EE 691: R & D

Final Report
Volatility Forecasting: Evidence from India

Bhavishya (19D070017)

May 4, 2023

Guided by **Prof. Piyush Pandey & Prof. V Rajbabu**
Indian Institute of Technology Bombay

Contents

1 Abstract

Volatility forecasting is crucial in finance for predicting uncertainty in financial markets. Advanced machine learning (ML) models, such as random forest (RF), artificial neural network (ANN), recurrent neural network (RNN), and long short-term memory (LSTM), are increasingly used due to their ability to capture complex non-linear relationships in financial data. This paper evaluates the performance of these models and compares them to the benchmark autoregressive integrated moving average (ARIMA) model using daily stock returns for major global stock indices. The study finds that ML models generally outperform the ARIMA model, with LSTM achieving the best results. Proper selection of input variables and model parameters is crucial for ML-based volatility forecasting. These findings have important implications for finance practitioners seeking to improve their predictions of volatility and make better investment decisions.

This is a team project (two students) with Aaryan Gupta. This report contains the part (done mainly by Bhavishya) in which the problem is approached through a Machine Learning point of view. Various methods are implemented and tried like Random Forests, Artificial Neural Networks, Recurrent Neural Networks, and LSTMs. Methods are then evaluated using appropriate metrics and compared with statistical methods (ARIMA). Lastly, various feature importance techniques are also implemented to get a deeper insight into the behavior of the market. The joint presentation for the project is here[1].

2 Time series forecasting

Time series forecasting is a crucial task for many industries, including finance, economics, and marketing. It involves using statistical models to predict future values of a variable based on its past behavior. ARIMA models are one of the most popular statistical models used for time series forecasting due to their ability to capture trends and seasonality in the data. In this article, we will explore what ARIMA models are, how they work, and how to use them for time series forecasting.

3 ARIMA, SARIMA and SARIMAX Models

3.1 Theory

When dealing with time series data, it is common to encounter patterns such as trend and seasonality. A trend refers to a long-term increase or decrease in the values of the time series data, while seasonality refers to a pattern that repeats itself over a fixed time interval.

To capture both trend and seasonality in the time series data, we can use a more complex version of the ARMA model called the Autoregressive Integrated Moving Average (ARIMA) model. The ARIMA model includes an additional parameter, d, which represents the degree of difference needed to make the time series data stationary. The "I" in ARIMA stands for "integrated", which refers to the differencing operation.

The Seasonal Autoregressive Integrated Moving Average (SARIMA) model is an extension of the ARIMA model that can handle time series data with both trend and seasonality. The SARIMA model includes additional parameters to capture the seasonal patterns in the data.

Seasonality refers to patterns in the data that repeat themselves over fixed time intervals, such as daily, weekly, or monthly. These seasonal patterns can have a significant impact on the time

series data and can be difficult to capture using traditional ARIMA models.

The SARIMA model can be written mathematically as:

$$y'_t = c + \phi_1 y'_{t-1} + \phi_2 y'_{t-2} + ... + \phi_p y'_{t-p} + \Phi_1 y'_{t-s} + \Phi_2 y'_{t-2s} + ... + \Phi_P y'_{t-Ps} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q} + \Theta_1 \epsilon_{t-s} + \Theta_2 \epsilon_{t-2s} + ... + \Theta_Q \epsilon_{t-Qs} \quad (1)$$

where y'_t is the differenced value of the variable at time t, c is a constant, $\phi_1, \phi_2, ..., \phi_p$ are the parameters of the autoregressive (AR) component, $\Phi_1, \Phi_2, ..., \Phi_P$ are the parameters of the seasonal autoregressive (SAR) component, $\theta_1, \theta_2, ..., \theta_q$ are the parameters of the moving average (MA) component, $\Theta_1, \Theta_2, ..., \Theta_Q$ are the parameters of the seasonal moving average (SMA) component, ϵ_t is the error term at time t, s is the length of the seasonal cycle, and p, q, P, and Q are the orders of the AR, MA, SAR, and SMA components, respectively.

Determining the optimal values for model order can be a challenging task. However, the auto arima function in Python's statsmodels library uses an iterative approach to automatically find the optimal values for these parameters based on minimizing the AIC (Akaike Information Criterion) or BIC (Bayesian Information Criterion). The auto arima function saves a significant amount of time and effort by automating the process of selecting the order of the SARIMA model.

The Seasonal Autoregressive Integrated Moving Average with eXogenous variables (SARIMAX) model is an extension of the SARIMA model that can handle time series data with both trend, seasonality, and exogenous variables. The SARIMAX model includes additional parameters to capture the impact of exogenous variables on the time series data.

Exogenous variables refer to external factors that can influence the time series data, such as weather patterns, economic indicators, or marketing campaigns. The inclusion of exogenous variables in the model can lead to more accurate forecasts by accounting for the impact of these external factors.

3.2 Implementation of SARIMA

For our data, the order is (2,1,2)(0,0,0,20). In our data, we have checked for seasonality using m = 20 (trading days in a month), so as to capture monthly repeating patterns.

Using a train-test-split of 0.9, the model has been trained on the train-data and used to forecast the test-data. The fitted diagnostics have been shown in Fig **??** and the forecast plot has been shown in Fig 1. The R2 score obtain was **96%** with an MSE of **1.07**.

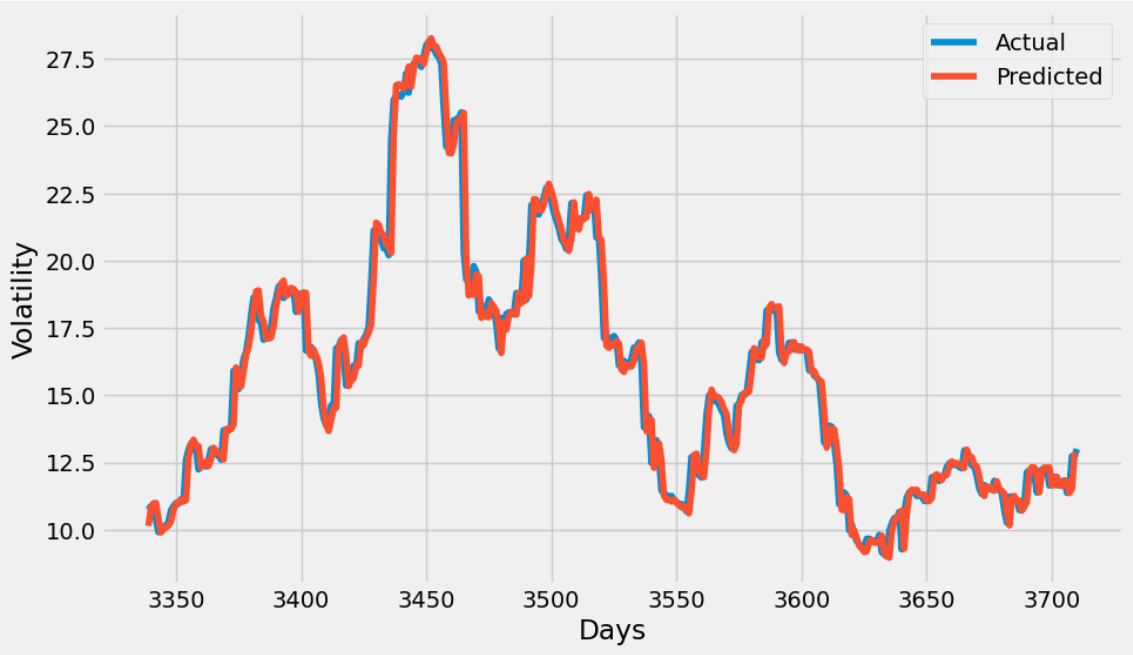


Figure 1: Forecast results

3.3 Implementation of SARIMAX

When selecting the optimal order of the SARIMAX model with exogenous variables, the same approach can be used as for a regular SARIMA model, by minimizing the AIC or BIC. The order of the SARIMAX model includes the number of exogenous variables to be included in the model, denoted by the variable x in the order notation (p, d, q, P, D, Q, m, x). The order of our model is (0,1,0,2,0,0,20).

3.3.1 Model Performance

Using a train-test-split of 0.9, the model has been trained on the train-data and used to forecast the test-data. The exogenous variable introduced have caused to increase the model performance, however, the effect is minimal.

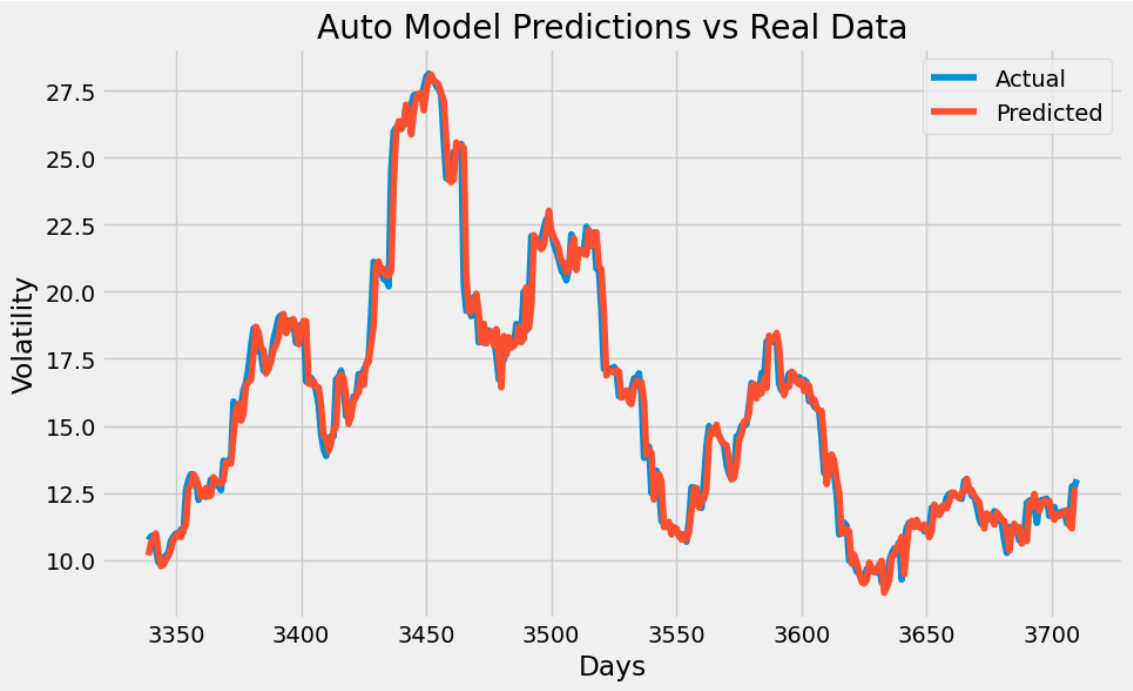


Figure 2: Model performance

Coming to the other major reason to implement this model, we analyze the statistical significance of each of these features (exogenous variables) to accurately determine which variables significantly affect Indian market volatility. This has been captured using p-value of the variables and we see that all the variables except Crude Oil futures and Gold have p-values less than 0.05, i.e., significant at 5% confidence. Even if these variables are removed no effect is seen on the model performance, the feature importance, or the significance of other exogenous variables.

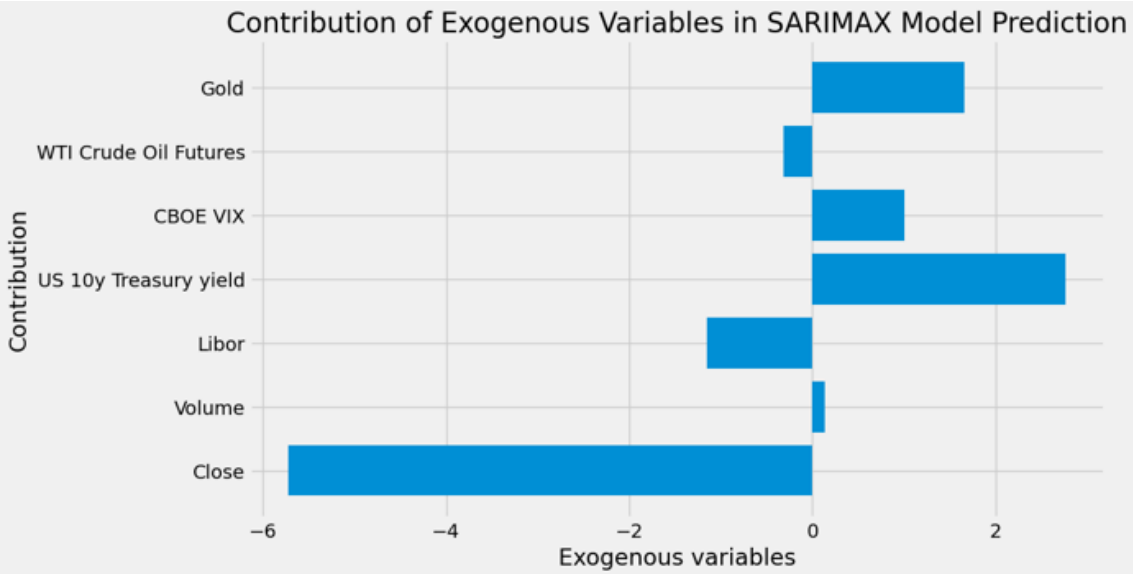


Figure 3: Feature Importance of Exogenous variables

4 Why Machine Learning?

Machine learning has become increasingly popular due to its flexibility and adaptability to complex patterns. It does not require explicit assumptions about the data distribution and can handle large datasets with a high number of variables. Moreover, machine learning models can capture temporal dependencies and long-term patterns, utilizing recurrent neural networks (RNNs) and convolutional neural networks (CNNs). However, there are some limitations to machine learning, including interpretability issues, large data requirements, and sensitivity to noise and outliers due to the absence of assumptions such as stationarity, linearity, and normality.

5 Data

The study used a dataset comprising daily data of Nifty30d volatility for the period of April 2008 to April 2023. Nifty30d volatility is a forward-looking measure of volatility in the Indian stock market, which is derived from the implied volatility of the next expiring options on the Nifty index. The dataset was obtained from Bloomberg.

The study also utilized several other variables which have an economic significance and are potential factors that could affect the Indian market volatility. These variables include:

- Nifty close prices: The daily closing prices of the Nifty 50 index were used as a proxy for the overall performance of the Indian stock market. Nifty close prices are considered a crucial factor in predicting future volatility as they reflect the behavior of the market participants.
- Nifty volume: The daily trading volume of the Nifty 50 index was included as a factor in the study as it is an important indicator of the level of market activity and liquidity. Higher trading volumes are often associated with higher volatility.

- CBOE VIX: The Chicago Board Options Exchange (CBOE) Volatility Index (VIX) was included in the study as a measure of the market's expectation of future volatility of the $S\&P$ 500 index. The VIX is often used as a proxy for global risk sentiment and can impact the Indian market volatility.
- US 10-year Treasury yield: The daily yield on US 10-year Treasury bonds was included in the study as a proxy for the global risk-free rate. The yield on the US Treasury bonds can impact the cost of capital for investors and hence affect the volatility of emerging market assets.
- LIBOR: The daily London Interbank Offered Rate (LIBOR) was included as a measure of the global interbank lending rate. Changes in LIBOR rates can affect the borrowing costs for market participants and impact the volatility of the Indian stock market.
- WTI crude oil futures: The daily prices of West Texas Intermediate (WTI) crude oil futures were included as a proxy for the global oil market. Changes in oil prices can impact the profitability and cost structure of firms in various sectors and can affect market volatility.
- Gold: The daily prices of gold were included in the study as a proxy for the global safe-haven asset. Changes in gold prices can reflect shifts in the global risk sentiment and can impact the volatility of emerging market assets.

6 Data Processing

Time series dataset is transformed into supervised learning using a sliding-window representation as shown in the figure. We have to take care that model must be trained on the past and predict the future. So, the training set is the part and test set is in the feature. There are three parameters

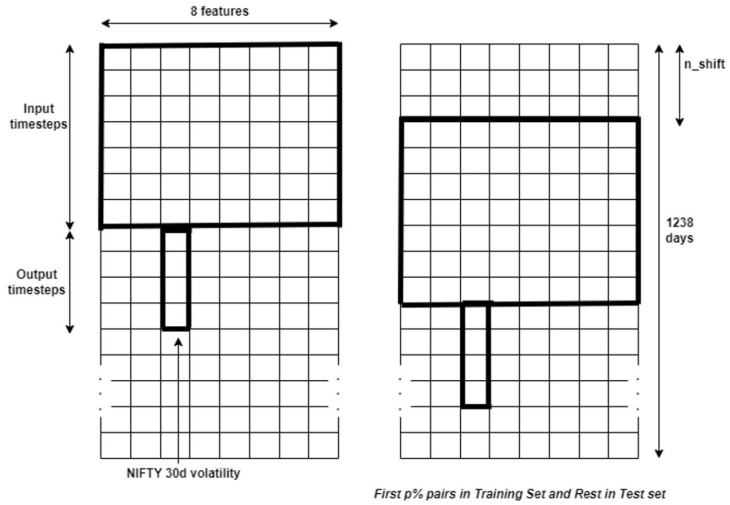


Figure 4: Time series dataset

used to make dataser which are Input Timesteps (number of days we are willing to go in past and use that information for prediction), Output Timesteps (number of days in the future we want to

predict volatility for), and N-Shift (the shift of days between two consecutive input-output pairs). Later in the project, these parameters are also varied and the trend are noted.

7 Random Forest

Random Forest is a machine learning algorithm that is commonly used for financial time series prediction. In a financial time series, the goal is to predict the future value of a financial variable, such as stock prices, based on historical data. Random Forest works by creating a large number of decision trees and then combining the predictions of each tree to obtain a final prediction. Each decision tree is trained on a random subset of the available data, and at each split in the tree, only a subset of the available features are considered. This helps to reduce overfitting and improve the generalization performance of the model. Random Forest can be used for both regression and classification tasks, depending on the nature of the problem. For financial time series prediction, it is typically used for regression tasks to predict the continuous values of the financial variable.

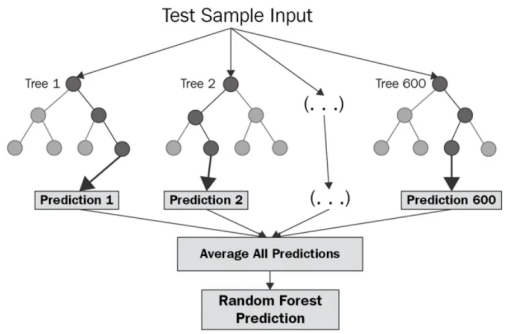


Figure 5: Source: Random Forrest Prediction [3]

7.1 Parameters and Results

Input timesteps = 125, Output timesteps = 1, Nshift = 1, Total number of features = 125 8 = 1000, Train:Test = 85:15

We evaluated each model using two metrics: mean squared error (MSE) and R2 score. MSE is a measure of the average squared difference between the predicted and actual values, while R2 score is a measure of how well the model fits the data compared to a simple baseline model. Mean square error of the output is found to be 31.3e-5, and R2 square is 0.9256.

7.2 Impurity-based Feature Importance

The feature importance in Random Forest is calculated using the average impurity method[2], which is based on the concept of Gini impurity. Gini impurity is a measure of the impurity or randomness of a node in a decision tree. The feature importance is calculated as the decrease in Gini impurity that results from splitting a node on a particular feature. When training a Random Forest model,

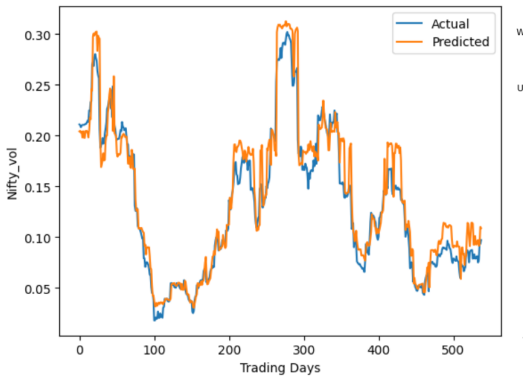


Figure 6: Prediction using Random Forrest

the algorithm creates a large number of decision trees, each of which is trained on a random subset of the available data and a random subset of the available features.

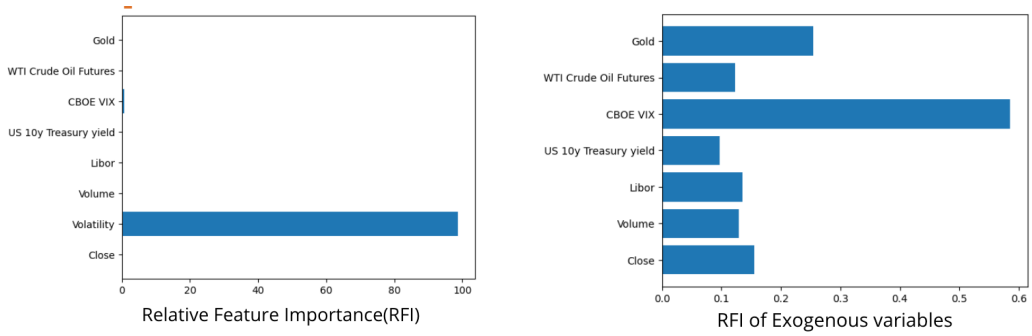


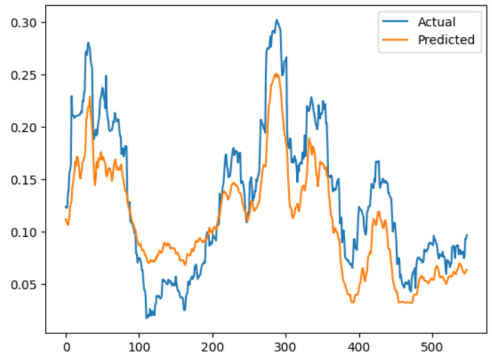
Figure 7: Results of Average-Impurity-based Feature Importance

To calculate the feature importance, the algorithm examines each decision tree and calculates the average decrease in Gini impurity for each feature over all the decision trees.The feature importance values range from 0 to 1, with higher values indicating greater importance. The feature with the highest importance is considered the most important feature for predicting the target variable.

8 Neural Networks

Artificial neural networks (ANNs) are a type of machine learning algorithm that can be used for time series prediction. ANNs are particularly well-suited for this task because they can learn complex non-linear relationships between input features and the target variable, making them useful for modeling time-dependent processes.

Some authors suggest that ANN can adequately adjust both seasonality and the linear trend of a time series, based on the fact that ANN are capable of modelling any arbitrary function *(Franses Draisma, 1995)*. Other authors claim that despite being universal function approximators, ANN can benefit from the previous elimination of systematic components, thereby focusing on learning the most complex aspects of the series *(Nelson, Hill, Remus, O'Connor, 1999)*



Layer (type)	Output Shape	Param #
dense (Dense)	(None, 32)	15904
dense_1 (Dense)	(None, 16)	528
dense_2 (Dense)	(None, 8)	136
dense_3 (Dense)	(None, 4)	36
dense_4 (Dense)	(None, 1)	5

Total params: 16,609
Trainable params: 16,609
Non-trainable params: 0

Figure 8: Left: Prediciton results using ANNs (x-axis are days, and y-axis is volatility)
Right: Model Specifications

8.1 Parameters and Results

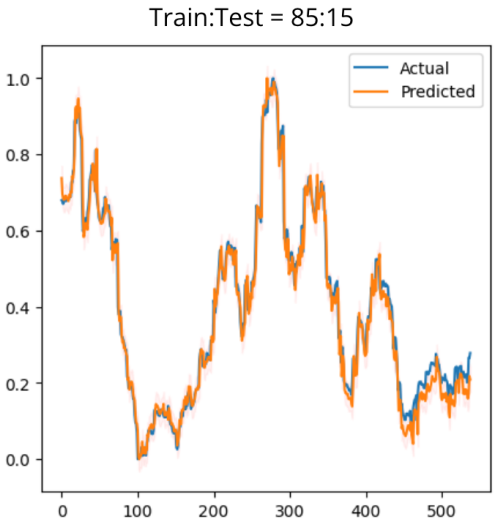
- Train:Test = 85:15
- 128 input days
- Output Timesteps = 1
- N Shift = 1
- MSE = 151e-5
- R2 score: 0.6896

9 RNN-LSTMs

Recurrent neural networks (RNNs) are a type of artificial neural network that are commonly used for processing sequential data such as time-series. The key feature of RNNs is that they have a feedback loop that allows information to be passed from one time-step to the next, making them well-suited for modeling time-dependent processes. For the time-series forecasting task, we can think of your RNN model as a type of "memory machine" that is able to remember the previous values in the time-series and use them to make predictions about future values. This is a powerful capability that allows the RNN model to capture complex patterns and relationships in the data that may be difficult to detect using traditional statistical methods.

LSTMs, these are a type of RNN that are designed to address some of the limitations of traditional RNNs, such as the "vanishing gradient" problem that can occur when training deep networks. LSTMs use a more sophisticated architecture that includes a "memory cell" that can selectively remember or forget previous values in the time-series, making them well-suited for long-term dependencies.

LSTM+Attention model takes this one step further by adding an attention mechanism that allows the model to focus its attention on specific parts of the time-series that are most relevant for making predictions. The attention weights on rows select those variables that are helpful for



128 days, Output_Timesteps = 1, N_Shift = 1

Model	MSE	R2 Score
RNN	.00226	.9599
LSTMs	.00141	.9779
LSTM + Attention	.00432	.9333
BiLSTM + Attention	.00126	.9794

Figure 9: (Left: Prediction using LSTMs), (Right: Individual Performances of the four models)

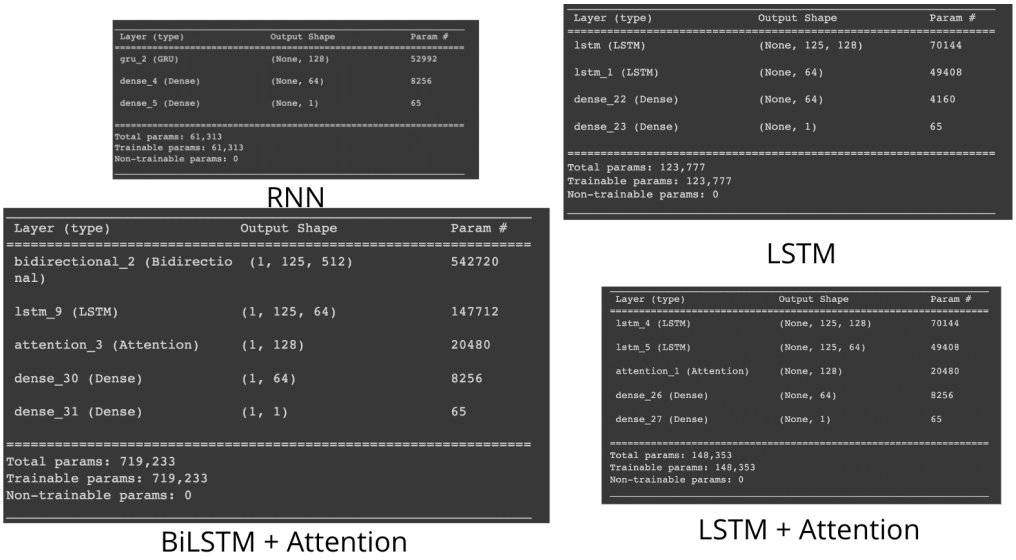


Figure 10: (Left: Prediction using LSTMs), (Right: Individual Performances of the four models)

forecasting. Since the context vector vt is now the weighted sum of the row vectors containing the information across multiple time steps, it captures temporal information *(Shun-Yao Shih et al.)* This is a useful capability when dealing with complex time-series data where different parts of the time-series may have different levels of importance.

BiLSTM+Attention model is a bidirectional LSTM that is able to process the time-series both forwards and backwards. This allows the model to capture information from both past and future time-steps, making it well-suited for tasks such as predicting trend changes or sudden spikes in volatility.

9.1 Other Parameter Trends

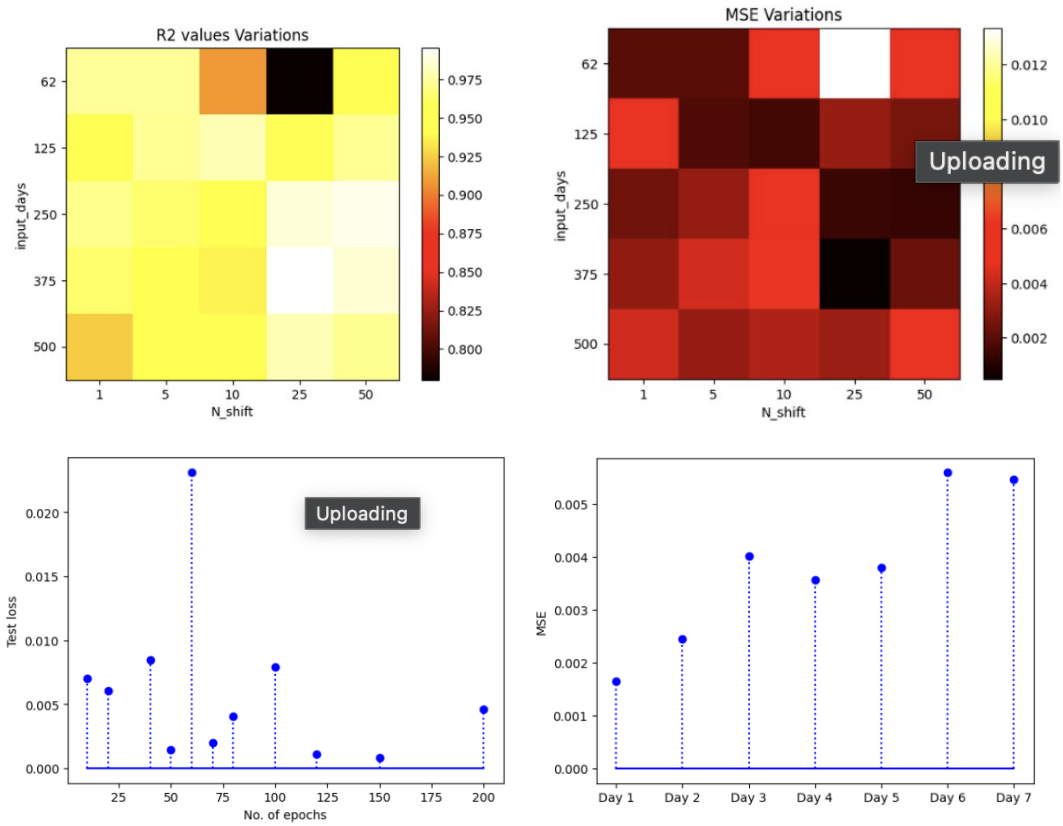


Figure 11: (Above: R2 value and MSE values on varying input-days and N-shift scales), (Below-Left: Test loss on varying number of training epochs), and (Below-Right: Individual MSEs from Day1 to Day7 when output-timesteps is 7)

9.2 Permutation Feature Importance

Permutation feature importance[4] is a popular method for determining the importance of features in a machine learning model. This method involves measuring the impact of permuting or shuffling the values of a feature on the model's performance. The idea is that if a feature is important, shuffling its values should significantly reduce the model's performance, while shuffling an unimportant feature should have little effect.

To calculate permutation feature importance, the values of a single feature in the test set are randomly permuted and then the model is used to predict the target variable using the permuted data. The resulting reduction in the model's performance is then used as an indication of the feature's importance. This process is repeated for all features in the model to determine their relative importance.

Permutation feature importance is a powerful technique because it provides a model-agnostic way to evaluate feature importance. This means that it can be applied to any machine learning

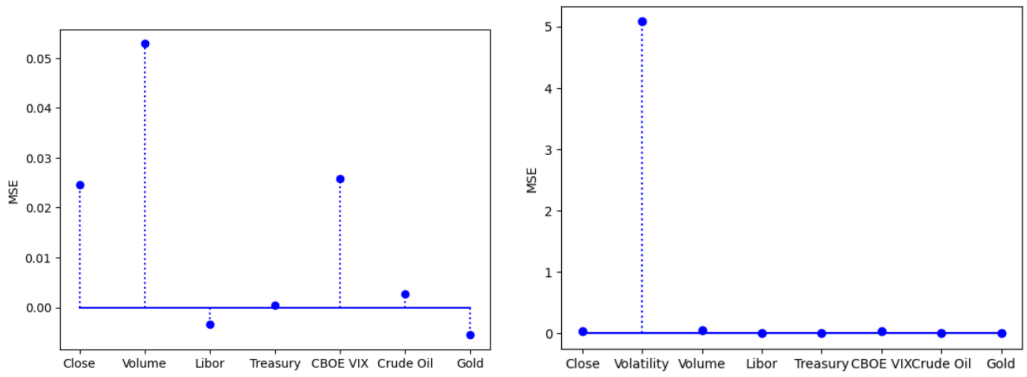


Figure 12: (Left: Relative Feature Importance of Exogenous variables), (Right: Feature Importance of all variables)

model regardless of the underlying algorithm. It also has the advantage of being computationally efficient and easy to interpret.

9.3 Exclusion-based Feature Importance

Finding Feature Importance through Exclusion [5] is a method for estimating the importance of a feature in a machine learning model. This method involves training the model multiple times with different features excluded, and then measuring the impact of each exclusion on the model's performance.

To use this method, one can start by training the model with all features included. Then, the model is trained again with one feature excluded, and the resulting loss is compared to the loss from the original model. The larger the increase in loss when a feature is excluded, the more important that feature is considered to be.

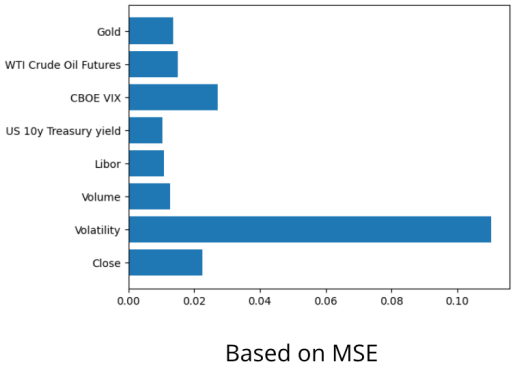


Figure 13: Feature Importance through Exclusion

This process is repeated for each feature in the model, allowing the researcher to rank the features in order of importance based on the magnitude of the increase in loss when each feature is excluded.

The advantage of this method is that it is model-agnostic and can be applied to any type of

machine learning model. It also provides a direct measure of feature importance, allowing for easy interpretation of the results.

10 Summary of Results

In the following table, we have summarized the performances of all techniques and compared MSE and R2 score results for the same. We can observe that LSTMs have the maximum R2 score and the model can therefore explain the variability of the time series by the maximum amount.

Technique	MSE	R2 score
ARIMA (benchmark)	0.0013	.9776
Random Forest	31.3e-5	.9256
ANN	151e-5	.6896
RNN	.00226	.9599
LSTMs	.00141	.9779
LSTM + Attention	.00432	.9333
BiLSTM + Attention	.00126	.9794

Figure 14: Comparison of performances of each technique

11 Conclusion

In this paper, we have examined the effectiveness of various advanced machine learning models in forecasting the volatility of the Indian stock market, using the Nifty30d volatility index as a forward-looking measure. We compared the performance of Random Forest, Artificial Neural Network, Recurrent Neural Network, and Long Short-Term Memory models with the benchmark ARIMA model. Our results show that the advanced machine learning models outperform the ARIMA model in terms of forecasting accuracy, with LSTM performing the best.

We also included several exogenous variables with potential economic significance, including Nifty close prices, Nifty volume, CBOE VIX, US 10-year Treasury yield, LIBOR, WTI crude oil futures, and gold. The inclusion of these variables in the SARIMAX model helped us gain insight into how different macroeconomic factors might affect volatility in emerging markets.

Our findings suggest that advanced machine learning models can be powerful tools for volatility forecasting in the Indian stock market. The inclusion of exogenous variables in the modeling process can further enhance the accuracy of the forecasts and give us better control of the model.

Overall, our study provides useful insights for traders and investors who are interested in volatility trading strategies in the Indian stock market. Our results demonstrate the potential for machine learning models to improve volatility forecasting and provide more accurate predictions of market conditions, which can inform investment decisions and ultimately lead to better financial outcomes.

12 References

References

[1] https://docs.google.com/presentation/d/1yLr2yDxN81pndBkxSa71PRHyi9rrW5HdR2va-O_k8/edit?usp=sharing

[2] L. Breiman, "Random Forests", Machine Learning, 45(1), 5-32, 2001.

[3] https://corporatefinanceinstitute.com/resources/data-science/random-forest/

[4] Tabatabaei Yazdi, S. A., Naseri, N., Rezaei, M. (2020). Predicting Financial Distress and Corporate Failure: A Review from the State-of-the-Art Machine Learning Algorithms. Journal of Risk and Financial Management, 13(5), 104.

[5] Ghosh, S., Sarker, S., Choudhury, T. (2021). Identifying Drivers of Bitcoin Volatility: An Empirical Study Using Machine Learning. Journal of Financial Data Science, 3(1), 31-45.

[6] Hosker, James; Djurdjevic, Slobodan; Nguyen, Hieu; and Slater, Robert (2018) "Improving VIX Futures Forecasts using Machine Learning Methods," SMU Data Science Review: Vol. 1: No. 4, Article 6.

[7] L. Breiman, "Random Forests", Machine Learning, 45(1), 5-32, 2001.

[8] https://corporatefinanceinstitute.com/resources/data-science/random-forest/

[9] Tabatabaei Yazdi, S. A., Naseri, N., Rezaei, M. (2020). Predicting Financial Distress and Corporate Failure: A Review from the State-of-the-Art Machine Learning Algorithms. Journal of Risk and Financial Management, 13(5), 104.

[10] Ghosh, S., Sarker, S., Choudhury, T. (2021). Identifying Drivers of Bitcoin Volatility: An Empirical Study Using Machine Learning. Journal of Financial Data Science, 3(1), 31-45.

[11] Hosker, James; Djurdjevic, Slobodan; Nguyen, Hieu; and Slater, Robert (2018) "Improving VIX Futures Forecasts using Machine Learning Methods," SMU Data Science Review: Vol. 1: No. 4, Article 6.

[12] Ahoniemi, Katja. (2008). Modeling and Forecasting Implied Volatility - an Econometric Analysis of the VIX Index. 10.2139/ssrn.1033812

[13] StockViz - Invest Without Emotions - ARMA + GARCH to Predict VIX - by Shyam

[14] Permutation Feature Importance for ML Interpretability by Seth Billiau

[15] Stock Price Volatility Prediction with LSTM Neural Networks, Jason C. Sullivan