Forecasting Indexes Volatilities by using Machine Learning Techniques, Econometric and Randomized Models

Francisco M. Bettencourt

Professor Bruno Damásio

NOVA Information Management School, Instituto Superior de Estatística e Gestão de Informação, Universidade Nova de Lisboa, Portugal

ARTICLE INFO

Keywords:
Stock Volatility Forecasting
Machine Learning Stock Forecast
Monte Carlo Stock
GARCH Volatility Forecast
LSTM Stock

ABSTRACT

Predicting the volatility of returns for a stock index is an attractive and defying task in the field of Machine Learning (ML). The comparison of Machine Learning models, and their resulting predictions, with several Time Series algorithms and Monte Carlo simulations, could provide valuable insight regarding the advantage of using more recent Machine Learning methods to predict stock index volatility. In this article, a study is presented on the various models' ability to predict for five worldwide Indexes, the returns and therefore, their volatilities, at the beginning of the Ukraine's conflict. By applying and comparing the performance of different algorithms, this study aims to investigate if recent ML models could lead to enhanced predictive capabilities, when in comparison to more established and frequently used statical methods and/or random models. Therefore, as mentioned above, this study will be based on five indexes, namely the Euronext 100 (Europe), the National Stock Exchange India (India), the São Paulo Stock Exchange (South America), the NASDAQ (North America) and the Hang Seng Index (Hong Kong), and the data source will be the financial information, explained in detail in section 3, from January 1st 2015 until the March 4th 2022. The study and forecasting of volatility are of high value, since Pension/Investment funds, as well as other stakeholders in Financial Markets, recognize that the risk should be minimized to the maximum level, and be within the standards that Pension/Fund members agreed upon. With this being said, the main focus of this project will not be to try to obtain the most accurate model to predict the daily volatility, but to compare how different models said volatility and if their predictions fall very far from one another. The main finding of the study was that multivariable models had performed better than univariable and randomized models. Also, models that include data with different levels of frequency (daily, monthly, quarterly) have a better forecasting capacity.

1. Introduction

Forecasting of financial assets has always been a vital topic in finance. Giving the ability to overperform the market, and therefore, break the market efficiency theory, a precise forecast could generate huge profits to those who would be able to achieve it, as it is corroborated by Poon and Granger (2001). From the beginning of the history of the stock market and trading, the evolution of technology has narrowed the gap to a reliable future value prediction. Nowadays, with the widespread use of Machine Learning algorithms and Auto Regressive models, and due to the recent computational power increase and ease of access, big funds and banks are trying to get to the "perfect" prediction, as a way to increase profits and attain a better understanding of their risk exposure. The increased easiness of access to real live market data has led to a multiplication of the number of models, statistics and key-risk metrics available. The above means, that even the small investment funds or the single investor that likes to go to the markets by himself, is able to have reliable and trustworthy information on a daily basis, which allows them to have a better understanding of the risk and profit opportunities that they are exposed to, such as it is described by Ma, Xiong and Feng (2021). There is evidence to show that, in the last few years a lot of small investors, especially those that have a background in Finance and Engineering have started to trade on their own. By using and creating machine learning algorithms that allow them to sometimes, even outperformed big Investment funds and the S&P500 in terms of returns for example, (Torre-Torres, Venegas-Martínez, & Martínez-Torre-Enciso, 2021). Nevertheless, and even acknowledging that returns are one of the most important factors that weighs on investors investment decision, as it is explained by Chaudhuri and Koo (2001), it is also important to understand that different investors have a different risk profile and appetite. If some are willing to undertake a significative risk on the longer and shorter term, others are not. Due to this, it is really important that all the stakeholders on the process have a clear view of which are the levels of risk they are exposed to, and if this level is the go-to level for the Investor. Nowadays, is also remarkably important to acknowledge the weight and influence that some Externalities and Macro environment factors have on investment decisions. Social and environmental awareness is increasing at a really fast pace, which can result in big, unexpected movements in the market, such as, the Ukraine/Russia Crisis. This one has led to an unprecedented disinvestment on Russian Companies and assets. However, not even the best risk metrics could predict what the impact of such conflict would be for the world economy, or for the European Central Bank/American Federal Bank, even if there is no direct exposure to financial Russian Assets. For example, oil prices have increased meaning that Energy companies within these Indexes could be facing relatively bigger market movements than what was expected, (Engelhardt, Ekkenga, & Posch, 2021).

With all that being said and recognizing that a fundamental basis is always really important to be able to understand and calculate the expected volatility for a given period of time, in this article, the main goal is to obtain a prediction as accurate as possible of these Indexes volatilities. To do so, data will be trained and tested for a series of models and their outputs will be compared. Some of the models, such as Time Series forecasting models rely solely on the past values of the Indexes prices, whether these are low or high frequency, such as GARCH and GARCH-MIDAS. Whilst others, such as Machine Learning algorithms, allow for additional information to be considered when forecasting future values.

2. **Background study**

This section details the literature related to the forecasting process used under each predictive model.

2.1. Risk and Returns Metrics

The returns for a given stock/index should be given by *Equation 1* in which P represents the closing price of the asset, and t represents the day of the Price. The mean return is given by dividing the sum of all daily returns by the number of days N, as it is set on *Equation 2*. One way to calculate the volatility is by assuming that the return distribution is Normal distributed, and with that calculate the squared root of the variance, as it is showed in *Equation 3*.

$$r_t = Log(\frac{P_t}{P_{t-1}}) \quad (1) \qquad \mu = \frac{\sum r_t}{N} \quad (2) \qquad \sigma = \sqrt{\frac{\sum |r_t - \mu|^2}{N}} \quad (3)$$

2.2. Data Analysis in Financial Terms

Since the beginning of the 21st century, data driven companies and data driven business models have been one of the most profitable, (Deevi, 2015). As such, and defining data as an individual set of facts, statistics, and information, that is fitter for a deep analysis and allows to achieve conclusions from it, sometimes, and by using predictive methods, it allows data managers and data scientists to achieve a high level of accuracy when predicting future outcomes. It this being said, is expected that some type of information that exists on the Financial Markets, with the help of this same predictive methods, could be used by investment managers in order to take decisions. In the concrete case of this article, the data to be used across all models are the actual prices of the selected indexes, since they are the base for return calculation and consequently for volatility as well. When using a predictive method for forecasting it is always necessary to split the data set into 2 Datasets, namely the Training and the Test set, (Xu & Goodacre, 2018). The Training Dataset consists of building the model with multiple model parameter settings and then each trained model is challenged with the validation set (not to be used)**.** The Test Dataset is the last set of data, that should be a set with new data that was never considered when drafting the model, and the actual accuracy of the model on this set, will determine the actual prediction capacity of the same.

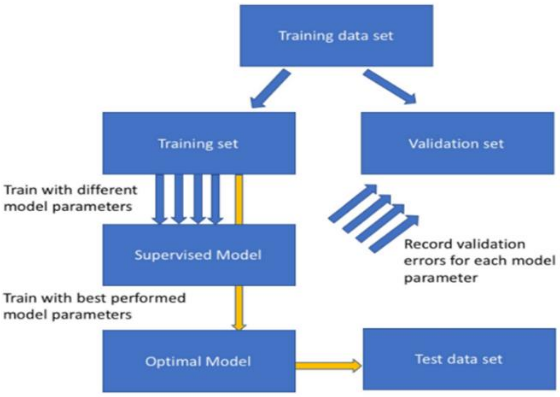

Training data set
Training set
Validation set
Train with different model parameters
Supervised Model
Record validation errors for each model parameter
Train with best performed model parameters
Optimal Model
Test data set

Figure 1- General flowchart used for model selection (Xu & Goodacre, 2018)

By considering prices of the Indexes, as the main source of data for the model, it is also important to denote that these ones are sequential, meaning that the order in which they are presented on the data set affects the outcome of the model. In this type o situations, and basing that the data should be ordered by day, i.e., the first observation should be the day of the first Price used, is expected that the training set should be the older prices, and the test set the most recent ones, being that the main objective of the article is to corroborate which of the models used, if any, could help to predict volatility in the future, and in this way help Investment Managers and small investors to be more aware of the risk they are facing.

2.3. Models Accuracy and Prediction Capacity on Timeseries Financial Data

There are different types of models, that have different types of assumptions that can be used to predict target variables. Some models, require less information, i.e., they only need to be supplied with stock data, or the main Key Statistics that are based on the stock data, whilst some other models, may require a bit more information, in order to also produce, what could equate into a more accurate result. Since this type of data is a Time Series data, what can also be understand as a collection of values obtained from sequential measurements over time (Esling & Agon, 2012), in this article, was decided to split the models that can predict a time series in three. First, the Econometric models, that are models that are able to describe the application of statistical methods to the quantification and critical assessment of hypothetical relationships using data (Dougherty, 2016). Second, randomized models, i.e., models that based on given assumptions of the overall distribution, will randomly provide values for the target variable. One of the most famous is the Monte Carlo Simulation based on a Geometric Brownian Motion. At last, Machine Learning algorithms will be used, these ones can be based on the actual price of stocks, i.e., they will account every single observation in the model, they can be based solely on the distribution moments of all observations combined, or they can use multiple variables and their key statistics in order to predict the target variable. Since the models described above are predictive models, could be therefore assumed, that these ones are able to predict values that could be accurate or highly inaccurate. Therefore, accuracy models can be used to fairly compare the accuracy capacity between models. The models that are better explaining relationships between variables or assumptions used, should be the ones with a higher accuracy rate.

2.4. Econometric Models

It is possible to denote that the most well-known Econometric models to be used are the Generalized Autoregressive Conditional Heteroskedasticity (GARCH), (Andersen, Davis, Kreiss, & Mikosh, 2009). In this work it will be presented two type of GARCH Models, namely the GARCH, which is univariable model, that works directly with the returns and their distribution parameters, and also the GARCH-MIDAS, in which MIDAS stands for Mixed Data Sampling, and is based on the same 'basis' as the other one, but have the peculiarity of allowing low frequency variables to provide thoughtful insights regarding the parameters (Conrad, Custovic, & Ghysels, 2018). The GARCH model, is a model for the variance of a time series. Despite their capacity to predict long run volatility, they actually tend to perform a more accurate prediction result, when accessing short term volatility. The long run variance on the model is given by γ, and both α and β are the respective weights that should be attributed to the μ_{t-1}^2 and σ_{t-1}^2 respectively. Therefore, the volatility value calculated under the GARCH model should be the square root of (5)
for any given t day.

$$\sigma_t^2 = \gamma V_L + \alpha\mu_{t-1}^2 + \beta\sigma_{t-1}^2 \quad (4) \qquad \gamma + \alpha + \beta = 1 \quad (5)$$

Regarding the technical details of GARCH-MIDAS models, in which the conditional variance is multiplicatively decomposed into a short-term (High frequency) and a long term (Low frequency) component. The short-term component is the returns of each Index, and the low frequency are the monthly/quarterly explanatory variables (Engle, Ghysels, & Sohn, 2013). For the return calculation, t denotes the Low frequency, and based on this $i = 1, \dots, N_t$, denotes the number of days in the Low frequency variable. The conditional mean of returns is constant.

$$r_{i,t} = LN(\frac{P_{i,t}}{P_{(i-1,t)}}) \quad (6) \qquad r_{i,t} = \mu + \varepsilon_{i,t} \qquad \varepsilon_{i,t} = \sqrt{h_{i,t}\tau_t \mathrm{Z}_{i,t}} \quad (7)$$

The innovation $\mathrm{Z}_{i,t}$ is assumed to be independent and identical distributed with mean zero and variance one. $h_{i,t}$ and τ_t denote the short- and long-term component of the conditional variance, respectively. The short-term component, $h_{i,t}$ varies at the daily frequency and follows a unit-variance GARCH (1,1) process where α >0, $\beta \geq 0$, and $\alpha + \beta < 1$. The Long-term Component varies at the quarterly frequency.

$$h_{i,t} = (1 - \alpha - \beta) + \alpha\frac{\varepsilon_{i-1,t}^2}{\tau_t} + \beta h_{i-1,t} \quad (8)$$

⍰
⍰
⍰
⍰
=
⍰
⍰

Where X_t denotes the explanatory variable and $\varphi_k(w_1, w_2)$ a certain weighting scheme. For this case the weighting scheme to be used will be the Beta weighting Scheme.

$$\varphi_k(w_1, w_2) = \frac{\left(\frac{k}{K+1}\right)^{w_1-1} .(1-\frac{k}{K+1})^{w_2-1}}{\sum_{j=1}^{K}\left(\frac{j}{K+1}\right)^{w_1-1} .(1-\frac{j}{K+1})^{w_2-1}} \quad (10)$$

All the methodology above is benchmarked on the work of Engle et al 2013 and provide thoughtful insights regarding this approach to mixed data sampling variables, in order to forecast volatility.

2.5. Monte Carlo Simulation

By acknowledging that the return of prices follow a given distribution, in this case, a normal distribution, it may be assumed that generating random variables, for the target variable, should not be totally random. A Geometric Brownian motion is often used to explain the movement of time series variables and, when adapted to corporate finance, explains the movement of asset Prices (Reddy & Clinton, 2016), in this concrete case, a Stock Market Index. Since volatility of an asset is measured by its returns, which are based on the logarithmic difference between the price of an asset in a day and the day immediately before that, it may be assumed that the returns distribution for the long term follows an uncertain distribution (random walk), that will probably be approximately normal within a width range of samples. For the Geometric Brownian assumption to be effective regarding modeling stock price, or Index price, in a time series, the following conditions must be verified, (Sengupta, 2004):

- The underlying asset must be continuous into time and value.
- A stock must follow a Markov process, meaning that only the current stock price is relevant for predicting future prices.
- The proportional return of a stock is Log-Normally distributed
- The continuously compounded return for a stock is normally distributed.

$$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t \quad (11)$$

The left side of the *Equation 11* is the certain component and the right one is the uncertain or variable component. The first part is called the drift of the stock and it is assumed as the return that a stock will earn over a short period of time. The uncertain component represents a stochastic process that includes the volatility of returns on an Index, and also a Wiener process which is the stochastic component (Reddy & Clinton, 2016). For each random number generated from a normal distribution, and this distribution is used due to the fact that returns are normally distributed, the Wiener process consists of the multiplication of this random number by the square root of time, which in turn creates the stochastic process. When it comes to a Monte Carlo simulation, it is a process that consists in simulating values, for a given variable, n times, in order to predict the most probabilistic outcome, i.e., the one that appears the most times within the simulation. When applying the Monte Carlo simulation to the Geometric Brownian Motion, it should be applied the drift value

and the volatility (Brewer, Feng, & Kwan, 2012). By using a Monte Carlo Simulation, it is possible to generate a Price for a given day, and from that price calculate the return and volatility. Where:
Z is given by a random normal distribution, with x number of simulations, and assuming that mean is zero and standard deviation is one.
Wt is described as the Wiener process and is given by multiplying the square root of time by the Z variable.
St (Spot price at time t) is given by multiplying stock price at time zero by the base of natural logarithm (e) raised to the power of the log normal distribution, i.e., $Drift(\hat{\mu}) - \sqrt{\sigma}$, multiplied by time, plus standard deviation multiplied by the Wiener Process.

$$Z = (x, \mu = 0, \sigma = 1) \quad (12) \qquad W_t = \sqrt{T} * Z \quad (13)$$

$$St = S_0 e^{[\,(\hat{\mu} - 0.5 * \sigma^2)T + \sigma * W_t]} \quad (14)$$

2.6. Machine Learning Models

2.6.1 Support Vector Regression

Is similar to Support Vector Machine, offers a principled approach to machine learning problems because of its mathematical foundation in statistical learning theory. SVM constructs its solution in terms of a subset of the training input and has been extensively used for classification, regression, novelty detection tasks, and feature reduction (Awad & Khanna, 2015). Vapnik-Chervonenkis (VC) theory proves that a VC bound on the risk exists. VC is a measure of the complexity of the hypothesis space. The VC dimension of a hypothesis H relates to the maximum number of points that can be shattered by H. H shatters n points, if H correctly separates all the positive instances from the negative ones. In other words, the VC capacity is equal to the number of training points n that the model can separate into 2^n different labels. This capacity is related to the amount of training data available (Awad & Khanna, 2015).Based on the above, the VC dimension a h affects the generalization error, as it is bounded by $\|\omega\|$ where ω is the weight vector of separating hyperplane and the radius of the smallest sphere R that contains all the training points. The overall error of a machine learning model consists of the sum of the individual errors, where ε_{emp} is the training error, and $\varepsilon_g$is the generalization error.

$$h < \frac{R^2}{\|\omega\|^2} \quad (15) \qquad \varepsilon = \varepsilon_{emp} + \varepsilon_g \quad (16)$$

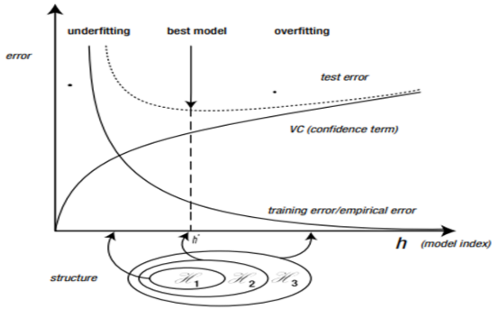


Figure 2- Relation between Error and Model Index (Awad & Khanna, 2015).

Bearing this in mind, the actual difference between SVM and SVR, is that the regression problem is a generalization of the classification problem, in which the model returns a continuous-valued output, as opposed to an output from a finite set (Awad & Khanna, 2015). The Support Vector Regression formula should be given by *Equation 17,* or by augmenting x by one and include b in the ω vector, *Equation 18* is obtained.

$$y = f(x) =< \omega, x > +b =$$
$$= \sum_{j=1}^{M} \omega_j x_j + b, y, b \in \mathbb{R}, x, \omega, \in \mathbb{R}^M \quad (17)$$

$$f(x) = \begin{bmatrix} \omega \\ b \end{bmatrix}^T \begin{bmatrix} x \\ 1 \end{bmatrix} = \omega^T x + b \;\; x, \omega \in \mathbb{R}^{M+1} \quad (18)$$

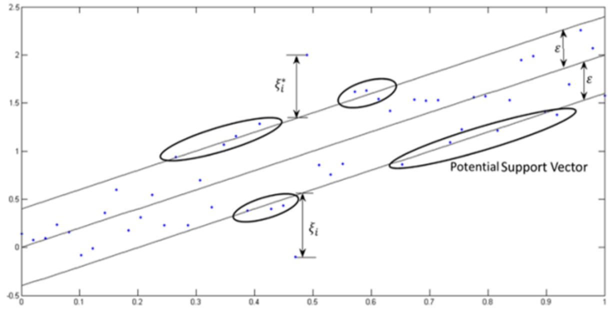


Figure 3- Support Vector Regression example (Awad & Khanna, 2015)

2.6.2 Long-Short Term Memory

Is a recurrent neural network. Recurrent or very deep neural networks are difficult to train, as they often suffer from the exploding/vanishing gradient problem (Houdt, Mosquera, & Nápoles, 2020). Overall, this can be prevented by using a "Constant Error Carousel" (CEC), which maintains the error signal within each unit's cell. The input gate and output gate, form the memory cell. The self-recurrent connections indicate the feedback with a lag of one-time step. A plain vanilla LSTM unit is composed of a cell, an input gate, an output gate and a forget gate, that allows the network to reset is state. In short, the architecture of a LSTM model, is based in a set of recurrently connected sub-networks, also known as, memory blocks. The main function of this blocks is to maintain its state over time and regulate the information flow through non-linear gating units (Houdt, Mosquera, & Nápoles, 2020).

Block input- this step is devoted to updating the block input component, which combines the current inputs $x^{(t)}$ and the output of that LSTM unit $y^{(t-1)}$ in the last iteration, where, W_Z and R_Z are the weights associated with $x^{(t)}$ and $y^{(t-1)}$ respectively, whilst b_Z represents the bias weight vector.

$$Z^{(t)} = g(W_Z x^{(t)} + R_Z y^{(t-1)} + b_Z) \quad (19)$$

Input Gate- it combines the current input $x^{(t)}$, the output of that LSTM unit $y^{(t-1)}$ and the cell value, $c^{(t-1)}$ in the last iteration, where ■ denotes the point-wise multiplication of two vectors, W_i, R_i, p_i are the weights provided to $x^{(t)}, y^{(t-1)}, c^{(t-1)}$

respectively, whilst b_i represent the bias vector of the component.

$$i^{(t)} = \sigma\left(W_i x^{(t)} + R_i y^{(t-1)} + \ p_i \blacksquare c^{(t-1)} + b_i\right) \quad (20)$$

Forget Gate- The LSTM unit determines which information should be removed from its previous cell states $c^{(t-1)}$. Therefore, the activation values, $f^{(t)}$, of the forget gates at time step t, are calculated based on the current input $x^{(t)}$, the outputs $y^{(t-1)}$, and the state $c^{(t-1)}$ of the memory cells ate previous time step $(t-1)$, and b_f is the bias terms of the forget gates, where ■ denotes the point-wise multiplication of two vectors, W_f, R_f, p_f are the weights provided to $x^{(t)}, y^{(t-1)}, c^{(t-1)}$ respectively.

$$f^{(t)} = \sigma\left(W_f x^{(t)} + R_f y^{(t-1)} + \ p_f \blacksquare c^{(t-1)} + b_f\right) \quad (21)$$

Cell- this step computes the cell value, which combines the block input $Z^{(t)}$, the input gate $i^{(t)}$ and the forget gate $f^{(t)}$, with the previous cell value.

$$c^{(t)} = Z^{(t)} \blacksquare\ i^{(t)} + c^{(t-1)} \blacksquare f^{(t)} \quad (22)$$

Output Gate- is a combination of the current input $x^{(t)}$, the output of that LSTM unit $y^{(t-1)}$ and the cell value $c^{(t-1)}$ in the last iteration, where ■ denotes the point-wise multiplication of two vectors, W_o, R_o, p_o are the weights provided to $x^{(t)}, y^{(t-1)}, c^{(t-1)}$ respectively, whilst $b_o$represent the bias of the weight vector.

$$o^{(t)} = \sigma\left(W_o x^{(t)} + R_o y^{(t-1)} + \ p_o \blacksquare c^{(t-1)} + b_o\right) \quad (23)$$

Block Output- combines the current cell value $c^{(t)}$ with the current output gate, where in the steps above mentioned, $\sigma, g\ and\ h$ denote point-wise non-linear activation functions. The logistic Sigmoid is used as a gate activation function.

$$y^{(t)} = g(c^{(t)}) \blacksquare\ o^{(t)} \quad (24)$$

Equation 25 refers to the LSTM logistic sigmoid, while *Equation 26* represents the Hyperbolic Tangent, which is often used as the block input and output activation function.

$$\sigma(x) = \frac{1}{1+e^{1-x}} \quad (25) \qquad h(x) = g(x) = \tanh(x) \quad (26)$$

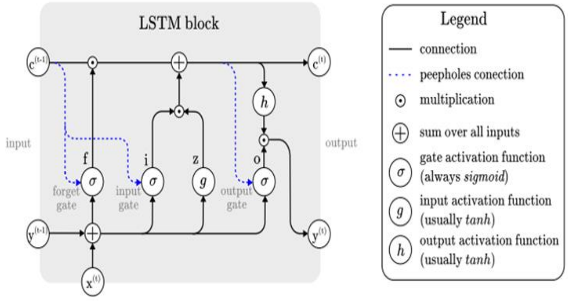


Figure 4- LSTM process (Houdt, Mosquera, & Nápoles, 2020)

The entire process above described, as well as all formulas were based solely on the research performed under the publication article “A review on the long short-term memory model” (Houdt, Mosquera, & Nápoles, 2020).

2.7. Accuracy Measurement Models

Since all the models above described, are used to make a prediction, i.e., based on a multitude of assumptions, these models will predict a value for the Target Variable, it could be acknowledged that they will sometimes be right, and sometimes wrong.

Mean absolute error- it involves summing the magnitudes (absolute values) of the errors in order to obtain the total error, and the dividing it by n (Willmott & Matsuura, 2005). This measures the absolute average difference between the real data and the predicted data, but it usually tends to fail to punish large errors in prediction, and by assuming n as the number of observations, x_i is the output generated from the model, x is the actual, observed value and $|x_i - x|$ is the absolute error.

$$MAE = \tfrac{1}{n}\textstyle\sum_{i=1}^{n} |x_i - x| \quad (27)$$

Mean Squared error- This one, is really similar to the one above, but since with will square absolute error, the geometric difference between both observations will be emphasized. By assuming n as the number of observations, x_i is the output generated from the model, x is the actual, observed value and $|x_i - x|$ is the absolute error.

$$MSE = \tfrac{1}{n}\textstyle\sum_{i=1}^{n} |x_i - x|^2 \quad (28)$$

Root Mean Squared error- also very similar to the one above, this one is able to explain the second moment of the error distribution, i.e., the standard deviation of the error. By assuming n as the number of observations, x_i is the output generated from the model, x is the actual, observed value and $|x_i - x|$ is the absolute error.

$$RMSE = \sqrt{\tfrac{1}{n}\textstyle\sum_{i=1}^{n} |x_i - x|^2} \qquad (29)$$

Measure	Symbol	Advantages	Limits
		Scale-Dependent Measures	
Mean Absolute Error	MAE	Oftentimes, the RMSE is preferred to the MSE, as it is on the same scale as the data. Historically, the RMSE and MSE have been popular, largely because of their theoretical relevance in statistical modeling. The RMSE is useful as a relative measure to compare forecasts for the same series across different models. The smaller the error, the better the forecasting ability of that model according to the RMSE criterion. The mean absolute error (MAE) is less sensitive to large deviations than the usual squared loss.	Scale-dependent measures are on the same scale as the data. Therefore, none of them are meaningful for assessing a method's accuracy across multiple series. The sensitivity of the RMSE to outliers is the most common limitation of using of this measure.
Mean Square Error	MSE		
Root Mean Square Error	RMSE		

Figure 5- Error Measurements Pros and Cons

2.8. Diebold-Mariano Test and Harvey, Leybourne, Newbold Test

The Diebold-Mariano test can discriminate the significant differences of forecasting accuracy between different models based on the scheme of quantitative analysis. It works on a Hypothesis basis and allows to check if the results are statistically significant or not for a forecasted series (Constantini & Knust, 2011). The Diebold Mariano test is given by a function in which y_t are the actual data series values, $\hat{y}_{i,t}^{h}$ are the ith competing h-step forecasting series. The forecasting errors from the ith competing models are denoted by $e_{i,t}^{h}(i = 1,2,3,\dots,m)$ in which m is the number of forecasting models, (Buturac, 2021).

$$e_{i,t}^{h} = y_t^h - \hat{y}_{i,t}^{h} \quad (30)$$

The forecast accuracy is presented in *Equation 31* and the Null hypothesis of equal forecast accuracy is presented in *Equation 32,* and the loss differential between both Datasets is presented on *Equation 33* and based on it the loss mean differential ($\bar{d}$) is defined in *Equation 34.*

$$g\left(y_t^h, \hat{y}_{i,t}^{h}\right) = g\left(e_{i,t}^{h}\right) \quad (31)$$

$$H_0\colon \mathrm{E}\left[g_{i,t}\right] = \mathrm{E}\left[g_{j,t}\right] \; or \; \mathrm{E}[d_t] = 0 \quad (32)$$

$$d_t = g\left(e_{i,t}\right) - g\left(e_{j,t}\right) \quad (33)$$

$$\bar{d} = \frac{1}{T}\sum_{t=1}^{T}\left[g\left(e_{i,t}\right) - g\left(e_{j,t}\right)\right] \quad (34)$$

And finally, the Diebold Mariano is given by *Equation 35*, in which $2\pi\hat{f}_d(0)$ is a consistent estimator of the asymptotic variance of $\sqrt{T\bar{d}}$. Because DM tests converge the actual distribution to a Normal distribution, it is possible to reject the Null Hypothesis at 5% confidence level (Buturac, 2021).

$$DM = \frac{\bar{d}}{\sqrt{\frac{2\pi\hat{f}_d(0)}{T}}} \xrightarrow{d} N(0,1) \quad (35)$$

Despite the fact that the model above mentioned is one of the most well-known and used, when it comes to perform on small data samples, such as the one in the Test dataset, which include less than a dozen of observations, this model tends to reject the Null Hypothesis, confirming that both models have different forecasting capacity. In order to prevent that, Goran Buturac, (Measurement of Economic Forecast Accuracy: A systematic Overview of the Empirical Literature, 2021), refers to a couple of other methods/models that can predict with higher accuracy on these small data sets. The one that is going to be used is the HLN model, that through a set of modifications on the linear regression theory, the HLN modifies the DM test in the following form, (Harvey, Leybourne, & Newbold, 1997):

$$DM^* = \frac{DM}{\sqrt{\frac{\left[T+1-2h+\frac{h(h-1)}{T}\right]}{T}}} \quad (36)$$

The HLN test, since it is usually proper for small data samples, in which is not possible to presume a distribution, and the sample size does not allow to conclude that it could be approximately Normal, it should use the t-distribution with $(T-1)$ degrees of freedom.

2.9. Similar Articles and Results

This article tends to compare the volatility predictive capacity, across five Worldwide indexes, by considering different models. Previous articles have also followed a similar approach, sometimes only by focusing in one type of model, such as Machine learning for example, or by using a multitude of models. With this subsection of the Background Study, it is intended to show the main results/outcomes, that those articles had, and with this have enough empirical knowledge to compare them with the results from this article. It is also needed to understand that the comparisons should not be straight forward, as there are a multitude of factors that impact the outcomes, either by the Macro environment factors, by the country, date that was performed or even by the model used. With this being said, the following articles should work as a comparison basis for the expected outcomes:

"Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models" (Kim & Won, 2018), is an article based on the KOSPI 200 stock index (South Korea), and considers a data period ranging from January 1, 2001 to September 30, 2011, and the main objective was to predict the daily volatility until January 2, 2017. It was also used other variables beside the Stock data, such as the Korean Treasure bonds and the 3-year AA-grade corporate bond. In addition, other variables, commodities, such as Oil and Gold, were also used as a variable that would help to determine the target variable. Models such as the MSE, MAE and others, were used to compare the prediction capacity. The GARCH model performed within the expected values that the researchers expected, nevertheless, the LSTM model outperformed the GARCH model.

Our assumption for this one, is that GARCH models accrue a value to the long run variance, that might have a significant impact in a such long time period, and also the LSTM by using other variables, beside the prices and statistics of the same, might have a better capacity to explain the price movements and therefore the volatility. Statistics such as correlation between variables, could have a big impact in the predicted outcome, i.e., by assuming that variables are heavily correlated, a change in variable A could equate in an immediate change on the Target variable, by a given, believed, amount.

The article "*A machine learning approach to predicting stock returns" (Silva, 2021)*, is based on the monthly returns of the NASDAQ and uses a couple of machine learning techniques that help to understand the behavior of volatility. It also, considers other variables such as technical indicators, being these ones, based on price movement and statistics. In the concrete case of this article, the only two models that will be undertaken are the Linear regression and the CART, in which the Linear regression had the worst perform of both, showing a MSE almost three times bigger than the CART model, despite that in the Training have performed better.

On the article *"Multi-Transformer: A New Neural Network-Based Architecture for forecasting S&P Volatility (2021)"* it is possible to understand how different neural networks, such as Transformer, Multi-Transformer, LSTM and ANN can, by using a wide range of GARCH models, such as EGARCH, FIGARCH and so on, improve these ones, by providing thoughtful inputs to the models, that in the end will equate in a much higher prediction capacity. I consider this article to be of crucial relevance for *any further work*, because it not only provides an in-depth study about a recent Neural Network-Architecture, as well as combine a couple of them in a multi-Transformer model, and can still, based on "older" methods such as the GARCH, improve significatively the Forecasting capacity. By acknowledging that this study, as any other has some limitations, the study above mentioned shows different ways to access the same problem, by exploring some new models and adapting old ones.

Summing up, this three articles create the possibility to understand that, models that have a better $R^{2,}$ i.e., use a combination of variables, and weight attributed to the same variables, have a larger capacity prediction, in the sense that they generate outcomes that are based not solely on the previous behavior of the Target variable, and that models that not consider a long run variance, will be able to predict better in the shorter period. It is mine personal belief that, a priori, the LSTM model will surpass all the other models for all sets of data under analysis. This is due to fact that it's the most recent one, and also, the one that seems more sophisticated.

3. Methodology

"A research design is the strategy for a study and the pan by which the strategy is to be carried out. It specifies the details of how the project should be conducted in order to fulfil the research objective" (Falinouss, 2007, p. 90). In this part of the article, it is intended to provide a full description on how the results will be achieved, and with that, the answer to the research problem.

3.1. Research approach and Design Strategy

Explanatory research aims to develop an initial hunch or insight, and to provide direction for any further research needed. The primary purpose of explanatory research is to shed light on the nature of the situation and to identify any objectives or data that needs to be addresses through additional research, working as a sort of benchmark (Falinouss, 2007). By acknowledging that the time period under research will be from 01/01/2015 until 04/03/2022, it is expected, due to large macrosocial events that occurred during the same period, that volatility levels turn out to be directly impacted by those, i.e., the market was exposed to large amounts of external factors beside the normal trading that occurs under the expected levels. The main research question that this article tries to answer, is which type of model, i.e., from the ones above described, predicted the daily volatility of the first two weeks since the start of Ukraine's conflict (for reference it will be assumed the 20th of February (Sunday) as the first day of conflict) with the highest accuracy level. This would help also to understand if models that allow more data than the single Returns and their distribution, so exogenous variables, could have a higher level of forecasting, by attributing weights to other variables. Furthermore, in order to provide more globalized research, five different indexes from five different regions will be used, namely the Euronext 100 (Europe), the National Stock Exchange India (India), the São Paulo Stock Exchange (South America), the NASDAQ (North America) and the Hang Seng Index (Hong Kong), as mentioned in the Abstract.

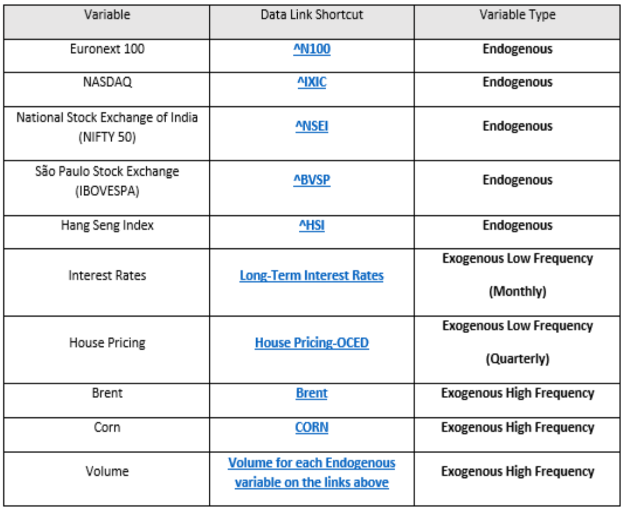
3.2. Data Collection

It is widely believed that the success of any data solution, is based on the quality of the data that the models use. With this being assumed, is totally crucial to achieve the maximum amount of information during the time period under analysis. For this article, two type of variables will be used, the endogenous and the exogenous ones.

For the Endogenous variables, it will be considered the actual closing daily prices of each index during the period under analysis. Based on these prices will be possible to determine multiple metrics that will be accounted for in every single model, as explained below, and this should be considered the most valuable variable of every single model. This information is publicly available, either through “Yahoo Finance”, “Bloomberg” or any other information provider.

Regarding Exogenous variables, these ones will be split between two sets, in order to allow for different frequencies. Low frequency variables, in which it will be used the “Interest-rate” as a percentage change on a monthly basis and “House Pricing” as a percentage change from quarter to quarter. The information above, will be region linked, for example, House pricing changes in India, will be used only when the NSEI is being predicted, and in no other region Index. All this information is publicly available through the OCED Public Website. Regarding the High frequency variables, Commodities prices will be used , namely Corn and Brent, since changes on both of them are heavily correlated with the war on Ukraine, being both Ukraine and Russia large producers and exporters, and without their role there is a shortage on the supply chain. Finally, the volume, i.e., number of shares traded in one day, will also be considered as a variable in the model, but in this concrete case, every country Index will account their volume only, so for any Indian model Brazil’s Volume will not be used, only the NSEI Volume. All this information is widely and publicly available through “Yahoo Finance” or any other free provider.

The Figure 6, provide a direct web link from each it is possible to obtain the same data that it was used on the models (link in Appendix).



Variable	Data Link Shortcut	Variable Type
Euronext 100	^N100	Endogenous
NASDAQ	^IXIC	Endogenous
National Stock Exchange of India (NIFTY 50)	^NSEI	Endogenous
São Paulo Stock Exchange (IBOVESPA)	^BVSP	Endogenous
Hang Seng Index	^HSI	Endogenous
Interest Rates	Long-Term Interest Rates	Exogenous Low Frequency (Monthly)
House Pricing	House Pricing-OCED	Exogenous Low Frequency (Quarterly)
Brent	Brent	Exogenous High Frequency
Corn	CORN	Exogenous High Frequency
Volume	Volume for each Endogenous variable on the links above	Exogenous High Frequency

Figure 6- Information Source

3.3. Data Preparation

The data will be collected is in the pure form of prices, i.e., the values are still in their brute form, and in order to get them in the perfect shape for the model data analysis work is needed. For all missing values, i.e., prices that are not available for a single business day, this day will be omitted from the study, and the return for the day immediately after the missing value, will be given by the logarithmic difference between the missing value day plus one and the missing value day minus one.

$$r_{i+1} = \ln(\frac{P_{i+1}}{P_{i-1}}) \quad (37)$$

Firstly, since this is a time series data set, the split between training and test data set should be performed on a basis of temporal continuity, meaning., not randomly, or in a percentage of the total set. Since the main goal in here is to forecast the volatility for the first two business weeks of war in Ukraine, starting on 20th February 2022, the training set will account for all observations since 01/01/2015 until 18/02/2022. Therefore, the test set, in which the trained models will perform, will be from 21/02/2022 until 04/03/2022.

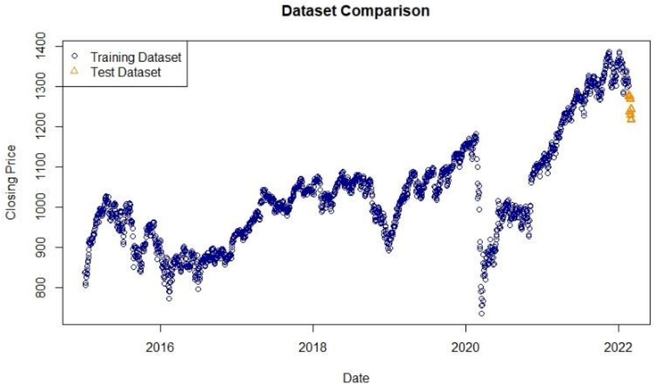


Figure 7- Split Between Training and Test Dataset Euronext 100 Example

Secondly, starting with the endogenous variables, these are provided in a Price format. When analyzing stock data, it is highly recommended to convert the same to returns, especially daily, in order to be possible to have a relevant sample size, that will be approximately Normal Distributed, and therefore, easier to work with. Otherwise, prices tend to be Log-Normal distributed, and the key statistical parameters will be harder to calculate (Hull J. C., 2018). Bearing this information in mind, the first step is to calculate the daily return of the Stock index, based on the “Closing Value”, for the entire range of observations. Then, the mean return can be calculated, as well as the standard deviation for the period. Thirdly, when it comes to the exogenous variables, for the high frequency ones, since they are in the same time basis that the endogenous ones, i.e., they are also daily and the number of observations matches between both variables, for corn and Brent the daily difference will be used, so same as return for a normal stock, and for volume the same process will be used, which means, the daily difference. For low frequency variables, since these ones are on a different time basis that the target variable, they will be under a process noted as MIDAS, described above on the Background Study chapter, and provide quarterly information, in this concrete case, the percentual change from each month/quarter, that will impact the daily returns by a given percentage under the model, and in the end achieving different parameter values. Fourthly, the data above described will not be considered under all of the models. In order to better understand this, the Figure 8, in which is defined the type of variables that each model will use.

Model	Daily returns	Statistic Parameters of Returns	High Frequency Indicators	Low Frequency Indicators
Monte Carlo	Yes	Yes	No	No
GARCH	Yes	Yes	No	No
GARCH MIDAS	Yes	Yes	No	Yes
SVR	Yes	N/A	Yes	No
LSTM	Yes	N/A	No	No

Figure 8- Variable Inputs by Model

3.4. Results Obtention

Now that the type of data used by each model has been defined, it is time to define the process of achieving the results. Information about each model has been previously described in the Background Study chapter, but in order to obtain a more graphical example, with will be covered here as well, the main characteristics of the process.

***Preliminary Analysis*->** For each given stock index under the framework of this dissertation, it should be downloaded the information from the provider, in this case “Yahoo Finance”. It should only be taken into consideration both “Date”, “Close”, and “Volume”. After, the data should be divided between training and test, for the time periods above mentioned. Once the data is divided, it should be calculated the returns for both datasets as well as the mean and standard deviation. Moving forward, the returns for the test data will work only as a basis for comparison and should never been accounted under the models (forecasting part).

Monte Carlo Simulation -> At this moment in time, should be assumed that the returns for a given index are already calculated as well as the distribution of the same, and with that the underlying moments, meaning, the first and second moment, equating in the mean and standard deviation. Whenever the model produces a random daily price, by using *Equation 14*, it is obtained only one price. In order to predict for two weeks, it is needed to repeat the process on *Equation 14* for the length of the Test Dataset. By doing so, the prices obtained will not follow a specific time frame, i.e., they will not be in sequential order. To overcome this difficulty, it was determined that the simulation number one, will equate in the price for the 21st of February 2022, and so on, until the last simulation equates in the price at 4th of March 2022. The output data is now under a continuous, sequential basis, and the calculus of the returns is possible to be obtained. Since the main goal of a Monte Carlo Simulation is to simulate a huge number of times, the process above will be simulated one thousand times, meaning, that in the end will be obtained an array with ten/nine rows (number of business days during the Test Dataset timeframe), each one representing the daily return of the given day, and will have one thousand columns for the length of the Test Dataset (which could differ based on the specificities of each region/country) and the ten days path repeats itself, over one thousand simulations. Afterwards, the mean return for each row will be calculated, and that should be the return used to compare with the actual return. Then the absolute error and squared error will be calculated in order to obtain the performance metrics.

GARCH (1,1) -> Once the daily returns are calculated, it is now possible to obtain the estimations for the GARCH model. Before going into detail regarding the GARCH Model characteristics, since the main goal is to forecast into the future, the returns need to be lagged on the same basis as the length of Test Dataset, meaning that the GARCH value for n^{th} day should give by *Equation 38*.

$$\sigma_n^2 = \gamma V_L + \alpha\mu_{n-length(test)}^2 + \beta\sigma_{n-length(test)}^2 \quad (38)$$

The GARCH model, as demonstrated on (5)
works on the basis of weight being attributed to the given variables, as shown in *Equation 5.* The Long run variance is given by deducting one per Alpha and Beta, and these parameters should be calculated using a solver system, in which the objective is to attribute values to either $\gamma, \alpha\ and\ \beta$, being that the long run variance weight allocation is always higher than zero, and both $\alpha\ and\ \beta$ range between one and zero. The target variable on the solver system is the likelihood, that should be maximized by changing the weights between the above-mentioned variables. The only step that is still missing is to define the GARCH model parameters, so how many moving averages and auto regressions should be accounted for in the simulation. The best and proper way to check this is by using the Autocorrelation function, with lag equal to one, and check for each point if it shows a major breakdown and it is needed to check the probability of P being higher than the $p - value$. If this value is higher than, let’s assume, 0.05, then this variable is not significant for the model, and the number of Moving Averages and Auto regressions used should be changed. In this concrete case, the GARCH (1,1) model will be used across all countries, since the main goal is to compare the accuracy capacity of each model, under each country, on the same given conditions. With $\gamma, \alpha\ and\ \beta$ defined, one only needs to apply *Equation 38* and both Training and Test target values will be available.

GARCH-MIDAS -> as explained above in the Background Study chapter, this one model is pretty similar to the standard GARCH model but allows for low frequency variables that will be accounted for on each day’s return. By using these new returns, all other metrics will be also changing due to the weight each variable has to predict the target variable. Since in this article there are two type of low frequency variables, being the *Interest Rates* in a monthly basis and *House Pricing* in a quarterly basis, by using a Beta weighting scheme, both variables will have a different weight on the Target variable, i.e., the return. It is also needed to equate the time periods, for example for Q1-2015, all the daily returns between 01/01/2015 and 31/03/2015 should have the same House pricing return, i.e., since they fall under the same timeframe. Finally, the Lag for each low frequency time variable needs to be defined. In this study, monthly variables have a lag of twelve and Quarterly variables have a lag of four, since this is the number of times the events occur during a year, and the model is set up in that way. Also, it's recommended to use the value of returns as a percentage, meaning, the actual value multiplied by one hundred. Once the model is provided with the key parameters, it can calculate for each day a value for $h \text{ and } \tau$, that when multiplied by one by another, provide the variance for the given day, as it is mentioned in *Equation 7*. Once these new, now weighted returns are calculated, all is left to do is to apply the same *GARCH* process as above described, obtain the $\gamma, \alpha \text{ and } \beta$ and then calculate the *GARCH* value for each single observation.

Support Vector Regression -> This one is the model that requires and allows for the most data being provided. This model will use not only the lagged daily Returns, but also the Corn and Brent lagged daily returns, as well the lagged daily change on Volume. Being that this one is a Multivariable model, the target value will be defined by a specific coefficient attributed to each one of the variables, and the model will work on maximizing the best performance possible, by reducing the Mean Squared Error of the model as much as possible. This could and should be performed by using a statistical software such as RStudio or Python, due to the need of high computational power. Once the model has defined its parameters and coefficients for each variable, one only needs to apply the same ones to the lagged variables that will impact on the Dataset and the forecasting process is completed. Once these forecasted values are obtained, same as above, they should be compared with the actual values, and with that, measured the errors on the model's forecasting capacity.

LSTM -> Similar to the above model, this one is also a Machine Learning Model, being that this one is based on a layer process. For this model only the lagged returns by the length of the Test Dataset will be used, and it should be also used a statistic software, especially one that allows such complex model to run in an efficient amount of time. The model will try under each layer to reduce as much as it can the Mean Squared Error, until it gets to the point that for each layer, the reduction in the same begin to be zero.

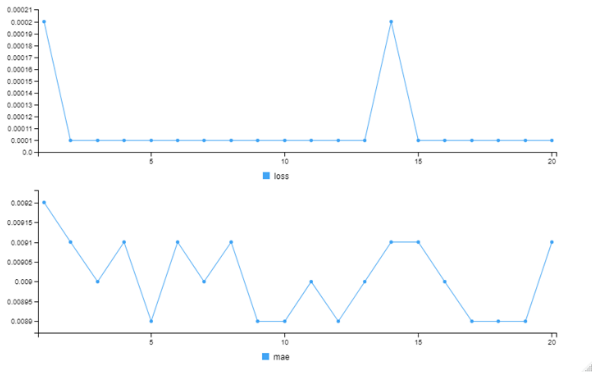


Figure 9- LSTM Error loss Function

For forecasting the model will use the "best version" of itself and will try to predict, based on the lagged returns, the forecasted value of the actual returns. Once that occurs, one just needs to compare them and measure the differences.

Dataset Statistical Comparisons -> As explained in the methodology chapter, chapter 3, the Diebold Mariano should be used for all sets of data that are considered Normal, or approximately Normal, this meaning that for the Training Dataset, it will be used for each individual error, by model, by country, a DM statistical test, to see if both forecasts have equal accuracy. For the Test Dataset, the HLN statistical test will be used, since the sampling length is, on average, 9 observations (days), and could be assumed to follow a T distribution.

4. Analysis of Results

Based on the study performed, it was possible to conclude that during the period under analysis, different indexes had different volatility ranges, being that some were more volatile than others. The period during which all of them present the biggest negative return was the during the Covid-19 pandemic, as it is possible to see on the example presented on Figure 10.

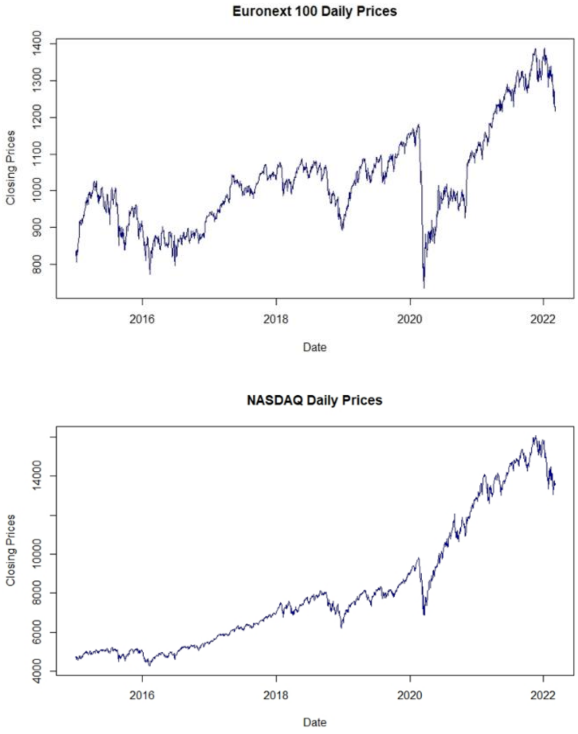
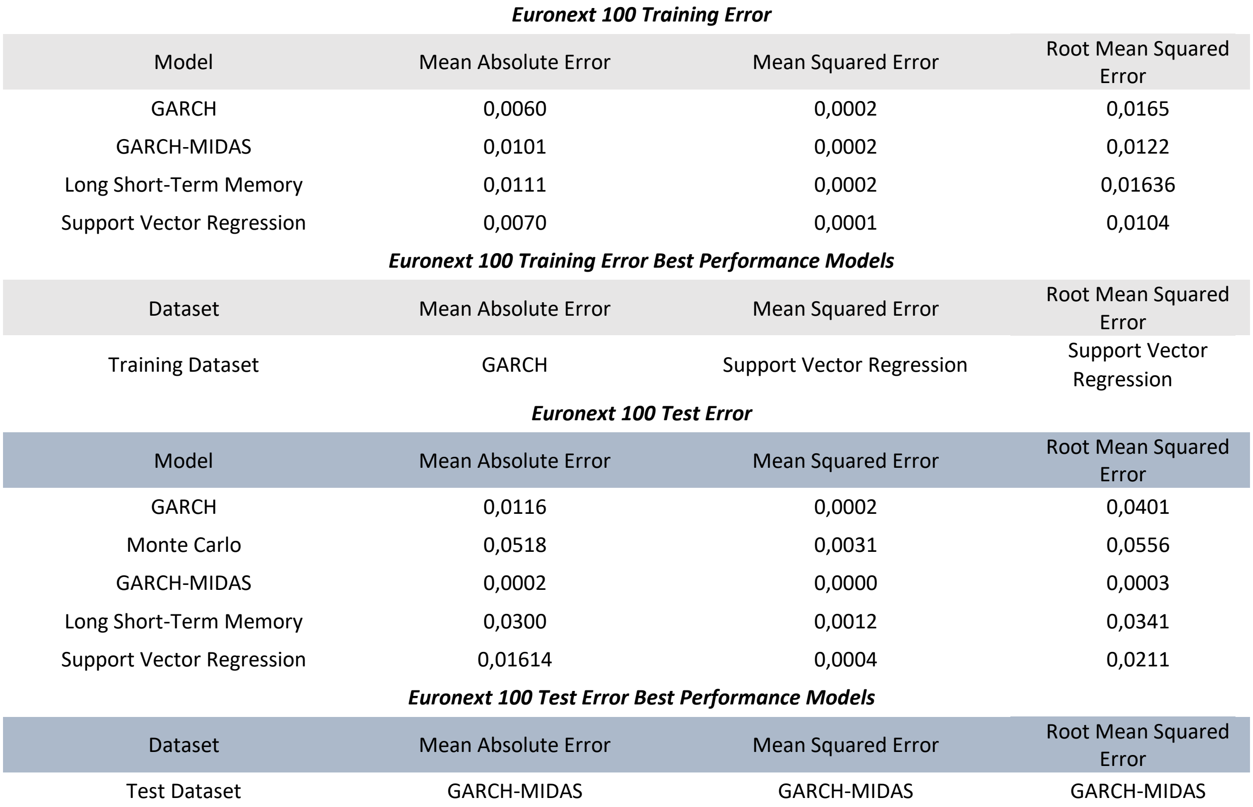


Figure 10- NASDAQ and Euronext 100 Daily Closing Prices

Since the main goal of this article is to compare the forecasting capacity of a series of models, across five different countries/regions, the figures below present in detail which were the Training and Test errors, by Accuracy Measurement Model, by Index.



Euronext 100 Training Error

Model	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
GARCH	0,0060	0,0002	0,0165
GARCH-MIDAS	0,0101	0,0002	0,0122
Long Short-Term Memory	0,0111	0,0002	0,01636
Support Vector Regression	0,0070	0,0001	0,0104

Euronext 100 Training Error Best Performance Models

Dataset	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Training Dataset	GARCH	Support Vector Regression	Support Vector Regression

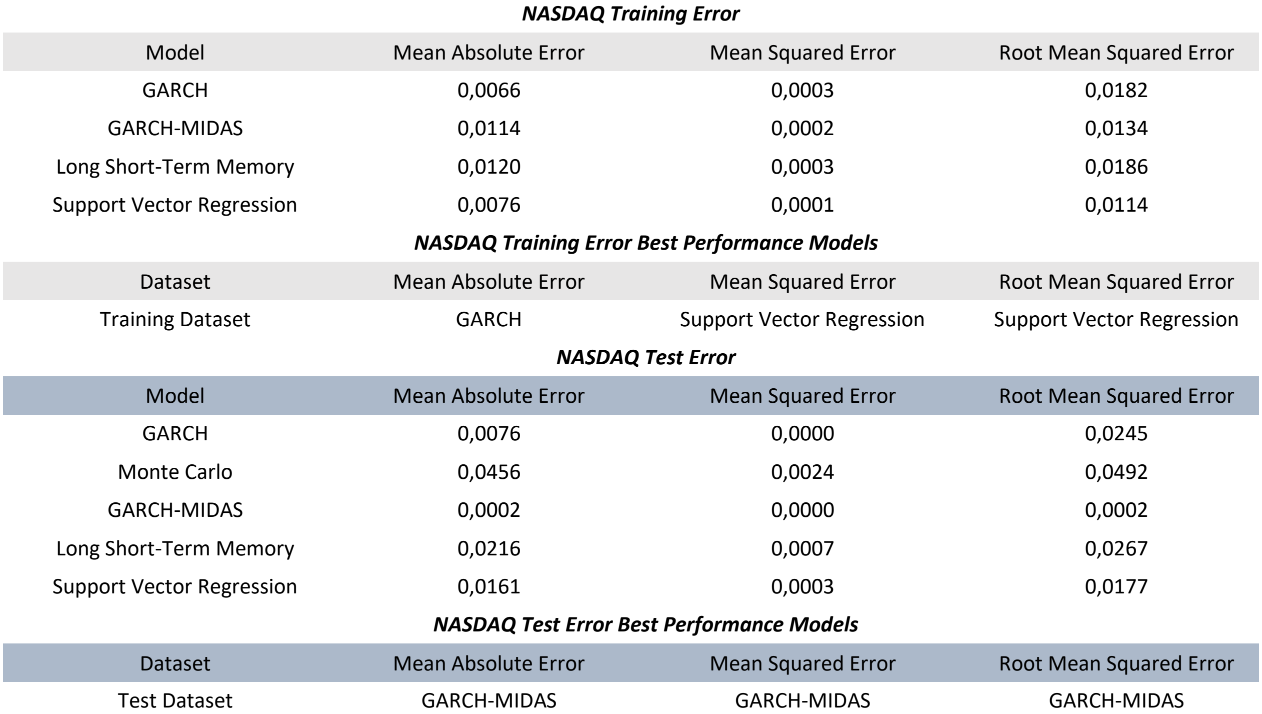
Euronext 100 Test Error

Model	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
GARCH	0,0116	0,0002	0,0401
Monte Carlo	0,0518	0,0031	0,0556
GARCH-MIDAS	0,0002	0,0000	0,0003
Long Short-Term Memory	0,0300	0,0012	0,0341
Support Vector Regression	0,01614	0,0004	0,0211

Euronext 100 Test Error Best Performance Models

Dataset	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Test Dataset	GARCH-MIDAS	GARCH-MIDAS	GARCH-MIDAS

Figure 11- Euronext 100 Forecast Error



NASDAQ Training Error

Model	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
GARCH	0,0066	0,0003	0,0182
GARCH-MIDAS	0,0114	0,0002	0,0134
Long Short-Term Memory	0,0120	0,0003	0,0186
Support Vector Regression	0,0076	0,0001	0,0114

NASDAQ Training Error Best Performance Models

Dataset	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Training Dataset	GARCH	Support Vector Regression	Support Vector Regression

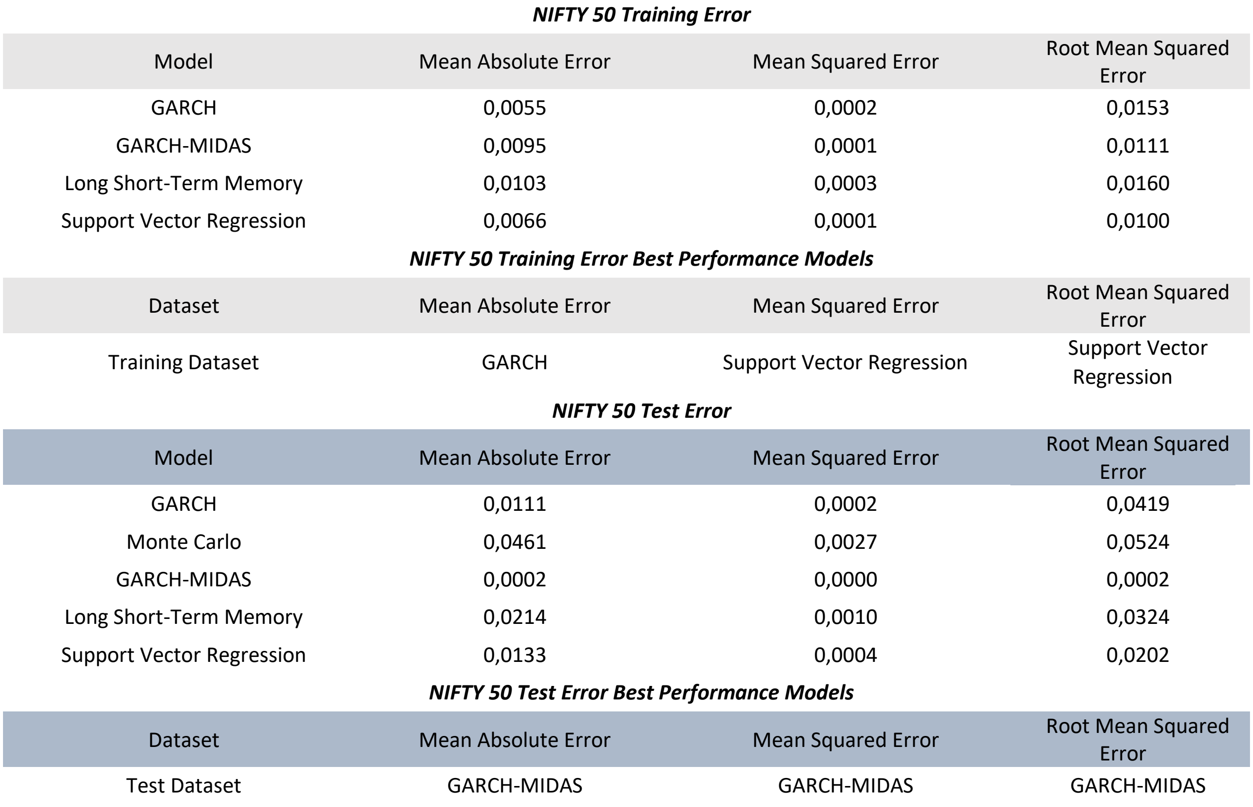
NASDAQ Test Error

Model	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
GARCH	0,0076	0,0000	0,0245
Monte Carlo	0,0456	0,0024	0,0492
GARCH-MIDAS	0,0002	0,0000	0,0002
Long Short-Term Memory	0,0216	0,0007	0,0267
Support Vector Regression	0,0161	0,0003	0,0177

NASDAQ Test Error Best Performance Models

Dataset	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Test Dataset	GARCH-MIDAS	GARCH-MIDAS	GARCH-MIDAS

Figure 11- NASDAQ Forecast Error



NIFTY 50 Training Error

Model	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
GARCH	0,0055	0,0002	0,0153
GARCH-MIDAS	0,0095	0,0001	0,0111
Long Short-Term Memory	0,0103	0,0003	0,0160
Support Vector Regression	0,0066	0,0001	0,0100

NIFTY 50 Training Error Best Performance Models

Dataset	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Training Dataset	GARCH	Support Vector Regression	Support Vector Regression

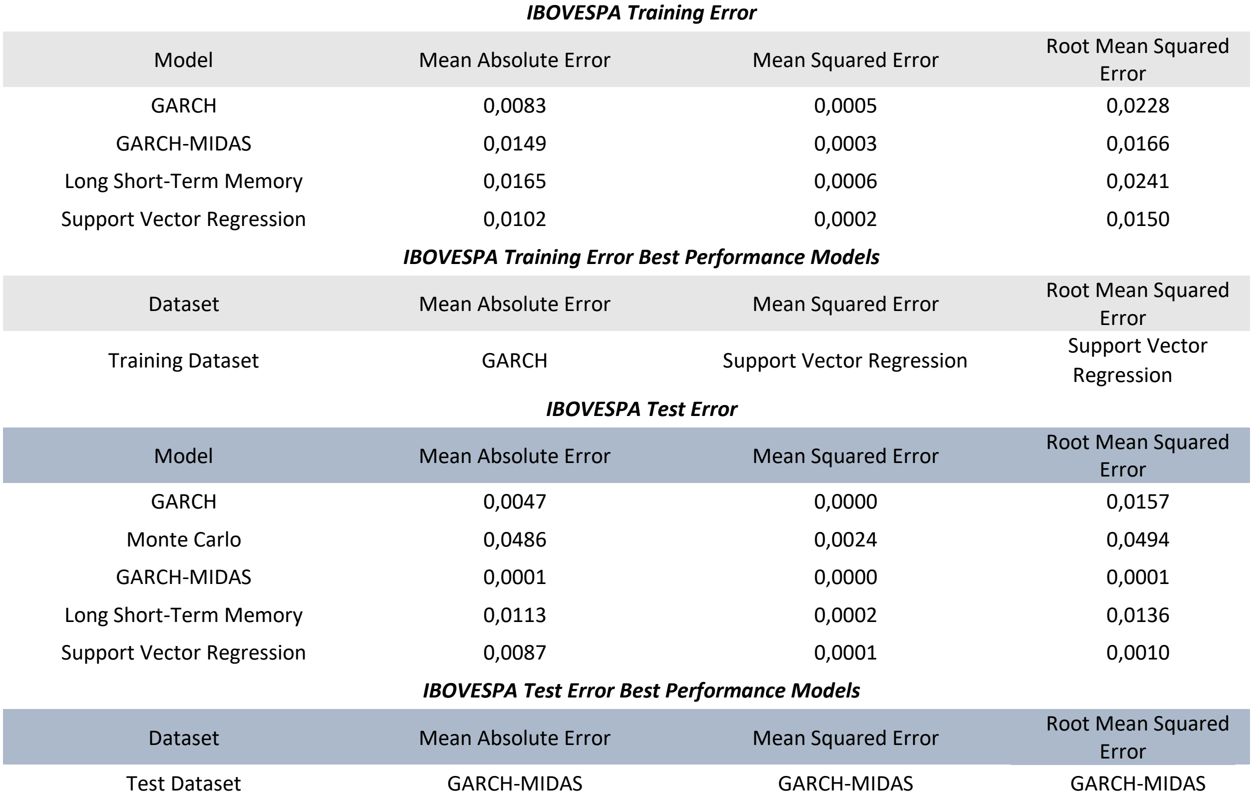
NIFTY 50 Test Error

Model	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
GARCH	0,0111	0,0002	0,0419
Monte Carlo	0,0461	0,0027	0,0524
GARCH-MIDAS	0,0002	0,0000	0,0002
Long Short-Term Memory	0,0214	0,0010	0,0324
Support Vector Regression	0,0133	0,0004	0,0202

NIFTY 50 Test Error Best Performance Models

Dataset	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Test Dataset	GARCH-MIDAS	GARCH-MIDAS	GARCH-MIDAS

Figure 13- NIFTY 50 Forecast Error



IBOVESPA Training Error

Model	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
GARCH	0,0083	0,0005	0,0228
GARCH-MIDAS	0,0149	0,0003	0,0166
Long Short-Term Memory	0,0165	0,0006	0,0241
Support Vector Regression	0,0102	0,0002	0,0150

IBOVESPA Training Error Best Performance Models

Dataset	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Training Dataset	GARCH	Support Vector Regression	Support Vector Regression

IBOVESPA Test Error

Model	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
GARCH	0,0047	0,0000	0,0157
Monte Carlo	0,0486	0,0024	0,0494
GARCH-MIDAS	0,0001	0,0000	0,0001
Long Short-Term Memory	0,0113	0,0002	0,0136
Support Vector Regression	0,0087	0,0001	0,0010

IBOVESPA Test Error Best Performance Models

Dataset	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Test Dataset	GARCH-MIDAS	GARCH-MIDAS	GARCH-MIDAS

Figure 14- IBOVESPA Forecast Error

Hang Seng Training Error

Model	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
GARCH	0,0068	0,0003	0,0170
GARCH-MIDAS	0,0113	0,0001	0,0116
Long Short-Term Memory	0,0129	0,0003	0,0171
Support Vector Regression	0,0080	0,0001	0,0110

Hang Seng Training Error Best Performance Models

Dataset	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Training Dataset	GARCH	Support Vector Regression	Support Vector Regression

Hang Seng Test Error

Model	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
GARCH	0,0101	0,0001	0,0326
Monte Carlo	0,0524	0,0030	0,0551
GARCH-MIDAS	0,0002	0,0000	0,0003
Long Short-Term Memory	0,0212	0,0006	0,0237
Support Vector Regression	0,0111	0,0003	0,0164

Hang Seng Test Error Best Performance Models

Dataset	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Test Dataset	GARCH-MIDAS	GARCH-MIDAS	GARCH-MIDAS

Figure 15- Hang Seng Forecast Error

On the below Figures, is possible to observe the statistical significance between the forecasting capacity, by the Mean Squared Error, by Model, by Index. All values with P-value bigger than 0.05 are highlighted as red.

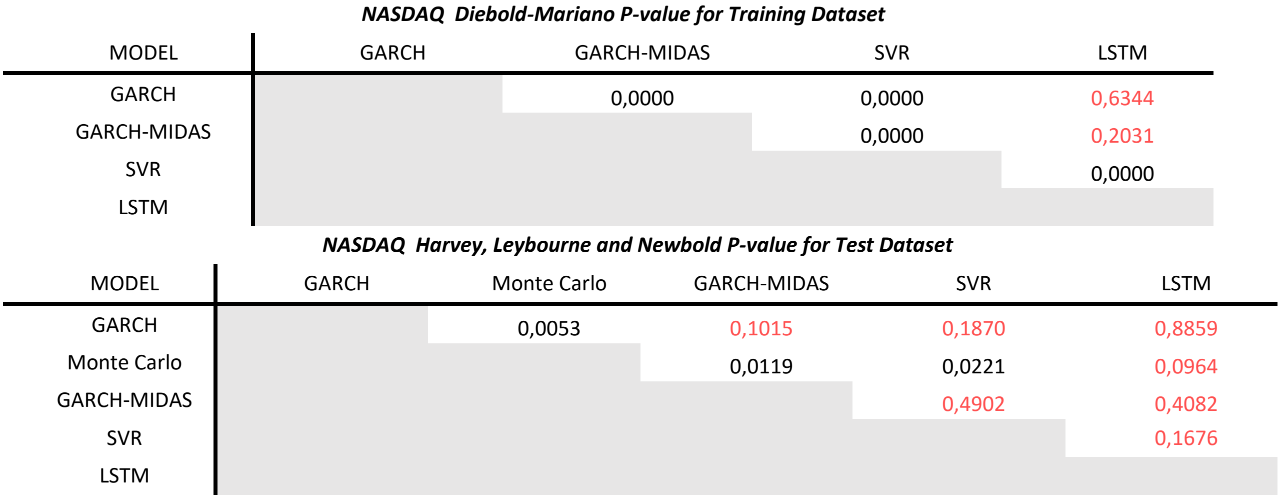
Euronext 100 Diebold-Mariano P-value for Training Dataset

MODEL	GARCH	GARCH-MIDAS	SVR	LSTM
GARCH		0,0484	0,0000	0,7995
GARCH-MIDAS			0,0000	0,4472
SVR				0,0000
LSTM				

Euronext 100 Harvey, Leybourne and Newbold P-value for Test Dataset

MODEL	GARCH	Monte Carlo	GARCH-MIDAS	SVR	LSTM
GARCH		0,0029	0,1987	0,2064	0,4125
Monte Carlo			0,0028	0,0063	0,0902
GARCH-MIDAS				0,2818	0,3574
SVR					0,0980
LSTM					

Figure 16- Euronext 100 Statistic Test DM and HLN



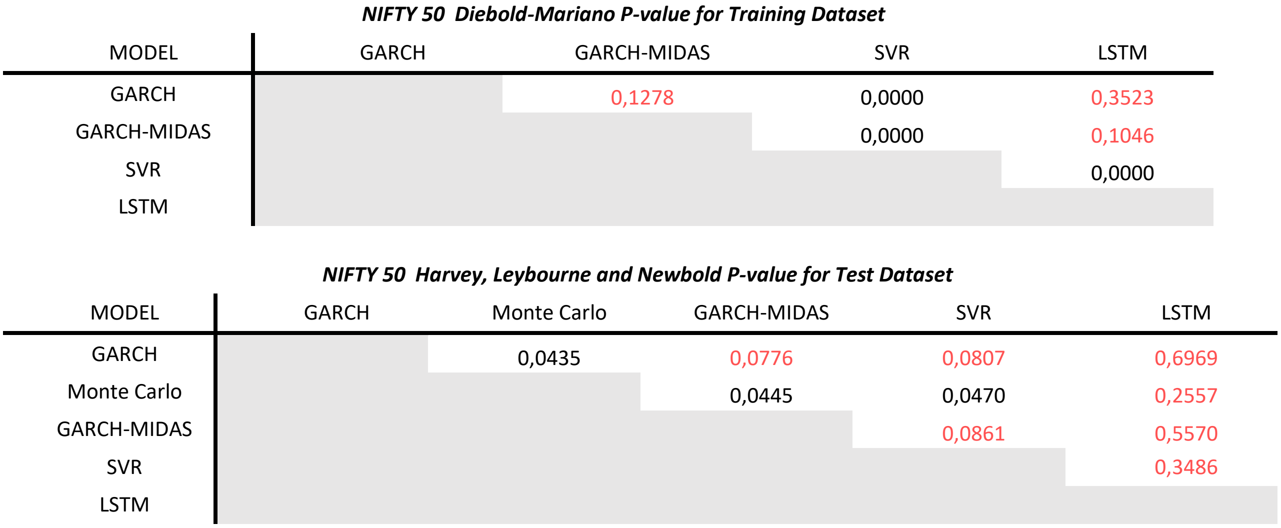
NASDAQ Diebold-Mariano P-value for Training Dataset

MODEL	GARCH	GARCH-MIDAS	SVR	LSTM
GARCH		0,0000	0,0000	0,6344
GARCH-MIDAS			0,0000	0,2031
SVR				0,0000
LSTM				

NASDAQ Harvey, Leybourne and Newbold P-value for Test Dataset

MODEL	GARCH	Monte Carlo	GARCH-MIDAS	SVR	LSTM
GARCH		0,0053	0,1015	0,1870	0,8859
Monte Carlo			0,0119	0,0221	0,0964
GARCH-MIDAS				0,4902	0,4082
SVR					0,1676
LSTM					

Figure 17- NASDAQ Statistic Test DM and HLN



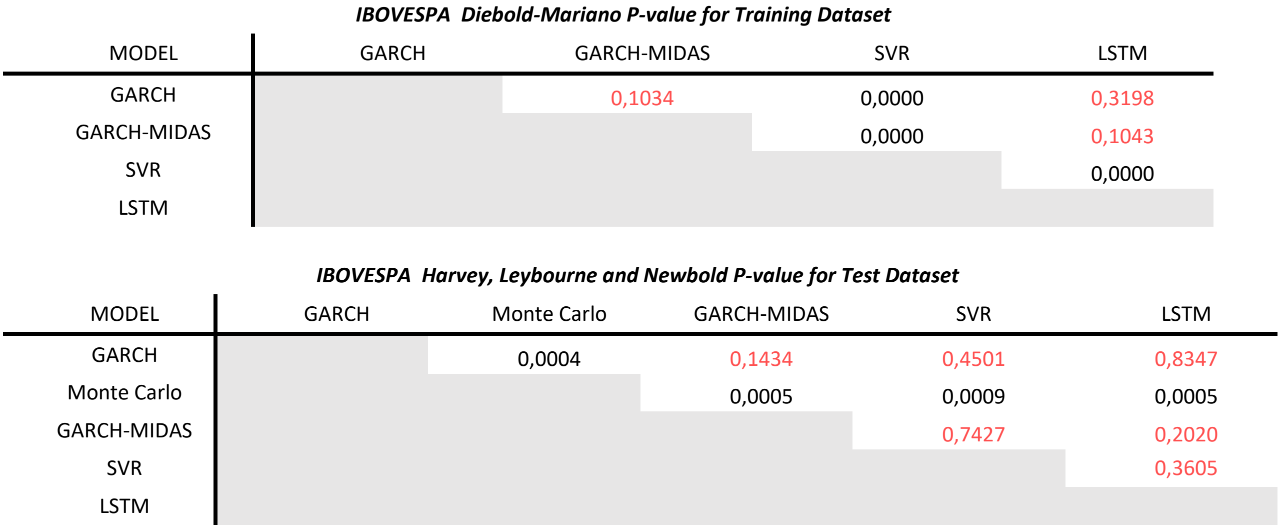
NIFTY 50 Diebold-Mariano P-value for Training Dataset

MODEL	GARCH	GARCH-MIDAS	SVR	LSTM
GARCH		0,1278	0,0000	0,3523
GARCH-MIDAS			0,0000	0,1046
SVR				0,0000
LSTM				

NIFTY 50 Harvey, Leybourne and Newbold P-value for Test Dataset

MODEL	GARCH	Monte Carlo	GARCH-MIDAS	SVR	LSTM
GARCH		0,0435	0,0776	0,0807	0,6969
Monte Carlo			0,0445	0,0470	0,2557
GARCH-MIDAS				0,0861	0,5570
SVR					0,3486
LSTM					

Figure 18- NIFTY 50 Statistic Test DM and HLN



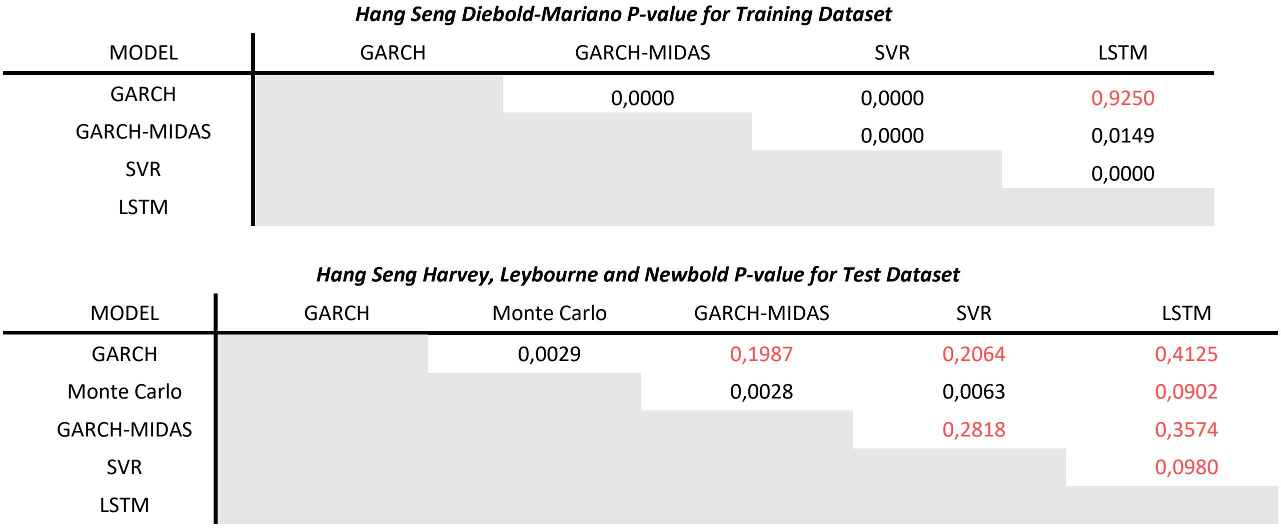
IBOVESPA Diebold-Mariano P-value for Training Dataset

MODEL	GARCH	GARCH-MIDAS	SVR	LSTM
GARCH		0,1034	0,0000	0,3198
GARCH-MIDAS			0,0000	0,1043
SVR				0,0000
LSTM				

IBOVESPA Harvey, Leybourne and Newbold P-value for Test Dataset

MODEL	GARCH	Monte Carlo	GARCH-MIDAS	SVR	LSTM
GARCH		0,0004	0,1434	0,4501	0,8347
Monte Carlo			0,0005	0,0009	0,0005
GARCH-MIDAS				0,7427	0,2020
SVR					0,3605
LSTM					

Figure 19- IBOVESPA Statistic Test DM and HLN



Hang Seng Diebold-Mariano P-value for Training Dataset

MODEL	GARCH	GARCH-MIDAS	SVR	LSTM
GARCH		0,0000	0,0000	0,9250
GARCH-MIDAS			0,0000	0,0149
SVR				0,0000
LSTM				

Hang Seng Harvey, Leybourne and Newbold P-value for Test Dataset

MODEL	GARCH	Monte Carlo	GARCH-MIDAS	SVR	LSTM
GARCH		0,0029	0,1987	0,2064	0,4125
Monte Carlo			0,0028	0,0063	0,0902
GARCH-MIDAS				0,2818	0,3574
SVR					0,0980
LSTM					

Figure 16- Hang Seng Statistic Test DM and HLN

5. Conclusions

This was an overly broad study, in which it was included five different models, from five different model "families". The period under analysis was wide and was also subject to an abnormal level of volatility, during the Covid-19 pandemic Period, as it is possible to conclude from the figures on the appendix, in which all Indexes were highly impacted, albeit some more than others. When it comes to the study, by itself, it is possible to understand that both GARCH Models were capable of predicting with an high level of accuracy, being the GARCH-MIDAS Model, the best predictive model across all Indexes Test Dataset with low frequency data being assumed on the returns, as a function of the weight each one have on the Target Variable (Beta Weighting) and accounting for Macro Financial information such as the House Pricing and the Long Term Interest Rate, by the Index Origin (Geographical Location). This article also allows to conclude, and the reader to comprehend, that both Econometric Models and Machine Learning Models, provide a more accurate prediction of the actual returns, than a Monte Carlo Simulation (Random Model), which, even by considering the key statistics moments of the return's distribution, it was the worst predictive model, under all scenarios, proving that sophisticated models, that allow for multivariable, and that provide different weights to more recent data when predicting the Target Variable, have a more accurate predictive capacity than a model that considers the same weight per observation over the entire length of the period under analysis. When observing the Diebold-Mariano and Harvey, Leybourne and Newbold figures, is possible to note that there are a significant number of red values, meaning, that the probability of the value being higher than 0.05 is significant, and with that should provide enough information to reject H_0, meaning that it could be considered that both models have different capacity accuracy. In theoretical terms, that could equate to saying that it is not possible to conclude which model did in fact predict with the highest accuracy, nevertheless, in Graphical Terms (Appendix figures) and by using the Accuracy Measurement Models (MAE, MSE,RMSE) it is possible to detect a pattern, and with that have enough confidence to assume that the forecasting "Scores" are acceptable. The reason why I personally believe that the GARCH-MIDAS, was the best performing model on the Test Dataset for all Indexes, is due to the fact that, the forecasted values under GARCH-MIDAS were almost a constant in the Test Dataset, meaning that it was not "trying" to keep the actual value of the return, but was actually working as a continuation of the Training Dataset, i.e., if the Test Dataset Returns, follow the volatility pattern as the ones in Training Dataset, the GARCH-MIDAS has a high degree of accuracy. But if the difference turns out to be bigger, let's assume, once again, in my personal belief but also based on empirical knowledge acquired whilst writing this article, models such as the Support Vector Regression and Long Short-Term Memory have a higher capacity of "catching" these movements, and with that have a higher forecasting capacity.

6. Limitations and Recommendations for Future Works

This study, as explained in the Chapter 5, was very broad, in which the main goal was by using small tweaks to the dataset, try to understand the impact on the predictive capacity of each type of model. As a first limitation is the fact that for some models, only two or three exogenous variables were used, when these ones allow for much more, and with that even achieve a higher level of accuracy. For example, an Investment Manager that works directly with the India Stock Market, that knows which type of Exogenous variables, such as Macro Variables, other stocks and stocks indexes correlated with the NIFTY 50, could provide a much higher level of information to the model, and with that even achieving a higher accuracy level on its prediction. One recommendation is to try these models, not only with different variables, but also with different forecasting time frames, whether higher or smaller timeframes. Sometimes Investment decisions need to be performed under a high level of pressure, and one- or two-days difference could equate to an extreme situation for some members of the investment/Pension fund, that do not wish to have a risk profile whilst investing, and without a daily study, they could be put under that scenario. Other limitation, and this one could have impacted more the result for the best model, is the computational capacity, especially for models such the Support Vector Regression and the Long Short-Term Memory. The last one is a Recurrent Neural Network, and working by layers, going back and forth trying to allocate different weights to the given observations, in order to achieve the most accurate forecast as possible. Regarding what I've personally observed whilst coding the Support Vector Regression Model, was that by increasing the "Cost" of the function, the Forecasting Capacity increases significantly. Nevertheless, this is a really heavy model, that takes a really long time to run, and would be practically impossible, under the current situations, to write this article based on that, so the "Cost" was reduced to be limited to an acceptable value, in which the model is capable of running on a not abnormal time.

Our main recommendations to any reader that wishes to continue this work, or use it as a basis for, not only the empirical knowledge to anchor their work, but also for ideas, is to firstly, try to use any other type of GARCH Model, such as FIGARCH, try to use other variables on the GARCH-MIDAS, try to use a Test Dataset that steps a bit beside the pattern of the Training Dataset, try to input exogenous variables on the Long Short-Term Memory Model, since this one, based on the literature available, is one of the most famous models used to predict returns and stock prices, and also on the same basis, try to apply more recent models, such as the Transformer or Multi Transformer. For those readers that wish to go a step even further, I will recommend trying to create a new model, an ensemble model, in which you combine the predictions of five or six models already included in your work, and by allocating different weights to them and then this new model will be able to maximize the training forecasting capacity. Finally, the use of sentiment analysis, i.e., human behavior and the way it affects the stock market, is a field of the literature that is rapidly increasing, and by using it as an input, it could generate an unprecedent forecasting capacity level.

Acknowledgments

This research was supported by NOVA IMS, Universidade de Lisboa, with the Support of Professor Bruno Damásio with the main objective of working as a dissertation paper that would allow me to finish my personal master degree in *"Statistics and Information Management, with specialization in Risk Management".*

R Code and Excel Spreadsheets that support the Study.

Supplementary material related to this article can be found online at FB GitHub-> Forecasting Volatility.

Appendix with detailed information about each model.

FB GitHub-> Appendix

References

Andersen, T., Davis, R., Kreiss, J.-P., & Mikosh, T. (2009). *Handbook of Financial Time Series.* Berlim: Springer.

Awad, M., & Khanna, R. (2015). *Efficient Learning Machines.* Apess Open.

Brewer, K., Feng, Y., & Kwan, C. (2012). Geometric Brownian motion, option pricing, and simulation: some spreadsheet-based exercisies in financial modelling. *Spreadsheets in Education*.

Buturac, G. (2021). Measurement of Economic Forecast Accuracy: A systematic Overview of the Empirical Literature. *Journal of Risk and Financial Management*.

Conrad, C., Custovic, A., & Ghysels, E. (2018). Long- and Short-Term Cryptocurrency Volatility Components: A GARCH-MIDAS Analysis. *Journal of Risk and Financial Managment*.

Constantini, M., & Knust, R. M. (2011, November). On the Usefulness of the Diebold-Mariano Test in the Selection of Prediction Models: Some Monte Carlo Evidence.

Deevi, S. (2015, June 01). The Big Data-Driven Business: How to Use Big Data to Win Customers, Beat Competitors, and Boost Profits.

Dougherty, C. (2016). *Introduction to Econometrics* (Vol. 5). United Kingdom: Oxford University Press.

Engle, R. F., Ghysels, E., & Sohn, B. (2013). Stock market volatility and macroeconomic fundamentals. *Review of Economics and Statistics*, 776-797.

Esling, P., & Agon, C. (2012). Time-Series Data Mining.

Falinouss, P. (2007, June). Stock Trend Prediction Using News Articles. p. 90.

Harvey, D., Leybourne, S., & Newbold, P. (1997). Testing the Equality of Prediction Mean Squared Errors. *International Journal of Forecasting*, 281-291.

Houdt, G. V., Mosquera, C., & Nápoles, G. (2020, May 13). A review on the Long-Short Term Memory Model. p. Springer Nature.

Hull, J. C. (2018). *Risk Management and Financial Institutions.* Wiley.

Kim, H. Y., & Won, C. H. (2018, March 06). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models .

Ma, J., Xiong, X., & Feng, X. (2021). News Release and the Role of different types of investors. *International Review of Financial Analysis*.

Poon, S.-H., & Granger, C. (2001, June 11). Forecasting Financial Market Volatility.

Reddy, K., & Clinton, V. (2016). Simulating Stock Prices Using Geometric Brownian Motion: Evidence from Australian Companies. *Australasian Accounting, Business and Finance Jornal*, p. 27.

Sengupta, C. (2004). *Financial Modeling Using Excel and VBA.* Sidney: Wiley Finance.

Silva, F. O. (2021, May 21). A MACHINE LEARNING APPROACH TO PREDICTING STOCK RETURNS.

Torre-Torres, O. V., Venegas-Martínez, F., & Martínez-Torre-Enciso, M. I. (2021, 01 18). Enhancing Portfolio Performance and VIX Future Trading Timing with Markov-switching GARCH Models.

Willmott, C. J., & Matsuura, K. (2005, December 19). Advantages of the mean absolute error over the root mean square error in assesing average model performance. pp. 79-82.

Wong, Z. Y., Chin, W. C., & Tan, S. H. (2016, December 19). Daily value-at-risk modeling and forecast evaluation: The Realized volatility approach.

Xu, Y., & Goodacre, R. (2018). On Splitting Training and Validation Set: A Comparative Study of Cross Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *Journal of Analysis and Testing*.