Volatility Forecasting for Equity Options on the National Stock Exchange of India

Nicholas Pittman

Tippie College of Business

The University of Iowa

May 11, 2025

Abstract

This report applies four return-based volatility estimation methods to 64 stocks from 2010 to 2020, finding that the Root Sum of Squares (RSS) method most accurately tracks realized volatility for short-dated options. Seven forecasting methods were evaluated, including regression, machine learning, and GARCH time series models. Decision trees and random forests significantly outperformed the time series approaches, with momentum factors like historical volatility and total return emerging as the most predictive features. These forecasts were applied to a portfolio trading strategy, but neither model delivered meaningful long-term returns, indicating the need for a more robust approach to consistently outperform the market.

1. Introduction

Volatility, or the statistical measure of dispersion of returns for a given security, is an integral component across all financial markets, given its impact on security pricing, risk management, and derivatives. As the only variable in the Black-Scholes algorithm that is not directly observed in the market, numerous assumptions and calculations exist for how to derive this number. This analysis focuses on forecasting volatility over the life of an option contract, which is a derivative instrument that gives the buyer the right, but not the obligation, to buy or sell the underlying asset at a predetermined price (strike price K). Given that volatility and the option's price have a direct relationship, it can be assumed that for options where the forecasted volatility exceeds the option's implied (or current) volatility, the option is undervalued. Thus, a trading strategy can be developed to analyze its performance within a broader portfolio.

The focus of this analysis centers around the National Stock Exchange of India (NSE), a newer exchange formed in 1992 to succeed the Bombay Stock Exchange by incorporating modern advancements such as electronic trading, increased transparency, and more efficient prices. However, the NSE has been riddled with controversy, where several individuals have been convicted of market manipulation, price gouging, and data leaks (Rangan, 2019; SBI Securities, 2024). Combined with the overwhelming presence of young, high-growth companies and macroeconomic tailwinds, numerous securities exhibited significantly higher volatility than is typically observed in the more established Western markets.

Therefore, the implications of this analysis are particularly relevant for asset managers seeking more accurate volatility forecasts to improve their short-term risk management strategies. While extensive research exists for developed markets, relatively little has been done in emerging markets such as India. This study aims to apply widely used volatility estimation and forecasting methods to assess their effectiveness in this context, with the goal of producing forecasts that outperform simple averages.

This report is structured as follows. Section 2 outlines the data sources, preprocessing steps, and calculations used to estimate historical volatility. Section 3 details the forecasting methods and assumptions employed, including regression models, machine learning algorithms, and time series analyses. Section 4 presents the results from these models, along with the performance of a backtested portfolio trading strategy. Finally, Section 5 concludes by analyzing the implications of these results, acknowledging limitations, and opportunities for future research.

2. Data

The data collected for this analysis originated from three sources. Options data from the NSE was sourced by Tanay Agarwal on Kaggle, listing daily contract information from 2000 to 2020 (Agarwal, 2020). The data set is rich with various financial derivatives for futures and options on individual securities, currencies, and indices. Risk-free interest rate data was sourced from the Federal Reserve of Economic Data (FRED), representing the one-year interest rate on government bills (International Monetary Fund, 2025). Finally, stock price data was sourced from the Yahoo! Finance API in Python.

A series of sampling, filtering, joining, and feature engineering was performed to prepare the data for modeling and to minimize the impact of data quality issues. In general, the structure of this preprocessing was based on the work of Amit Goyal and Alessio Saretto for predicting the cross-sectional distribution of implied volatilities (2009).

To begin, the dataset was filtered to only consider stock options, ignoring those related to indices or currencies. A random sample of 75 stocks was generated to create a small, yet representative, subset of companies. All stocks considered must have been actively traded between 2007 and 2020 under the same ticker to avoid inconsistencies.[1] The sample was then filtered further to only consider European option contracts with 27 to 63 days to expiry (Goyal & Saretto, 2009). This was applied to avoid the potential of an option holder exercising his/her option before maturity, allowing for a more consistent framework when calculating profit and loss in the simulated trading strategy. After these filters were applied, a sample of 64 stocks remained.

Daily stock prices were then downloaded for each of these securities from Yahoo! Finance. Risk-free interest data from FRED was aggregated monthly to remove excess noise and

[1] Stock options did not become available on the NSE until 2001, so 2007 was used as the initial sampling period to generate a sufficiently large population of securities for analysis. Based on these criteria, a population of 146 stocks remained.

joined back to the options data frame based on the month of the observation date of each option contract.

Finally, these options were filtered to only consider contracts that were at-the-money (ATM), represented by a moneyness (S/K) between 0.99 and 1.01 to ensure all contracts have a roughly equal chance of expiring in-the-money (ITM) or out-of-the-money (OTM). After applying the above-mentioned filters, the final sample consisted of 137,802 daily observations, where the average time to expiry was 44 days, and 51% of call options and 49% of put options expired in the money.

The next preprocessing steps consisted of calculating the implied volatility of each option, the realized volatility over the option's term, and the realized volatility and total log returns in the 90 days leading up to the observation date. Each stock's sector was also pulled from Yahoo! Finance for use in descriptive analytics and model building.

Implied volatility (IV) is a forward-looking measure of how much the market expects the price of a security to fluctuate over the contract period (Ganti, 2024). In other words, it is the volatility that represents the current price of the option by reverse-engineering the Black-Scholes formula. There are several methods for calculating IV, but the Newton-Raphson method was used given its faster computation and simple approach (Quant Next, 2023). The algorithm makes an initial volatility guess before using the derivative of the option price function to converge to the Black-Scholes price. This algorithm is summarized in **Formula 1**, where σ_n represents the initial volatility estimate and $f(\sigma_n)$ represents the difference between the option value using Black-Scholes and the market price. While the model can struggle with options that are deeply ITM or OTM, this analysis focused exclusively on at-the-money (ATM) contracts, for which the model converged without issue.

$$\sigma_{n+1} = \sigma_n - \frac{f(\sigma_n)}{f'(\sigma_n)}$$

Formula 1: Newton-Raphson Method for Calculating Implied Volatility

As there are several ways to calculate realized volatility, this analysis considered the four most popular return-based methods in the financial services industry (Gurung, 2025) – sample standard deviation, a five-day rolling average, an exponentially weighted moving average, and the root sum of squares. Utilizing daily log returns, calculations were inflated to an annual level to reflect the scale of implied volatility. The equations for each method are listed below in **Formula 2**, **Formula 3**, **Formula 4**, and **Formula 5**, where r is the daily log return, $\overline{r}$ is the mean of daily returns over the option's period, n is the number of observations, α is the smoothing constant,[2] and t is the time to expiry. These same methods were also used to calculate the 90-day historical annualized volatility leading up to the observation date. Finally, the 90-day historical return was calculated as the cumulative log return of the underlying stock over the most recent 90 calendar days.

$$\sigma = \sqrt{\frac{\sum(r - \overline{r})^2}{n-1}} * \sqrt{252}$$

Formula 2: Standard Deviation volatility (STD)

$$\sigma = \frac{\sum\sqrt{\frac{\sum(r - \overline{r_5})^2}{n-1}}}{5} * \sqrt{252}$$

Formula 3: 5-day Rolling Average volatility (ROLL)

$$\sigma = \sqrt{\alpha * \sigma_{t-1}^2 + (1-a) * r_{t-1}^2} * \sqrt{252}$$

Formula 4: Exponentially Weighted Moving Average volatility (EWMA)

$$\sigma = \sqrt{\sum r^2} * \sqrt{\frac{252}{t}}$$

Formula 5: Root Sum of Squares volatility (RSS)

Table 1 presents the distribution of realized volatility (RV) across all options, alongside the average deviation from each option's implied volatility (IV). Overall, the methods yield similar results, with an average realized volatility around 30%. However, they consistently underestimate the level of implied volatility, driven by the pronounced positive skewness of IV (2.96), as illustrated in **Figure 1**. Despite these comparable results, there still exists enough

[2] An α = 0.2 was used for this analysis based on tuned results.

variation between groups to be meaningful in a predictive model, and thus, all will be considered.

	Mean	Median	Std. Dev.	Minimum	Maximum	$\overline{(RV - IV)}$
STD RV	0.3132	0.2818	0.1412	0.0568	3.2176	-0.2260
EWMA RV	0.3013	0.2728	0.1309	0.0473	2.2544	-0.2368
ROLL RV	0.2930	0.2651	0.1270	0.0462	2.6745	-0.2464
RSS RV	0.2678	0.2404	0.1204	0.0495	2.7630	-0.2699

Table 1: Summary Statistics of Realized Volatility (RV) Calculations.

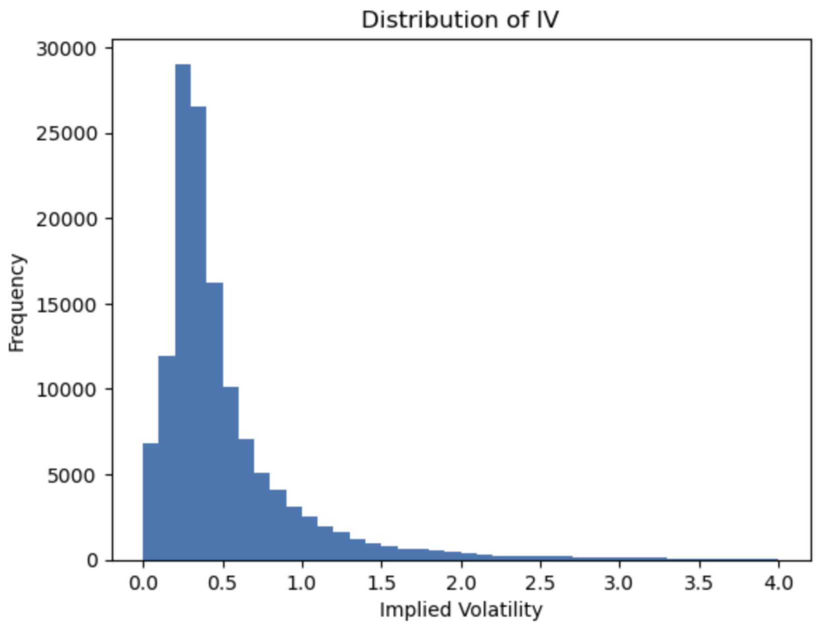

Figure 1: Distribution of Implied Volatility (IV)

Considering idiosyncratic factors can be extremely valuable when forecasting a security's volatility, thus each stock's sector was included in the analysis. **Figure 2** shows the distribution of realized RSS by sector, emphasizing the high-growth and high-volatility areas such as Industrials and Financial Services. Within Financial Services specifically, its long tail of outliers can be attributed to the Yes Bank Crisis, where the company's share price fell from ₹404 to ₹17 in 17 months. Driven by a slew of poor direct lending decisions, the company was placed under moratorium by the Reserve Bank of India in March 2020. The events significantly damaged investor sentiment, negatively impacting dozens of mid- and small-cap stocks (ET Online, 2020).

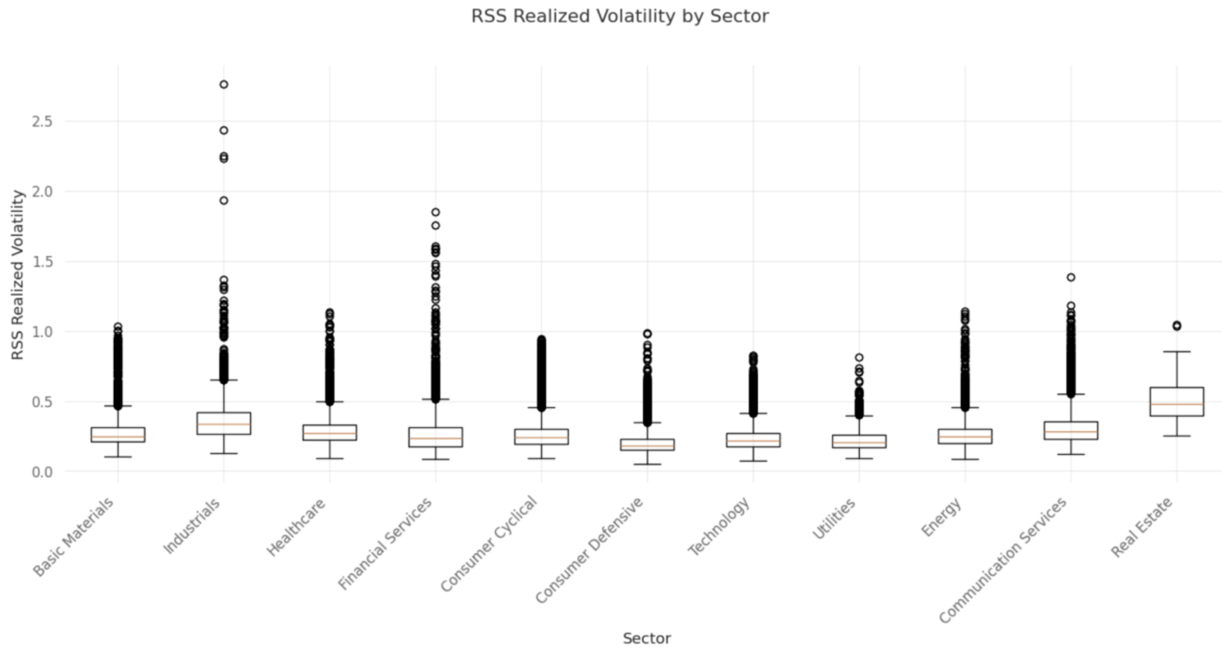

Figure 2: Distribution of Root Sum of Squares (RSS) Realized Volatility (RV) by Sector

As previously noted, the NSE has a history of frequent volatility, offering buyers the potential to earn exceptional returns on option contracts. For exploratory purposes, the payoff, profit, and return on investment (ROI) were calculated for each stock. Similar to implied volatility, the distribution of returns is highly right-skewed (57.51), inflating the expected payoff for each option type. The ten highest ROIs in the dataset were put options for stocks in the Healthcare, Financial Services, Technology, and Consumer Cyclical industries, peaking at 497,800% for Strides Pharma Science (NSE: STAR) between April and June 2018. The stock experienced a 40-50% decline during this time, attributed to corporate restructuring through the demerger with Strides Shasun, along with US FDA compliance concerns and broader pharmaceutical industry headwinds (Ramammihy, 2018).

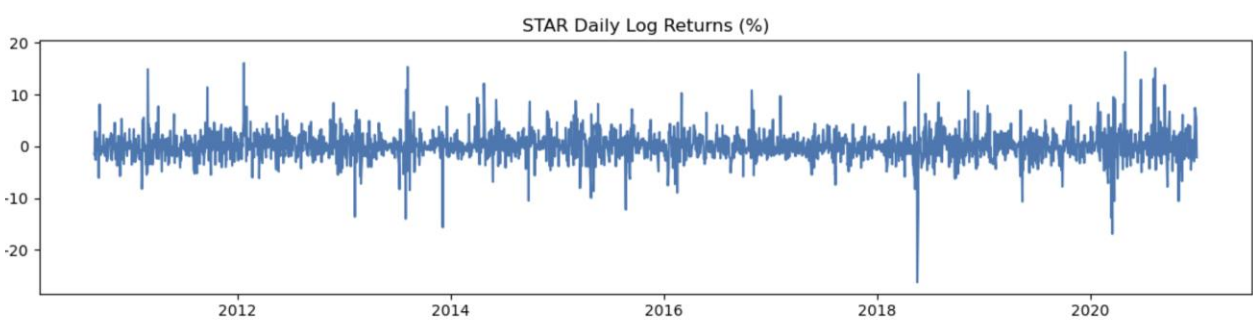

Figure 3: Strides Pharma Science (NSE: STAR) Daily Returns (08/2010 – 12/2020)

Although the median payoff for each option type is close to zero ($0.75 for calls and $0.00 for puts), the mean payoffs are significantly higher at $57.71 and $62.26, respectively. This discrepancy results in a median ROI of -100% (unexercised, worthless contracts), contrasted with a mean ROI of 190%. These figures highlight the asymmetric payoff structure of options and suggest that accurately identifying periods of heightened volatility could yield exceptionally high portfolio returns. With this potential in mind, the next section will discuss the methodology used to construct predictive models capable of identifying such opportunities.

3. Methods

3.1 Machine Learning Models

After all filters and transformations were applied, 137,802 option contracts with expiry dates between January 2011 and October 2020 remained. This data was split into a training and testing set using December 31, 2017, as the expiry date threshold. Each set of models would look to predict the realized future volatility for each measure (STD, EWMA, ROLL, RSS) using the option's implied volatility, natural log of implied volatility, option type (call or put), days to expiry, sector, 90-day historical return, and matching 90-day historical volatility.[3] The natural log of implied volatility was included to eliminate the strong skewness as discussed earlier. Other historical volatility measures were not included in the model, given the high degree of multicollinearity between features, as shown in **Table 2**.

	STD RV	EWMA RV	ROLL RV	RSS RV
STD RV	1.0000			
EWMA RV	0.9495	1.0000		
ROLL RV	0.9626	0.9368	1.0000	
RSS RV	0.9958	0.9438	0.9584	1.0000

Table 2: Pearson Correlation Matrix of Realized Volatility Measures

The motivation for this model extends from the ideas presented by Goyal and Saretto (2009) in that volatility shows a high degree of mean reversion and that lagged historical data could provide valuable information. The model created dummy variables for the option type and

[3] E.g., if STD is being predicted, the 90-day historical STD volatility was used.

sector features, using put options and the Consumer Cyclical (most frequent) sector as the baseline categories, respectively. All values were then rescaled and normalized between 0 and 1 to reduce the impact of outliers when calculating feature importances.

For each volatility measure, six models were built on the training data, or 24 models total, using the Scikit-learn Python libraries. This included an ordinary least squares regression (LR), an elastic net regression (EN), a decision tree (DTR), a random forest (RFR), a gradient boosted tree (GBR), and a neural network (NN). All models used five-fold cross-validation to prevent overfitting and root mean squared error (RMSE) as the scoring framework.[4] **Table 3** summarizes the final hyperparameters used for the most complex models, motivated by the work of Goyal and Saretto (2009), Culkin (2017), and Gan et al. (2020). In general, hyperparameters were selected to maximize predictive ability without wasting computational resources. For the RFR and GBR, 100 trees were constructed.

Model	Hyperparameters
DTR	Max Depth = 10, Min Samples per Split = 20, Min Samples per Leaf = 10, Random State = 1111
RFR	Max Depth = 10, Min Samples per Split = 20, Min Samples per Leaf = 10, Max Features = 'square root', Warm Start = True, Random State = 1111
GBR	Min Samples per Split = 20, Min Samples per Leaf = 10, Max Features = 'square root', Warm Start = 100, Random State = 1111
NN	Batch Size = 1000, Hidden Layer Sizes = (5,5,3,3), Max Iterations = 50, Alpha = 0.75, Warm Start = True, Learning Rate = 0.01, Random State = 1111, Early Stopping =True

Table 3: Hyperparameters Utilized in Machine Learning Models

After these models were trained, the best-performing model for each volatility metric was saved to be applied to the test data. The same predictive features and preprocessing steps were also applied, where the best overall model and volatility metric would be selected based on their RMSE.

[4] Root mean squared error (RMSE) was selected over R^2 as it represents a more unbiased error metric, particularly with time series data. Maximizing R^2 values often causes models to overfit while ignoring predictive accuracy. Further, RMSE does a better job at explaining the causality of a relationship. R^2 values were still calculated on test data as a point of reference.

3.2 GARCH Models

Being one of the most popular volatility forecasting methods in the financial services industry, a Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model was also constructed. The model rose to popularity given its ability to recognize volatility clustering, or the phenomenon where low and high volatility periods tend to persist. While traditional least squares (regression) models have homoskedasticity assumptions that are regularly violated in financial markets, GARCH treats this heteroskedasticity as the variance to be modeled, thereby improving its prediction accuracy (Engle, 2001).

The most commonly used specification is the GARCH(1,1) model, where the first term refers to the number of autoregressive lags (short-term persistence), and the second term refers to the number of moving average lags in the model (long-term persistence) (Engle, 2001). While widely adopted due to its simplicity, the GARCH(1,1) model assumes relatively fast mean reversion in volatility compared to higher-order variants. Thus, a GARCH(10,3) model was used, given the amount of historical data considered (seven years) and the high short-term persistence demonstrated by the underlying stocks. Daily log returns from the training data were used to forecast daily volatility throughout the testing period. This data was then sliced based on the holding period for each option, where the mean daily volatility from the period was calculated and scaled to an annual level to generate a prediction.

After computing the error metrics for both the machine learning and GARCH models on the test dataset, a simulated portfolio back test was conducted. The selected trading strategy was simple: if the predicted volatility exceeded the implied volatility, the investor would buy the option (call or put), otherwise, he/she would sell it. These trades were individually recorded in a separate data frame where the initial investment and resulting net profit were recorded.[5] This allowed for the calculation of portfolio returns under both modeling frameworks, which were then compared against a benchmark index (Nifty 50) (India, 2022).

[5] Since this analysis only considered European options, there existed no opportunity for early exercise.

4. Results

This section discusses the empirical results from the training, testing, and simulated trading exercises presented in the previous section. Since machine learning models are prone to overfitting, cross-validated training results were only considered when determining the best model to apply to the validation (testing) set. The performance of these models were scored on their root mean square error (RMSE) against the realized volatility observed over the life of the option. For the simulated trading strategy, portfolio returns were calculated and compared against the Nifty 50 benchmark index.

4.1 Regression and Machine Learning Results

Table 4 presents the performance metrics of the regression and machine learning (ML) models evaluated on the training set. Rows highlighted in yellow indicate the selected models, chosen based on their minimized root mean squared error (RMSE) and standard deviation (Std. Dev.). In general, the decision tree regressors (DTR) and random forest regressors (RFR) demonstrated the strongest performance with comparable results. Surprisingly, the neural networks (NN) underperformed, even relative to the baseline linear regression (LR) models. This outcome may be attributed to several factors, including a limited number of hidden layers, insufficient predictive features, or an inadequate number of training iterations. Additionally, the dataset contains a considerable number of outliers, which tend to favor tree-based methods due to their robustness to extreme values. Across all models, the DTR is preferred given its interpretability and ease of visualization.

	STD		**EWMA**		**ROLL**		**RSS**	
	RSME	Std. Dev	RSME	Std. Dev	RSME	Std. Dev	RSME	Std. Dev
LR	0.0919	0.0011	0.0889	0.0011	0.0835	0.0007	0.0793	0.0008
EN	0.1172	0.0005	0.1133	0.0008	0.1078	0.0002	0.1010	0.0003
DTR	0.0847	0.0011	0.0842	0.0015	0.0775	0.0006	0.0733	0.0009
RFR	0.0851	0.0010	0.0835	0.0011	0.0776	0.0007	0.0735	0.0008
GBR	0.0883	0.0009	0.0857	0.0011	0.0802	0.0006	0.0764	0.0008
NN	0.0924	0.0010	0.0896	0.0010	0.0841	0.0006	0.0804	0.0007

Table 4: Regression and ML Training Results

Figure 5 summarizes the performance of these selected models on the test data for options data between January 2018 and October 2020. The Root Sum of Squares (RSS) model demonstrates the strongest performance, exhibiting both the lowest RMSE and the lowest standard deviation among the models evaluated. On average, it produces a forecast error of 10.7%, which is particularly impressive given the high volatility of the underlying securities. These predictions will serve as the basis for the options trading strategy later in this discussion.

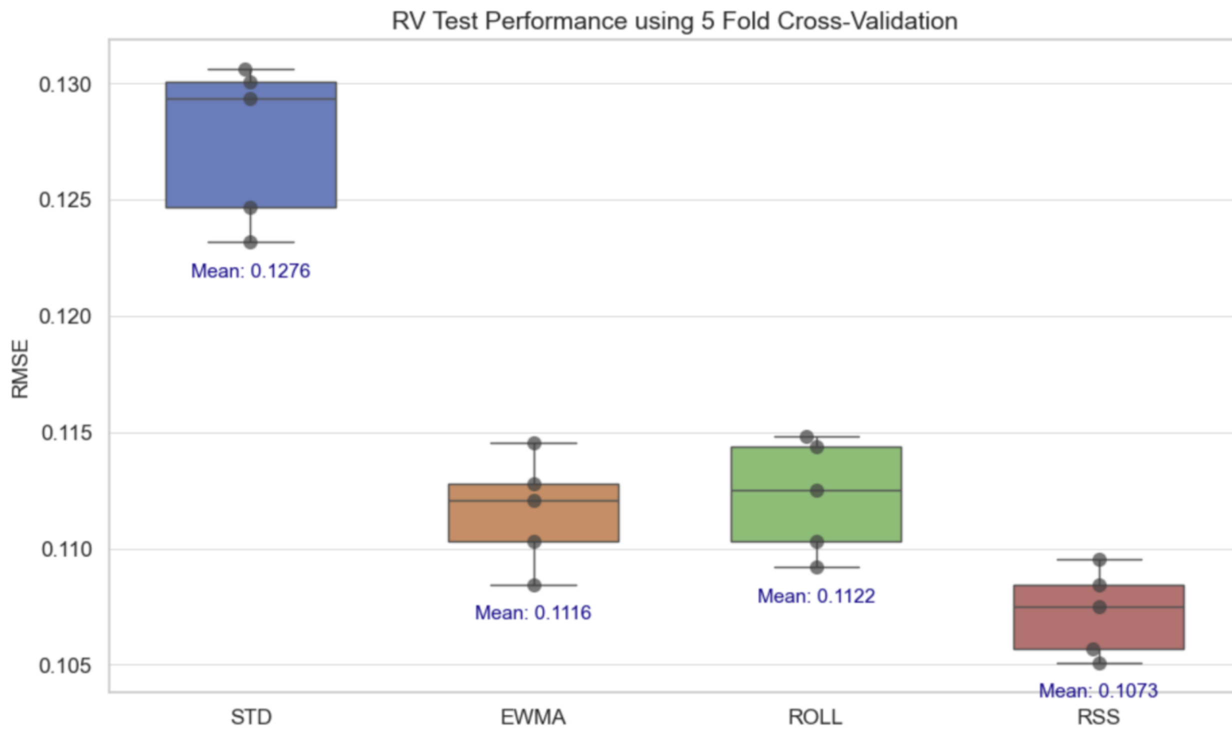

Figure 5: Regression and ML Testing Results

Since the RSS model was constructed using a decision tree, the details of its analysis can be expanded into a feature importance and tree diagram. **Figures 6 and 7** summarize this information, highlighting that the 90-day historical realized volatility and the 90-day historical return are the strongest predictors for predicting a stock's volatility. This clearly illustrates that momentum is a key factor for forecasting volatility, confirming the research done by Fama and French (2012). The Consumer Defensive and Industrials sectors emerged as the fourth and fifth most influential features, highlighting their contrasting sector composition and risk profiles. This was particularly evident in the Indian markets, where the Industrials sector experienced significant volatility over this decade (Nagaraj, 2025). Additionally, the natural log of implied volatility proved to be more informative than its raw value, confirming that the transformation effectively mitigated the impact of outliers and enhanced predictive performance.

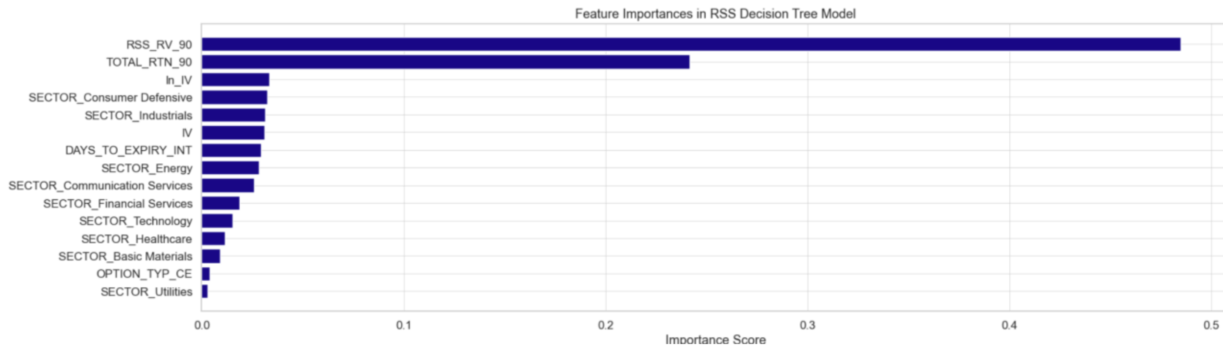

Figure 6: Feature Importance Plot of RSS Decision Tree

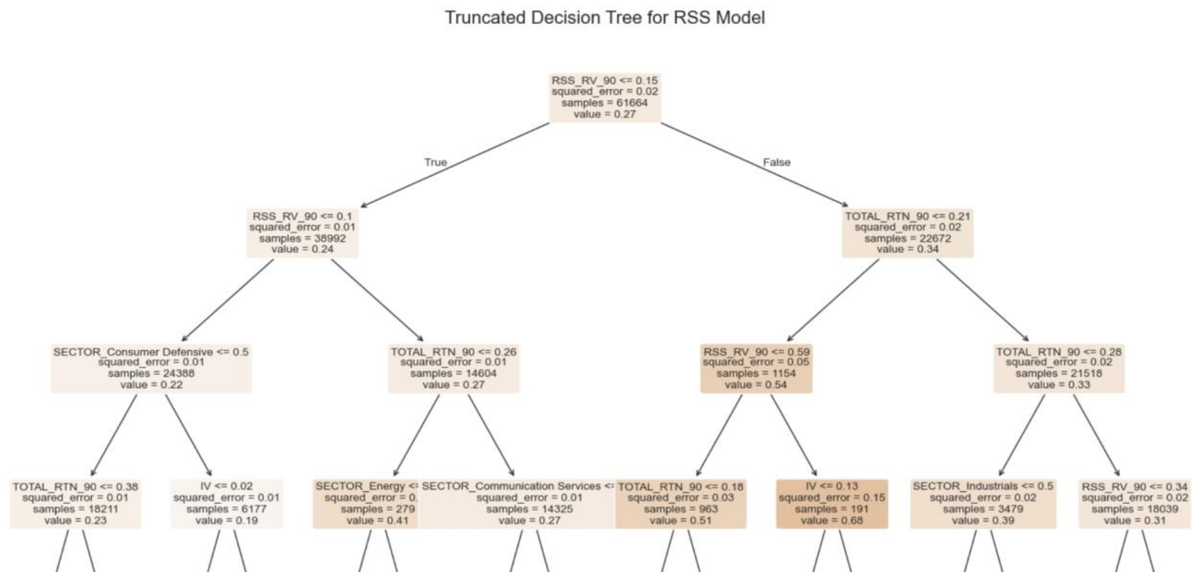

Figure 7: Truncated RSS Decision Tree

Figures 8 and 9 display the model's predictions compared to realized volatility, along with the distribution of forecast errors. The model performs well for securities whose realized volatility is less than 100%. While it tends to underperform for highly volatile securities – evident in the long right tail of the error distribution in **Figure 9** – it still produces reasonable forecasts, often anticipating volatility levels at or above 100%. This is particularly notable given that the model relies solely on descriptive numerical variables and does not incorporate a sentiment analysis of recent news or macroeconomic variables. The model is also unbiased, with a majority of forecast errors falling within ±4% and a median error of -1.38%.

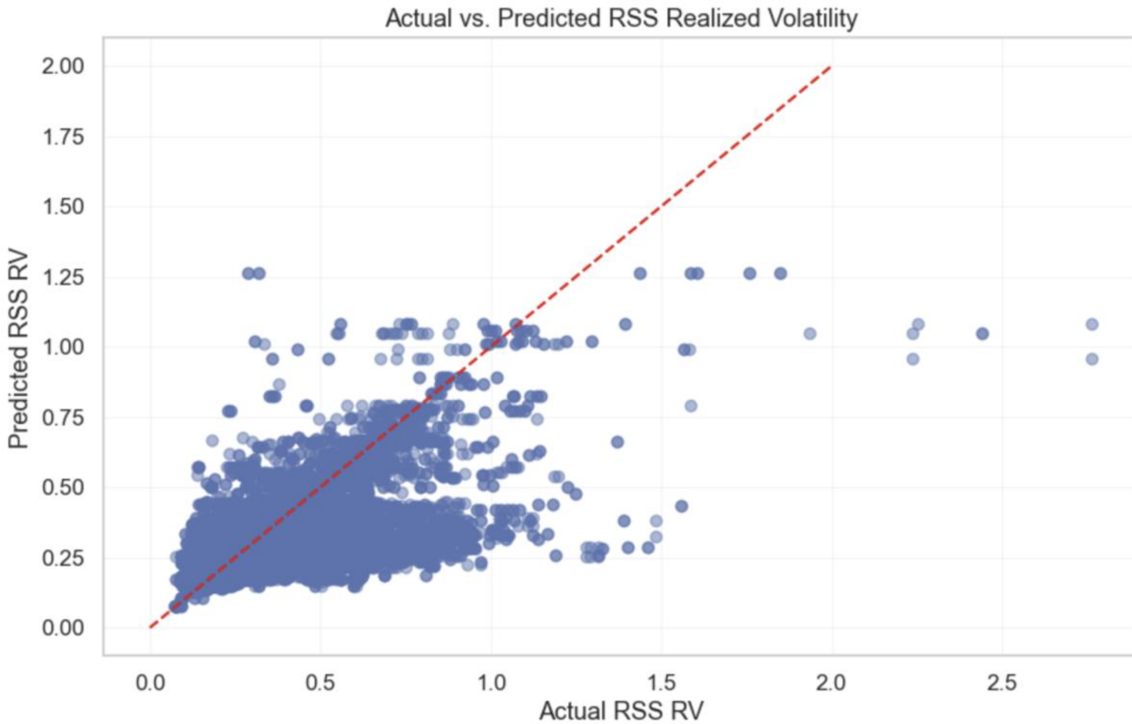

Figure 8: Comparison of Predicted RSS RV and Realized RSS RV

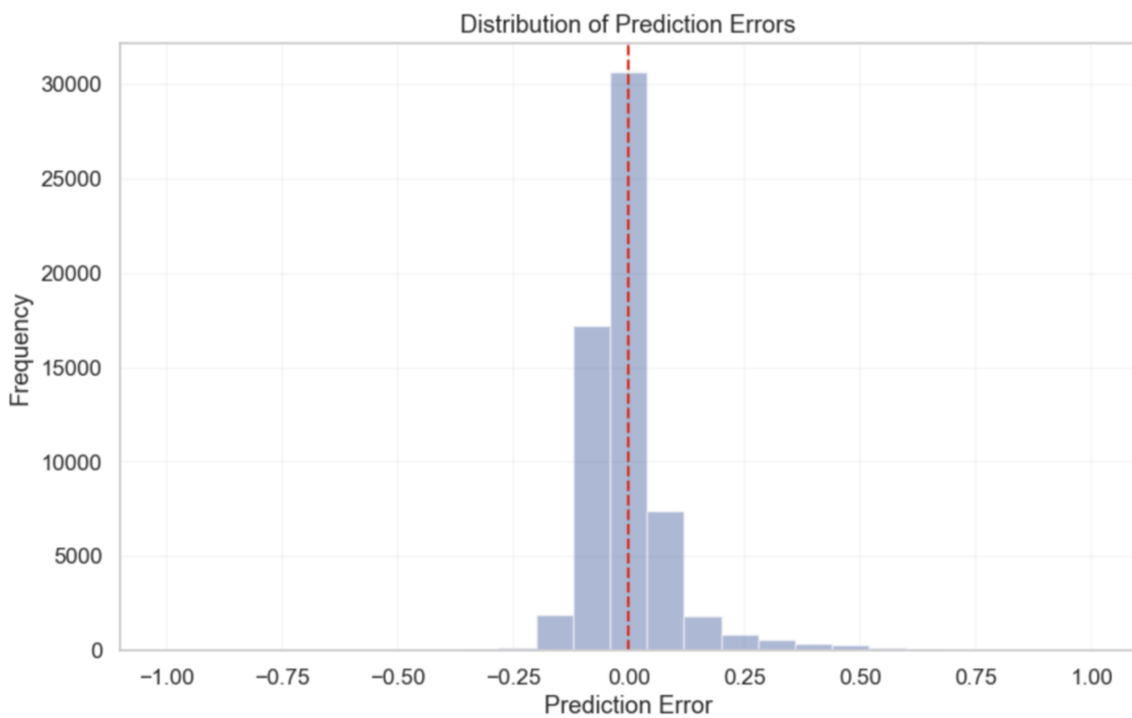

Figure 9: Distribution of Prediction Error for RSS Volatility Model

4.2 GARCH Results and Model Comparison

Figure 10 summarizes the forecasted GARCH volatility for one of the equities covered in this analysis. While a GARCH(1,1) model was initially developed as a baseline forecast, a GARCH(10,3) specification was ultimately employed to better account for noise in the data and capture the heightened variability of the underlying assets. However, as illustrated in the figure, the model continues to underperform over longer horizons, with volatility estimates converging to a steady state after approximately one month of observations.[6] In comparison to the RSS model, the GARCH(10,3) performs reasonably well through March 2020 in capturing average volatility. However, it struggles in the aftermath of COVID-19, when market conditions became highly erratic. Despite accounting for additional jumps in the data, the model remains overly stable and fails to adapt to the heightened underlying volatility, leading to weak performance across all evaluated metrics.

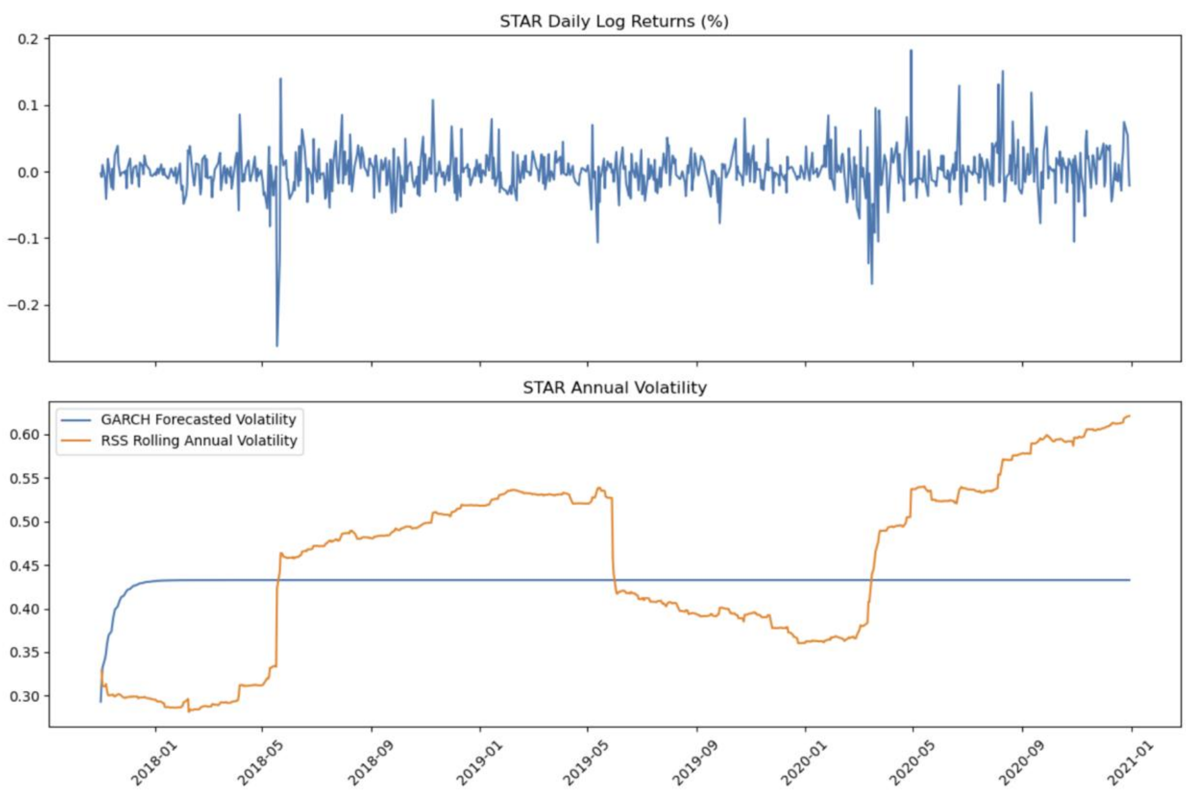

Figure 10: RSS and GARCH Volatility Forecasts against Underlying Price Movement

[6] This was an improvement over the GARCH(1,1) model, which reached its long-term average after two weeks of forecasting. Increased variability was shown across other equities, ultimately improving the model's predictive accuracy.

After all models were built and forecasted over the testing period, the R^2, Mean Absolute Error (MAE), and RMSE were collected to analyze their performance, which are summarized in **Table 5**. The RSS model (DT) showed encouraging performance, with forecast errors of 6% and an R^2 of 46%. The GARCH model ultimately struggled due to its inability to capture underlying asset trends beyond a one-month horizon. This limitation suggests the potential benefit of constructing additional models that layer or ensemble previous results to improve long-term forecasts. In contrast, the RSS model delivered the most impressive performance, especially considering the limited scope of input features and the high volatility of the assets involved. These results indicate that the RSS model may contribute to generating a positive alpha within a portfolio trading strategy.

	RSS	**GARCH**
R^2	0.4598	-0.0293
MAE	0.0612	0.1072
RMSE	0.1016	0.1402

Table 5: Final Model Performance Results

4.3 Simulated Trading Strategy

To apply the results of the best-performing models, a simulated trading strategy was conducted to determine if using these forecasts can outperform the market. Running profit and returns metrics were calculated for a ₹100M portfolio, where the monthly results are displayed in **Figure 11**. All trades assumed the investor bought or sold a single contract. Over the 34 months of data tested, the RSS Portfolio returned -0.7%, the GARCH portfolio returned -2.8%, and the Nifty 50 Index returned 6.0%. Overall, both trading strategies were stable, regularly offsetting any gains with losses in the following months. Since this analysis only considered ATM options, these results emphasize the unpredictability that exists when trading on contracts with short expiry periods. Monthly profits for both strategies are summarized in **Table 6**, illustrating the difficulties each of these models faced. In the 34 months tested, the GARCH strategy recorded a positive profit in five of them, compared to 14 for RSS. Although the models initially outperformed the broader index through the onset of COVID-19, the index's subsequent rebound enabled it to surpass these strategies by July 2020. Despite the heightened volatility observed after March 2020, the RSS and GARCH strategies continued to yield similar results. These findings suggest the need for a more sophisticated approach, such as restricting trades to securities for which both models recommend the same action. In general, despite the strong performance from the RSS model in out-of-sample forecasting, both models struggled when applied to a portfolio backtesting framework.

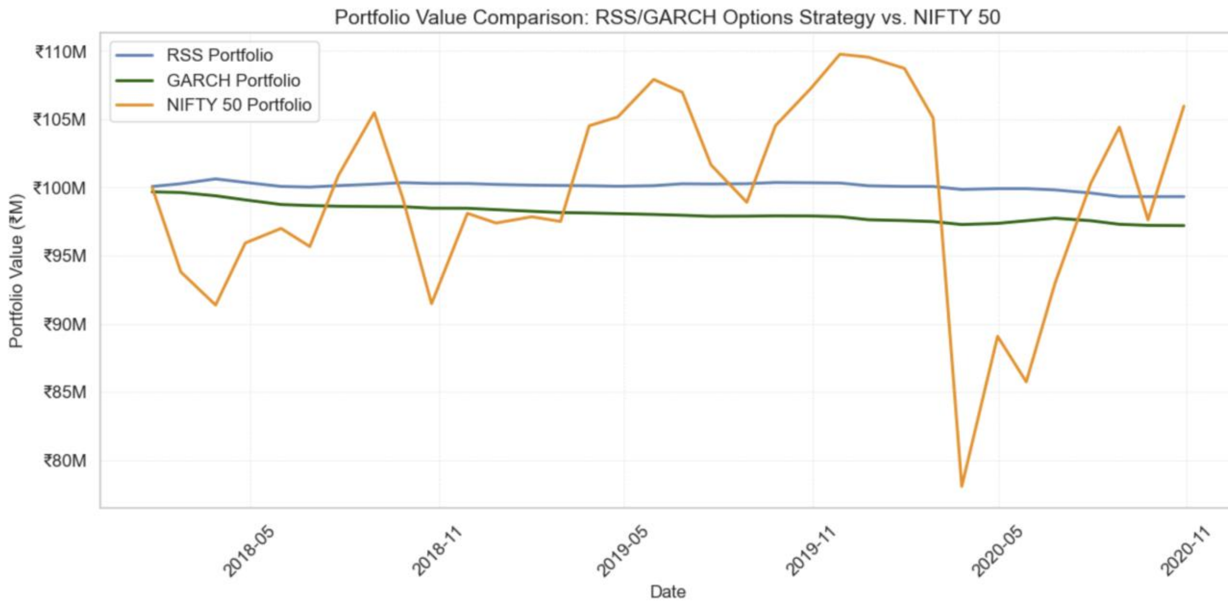

Figure 11: Backtest Results from Proposed Trading Strategy

	Mean	Median	Std. Dev.	Minimum	Maximum
RSS Portfolio (₹)	-19,940	-17,256	136,759	-296,372	354,028
GARCH Portfolio (₹)	-82,540	-62,190	124,621	-338,937	191,747

Table 6: Monthly Profit/Loss Statistics for Proposed Trading Strategy

5. Conclusion

This analysis has demonstrated numerous takeaways and the potential for a more thorough analysis in the future. Among the four volatility metrics evaluated, the Root Sum of Squares (RSS) algorithm consistently demonstrated superior performance when implemented through a decision tree framework. Momentum was the most influential predictor, focusing on historical volatility and total return from the previous 90 days. The stock's implied volatility and appearance in the consumer defensive or industrials sectors also proved as key indicators. Although a GARCH model was built in comparison, its performance lagged considerably, failing to capture key underlying trends beyond a one-month horizon. The applied trading strategy revealed that neither model could generate substantial returns over the long run. Despite RSS's lower error metrics, it was unable to produce consistent gains, even during periods of heightened volatility.

In general, these results indicate that additional features such as consumer sentiment derived from news articles, correlated asset statistics, or macroeconomic factors could improve the predictive ability of the model. As this analysis was centered around short-dated, ATM options, expanding to consider options with longer expiries (up to one year), different moneyness (ITM or OTM), different contract styles (American), and a longer training period could improve robustness. Further, the inclusion of price-based volatility metrics such as Parkinson's or Garman-Klass could capture more nuanced dynamics. The GARCH model could benefit from the most refinement, potentially through additional tuning of its lagging parameters or stacking models in three- to six-month intervals to mitigate mean reversion effects. Subsequent training should utilize more historical data or the application to a different stock exchange. More complex architectures would also benefit neural networks, which notably struggled in this analysis.

Nevertheless, the performance of RSS given these few features is noteworthy, indicating that the most significant improvement can be made to the trading strategy. The strategy was very straightforward, ignoring the option to take no action. This is arguably unrealistic, especially when transaction fees are considered for portfolios of this magnitude. Future work should consider only making a trade recommendation if the models agree, or the forecasted volatility deviates significantly from the implied volatility (by a margin of at least 10%).

This study reinforces the understanding that accurately modeling future volatility does not guarantee portfolio outperformance, and that a robust trading strategy is equally critical for success. Security weighting, portfolio concentration, and regulatory guidelines were ignored in this study, representing a significant limitation if portfolio optimization is to be addressed in subsequent research. Moving forward, building on these ideas can facilitate the development of more sophisticated volatility algorithms and trading strategies, enabling asset managers to more accurately predict market variability and effectively manage their risk.

References

Agarwal, T. (2020). *NSE India Futures & Options Daily (2000-20)*. Kaggle.com; Kaggle. https://www.kaggle.com/datasets/tanay001/nseindia-futures-options-daily

Culkin, R. (2017). Machine Learning in Finance: The Case of Deep Learning for Option Pricing.

Engle, R. F. (2001). *GARCH 101: An Introduction to the Use of Arch/Garch Models in Applied Econometrics*. SSRN. https://ssrn.com/abstract=1294571

ET Online. (2020, March 6). *Diamonds to Dust: How Yes Bank plunged from peak to bottom in just 17 months*. The Economic Times; Economic Times. https://economictimes.indiatimes.com/industry/banking/finance/banking/diamonds-to-dust-how-yes-bank-plunged-from-peak-to-bottom-in-just-17-months/articleshow/74507734.cms

Fama, E. F., & French, K. R. (2012). Size, value, and momentum in international stock returns. *Journal of Financial Economics*, *105*(3), 457–472. https://doi.org/10.1016/j.jfineco.2012.05.011

Gan, L., Wang, H., & Yang, Z. (2020). Machine learning solutions to challenges in finance: An application to the pricing of financial products. *Technological Forecasting and Social Change*, *153*, 119928. https://doi.org/10.1016/j.techfore.2020.119928

Ganti, A. (2024, June 12). *Implied Volatility - IV*. Investopedia. https://www.investopedia.com/terms/i/iv.asp

Goyal, A., & Saretto, A. (2009). Cross-section of option returns and volatility☆. *Journal of Financial Economics*, *94*(2), 310–326. https://doi.org/10.1016/j.jfineco.2009.01.001

Gurung, M. (2025, January 20). *Realized Volatility: What Is It, Calculation, Importance & More* (R. Rajan, Ed.). Investing.com. https://www.investing.com/academy/analysis/realized-volatility-definition/

India, N. (2022). *NIFTY 50 Live | NSE Nifty 50 Index Today - NSE India*. NSE India. https://www.nseindia.com/products-services/indices-nifty50-index

International Monetary Fund. (2025, January 1). *Interest Rates, Discount Rate for India*. FRED, Federal Reserve Bank of St. Louis. https://fred.stlouisfed.org/series/INTDSRINM193N

Nagaraj, R. (2025). India's premature deindustrialization and falling investment rate in the 2010s. *World Development*, *191*, 106954. https://doi.org/10.1016/j.worlddev.2025.106954

Quant Next. (2023, July 3). *Implied Volatility Calculation with Newton-Raphson Algorithm*. Quant Next. https://quant-next.com/implied-volatility-calculation-with-newton-raphson-algorithm/

Ramammihy, M. (2018, April 18). *Strides Shasun Demerger*. Strides Shasun. https://www.strides.com/pdf/investor_shareholderinfo/de_merger_of_commodity_api_business/intimation_to_se_on_cost_of_acquisition.pdf

Rangan, M. G. (2019, May 3). *The fall of NSE: Corruption or hubris?* The Economic Times; Economic Times. https://economictimes.indiatimes.com/markets/stocks/news/the-fall-of-nse-corruption-or-hubris/articleshow/69154289.cms

SBI Securities. (2024, October 10). *The Hidden Story behind BSE and NSE: Why India Needed Two Stock Exchanges*. LinkedIn. https://www.linkedin.com/pulse/hidden-story-behind-bse-nse-why-india-needed-two-stock-exchanges-nv9bc/