitive performance by effectively capturing both linear and nonlinear relationships in the data. Their adaptability to feature-rich datasets highlights their advantage in data-scarce environments. Nonlinear machine learning models, such as Gradient Boosting and Random Forest, provided moderate accuracy, but their slightly higher error rates suggest potential inefficiencies when applied to limited datasets.

Overall, these findings underline the importance of tailoring model selection to the specific constraints and characteristics of the cryptocurrency derivatives market. Simpler models like ARMA remain valuable for short-term forecasting with limited data, while feature-based machine learning approaches, such as LASSO and MLP, offer a versatile and reliable option for implied volatility prediction. The results emphasize that combining simplicity with effective feature engineering can lead to more accurate predictions in data-scarce environments, providing crucial insights for practitioners and researchers in the evolving field of crypto derivatives.

Table 3: Comparison of Traditional Models and Machine Learning Models for Implied Volatility Prediction

Model Type	Model	MSE	RMSE	MAE
Traditional Models	ARMA	0.009912	0.099558	0.082060
	GARCH	0.049301	0.222037	0.199702
	EGARCH	0.049587	0.222682	0.200374
Machine Learning Models	Linear Regression (LR)	0.010484	0.102389	0.085148
	LASSO	0.009911	0.099556	0.082046
	K-Nearest Neighbors (KNN)	0.013811	0.117522	0.099263
	Decision Tree (CART)	0.025966	0.161140	0.135282
	Support Vector Regression (SVR)	0.012259	0.110720	0.090234
	Multilayer Perceptron (MLP)	0.010554	0.102735	0.082964
	AdaBoost (ABR)	0.010661	0.103251	0.086502
	Gradient Boosting (GBR)	0.014046	0.118514	0.096806
	Random Forest (RFR)	0.014404	0.120016	0.098942
	Extra Trees (ETR)	0.015748	0.125492	0.101202

Figure 2: Model Comparison : MSE, RMSE, MAE



4 Conclusion

This study investigates the effectiveness of traditional econometric models and machine learning approaches in forecasting the implied volatility of crypto derivatives, particularly in data-scarce environments. By analyzing the performance of ARMA, GARCH, EGARCH, and a range of machine learning models such as LASSO, Linear Regression, and Multilayer Perceptron, we uncover important insights into model selection for this unique market.

The results demonstrate that simpler models, such as ARMA and LASSO, are highly effective when the available historical data is limited. ARMA outperformed GARCH and EGARCH, highlighting that complex econometric models relying on long-term volatility clustering may not be suitable for short-term predictions in cryptocurrency markets. Conversely, machine learning models like LASSO and MLP leveraged engineered features, such as returns and rolling volatility, to achieve competitive performance, demonstrating their adaptability to nonlinear relationships and smaller datasets.

The findings emphasize the importance of matching model complexity to the data constraints of the cryptocurrency derivatives market. For practitioners and researchers, the choice of model should depend on the forecasting horizon and the availability of data. Simpler methods, such as ARMA and feature-based machine learning models, are better suited for short-term predictions and situations where historical data is scarce. In contrast, models like GARCH and EGARCH may still hold value for long-term forecasting when a larger dataset becomes available.

This research provides valuable insights for volatility forecasting in the rapidly growing field of cryptocurrency derivatives. It suggests that blending traditional econometric models with modern machine learning approaches offers a promising pathway for improving prediction accuracy. Future work could explore hybrid models that combine the strengths of both methodologies or focus

on expanding datasets to better leverage the capabilities of advanced econometric techniques like GARCH and EGARCH. These directions will be critical as the cryptocurrency market continues to evolve and demand for accurate risk management tools grows.

References

- [1] Oh, Dong Hwan, and Yang-Ho Park. *GARCH Option Pricing with Volatility Derivatives*. Board of Governors of the Federal Reserve System, Washington, D.C. Available online November 1, 2022.
- [2] Zhang, Wenjun, and Jin E. Zhang. *GARCH Option Pricing Models and the Variance Risk Premium*. Journal of Risk and Financial Management, 2020, 13(3), pp.51.
- [3] Wang, Tianyi, Yiwen Shen, Yueting Jiang, and Zhuo Huang. *Pricing the CBOE VIX Futures with the Heston–Nandi GARCH Model*. Journal of Futures Markets, first published November 9, 2016.
- [4] Christoffersen, Peter, K. *Evidence*. The Journal of Derivatives, 2013, 21, pp.8–41.
- [5] Venter, Pierre J., and Eben Maré. *GARCH Generated Volatility Indices of Bitcoin and CRIX*. Journal of Risk and Financial Management, 2020, 13(6), pp.121.
- [6] McAleer, Michael, and Christian Hafner. *A One Line Derivation of EGARCH*. Econometrics, 2014, 2, pp.92–97.
- [7] Genovese, Christopher R., et al. *A comparison of the lasso and marginal regression*. The Journal of Machine Learning Research, 2012, 13(1), pp.2107–2143.
- [8] Akaike, Hirotugu. *A New Look at the Statistical Model Identification*. IEEE Transactions on Automatic Control, 1974, 19, pp.716–723.
- [9] Schwarz, Gideon E. *Estimating the Dimension of a Model*. Annals of Statistics, 1978, 6, pp.461–464.