27 April 2024

Enhancing Trading Performance: Leveraging ARIMA-GARCH Strategies for Volatility Forecasting

Asad Qureshi asq10@scarletmail.rutgers.edu

Kasper Kujala kasper.kujala@rutgers.edu

Adam Tyler adam.tyler@rutgers.edu

Subha Bose sb2370@scarletmail.rutgers.edu

Abstract

We combine ARIMA and GARCH to create a trading strategy for the S&P 500 Index with the objective of outperforming the buy and hold approach. We systematically determine the best ARIMA model and use a GARCH(1,1). We utilize parallelization to improve computational efficiency. Our strategy outperforms the buy and hold approach, particularly during periods of high volatility.

1. Introduction

In the world of quantitative finance, accurately forecasting volatility is a pivotal undertaking for novice and seasoned traders alike, as it can help individuals and entities make informed decisions regarding risk management, asset allocation, derivative instrument pricing, and fiscal policies. The predictability of volatility is essential in each of these contexts, as it can directly influence a traders' ability to access profit opportunities. For example, a risk manager may seek to gauge the likelihood of their portfolio value increasing or decreasing within a certain timeframe, while an option trader aims to anticipate future contract volatility. Understanding forecasted volatility is also crucial for hedging purposes. By accurately anticipating future levels of volatility, investors and financial institutions can actively adjust their hedging positions, such as derivatives contracts or portfolio allocations, to effectively offset adverse price movements and safeguard against market fluctuations. Similarly, portfolio managers may wish to divest assets before they become excessively volatile, and market makers adjust bid-ask spreads based on anticipated future volatility.

1.1. Volatility Characteristics

In order to comprehend the principles of volatility, it is essential to emphasize its *defining characteristics:*

Despite its impact on market dynamics, volatility itself is *not directly observable*, with no single value representing its true nature.[1] Instead, we distinguish its influence through observable changes in prices, acknowledging the absence of a definitive measure for volatility itself. In addition, estimating volatility is confined to specific time intervals, compounding the difficulty in assessing its accuracy, even with multiple reference points.

Moreover, volatility exhibits a *dynamic nature*, thereby complicating its modeling process.[1] The conventional Black-Scholes model that is used for pricing options, for instance, falls short when it assumes constant volatility, which was a simplification adopted for modeling convenience rather than reflecting real-world fluctuations. Modern methodologies recognize volatility as a stochastic or time-varying process (a random walk in a time series, for example), aligning it closer to reality but still posing great challenges when it comes to modeling its behavior.

When estimating volatility, you need to take into account both its *continuous and discrete elements*, each requiring distinct modeling approaches due to asset trading session constraints.[1] While theoretically, volatility is continuous, the absence of price data during certain periods necessitates reconciliation with discrete market events, creating a jump in the pricing data that must be incorporated into the volatility forecast. Ignoring its discrete or continuous behavior can lead to mismatching theoretical constructs with practical outcomes, which can result in significant estimation discrepancies.

Finally, volatility's *auto regressive tendencies* offer insight into future behavior, as past volatility serves as a good predictor of future movements.[1] Clustering around events and exhibiting persistence over time, volatility affords model makers effective forecasting methods compared to the more elusive task of predicting returns. Even a simple historical standard deviation of returns proves valuable in forecasting tomorrow's volatility, leveraging its autoregressive properties to inform future expectations.

1.2. Volatility Persistence

To understand the nature in which to model volatility, it is important to develop a fundamental understanding of its auto regressive tendencies. This leads us to the concept of volatility persistence, a notable phenomenon observed across financial markets, characterized by the tendency of volatility levels to exhibit clustering over time. This type of clustering often leads to prolonged periods of high or low volatility, therefore exerting a profound influence on financial decision-making. To address volatility persistence in a financial time series analysis, the GARCH model emerges as a vital tool that can capture this clustering nature through its autoregressive structure.

The GARCH (Generalized Autoregressive Conditional Heteroskedasticity) model is a statistical framework commonly used in financial mathematics to model and forecast time-varying volatility in asset returns. It is an extension of the ARCH (Autoregressive Conditional Heteroskedasticity) model by incorporating past conditional variances in addition to squared residuals, allowing for a more comprehensive representation of volatility dynamics. By capturing the persistence of volatility and its clustering tendencies over time, the GARCH model provides valuable insights into the behavior of financial markets and helps improve the accuracy of risk management, asset pricing, and derivative instrument valuation. A typical GARCH model is represented as GARCH(p,q), where p represents the number of lag variances while q represents the number of lag residual errors.[2] Incorporating a moving average component which the ARCH model lacks enables us to capture both the conditional variation over time and fluctuations in the time-dependent variance.

$$\boldsymbol{\sigma}_t^2 = \alpha_0 + \alpha_1 a^2_{t-1} + \beta_1 \boldsymbol{\sigma}^2_{t-1} \qquad (1)$$

Equation 1 captures a GARCH (1,1) and the terms of the equation represent the following:

σ_t^2: conditional variance of the time series at time t
$\boldsymbol{\sigma}^2_{t-1}$: conditional variance of the time series at the previous time step, t-1
a^2_{t-1}: squared error term at the previous time step, t-1
α_0: intercept
α_1, β_1: coefficients for their respective terms

Equation 1 suggests that the current conditional variance (σ_t^2) is influenced by both the past squared error term (a^2_{t-1}) and the past conditional variance ($\boldsymbol{\sigma}^2_{t-1}$), with their respective coefficients determining the magnitude of their impact. To summarize, this equation captures the idea that volatility tends to cluster over time, with periods of high volatility often followed by additional periods of high volatility, and vice versa.

1.3. ARIMA

In addition to the GARCH model, the ARIMA model is also a prominent tool used in time series analysis. The Autoregressive Integrated Moving Average (ARIMA) model is a powerful tool in time series analysis widely used to understand and forecast data exhibiting temporal dependencies. In essence, the ARIMA model combines autoregressive (AR), differencing (I), and moving average (MA) components to capture the underlying patterns in the data. The AR component represents the linear relationship between an observation and a certain number of lagged observations, reflecting the series' own past values. The differencing component accounts for non-stationarity in the data by transforming it into a stationary series, thereby removing any trends or seasonal patterns that may exist. Lastly, the MA component captures the relationship between an observation and the residual errors from a moving average model applied to lagged observations. By integrating these components, the ARIMA model can effectively model and forecast various time series phenomena, making it a valuable tool for economists, statisticians, and analysts across diverse fields. Its versatility and flexibility in handling a wide range of data types and patterns make the ARIMA model a cornerstone in time series analysis and forecasting. By integrating the ARIMA and GARCH models, you are able to address both the mean and volatility aspects of the data, enhancing the forecast's ability to capture the particular nuances of the economic climate being studied.

2. Dataset

The dataset utilized for our project is daily closing prices of the S&P 500 index. The data was retrieved from Yahoo Finance. We specifically chose the "Close" price of the index, which represents the final trading price of the index for each day. This metric is particularly relevant for our analysis as it reflects the market's consensus price of the index at the end of the trading day, summarizing the day's market movements and investor sentiments. The data begins from 01/01/2003 to 12/31/2023. providing a comprehensive view of the index's performance over a decade, which includes varying market conditions such as bullish trends, bearish downturns, and periods of high volatility. This time frame allows for a thorough assessment of the ARIMA+GARCH model's predictive abilities across various market scenarios, facilitating testing under different market conditions throughout the time period.

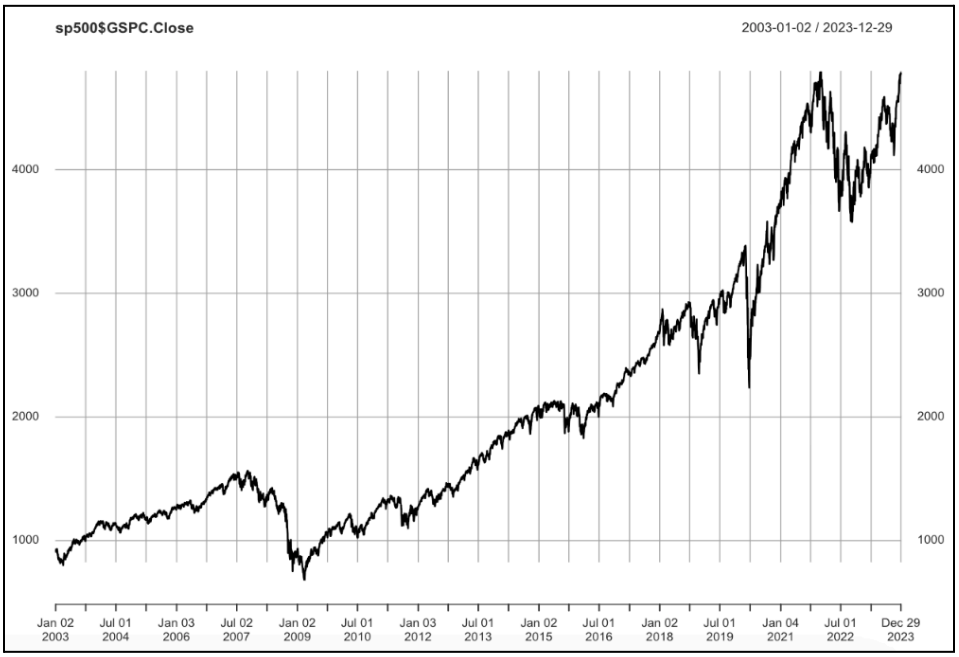

Figure 1: S&P 500 Closing Price From January 2003 - December 2023

2.1. Data Preprocessing

Initially we check for any missing values in our closing price data. To normalize the influence of extreme price fluctuations, we apply a logarithmic transformation to the prices, which also helps stabilize the variance. Further, we calculate daily returns and use these as a basis for our ARIMA and GARCH models instead of raw price data, as logarithmic returns are often more stationary and thus more suitable for our analytical methods. To confirm stationarity of the S&P 500 log returns we utilize the Augmented Dickey Fuller (ADF) test with a significance level of 0.05. The ADF test yields a p-value less than 0.01, indicating that the log returns are stationary.

```
        Augmented Dickey-Fuller Test

data:  sp500daily_log
Dickey-Fuller = -10.946, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary
```

Figure 2: ADF Test of S&P 500 Log Returns from January 2003 - December 2023

3. Trading Strategy

3.1. Strategy Overview

Our trading strategy is designed with the primary objective of accurately predicting future price movements and volatility in the S&P 500 index, with the overarching goal of surpassing the performance of a traditional buy and hold approach. To achieve this objective, we employ a sophisticated modeling approach that combines ARIMA and GARCH techniques.

The ARIMA model is utilized to capture underlying price trends in the S&P 500 index, while the GARCH model is used to model and forecast the volatility dynamics associated with these price movements. By integrating these two models, we aim to gain a comprehensive understanding of both the directional movement and the level of volatility present in the market.

Trading signals are generated based on the forecasts provided by the ARIMA and GARCH models. A positive forecast from the model prompts a buy signal (+1), indicating a favorable outlook for future price appreciation, while negative forecasts trigger sell signals (-1), indicating an expectation of price decline. Additionally, if consecutive daily forecasts yield identical signals, the existing position is maintained, reflecting our confidence in the persistence of the market volatility.

In this paper, we seek to demonstrate the effectiveness of our trading strategy by implementing it to either short or long our entire portfolio. By actively adjusting our portfolio positions based on the signals generated by our ARIMA+GARCH model, we aim to showcase the strategy's ability to capitalize on both upward and downward price movements in the S&P 500 index, thereby potentially achieving superior returns compared to a passive buy and hold strategy.

3.2. ARIMA model

In our trading strategy, the ARIMA component is designed to adaptively fit the model using a rolling window approach. This method ensures that the model continually updates and bases its forecasts on the most recent data, enhancing its responsiveness to market changes. For each window, we systematically determine the optimal parameters for the ARIMA model by iterating through potential values for p (autoregressive terms) and q (moving average terms). We select the combination that yields the lowest Akaike Information Criterion (AIC) score, which balances model fit and complexity to prevent overfitting. The differencing order d is fixed at zero, as we have already differenced the S&P 500 daily close price data.

3.3. GARCH model

The GARCH model complements the ARIMA model's price forecasts by providing a measure of the expected volatility associated with those forecasts. Our GARCH model specifically utilizes a (1,1) order, implying that it includes one lag each for the autoregressive and moving average components of volatility. This configuration is typically sufficient to capture the volatility clustering commonly observed in financial markets—where high-volatility days tend to cluster together, as do lower-volatility days. The integration of GARCH with the outputs from the ARIMA model is a key aspect of our strategy. The ARIMA model forecasts future price movements, and the GARCH model assesses the expected volatility around these forecasts. By combining these two perspectives, we can tailor our trading signals not only based on the direction and magnitude of the price movements but also considering the associated uncertainty or risk as indicated by the volatility forecasts.

3.4. Implementation

In our study, we utilized several commonly used R libraries, namely quantmod, rugarch, and tseries, to conduct comprehensive analyses of financial market data. Specifically, we focused on computing daily log returns for the S&P 500 index, which is a widely tracked benchmark in the financial industry. By leveraging the capabilities of these libraries, we were able to fit ARIMA and GARCH models to the data, allowing us to capture both price trends and volatility dynamics.

We implement the ARIMA part of our model with a custom function. We systematically explore ARIMA models within the defined parameter space (p and q ranging from 0 to 5, excluding the trivial (0,0) case). This function employs a tryCatch mechanism to handle potential errors during model fitting. The custom function iteratively fits ARIMA models and records the one with the lowest AIC, identifying the model that best captures the underlying patterns in the data.

We implement the GARCH part of our model also with a custom function, we fit the GARCH(1,1) and fit the ARIMA order given from our ARIMA function.If the ARIMA model is null, indicating a failure in its fitting process, the GARCH model fitting is skipped, as indicated by the error handling mechanism in place. This ensures that our volatility forecasts are always based on reliable and valid preliminary analyses. Error handling within the function is managed through tryCatch blocks that capture and warn of any issues during the fitting process, such as non-convergence errors. This allows for easier troubleshooting of the model's reliability. We use the Skewed Generalized Error Distribution (SGED) to accommodate the skewness and heavier tails typically observed in financial return data.

Subsequently, we developed forecast functions based on these models to generate predictions for future market movements. The function begins by checking the integrity of the GARCH model fit. If the model fitting process fails, indicated by a null garch_fit, the function immediately returns a 'no convergence' status along with the date of the forecast attempt. This step is crucial for maintaining the reliability of our trading signals, ensuring that no decisions are based on flawed or incomplete data analyses. Once a valid GARCH model fit is confirmed, the function proceeds to generate a forecast for the next day's market return using the ugarchforecast method. This forecast includes an estimation of the return and volatility of the market's movement. The forecast output, specifically the predicted market return, is used to generate trading signals. If the predicted value indicates a positive return , a buy signal (1) is generated. Conversely, a predicted downward movement triggers a sell (-1) signal. If the signal is the same as the previous day the position stays unchanged.

To optimize efficiency, we implemented the parallel library, enabling us to speed up calculation tasks involved in offering predictions. We configure our cluster by initializing a cluster object. Within each cluster node, essential libraries such as quantmod, rugarch, and t series are loaded. Using parLapply, a parallel version of the lapply function, we distribute the task of generating trading signals across the multiple cores. Each core processes a portion of the total forecast length, applying our ARIMA and GARCH model fitting functions, followed by the forecast generation.

Finally we calculated strategy returns and plotted them against buy-and-hold returns, providing insights into the effectiveness of our trading strategy. The implementation of our trading strategy showcases the practical application of these R libraries in quantitative finance research, demonstrating their utility in analyzing and forecasting financial market dynamics.

4. Results

```
# getting S&P 500 data
sp500 <- getSymbols('^GSPC', src = "yahoo", from = '2003-01-01', to = '2023-12-31', auto.assign = FALSE)
plot(sp500$GSPC.Close)

# check for missing values
print(sum(is.na(sp500)))

# compute daily log returns
sp500daily_log <- dailyReturn(sp500$GSPC.Close, type = 'log')
print(sum(is.na(sp500daily_log)))
plot(sp500daily_log)

window_length <- 200 #rolling window
forecast_length <- length(sp500daily_log) - window_length
forecasts <- vector(mode = 'character', length = forecast_length)

# fit ARIMA model function
fit_best_arima <- function(data) {
  best_aic <- Inf
  best_arima <- NULL
  for (p in 0:5) {
    for (q in 0:5) {
      if (p == 0 && q == 0) {
        next
      }
      arimaFit = tryCatch(arima(data, order = c(p, 0, q)), error = function(err) NULL, warning = function(err) NULL)
      if (!is.null(arimaFit)) {
        current_aic <- AIC(arimaFit)
        if (current_aic < best_aic) {
          best_aic <- current_aic
          best_arima <- arimaFit
        }
      }
    }
  }
  return(best_arima)
}
```

Figure 3: Code Snippet of Data and ARIMA model

```r
fit_garch_model <- function(data, arima_model) {
  if (is.null(arima_model)) {
    warning("ARIMA model is NULL, skipping GARCH fitting.")
    return(NULL)
  }
  spec <- ugarchspec(
    variance.model = list(garchOrder = c(1, 1)),
    mean.model = list(armaOrder = c(arima_model$arma[1], arima_model$arma[3]), include.mean = TRUE),
    distribution.model = "sged"
  )
  fit = tryCatch(
    ugarchfit(spec, data, solver = 'hybrid'),
    error = function(e) { warning("GARCH fitting failed: ", e$message); NULL },
    warning = function(w) { warning("GARCH fitting warning: ", w$message); NULL }
  )
  return(fit)
}

# create forecast function
create_forecast <- function(garch_fit, data) {
  if (is.null(garch_fit)) {
    return(paste(index(data[length(data)]), "no convergence", sep = ","))
  }
  fore = ugarchforecast(garch_fit, n.ahead = 1)
  predicted_value = fore@forecast$seriesFor
  return(paste(index(data[length(data)]), ifelse(predicted_value[1] < 0, -1, 1), sep = ","))
}

# create cluster for running loop
cl <- makeCluster(detectCores() - 1)  # use all cores but one
clusterEvalQ(cl, {
  library(quantmod)
  library(rugarch)
  library(tseries)
})
clusterExport(cl, c("sp500daily_log", "window_length", "fit_best_arima", "fit_garch_model", "create_forecast"))

# loop for signals
results <- parLapply(cl, 1:forecast_length, function(d) {
  sp_returns_offset = sp500daily_log[d:(d + window_length - 1)]
  final_arima = fit_best_arima(sp_returns_offset)
  final_garch = fit_garch_model(sp_returns_offset, final_arima)
  create_forecast(final_garch, sp_returns_offset)
})
```

Figure 4: Code Snippet of the GARCH Model and Signal Creation

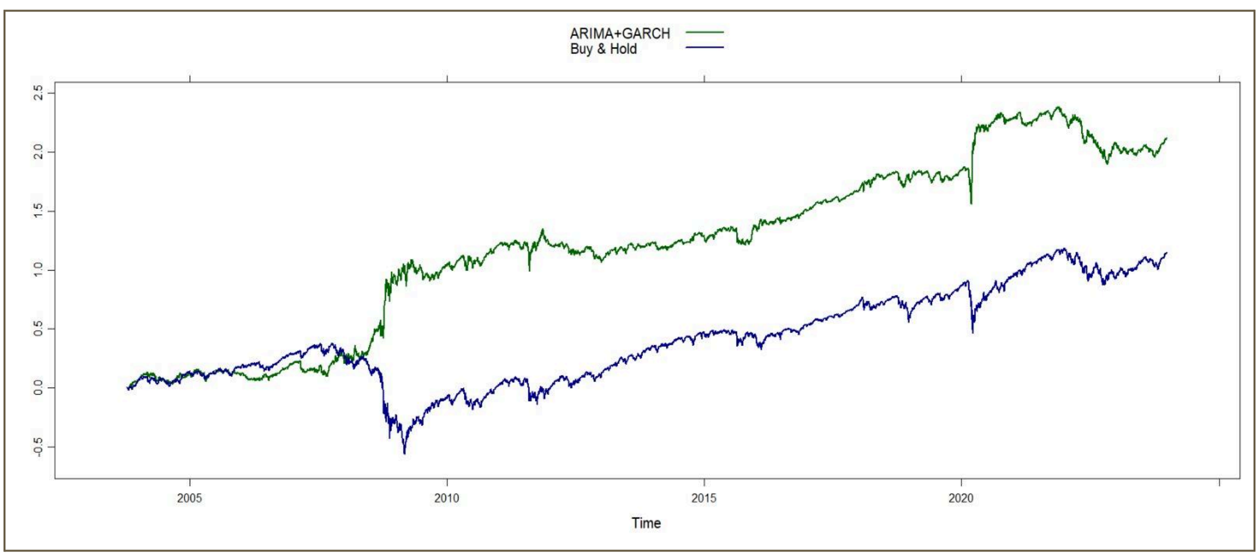

Figure 5: Equity Curve of Trading Strategy vs Buy and Hold from 2003-2023

2023-12-01	1
2023-12-04	1
2023-12-05	1
2023-12-06	1
2023-12-07	1
2023-12-08	1
2023-12-11	1
2023-12-12	1
2023-12-13	1
2023-12-14	1
2023-12-15	1
2023-12-18	1
2023-12-19	1
2023-12-20	1
2023-12-21	1
2023-12-22	1
2023-12-26	1
2023-12-27	1
2023-12-28	1

Figure 6: Trading Signals From December 2023

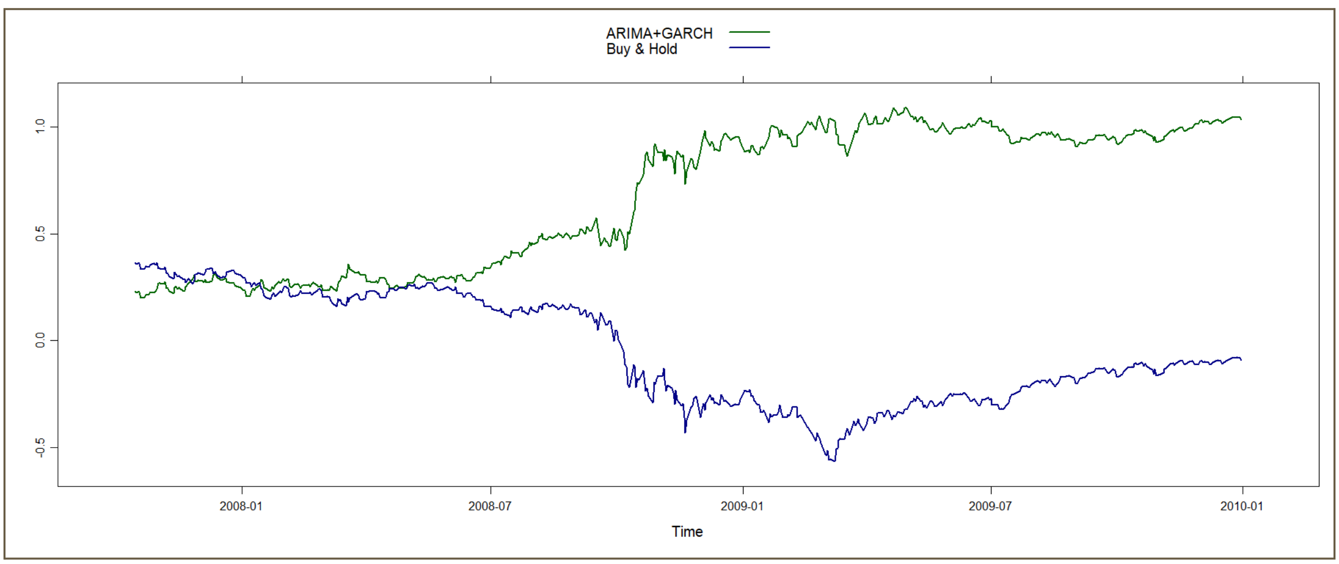

Figure 7: Equity Curve of Trading Strategy vs Buy and Hold During 2008/2009 Financial Crisis

2008-10-07	1
2008-10-08	-1
2008-10-09	-1
2008-10-10	1
2008-10-13	1
2008-10-14	-1
2008-10-15	-1
2008-10-16	1
2008-10-17	1
2008-10-20	1
2008-10-21	-1
2008-10-22	-1
2008-10-23	1
2008-10-24	1
2008-10-27	1
2008-10-28	1
2008-10-29	-1
2008-10-30	-1
2008-10-31	-1
2008-11-03	-1
2008-11-04	-1
2008-11-05	-1
2008-11-06	1
2008-11-07	1
2008-11-10	1
2008-11-11	1
2008-11-12	1

Figure 8: Trading Signals From October 2008

2010-01-04	1
2010-01-05	1
2010-01-06	1
2010-01-07	1
2010-01-08	-1
2010-01-11	1
2010-01-12	1
2010-01-13	1
2010-01-14	1
2010-01-15	1
2010-01-19	1
2010-01-20	-1
2010-01-21	1
2010-01-22	1
2010-01-25	1
2010-01-26	1
2010-01-27	1
2010-01-28	1
2010-01-29	1
2010-02-01	1
2010-02-02	1
2010-02-03	1
2010-02-04	1
2010-02-05	1
2010-02-08	1
2010-02-09	1
2010-02-10	-1

Figure 9: Trading Signals From January 2010

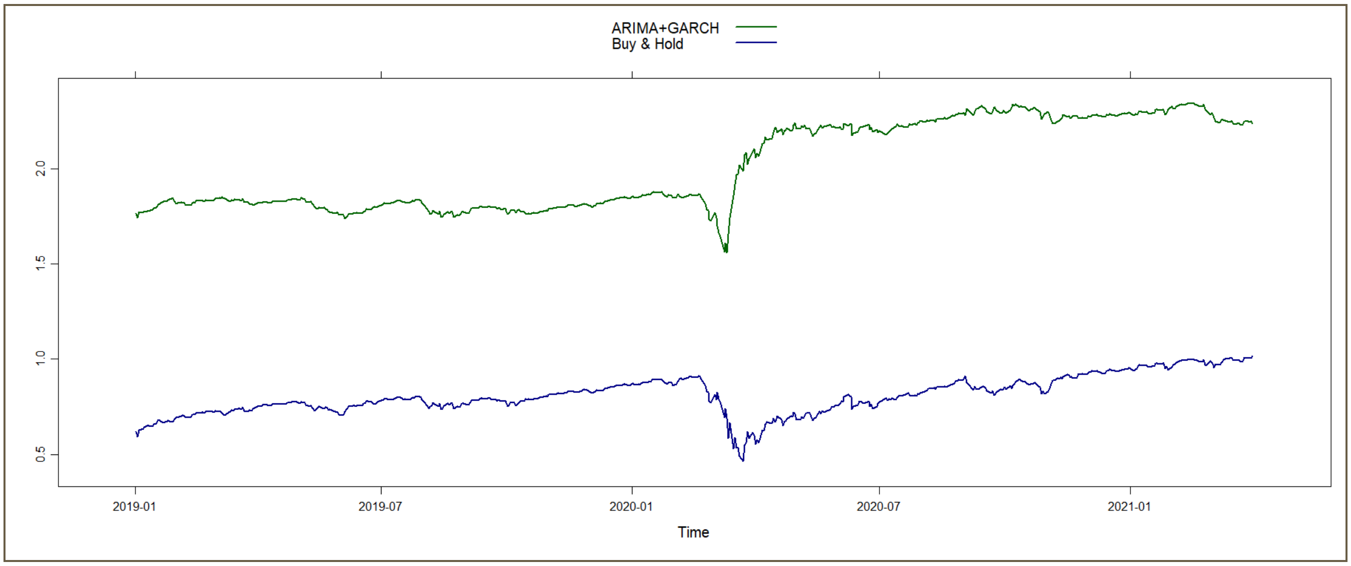

Figure 10: Equity Curve of Trading Strategy vs Buy and Hold During Onset of COVID

2020-03-17	1
2020-03-18	-1
2020-03-19	1
2020-03-20	-1
2020-03-23	1
2020-03-24	1
2020-03-25	1
2020-03-26	-1
2020-03-27	-1
2020-03-30	1
2020-03-31	-1
2020-04-01	1
2020-04-02	1
2020-04-03	1
2020-04-06	1
2020-04-07	-1
2020-04-08	1
2020-04-09	-1
2020-04-13	-1
2020-04-14	1
2020-04-15	-1
2020-04-16	1
2020-04-17	-1
2020-04-20	-1
2020-04-21	1
2020-04-22	1
2020-04-23	1

Figure 11: Trading Signals from March-April 2020

2021-03-01	-1
2021-03-02	-1
2021-03-03	1
2021-03-04	1
2021-03-05	-1
2021-03-08	1
2021-03-09	1
2021-03-10	1
2021-03-11	-1
2021-03-12	1
2021-03-15	-1
2021-03-16	-1
2021-03-17	1
2021-03-18	1
2021-03-19	1
2021-03-22	1
2021-03-23	1
2021-03-24	1
2021-03-25	1
2021-03-26	1
2021-03-29	1
2021-03-30	1
2021-03-31	1
2021-04-01	-1
2021-04-05	-1

Figure 12: Trading Signals from March-April 2021

Date	ARIMA+GARCH Cumulative Log Return	Buy and Hold Cumulative Log Return
2010-01-01	1.0507563	-0.073639230
2020-05-01	2.208999	0.6838390
2023-12-31	2.120745	1.1495660

Table 1: Cumulative Log Return at Key Points

5. Conclusion

5.1. Final Analysis

Assessing the performance of our trading strategy, which integrates ARIMA and GARCH models, we observe an advantage over the traditional buy and hold strategy. Looking at Figure 5 we can observe how our model outperforms the buy and hold approach during periods of high volatility, like during the financial crisis and 2020 during the pandemic.

While this strategy does show promising returns, especially in periods of high volatility, what it does not take into consideration are transaction costs, interest fees and tax implications. It can be seen from Figure 11 that frequently the strategy gives different signals one day after another meaning that if this strategy was implemented it would need to be traded fully each day. Going from a completely long position to a completely short position will have huge transactional cost implications. Depending on the amount of capital allocated to this strategy it may even be infeasible to switch the direction of the portfolio from day to day. The other huge implication is the tax burden this strategy will incur. Because the buy-and-hold strategy can be assumed to be taxed on a long-term horizon its tax burden is not especially large and is what any investor would expect. Our strategy potentially trades each day creating a taxable event. This event is also taxed at the short-term rate which is substantially higher. The last point that can be detrimental to this strategy is the cost of shorting. For each day the strategy forecasts we must be short we will need to borrow these shares and we will incur the cost of borrowing these shares. For the S&P 500, there should be ample shares available to short as it is an extremely popular index so we do not foresee any issue with finding available shares to short.

5.2. Future Considerations

One point we can further expand upon is the size of the rolling window. Currently, we use a 200-day rolling window which may or may not be the most optimal. Our decision for selecting the 200-day rolling window was that the calculation of signals is computationally limited and using any larger window would have been impossible given our technological limitations. It would be interesting to further explore this idea with a smaller rolling window. We suspect that this will lead to even more frequent changing of the signal forecasted by the model which will only further increase the transactional costs discussed earlier. Although the results of this are untested, we suspect that a window that is too short may lead to the model forecasting signals that do not align with the current trend of the market and will lead to substantial losses in periods of high volatility which is one of our main goals to avoid.

Another future consideration, instead of reassessing the p and q of the ARIMA order for each forecast, instead we propose adjusting these parameters less frequently. Potentially every quarter or every six months. This will make the model less computationally difficult to run. Allowing us to adapt to underlying trends in S&P 500 index while avoiding the computational overhead associated with the constant checking for the best ARIMA order.

For the GARCH model, we elected to go with the parameters (1,1) which may not always be optimal. We can expand this research further by changing the parameters based on a specific time period. Similarly to how we choose our ARIMA parameters, we could select a window and test during that period to produce the best GARCH model based on the AIC score. Another way to look at it is if we still have static parameters for the GARCH model but test more parameters to see if we have significantly greater performance than the performance generated using the (1,1) parameters.

In conclusion, this paper has demonstrated the importance of volatility modeling in financial time series analysis and forecasting. By integrating techniques such as the ARIMA and GARCH models, researchers and model makers alike can gain valuable insights into both the mean and volatility components of the data, enhancing their ability to make informed decisions in dynamic market environments.

References

Salt Financial Indices. (2021). The Layman's Guide to Volatility Forecasting: https://saltfinancial.com/static/uploads/2021/05/The Laymans Guide to Volatility Forecasting.pdf

Salt Financial Indices. (2021). The Layman's Guide to Volatility Forecasting: https://saltfinancial.com/static/uploads/2021/05/The Laymans Guide to Volatility Forecasting.pdf