ERASMUS UNIVERSITY ROTTERDAM
ERASMUS SCHOOL OF ECONOMICS
Bachelor Thesis Economics & Business
Specialization: Financial Economics

Volatility forecasting models for Value-at-Risk estimation

A comparative study of GARCH, SVR-GARCH, and LSTM

Author: Filip Borzęcki
Student number: 611297
Thesis supervisor: Philip Messow
Second reader: Kan Ji
Finish date: 30.06.2024

The views stated in this thesis are those of the author and not necessarily those of the supervisor, second reader, Erasmus School of Economics or Erasmus University Rotterdam.

ABSTRACT

This paper analyses three prominent models – GARCH, SVR-GARCH, and LSTM in volatility forecasting and one-day Value-at-Risk estimation. The three models are evaluated on two indexes – VTI, representing the general stock market and BND representing the general bond market. Out-of-sample performance of the model is evaluated using Mean Squared Error, whereas Value-at-Risk accuracy is evaluated using the binomial test and two loss functions. The study showed that a more sophisticated model, such as LSTM or SVR-GARCH can provide more accurate forecasts and Value-at-Risk estimates (under the loss function evaluation). The results of this study indicate that these models can provide superior forecasting ability and more reliable risk estimates for entities that are exposed to market risk and fluctuations.

Keywords: value-at-risk, volatility forecasting, machine learning, deep learning

TABLE OF CONTENTS

CHAPTER 1 Introduction

Value at Risk (VaR) is a fundamental tool for managing market risk. In simple terms, it estimates how much capital may be lost due to market fluctuations in each time interval and under a given confidence level (1-α) and the distribution that the returns are assumed to follow. Given this definition of VaR, one can also view it as a prediction of the α% quantile of a portfolio's return distribution (Bams et al., 2017). Since returns are inherently related to volatility, it is the forecasts of the latter that matter the most for the accuracy and reliability of VaR estimates. This paper will examine several methods of volatility forecasting - from the classic time-series model (GARCH) to a more sophisticated machine learning-based (ML) approach (SVR-GARCH) and deep learning (DL) approach (LSTM). The aim is to provide an overview of models that have been developed throughout the years and see which one is the most reliable for both general volatility forecasting and VaR estimation. The application of volatility models to VaR is of special interest since superior performance of a given model across the entire distribution does not necessarily imply superior performance at the tails, which are at the core of VaR estimation.

Many studies have been done on the topic of volatility estimation throughout the years. The naive method based on extrapolating historical average volatility into the future has been proven to be outperformed by implied volatility - volatility derived from a given option pricing model (Blair et al., 2010). Implied volatility, however, has been shown to exhibit several drawbacks as well. Bams et al. (2017) compared the performance of implied volatility-based VaR for the S&P500, DIJA and NASDAQ100 indices with GJR-GARCH and concluded that the latter outperforms the former. The motivation for this finding is that the implied volatilities are biassed due to volatility risk premia that they incorporate, making the VaR estimates of poor accuracy. The development of advanced statistical methods, broadly referred to as machine learning and deep learning models, extended the range of available approaches to volatility forecasting. Zhu et al. (2023) examined the performance of several such methods (Lasso, Random Forest, Deep Neural Network, and others). They discover that training the models on panel data, as opposed to individual time series, provides superior performance in forecasting. They also indicated that the ML models provided the most accurate estimates, due to their ability to capture the nonlinear character of stock market data as well as the interconnectedness of the features. One important feature of the data used by Zhu et al. (2023) is the removal of outliers, defined as 1% of trading days with the highest volatility. Such sampling would not be appropriate in VaR-related applications. Value at Risk models need to perform well also, or even primarily, in highly volatile periods, where the probability of extreme losses is the most significant. Another proof of the superior performance of ML approaches (and more precisely Neural Networks (NNs)) comes from Zhang et al. (2024). This research showed that NNs exhibit superior forecasting capabilities compared to tree-based models or linear regressions. These findings were derived from data for 93 stocks but, more interestingly, the training process exploited the commonalities across stocks by combining individual features with market features.

Current literature on volatility forecasting provides multiple proofs for the superior performance of more nuanced ML models compared to standard approaches. The recurring justification for this fact is the ability of these more complex models to handle latent interactions among variables and the nonlinear character of volatility. These findings, thus, suggest that similarly superior performance could be achieved in the context of Value at Risk. Some proofs for this hypothesis come from Karasan and Gaygısız (2020), who showed that SVR-GARCH provides superior VaR estimates compared to other models from the GARCH family. Adding other models to this analysis will allow for a more comprehensive comparison and more informed model selection for both practitioners and regulators. The question that this paper will attempt to answer is how the choice of volatility estimation model affects VaR performance and which model provides the best performance (among the models analysed).

In this paper, several models that have been tested so far will be applied to VaR estimates. The models were selected in such a way that each family of models is represented in the comparison. In particular, historical average (naive approach), GARCH (econometrics approach), SVR-GARCH (machine learning approach), and LSTM (deep learning approach) were chosen for this study. To examine how each model performs under different market conditions and on different asset classes, each model will be evaluated on two ETFs - VTI (Total Stock Market ETF) representing US equities and BND (Total Bond Market ETF) representing the universe of US bonds (both government and corporate with different ratings). Zhu et al., 2023 showed that realised volatilities (1-day, 5-day, 22-day) and downside realised semivariance (1-day, 5-day, 22-day) are the most important features in the training process. This paper will focus on the former. The dataset will span 16 years of daily sampled data and will be divided into training, testing, and validation subsets. The data will be sourced from Wharton Research Data Services. The back-tests for VaR estimates will be based on a binomial test and two loss functions, to provide a more nuanced perspective on the performance of the models.

Based on the studies described above, a superior performance of more sophisticated models should be observed. Back-tests of VaR estimates, which use ML or DL approaches should reveal that the ability of these models to capture interdependencies between variables and the nonlinear character of volatility results in more accurate estimates of volatility in the tails of the returns distribution and, thus, more reliable VaR estimates. The relative performance of the models from these two families, however, is not yet clear and will be assessed in this paper.

CHAPTER 2 Theoretical Framework

2.1 Volatility forecasting: the evolution of approaches

The volatile nature of asset prices is of the highest concern for anyone involved in the financial markets. Because of that, throughout the years many academics and practitioners have been trying to develop methods that would allow them to accurately estimate the volatility of their assets' prices in future periods, to make better trades or to manage their risk more effectively. If one assumes that a given time series follows a random walk-like pattern (i.e. each following observation deviates from the previous one by a random step that is identically and independently distributed), then the best forecast one can give for the next observation is the value of the current one. Asset returns are considered by many as such a time-series process (Solnik (1973) or Lee (1992)). Volatility, on the other hand, is quite different. Proofs of the predictable nature of volatility come from both developed (Harvey & Whaley, 1992) and emerging markets (Santis & Imrohoroglu, 1997).

The search for a robust model that would overperform naive approaches was initiated by Engle (1982), where the Autoregressive Conditional Heteroskedasticity (ARCH) family of models was introduced. The main, ground-breaking innovation of this model was that the assumption of a constant one-period forecast variance was relaxed and converted to an assumption stating that the variances can be conditional on the past and, thus, non-constant. This allowed for more realistic modelling of volatilities and was proven by the author to produce reliable results (based on the case of variance of inflation in the UK). Based on the ARCH model, several extensions were developed that allowed for more general applications with weaker assumptions about the data-generating process. Most notably, the Generalised Autoregressive Conditional Heteroskedasticity (GARCH) was developed by Bollerslev (1986). GARCH introduced the lags of the variance to the equation, allowing for a more parsimonious specification and more accurate predictions. The GARCH model will be derived and applied to Value at Risk estimation, serving as a benchmark in this analysis. Another innovation in the GARCH family was the GJR-GARCH model introduced by Glosten, Jagannathan, and Runkle (1993). This model accounts for the asymmetric reactions of financial markets to negative and positive news, with the more significant increases in volatility being caused by recent negative returns. Another step in the direction of non-linear models that account for the signs of past shocks was introduced by Taylor (2004). Adaptive (i.e. changing over time) exponential smoothing was proven to outperform the GARCH family of models. Similar results were found by McAleer and Medeiros (2008) and Liu et al. (2020). Finally, the Heterogeneous Autoregressive model for volatility was introduced by Corsi (2009). The main feature of this model is its ability to incorporate separate coefficients for volatility components realised over different time horizons. Such a design allows the model to successfully model the long memory feature of financial returns - shocks from the past have long-lasting effects in the future.

According to Engle and Patton (2007), a good volatility model can incorporate

1. persistence (i.e. volatility clustering) - large (small) moves are often followed by large (small) moves,
2. mean-reversion - periods of both abnormally high and abnormally low volatility will eventually revert to the long-term average level of volatility,
3. asymmetry - negative shocks have a higher impact on volatility than positive ones (leverage effect or risk premium effect are prominent explanations for this phenomenon),
4. effect of exogenous variables on volatility - a time series of observed volatilities or returns are not able to capture all influential events that happen in the market and that affect volatility,
5. significant kurtosis in the distribution of returns - returns are not normally distributed and exhibit heavy tails, indicating that a good volatility model should not require the assumption of Gaussian distribution of returns to be met.

Developments of the past decade in computational and data-processing power allowed for a new branch of volatility forecasting models to be developed. Machine learning (ML) and deep learning (DL) models have several features that make them prominent candidates to meet the requirements of a good volatility model. As observed by Prakash et al. (2020) modelling approaches based on unsupervised learning (where the data has no labels) present a significant ability to detect and separate multiple regimes within a time series. This indicated that such models should also perform well in volatility forecasting scenarios, where spotting volatility clusters is essential, as explained by Engle and Patton (2007). Thanks to the large number of parameters that are estimated in ML models and their ability to interpret nonlinearities in the data, patterns such as asymmetry and mean-reversion can also be efficiently captured and then used for more accurate predictions. This was shown in the example of SVR-GARCH by Peng et al. (2018). Moreover, the number and type of features that an algorithm is trained on are flexible, making a model more able to account for exogenous variables even when they are highly correlated with each other, as shown by Bucci (2020).

2.2 Volatility forecasting models for estimating Value at Risk

Value-at-Risk can be succinctly defined as a tool for measuring an entity's exposure to market risk (Linsmeier & Pearson, 1996). A plethora of methods that can be used to estimate this parameter have been developed throughout the years. A comprehensive overview of these methods compiled by Abad et al. (2014) distinguishes non-parametric, semi-parametric, and parametric methods. They differ primarily in the assumption they make about the distribution of the analysed time series. While the non-parametric models do not require the returns to follow the Gaussian distribution, the parametric ones do. This paper will focus on the latter type.

The GARCH model was chosen as a benchmark. It is a classic model that, according to Engle and Patton (2007), fulfils all the requirements of a good volatility-modelling device. Bams et al. (2017) found that the GARCH family of models outperforms other potential methods - option implied volatility, a method based on inverting an option pricing formula, assuming that the observed price is derived from that formula. According to the authors, the volatility risk premium embedded in implied volatilities caused the poor performance of the option-based Value-at-Risk. Byun and Cho (2013) confirmed this finding in a pure volatility forecasting setting on carbon futures products. Other studies, however, found significantly better forecasting performance of implied volatility compared to GARCH (see Mayhew and Stivers (2003) or Giot (2003)). However, since this study is specifically focused on Value at Risk estimation, the results of Bams et al. (2017) will be followed.

Literature also provided plenty of evidence for the superior volatility forecasting abilities of machine learning models. Vrontos et al. (2021) found that machine learning models show better results both in economic and statistical settings. They showed that on the VIX index (implied volatility index based on S&P 500 options), using several classic machine learning methods, such as Lasso, Elastic Net, or Trees. Following these findings, a model that appears in the literature very often as one that provides superior performance when compared to both econometric and other ML-based methods will be tested. Support Vector Regression-GARCH model is a combination of ML and econometric approaches. In this specification, the Support Vector Regression model, introduced by Vapnik and Cortes (1995), is used to estimate the parameters of a GARCH model. The source of SVR performance lies in its ability to nonlinearly map the input space onto a feature space of higher dimensionality and perform linear regression on the latter, thus capturing the character of the data more accurately (Chen et al., 2008). Current literature contains plenty of evidence for superior performance of this method for volatility forecasting, not only in the stock universe but also in other asset classes. Peng et al. (2018) showed that SVR-GARCH exhibits superior performance over GARCH and GARCH extensions (such as GJR-GARCH or EGARCH) in forecasting volatility on selected currencies and cryptocurrencies. This shows that the SVR-GARCH model can outperform also in highly volatile markets. Superior performance in general volatility forecasting does not immediately imply superior accuracy in Value-at-Risk estimation. Current literature on this use case, however, is more limited. Karasan and Gaygısız (2020) showed that applying SVR-GARCH to Value-at-Risk estimation gives more accurate results and can improve an entity's risk management capabilities. They showed this on a selection of 30 stocks from the S&P 500 index. Lux et al. (2017) noted that parametric models when used to estimate Value at Risk, may produce biased results or even underestimate the risk due to factors such as the time-varying nature of volatility or skewness and heavy tails of the financial returns distribution. As a solution to these issues, they proposed an SVR-GARCH specification with a kernel density estimation to account for the heavy tails of the returns distribution. These examples from the literature show that the superior performance of the SVR-GARCH model is replicable across asset classes and is persistent also when the use-case changes from general volatility forecasting to VaR estimation. This paper will put this method to the test against other prominent models, such as GARCH or LSTM.

The search for the best-performing volatility forecasting model does not stop here, however. Popularised in 1986 by Geoffrey Hinton, neural networks (NNs) also exhibit several characteristics that make them well-suited for volatility forecasting scenarios. Already in 1997, Miranda and Burgess (1997) showed that artificial neural networks, when designed to incorporate the stylised fact of persistence in volatility, outperform linear specifications. This superior performance, according to the authors, can be linked to NNs' flexibility and ability to account for non-linearities in the financial returns time series. A comparison of SVR and LSTM provided by Liu (2019) showed that, overall, the two models performed similarly in large intervals, both outperforming the GARCH model. The author noticed, however, that LSTM's potential advantage over SVR comes from its ability to interpret and learn from long sequences of data and its capacity to store that information in many hidden layers. This also indicates that LSTM could potentially provide more accurate VaR estimates. Similar evidence in favour of LSTM comes from Bucci (2020). In this paper, the author took advantage of LSTM's ability to handle large datasets that cover a wide range of features and to spot regime switches in time series of volatility. LSTM exhibited strong performance in interpreting non-linearities and linearities in the data, even when the data-generating process and the number of regime switches are unknown. The comparison of LSTM to classic econometric models showed that the former performs better in volatility forecasting scenarios on the S&P 500 index. Following these pieces of evidence from the literature, the LSTM model will be chosen for comparison in this paper and evaluated in the Value-at-Risk estimation use case.

Even though LSTM is seen in the literature the most often, it is not the only model that can provide significant forecasting abilities. Song et al. (2023) put a different kind of neural network architecture to the test and achieved improved performance both in the US and Chinese markets. The hybrid model they built was based on a combination of convolutional neural networks (CNN) and gated recurrent units (GRU). The CNN-based part of the model was able to extract significant features from the training data automatically, whereas the GRU-based part allowed the processing of both long- and short-term serially dependent features, which is crucial in any volatility-related scenario. Another architecture that is often used in volatility forecasting is trees. Teller et. al (2022) showed that Extreme Gradient Boosting, a model based on trees, performed better than both LSTM and HAR (on the data of selected major US stocks). How these models perform in the VaR estimation use case has not been examined yet.

CHAPTER 3 Data

3.1 Market indexes

The models' performance will be assessed on two indexes, representing two asset classes. Vanguard Total Bond Market ETF will be used as a proxy for the general US bond market. It provides exposure to both US government bonds and US corporate bonds of different ratings (ranging from AAA to BBB). Time-to-maturity of its holdings is also diverse, with around 40% of holdings maturing in 1-5 years, around 36% maturing in 5-10 years, and the remaining part maturing at even longer time horizons. Since historically bonds tended to exhibit lower volatility (Solnik 1996), testing the models on this market will allow us to examine how they perform under less volatile market conditions. To represent a more volatile market – the US equity market, Vanguard Total Stock Market ETF will be used as a proxy. Its holdings are widely diversified across industries and cover both value and growth stocks. The high liquidity of these assets throughout the sample allows to minimise of the liquidity risk aspect of risk management and keeps the analysis focused on market risk and volatility. Prices of the indexes were sampled daily, ranging from mid-2007 until the end of 2023. The data was sourced from CRSP, Wharton Research Data Services.

3.2 Volatility

Daily prices of BND and VTI indexes are converted to log-returns instead of simple returns since the former exhibit certain valuable properties such as additivity. Using formula (1):

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right)$$

daily returns of a given index at day t were computed. Variables representing volatility were computed using the close-close approach with an assumption of no drift of returns, which is also consistent with how Value-at-Risk will be computed in this analysis. Using the formula (2):

$$\sigma_t = \sqrt{\frac{1}{N}\sum_{i=1}^{N} r_i^2}$$

daily volatility estimates are obtained. The formula follows the work by Figlewski (1997). Since the mean of returns is assumed to be zero, the sum can be adjusted by dividing by N (and not by N-1, as it would be in a case of non-constant mean, where one observation would be lost to calculate that mean). Two volatility measures were computed using this formula – a 5-day rolling window and a 22-day rolling window. Following Cruz et. al (2003), the former will serve as the main measure of volatility and will be used as the target value (or a label) for the tested models. The latter will be used for the LSTM as a feature containing

a month of volatility information. The size of the rolling window was chosen following Zhu et al. (2023). Finally, 5-days volatility, 22-days volatility, squared returns, and implied volatility were lagged by 1-10 steps. These features will form the tensor that will be used for training the LSTM model.

Features such as downside realised semivariances turned out to exhibit high importance for the tested models in the study by Zhu et al. (2023) but they were excluded from this analysis for simplicity. They may, however, exhibit explanatory power for realised volatility, despite high correlation with the lags of volatility. Performance and liquidity measures (such as Sharpe ratios, trading volume, or market beta) were also included by the authors but turned out to be of lesser importance for most of the models. Relative signed jump, realised kurtosis, and realised skewness were not included in this study for the same reason.

3.3 Implied volatility

Data on implied volatility for both indexes was sourced from Bloomberg. It is computed by inverting the classic Black-Scholes formula (Black & Scholes, 1973) and based on at-the-money call options with 30 days maturity. Missing values for implied volatility were filled with the median value of the training set. This was done to limit the risk of data leakage into validation and test sets of the LSTM model. Moreover, each value was unannualised by using Bloomberg's default factor of 260 days. Adding implied volatility to the feature set was motivated by the findings of Busch et al. (2007), who showed that implied volatility is an important factor for forecasting realised volatility in several asset classes. Similar forecasting abilities were also proved by Szakmary et al. (2003) and Christensen et al. (1998).

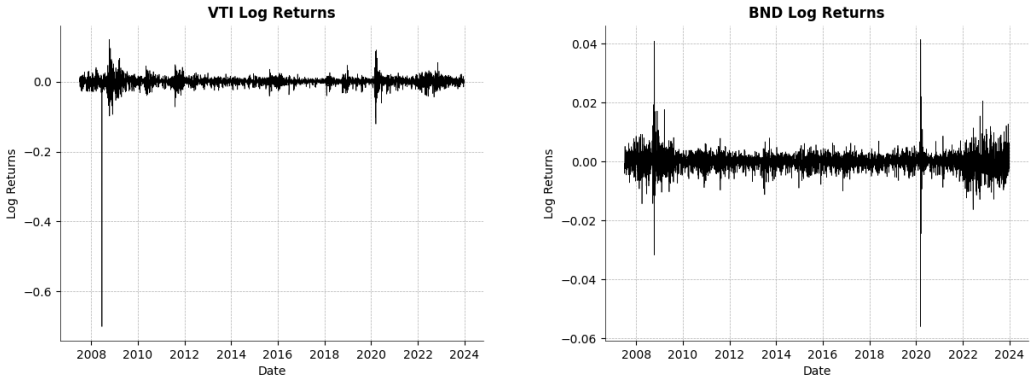
3.4 Training and testing sets

The full dataset ranges from July 2007 to December 2023 and consists of 4154 observations, each representing one trading day. The data was separated using the regular 80/20 splits, where roughly 80% of data is used for estimating parameters of a model and the remaining 20% is used for out-of-sample testing. The exact values were adjusted in such a way that the testing set is of size that is divisible by 20 (so that the number of expected Value-at-Risk violations at 5% confidence level is an integer). Finally, one extra observation was added to the testing set to maintain appropriate size after the first row is dropped, since no prediction will be generated for $t = 0$. This results in 3313 rows for training and 841 rows for testing (before $t = 0$ is dropped). Since the LSTM model requires an extra validation set, 841 rows were removed from the LSTM's training set to form a validation set. This results in 2472 rows for training, 841 rows for validation, and 841 rows for testing (before $t = 0$ is dropped). These values were chosen following the generally accepted ratio of 60/20/20 for training, validation, and testing, respectively.

3.5 Descriptive statistics

Figure 1

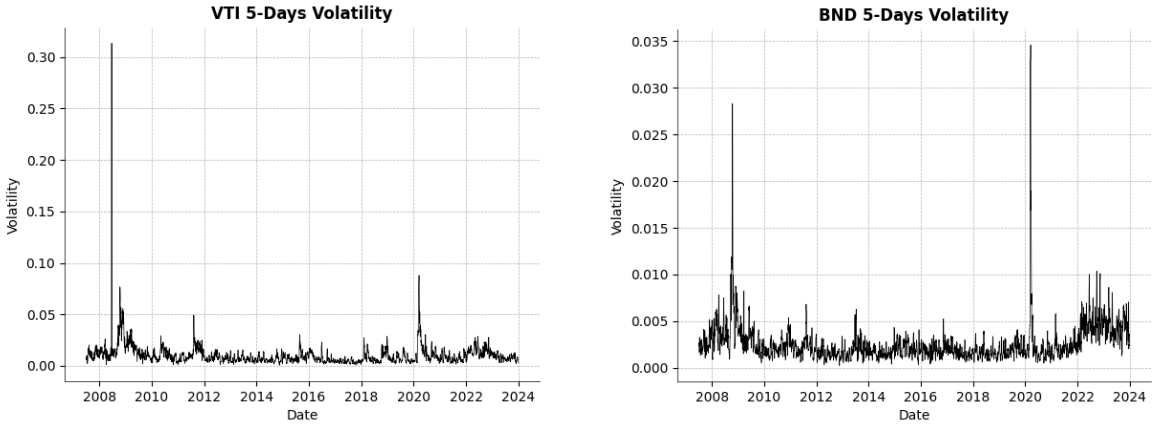
Daily log-returns of VTI and BND in the period July 2007 – December 2023



Note: Logarithmic returns of an asset i at time t are defined as a natural logarithm of the following quotient: price of an asset i at time t divided by price of an asset i at time t-1. VTI represents the Vanguard Total Stock Market Index and serves as a proxy for the general asset class of stocks, whereas BND represents the Vanguard Total Bond Market Index and serves as a proxy for the general asset class of bonds.

Figure 2

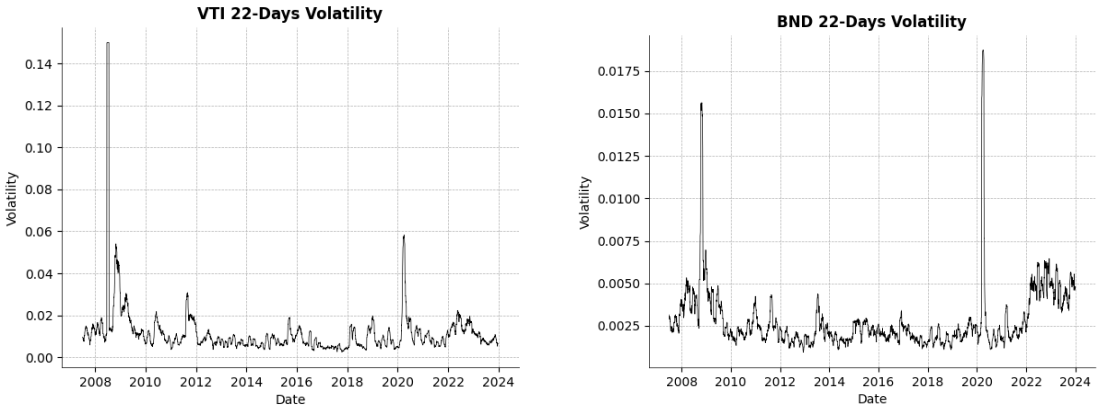
5-days rolling average volatility for VTI and BND in the period July 2007 – December 2023



Note: 5-days rolling average volatility at time t is defined by applying formula (2) on timesteps t,…, t-4. VTI represents the Vanguard Total Stock Market Index and serves as a proxy for the general asset class of stocks, whereas BND represents the Vanguard Total Bond Market Index and serves as a proxy for the general asset class of bonds.

Figure 3

22-days rolling average volatility for VTI and BND in the period July 2007 – December 2023



Note: 22-days rolling average volatility at time t is defined by applying formula (2) on timesteps t,…, t-21.. VTI represents the Vanguard Total Stock Market Index and serves as a proxy for the general asset class of stocks, whereas BND represents the Vanguard Total Bond Market Index and serves as a proxy for the general asset class of bonds.

Table 1

Descriptive statistics for VTI

	Log Returns	5-days Volatility	22-days Volatility	Implied Volatility
Mean	0.000111	0.010566	0.011477	1.104664
Standard Deviation	0.016920	0.013216	0.012436	0.520104
Minimum	-0.700527	0.000446	0.002565	0.246705
25%	-0.004496	0.005260	0.006359	0.755232
50%	0.000714	0.008199	0.008838	0.968060
75%	0.006177	0.012700	0.012752	1.317125
Maximum	0.120710	0.313407	0.149914	4.637659

Note: All statistics are based on a sample of 4154 observations, ranging from July 2007 to December 2023.

Table 2

Descriptive statistics for BND

	Log Returns	5-days Volatility	22-days Volatility	Implied Volatility
Mean	-0.000002	0.002637	0.002790	0.339073
Standard Deviation	0.003379	0.002110	0.001898	0.219884
Minimum	-0.055920	0.000247	0.000941	0.123415

25%	-0.001541	0.001541	0.001794	0.224999
50%	0.000123	0.002142	0.002229	0.265496
75%	0.001663	0.003133	0.003176	0.359081
Maximum	0.041335	0.034580	0.018730	2.851062

Note: All statistics are based on a sample of 4154 observations, ranging from July 2007 to December 2023.

CHAPTER 4 Method

4.1 Defining Value-at-Risk

Jorion (2001) defines the Value-at-Risk (VaR) measure as the worst expected loss during a given period and under a given confidence level. This intuitive and rather simple method has been an industry standard for all entities exposed to market risk. This paper will define Value-at-Risk using the following formula (3):

$$VaR_t = \sigma_t * \alpha$$

Following RiskMetrics (1996), the confidence level (α) will be derived from the conditionally normal distribution and approximated to 1.65. Moreover, the mean of returns that is often used in VaR calculations will be assumed to equal zero. This is in line with RiskMetrics (1996) guidelines and confirmed by Christoffersen (2012), who found that, when one-day horizons are in question, the mean of returns is non-distinguishable from zero and that the standard deviation is the main factor to consider.

4.2 Assessing performance of the models in estimating Value-at-Risk

VaR will be estimated based on volatility forecasts generated by each of the models. On each day of the testing set, the VaR estimate will be compared to observed returns for that day. If observed returns exceed those provided by VaR (i.e. they are more negative), a violation will be noted. The accuracy of the models will be assessed with a ratio of observed failures to expected failures. The number of expected failures is based on the confidence level of choice – a 95% confidence level means that exactly 5% of estimates should be marked as violations (resulting in a ratio of 1). A ratio smaller than 1 will indicate that a given model overestimates the risk, whereas a number larger than 1 will indicate that the model underestimates the risk. Both types of violations have their own negative consequences (such as unexpected losses for underestimation and uninvested capital for overestimation) and so an optimal model is not one that minimises the number of failures but one that provides a ratio closest to unity. To verify whether the deviation from the expected number of violations is statistically significant, a binomial test will be performed.

Let's define x as the observed number of violations, N as the size of the testing set, and p as 1 – confidence level. Then, Np represents the expected number of violations (equal to 42 in the case of 840 days in the testing set and a confidence level of 95%). Under the null hypothesis, the observed failure rate, defined as $\frac{x}{N}$, should converge to p. Rejection of the null would indicate that the observed failure rate $\frac{x}{N}$ is significantly different from the expected rate p. For sufficiently large samples, the Central Limit Theorem can be invoked to approximate the binomial distribution by the normal distribution, resulting in the following statistic (4):

$$Z = \frac{x - Np}{\sqrt{p(1-p)N}} \sim N(0,1)$$

This test, however, does not account for the magnitude of the loss that occurs in the case of a violation. This is a significant drawback, since a model that makes small mistakes but often, would be assessed as inferior to a model that makes huge mistakes but stays within the accepted range of violations. From a practical point of view, however, the former model would be considered more reliable and safer to use. This drawback of the binomial test was answered by Lopez (1999), who proposed a model based on a loss function. The loss function that is used in this study takes the following form (5):

$$L_1 = \begin{cases} 1 + (\varepsilon_{t+1} - VaR_{mt})^2 & if\ \varepsilon_{t+1} < VaR_{mt} \\ 0 & if\ \varepsilon_{t+1} \geq VaR_{mt} \end{cases}$$

The magnitude of a loss (i.e. the difference between actual returns and VaR estimate), if the VaR threshold is violated, now also enters the equation and provides additional information on the performance of a model. The function of the quadratic term, in this case, is to penalise larger deviations proportionally. It is worth, noting, however, that this loss function focuses only on the violations of VaR. To evaluate how accurately the VaR models estimate the losses, one of the loss functions of Caporin (2008) will also be used (6):

$$L_2 = |\varepsilon_{t+1} - VaR_{mt}|$$

This function is applied to each day of the testing set. The sum of the values will indicate how much a given model underestimates and overestimates the losses. This approach will provide a more detailed view of the performance of each model and answer some of the limitations of the other back-testing procedures.

4.3 Historical Average Method

The historical average method will serve as an example of a naive method, which simply relies on an average value of some arbitrary number of observations from the past. In this study, a forecast for t+1 will be the average volatility from the past 22 trading days (i.e. from t to t-21). Contrary to other methods, this method does not use any weighting of the prior observations, nor does it try to find reoccurring patterns in the data. Out-of-sample performance of this method will be evaluated using Mean Squared Error, whereas the accuracy of Value-at-Risk estimates will be evaluated using the method described above.

4.4 GARCH specification

For this study, the Generalized Autoregressive Conditional Heteroskedasticity (Bollerslev, 1986) model will be used as a benchmark. In its general form, this model consists of two equations. The first component

of a GARCH(p, q) model is the equation for the value of the time series at time t as a function of the mean of the series and an error term (7.1):

$$r_t = \mu + \epsilon_t$$

The second equation describes the variance of the time series and is the form of (7.2):

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-1}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2$$

In this case, conditional variance at time t is a function of the constant term, lags of the squared error term, and lags of the conditional variance. The number of lags for each of the two components is described by the p and q parameters of the model. The forecasting performance of the model will be evaluated using Mean Squared Error (i.e. the average squared distance between prediction and observed value at every timestep). Performance in the VaR estimation task will be evaluated using the method described above.

This study will test one of the simple specifications of the GARCH model, the GARCH(1,1) model with the mean assumed to be zero, following the argument of Christoffersen (2012), and the error term following the white noise process. The formulas of such a model are of the following form (8.1, 8.2, 8.3):

$$y_t = \epsilon_t$$
$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$
$$\epsilon_t | \mathcal{F}_{t-1} \sim N(0, \sigma_t^2)$$

Parameters of the model $(\alpha_0, \alpha_1, \beta_1)$ will be estimated on the training set. Then, the model with be tested on the testing set, producing a prediction for the next day on a rolling basis. The forecasting performance of the model will be evaluated using Mean Squared Error (i.e. the average squared distance between prediction and observed value at every timestep). Performance in the VaR estimation task will be evaluated using the methods described above.

4.5 SVR-GARCH specification

Secondly, an SVR-GARCH model will be employed. It is based on the GARCH(1,1) formulas for the mean and the variance but instead of applying the Maximum Likelihood to estimate the parameters, it uses a Support Vector Regression Model (SVR) – a supervised machine learning method for predicting continuous outcomes based on Support Vector Machines. The main advantage of such an approach is that SVR makes no assumptions about the distribution of the data (Bezerra & Albuquerque, 2017), making it more flexible

and, thus, more able to fit financial data, which often exhibits noisiness and nonlinearities (Cao & Tay, 2001).

The main mechanism behind SVR is based on mapping the input vector into a feature space of higher dimensionality and then running a linear regression on that feature space to get the final scalar output (Cortes & Vapnik, 1995). How the input vector is mapped into the higher-dimensional feature space is subject to the choice of the kernel function. This study will focus on two kernel functions – linear and radial basis functions. There are, however, several other kernels available that could achieve different results. The linear kernel computes the dot product of the input vectors and takes the following form (9.1):

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \boldsymbol{x_i} \cdot \boldsymbol{x_j}$$

Using this type of kernel, however, will make the model search for a linear relationship between the outputs and inputs, which may not be optimal for the case of volatility forecasting. That is why the radial basis function (RBF) kernel will be tested as well. RBF (or the Gaussian kernel) takes the following form (9.2):

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \exp\left(-\gamma \parallel \boldsymbol{x_i} - \boldsymbol{x_j} \parallel^2\right)$$

In this setting, the squared Euclidian distance between the input vectors is measured and adjusted by the gamma parameter, which dictates the weight of a single observation on the result. The choice of the RBF kernel was based on its wide usage in the literature (Bezerra & Albuquerque, 2017; Peng et al., 2018). The linear kernel was chosen to allow for a comparison of linear and non-linear specifications.

The performance of an SVR model is not only highly dependent on the choice of the kernel but also on its parameters. Apart from the gamma parameter, which is specific to the Gaussian kernel, the SVR specification also involves the regularisation parameter (C) and the epsilon parameter (ϵ). The former is used to balance the trade-off between the model's complexity and training error – large values of C will make the algorithm overfit and, thus, perform poorly out-of-sample (Bezerra & Albuquerque, 2017). The latter parameter controls the width of the epsilon-insensitive zone, which is the area around the fitted hyperplane. Points that fall outside of that zone are called support vectors and their distance from the boundaries of the epsilon-insensitive tube is represented by ξ_i and ξ_i^*, for positive and negative errors respectively (Smola & Schölkopf, 2004). These support vectors are the main drivers of SVR's parameters and shape the optimisation problem of the model through slack variables (ξ_i and ξ_i^*), whereas observations that fit within the epsilon-insensitive zone are ignored. The main optimisation problem of the model, then, takes the following form (10):

$$\min \{\frac{1}{2} \| \boldsymbol{w} \|^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i^*)\}$$

For detailed mathematical derivation of the SVR model and solution to equation (10) one can refer to Perez-Cruz et. al (2003) or Bezerra and Albuquerque (2017).

Finally, the SVR-GARCH model can be defined by estimating the GARCH(1,1) equations (6.1 and 6.2) with the SVR model. At time t, both the squared residuals from the mean and the proxy for the variance are obtained. These two features form the input vector of the following form (11):

$$[h_t \quad u_t^2]$$

The output scalar then becomes the one-step-ahead prediction for the variance (i.e. $\tilde{h}_{t+1}$). The training and testing data is normalised (so that each feature can be described by a zero mean and unit variance) using the following formula (12):

$$x_{scaled} = \frac{x - \mu}{\sigma}$$

To avoid data leakage to the testing set, the values of μ and σ are based solely on the training set. The Randomised Search method with 5-fold cross-validation was applied to find the best set of parameters. The training data is divided into 5 subsets and parameters chosen randomly during each iteration are tested on each of these subsets. Finally, rolling predictions are generated, rescaled, and aligned with observed values. Value-at-Risk estimates are computed based on the square root of the obtained predictions and compared to actual returns. The forecasting performance of the model will be evaluated using Mean Squared Error, whereas performance in the VaR estimation task will be evaluated using the binomial test and the loss function.

4.6 LSTM specification

The Long-Short-Term-Memory (LSTM) model was introduced by Hochreiter and Schmidhuber (1997) as a solution for the inability of other Recurrent Neural Networks (RNN) to capture long-term dependencies in the data. The long-term memory component comes from the memory cell, which obtains information from the input gate and outputs information through the output gate. These two additional gates are required to avoid the inflow and outflow of noise from the memory cell. Finally, the forget gate is used to decide, what stays in the memory and what should be deleted. Such an architecture is well suited for working with financial time-series data since it can effectively filter out irrelevant noise from the relevant signals and use the latter to learn long-term dependencies that the data exhibits. The LSTM model can take an almost

unlimited number of forms, due to the wide range of possible combinations of its hyperparameters. The LSTM specification in this study, however, will be based on the previous work of Liu et al. (2021), who tested a two-layer model in a volatility forecasting scenario, and then modified to fit the data of this study better.

Table 3

LSTM hyperparameters and their descriptions.

Hyperparameter	Description	Values
Units on a layer	Size of the vector that a layer outputs.	512, 512 (following Liu et al. (2021).
Activation function	A mechanism that determines if a neuron should be activated or not. Introduces non-linearity to the model.	tanH (following Liu et al. (2021).
Dropout rate	A regularisation mechanism that prevents overfitting. The dropout rate determines the fraction of units that are randomly set to zero in each training iteration.	Stocks: 0.6 Bonds: 0.7
Learning rate	The size of the step that the optimiser takes when looking for the optimal solution.	0.001 (following Liu et al. (2021).
Weight decay	A regularization technique that involves penalising large weights to prevent the model from being dependent on single units.	0.001 (following Liu et al. (2021).
Optimiser	A function that determines how the loss function is minimised.	ADAM (following Liu et al. (2021).
Number of epochs	The number of times the model sees the training set in its entirety during the training process.	10 (following Liu et al., (2021).
Batch size	The number of examples that are shown to the model at once during training.	16 (following Liu et al., (2021).

Note: If not specified otherwise, the same values of hyperparameters were used in models for the stock market and for the bond market. Values of hyperparameters were not re-tuned after the data on implied volatility and lags thereof were added to the models.

The final models are two-layer LSTM specifications with 512 units in the first and in the second layer. Dropout layers of 60% for bonds and 70% for stocks were added to the baseline specification to improve the out-of-sample performance, following Srivastava et al. (2014). The last layer of the models is a Dense layer, which produces scalar output.

The size of the training set decreased by around 800 rows. These rows formed a validation set, creating a 60/20/20 split for training, validating, and testing respectively. Each model specification is trained on a tensor of the following dimensions: (2472 ×11 ×3), representing the number of rows in the training set,

number of lags (lags from 1 to 10 and the current observation), and number of features (5-day volatility, 22-day volatility, daily squared returns). The training tensor will be normalised and scaled using the formula (12). Parameters of the scalers are evaluated on the training set only to avoid data leakage to other sets. Predictions generated on the test set are rescaled and aligned with their observed counterparts. Next, the models are trained on a training set extended with data on implied volatility, forming a tensor of the following size: (2472 ×11×4). The performance of the retrained models will be compared to previous specifications, potentially uncovering forecasting abilities of implied volatility, which is often viewed as a proxy for the sentiments of the market about future volatility. The forecasting performance of the models will be evaluated using Mean Squared Error, whereas performance in the VaR estimation task will be evaluated using the binomial test and the loss functions.

CHAPTER 5 Results & Discussion

5.1 Results

The models analysed in this study were tested and evaluated on two indexes representing two major asset classes – stocks (proxied with VTI) and bonds (proxied with BND). Their performance was evaluated on a testing set consisting of 840 observations. Two types of applications were tested and evaluated – the volatility forecasting task was evaluated using MSE and Value-at-Risk estimation was evaluated using a ratio of observed to expected failures and verified by the means of a binomial test.

Table 4

Out-of-sample performance evaluated using Mean Squared Error.

	Historical Average	GARCH(1,1)	SVR-GARCH (linear kernel)	SVR-GARCH (RBF kernel)	LSTM (without implied volatility)	LSTM (with implied volatility)
MSE (stocks)	1.312517	1.347470	0.807351	0.912018	0.842738	0.519421
MSE (bonds)	0.151171	0.098088	0.067440	0.157209	0.045414	0.052334

Note: The testing set contained consecutive 840 trading days. All reported values are in e^{-5}. First six digits of every score are reported.

The Mean Squared Error metric in the out-of-sample tests indicates that the SVR-GARCH specification can outperform the GARCH(1,1) model and the historical average method in predicting volatility. The two kernels performed similarly well on the stock market, but the linear kernel exhibited better performance on the bond market, under the chosen parameter values. The LSTM models performed similarly well to the SVR-GARCH specifications. Relative performance between the SVR-GARCH and the LSTM models can vary, depending on the choice of hyperparameters, but the overall tendency of these models to outperform the benchmark is noticeable. LSTM specification for the stock market improved its MSE score after the implied volatility feature and lags thereof were added to the training set (without retuning the hyperparameters), whereas that of the bond market model remained roughly the same.

Table 5

Results of the binomial test and loss function test for the VaR estimates on the stock market proxy.

	Historical Average	GARCH(1,1)	SVR-GARCH (linear kernel)	SVR-GARCH (RBF kernel)	LSTM (without implied volatility)	LSTM (with implied volatility)
Failure rate	1.21	0.81	0.86	0.90	74	47
Lopez Loss	51.003761	34.001960	36.002306	38.002436	74.004621	47.002893
Caporin Loss	15.892772	17.930120	17.487677	17.214953	16.223124	16.115606
$H_0: p = 0.05$	Not Rejected	Not Rejected	Not Rejected	Not Rejected	Rejected at 1%	Not Rejected

Note: Failure rate represents the ratio of observed failures to expected failures at 95% confidence level. Since the testing set contained 840 observations, the number of expected failures is 42. Second row of the table reports the values of the Lopez loss function. Third row reports the values for the Caporin Loss function. Fourth row of the table provides the p-values of the binomial test for the null hypothesis of the observed failure rate being sufficiently close to 1.

The evaluation of the models in the VaR estimation task on the stock market proxy shows that most of the models would be classified as reliable, based on the results of the binomial test. The only exception is the LSTM specification which did not use the implied volatility as one of the features. Looking at the Caporin (2008) loss function values, however, the VaR forecasts of the LSTM models followed the observed losses more accurately. This discrepancy is caused by the threshold nature of VaR back-tests that are based only on the number of violations. They do not account for the size of the violation, nor for the capital that is held unnecessarily when the VaR forecasts are too high.

Table 6

Results of the binomial test and loss function test for the VaR estimates on the bond market proxy.

	Historical Average	GARCH(1,1)	SVR-GARCH (linear kernel)	SVR-GARCH (RBF kernel)	LSTM (without implied volatility)	LSTM (with implied volatility)
Failure rate	1.26	1.24	1.50	1.07	1.64	1.60
Lopez Loss	53.000399	52.000399	63.000454	45.000326	69.000477	67.000429
Caporin Loss	5.161080	5.111020	5.216640	5.773789	5.052014	5.296770
$H_0: p = 0.05$	Rejected at 10%	Not Rejected	Rejected at 1%	Not rejected	Rejected at 1%	Rejected at 1%

Note: Failure rate represents the ratio of observed failures to expected failures at 95% confidence level. Since the testing set contained 840 observations, the number of expected failures is 42. Second row of the table reports the values of the Lopez loss function. Third row reports the values for the Caporin Loss function. Fourth row of the table provides the p-values of the binomial test for the null hypothesis of the observed failure rate being sufficiently close to 1.

In the case of the bond market proxy, the binomial test classified most of the models as not reliable – the null hypothesis was not rejected only for the GARCH(1,1) and SVR-GARCH with the RBF kernel. A closer look at the overall loss shows, however, that the LSTM specification (without implied volatility) minimises the Caporin (2008) function.

5.2 Discussion

The fact that more complex and newer models can perform better than parsimonious specifications is something to be expected. The question that remains, however, is whether these improvements are worth taking the risk that comes with employing a more complex model. Both SVR-GARCH and LSTM are highly dependent on the choice of hyperparameters – even slight changes in their values can significantly change the predictions, and, thus, the results. This issue becomes even more pronounced when one considers that, so far, a framework for robustly finding the optimal configuration has not been found. One of the solutions to this issue is to use ensembles of models – combining results of multiple specifications. This, however, adds another level of complexity to the forecasting procedure that may not be worth sustaining in practice. Moreover, the deep learning models are difficult to interpret. Due to the large number of parameters, one is usually not able to point out the drivers of a given forecasts or write-down the equation that led to the observed solution. For many institutions, especially those from the financial sector, this may not be acceptable. Whether one should choose a more complex specification or adhere to the classic methods is a question that should be answered on a case-by-case basis. What is clear, however, is that there are performance gains available for those who can accept these additional risks.

5.3 Limitations

The results of this study are valid only if the assumptions that they rely on are true. The generalisation to the entire stock and bond markets by means of a proxy may not reliably represent these markets. Moreover, practitioners usually build their portfolios around a selected number or type of positions. The results of this study may not extend to such cases - a reliable risk management strategy should be built directly for a specific portfolio. Moreover, several assumptions were made about the characteristics of the distribution of financial returns. Relaxing the assumptions of zero-mean returns could influence the performance of the models in both directions. However, as Table 1 and Table 2 show, in the analysed sample daily returns were, indeed, very close to zero. A stronger and less empirically justified assumption was made about the shape of the distribution of financial returns. Even though the standard normal distribution is often assumed in the literature, it has been empirically proven to fail to account for fat tails and asymmetry (Wang and Taaffe, 2015). Since the critical value for the Value-at-Risk estimates was based on the standard normal distribution, assuming other distributions would most definitely change the results of this study. Testing other classic distributions (such as the student-t) or empirical distribution functions (Moralles et al., 2013) is a subject for further research. Finally, the choice of the volatility proxy used in this study is also a subject for discussion. Using intraday data or more sophisticated estimators, such as that of Garman and Klass (1980) or Yang and Zhang (2000), could provide more accurate estimates and, thus, change the results.

The models analysed in this study could also be further tuned and modified to achieve better performance and accuracy. The simple GARCH(1,1) could be replaced by an extended specification with more lags of

both squared residuals and conditional variances and with different assumptions on the distribution of the residuals. The main drawback of the SVR-GARCH, on the other hand, is how the optimal parameters were found. The Randomised Search was chosen because of its computational efficiency but more accurate methods, such as that of Üstün et al. (2005), could provide more stable and, thus, more reliable results. Moreover, using more advanced kernels or combinations of kernels, as in Bezerra and Albuquerque (2016), could result in better performance. Finally, more extensive hyperparameter tuning and feature engineering would further improve the performance of the LSTM model. Simpler specifications, with fewer layers or fewer units per layer, are also likely to perform well in volatility forecasting tasks (Assaf et al., 2022). Adding variables that are known to be good predictors of volatility could exploit LSTM's ability to handle large datasets even further, as was done by Bucci (2020). Such specifications were not tested in this research, however.

To analyse the performance of the models in the VaR estimation task in an even more detailed manner, additional tests for independence of the violations could be used, such as that proposed by Christoffersen (1998). Similarly, to test if the exceptions are independent from other variables the Dynamic Quantile test could be applied, as proposed by Engle and Manganelli (2004). Moreover, verifying the forecasting performance of the models on different time horizons and different confidence levels would be required to generalize the findings of this paper.

CHAPTER 6 Conclusion

The main goal of this study was to implement and compare forecasting models of different types and test them on two different asset classes – the stock and the bond market (by means of a proxy in the form of ETFs). The historical average method, the GARCH(1,1), the SVR-GARCH with two types of kernel functions, and the LSTM models were tested in two scenarios – volatility forecasting and Value-at-Risk estimation. Results of this analysis have shown that the characteristics of the hybrid models (SVR-GARCH) and complex deep learning models (LSTM) – such as handling nonlinearities or distinguishing noise from relevant signals - can provide better performance than some of the classic approaches. The finding of superior forecasting ability of the SVR-GARCH and LSTM models, in comparison with GARCH-type models, is in line with prior research - Karasan and Gaygısız (2022) and Liu et al. (2022) report similar results. The additional complexity of the models comes with an additional cost, however. The performance of these models is very sensitive to the choice of hyperparameters, indicating that a robust tuning procedure is necessary to achieve optimal performance. Moreover, the high number of parameters that are estimated within most of the deep learning models significantly reduces the degrees of freedom, making it necessary to train the models on large samples to preserve generalisability.

The back-testing procedure of the Value-at-Risk estimates highlighted some of the limitations of this model. The binary evaluation of VaR turned out to be overly simplistic and rejected some of the LSTM and SVR-GARCH models, despite their superior MSE scores. Looking at the loss function, however, it was clear that superior performance in terms of MSE of the SVR-GARCH and LSTM specifications translates directly into more accurate VaR estimates. This discrepancy clearly shows that the choice of the VaR model strongly depends on the use case and priorities of each entity.

This study, despite certain limitations and assumptions, clearly showed that the path of machine learning and deep learning models in the search for reliable volatility and risk models is a promising one. Further work on aspects inherent to these models, that also make them difficult to implement in a robust way, such as hyperparameter tuning and feature engineering could reveal even better performance of these models and, thus, equip entities that are exposed to market risk and fluctuations with better tools to manage these challenges.

REFERENCES

Abad, P., Benito, S., & López, C. (2014). A comprehensive review of Value at Risk methodologies. *The Spanish Review of Financial Economics*, *12*(1), 15–32. https://doi.org/10.1016/j.srfe.2013.06.001

Assaf, O., Di Fatta, G., & Nicosia, G. (2022). Multivariate LSTM for Stock market Volatility Prediction. In *Lecture notes in computer science* (pp. 531–544). https://doi.org/10.1007/978-3-030-95470-3_40

Bams, D., Blanchard, G., & Lehnert, T. (2017). Volatility measures and Value-at-Risk. *International Journal of Forecasting*, *33*(4), 848–863. https://doi.org/10.1016/j.ijforecast.2017.04.004

Bezerra, P. C. S., & Albuquerque, P. H. M. (2016). Volatility forecasting via SVR–GARCH with mixture of Gaussian kernels. *Computational Management Science*, *14*(2), 179–196. https://doi.org/10.1007/s10287-016-0267-0

Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, *81*(3), 637–654. https://doi.org/10.1086/260062

Blair, B. J., Poon, S., & Taylor, S. J. (2001). Forecasting S&P 100 volatility: the incremental information content of implied volatilities and high-frequency index returns. *Journal of Econometrics*, *105*(1), 5–26. https://doi.org/10.1016/s0304-4076(01)00068-9

Bloomberg L.P. (2024). Implied volatility data for VTI and BND. Bloomberg Terminal.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*(3), 307–327. https://doi.org/10.1016/0304-4076(86)90063-1

Bucci, A. (2020). Realized Volatility Forecasting with Neural Networks. *Journal of Financial Econometrics*, *18*(3), 502–531. https://doi.org/10.1093/jjfinec/nbaa008

Busch, T., Christensen, B. J., & Nielsen, M. Ø. (2007). The role of implied volatility in forecasting future realized volatility and jumps in foreign exchange, stock, and bond markets. *Social Science Research Network*. https://doi.org/10.2139/ssrn.1148738

Byun, S. J., & Cho, H. (2013). Forecasting carbon futures volatility using GARCH models with energy volatilities. *Energy Economics*, *40*, 207–221. https://doi.org/10.1016/j.eneco.2013.06.017

Center for Research in Security Prices. (2023). CRSP US Stock Database. The University of Chicago Booth School of Business. https://www.crsp.org/products/research-products/crsp-stock-databases

Cao, L., & Tay, F. E. (2001). Financial forecasting using support vector machines. *Neural Computing & Applications*, *10*(2), 184–192.

Caporin, M. (2008). Evaluating value-at-risk measures in the presence of long memory conditional volatility. *The Journal of Risk*, *10*(3), 79–110. https://doi.org/10.21314/jor.2008.172

Chen, S., Jeong, K., & Härdle, W. K. (2008). Support Vector Regression Based GARCH Model with Application to Forecasting Volatility of Financial Returns. *Social Science Research Network*. https://doi.org/10.2139/ssrn.2894286

Christensen, B., & Prabhala, N. (1998). The relation between implied and realized volatility. *Journal of Financial Economics*, *50*(2), 125–150. https://doi.org/10.1016/s0304-405x(98)00034-8

Christoffersen, P., & Pelletier, D. (2003). Backtesting Value-at-Risk: A Duration-Based approach. *Social Science Research Network*. https://doi.org/10.2139/ssrn.418762

Christoffersen, P. F. (2012). *Elements of financial risk management.* Academic Press.

Corsi, F. (2008). A simple approximate Long-Memory model of realized volatility. *Journal of Financial Econometrics*, *7*(2), 174–196. https://doi.org/10.1093/jjfinec/nbp001

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/bf00994018

De Santis, G., & İmrohoroğlu, S. (1997). Stock returns and volatility in emerging financial markets. *Journal of International Money and Finance*, *16*(4), 561–579. https://doi.org/10.1016/s0261-5606(97)00020-x

Engle, R. F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, *50*(4), 987. https://doi.org/10.2307/1912773

Engle, R. F., & Manganelli, S. (2004). CAViaR. *Journal of Business & Economic Statistics*, *22*(4), 367–381. https://doi.org/10.1198/073500104000000370

Engle, R. F., & Patton, A. J. (2007). What good is a volatility model? In *Elsevier eBooks* (pp. 47–63). https://doi.org/10.1016/b978-075066942-9.50004-2

Figlewski, S. (1997). Forecasting volatility. *Financial Markets, Institutions & Instruments*, *6*(1), 1–88. https://doi.org/10.1111/1468-0416.00009

Garman, M. B., & Klass, M. J. (1980). On the Estimation of Security Price Volatilities from Historical Data. *The Journal of Business*, *53*(1), 67. https://doi.org/10.1086/296072

Giot, P. (2003). The information content of implied volatility in agricultural commodity markets. *Journal of Futures Markets*, *23*(5), 441–454. https://doi.org/10.1002/fut.10069

Glosten, L. R., Jagannathan, R., & Runkle, D. E. (1993). On the Relation between the Expected Value and the Volatility of the Nominal Excess Return on Stocks. *The Journal of Finance*, *48*(5), 1779–1801. https://doi.org/10.1111/j.1540-6261.1993.tb05128.x

Harvey, C. R., & Whaley, R. E. (1992). Market volatility prediction and the efficiency of the S & P 100 index option market. *Journal of Financial Economics*, *31*(1), 43–73. https://doi.org/10.1016/0304-405x(92)90011-l

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Jorion, P. (2000). *Value at risk: the new benchmark for managing financial risk*. https://openlibrary.org/books/OL7298414M/Value_at_Risk

J.P. Morgan/Reuters (1996). *RiskMetrics – Technical document.* J.P. Morgan.

Karasan, A., & Gaygısız, E. (2020). Volatility Prediction and Risk Management: an SVR-GARCH. *Social Science Research Network*. https://doi.org/10.2139/ssrn.4285524

Lee, U. (1992). Do stock prices follow random walk?: *International Review of Economics & Finance*, *1*(4), 315–327. https://doi.org/10.1016/1059-0560(92)90020-d

Linsmeier, T. J., & Pearson, N. D. (1996). Risk measurement: an introduction to value at risk. *Fainansu*. https://doi.org/10.22004/ag.econ.14796

Liu, M., Taylor, J. W., & Choo, W. (2020). Further empirical evidence on the forecasting of volatility with smooth transition exponential smoothing. *Economic Modelling*, *93*, 651–659. https://doi.org/10.1016/j.econmod.2020.02.021

Liu, Y. (2019). Novel volatility forecasting using deep learning–Long Short Term Memory Recurrent Neural Networks. *Expert Systems With Applications*, *132*, 99–109. https://doi.org/10.1016/j.eswa.2019.04.038

Lopez, J. A. (1999). Methods for evaluating value-at-risk estimates. *Econometric Reviews*, 3–17. https://EconPapers.repec.org/RePEc:fip:fedfer:y:1999:p:3-17:n:2

Lux, M., Härdle, W. K., & Lessmann, S. (2017). Data driven Value-at-Risk forecasting using a SVR-GARCH-KDE hybrid. *Social Science Research Network*. https://doi.org/10.2139/ssrn.3176951

Mayhew, S., & Stivers, C. (2003). Stock return dynamics, option volume, and the information content of implied volatility. *Journal of Futures Markets/□the □Journal of Futures Markets*, *23*(7), 615–646. https://doi.org/10.1002/fut.10084

McAleer, M., & Medeiros, M. C. (2008). A multiple regime smooth transition Heterogeneous Autoregressive model for long memory and asymmetries. *Journal of Econometrics*, *147*(1), 104–119. https://doi.org/10.1016/j.jeconom.2008.09.032

Miranda, F. G., & Burgess, N. (1997). Modelling market volatilities: the neural network perspective. *European Journal of Finance*, *3*(2), 137–157. https://doi.org/10.1080/135184797337499

Moralles, H. F., Rebelatto, D. a. D. N., & Sartoris, A. (2013). Parametric VaR with goodness-of-fit tests based on EDF statistics for extreme returns. *Mathematical and Computer Modelling*, *58*(9–10), 1648–1658. https://doi.org/10.1016/j.mcm.2013.07.002

Peng, Y., Albuquerque, P. H. M., De Sá, J. M. C., Padula, A. J. A., & Montenegro, M. R. (2018). The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with Support Vector Regression. *Expert Systems With Applications*, *97*, 177–192. https://doi.org/10.1016/j.eswa.2017.12.004

Pérez-Cruz, F., Afonso-Rodríguez, J. A., & Giner, J. (2003). Estimating GARCH models using support vector machines. *Quantitative Finance*, *3*(3), 163–172. https://doi.org/10.1088/1469-7688/3/3/302

Prakash, A., James, N., Menzies, M., & Francis, G. (2021). Structural clustering of volatility regimes for dynamic trading strategies. *Applied Mathematical Finance*, *28*(3), 236–274. https://doi.org/10.1080/1350486x.2021.2007146

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199–222. https://doi.org/10.1023/b:stco.0000035301.49549.88

Solnik, B., Boucrelle, C., & Fur, Y. L. (1996). International market correlation and volatility. *Financial Analysts Journal*, *52*(5), 17–34. https://doi.org/10.2469/faj.v52.n5.2021

Solnik, B. H. (1973). Note On The Validity Of The Random Walk For European Stock Prices. *The Journal of Finance*, *28*(5), 1151–1159. https://doi.org/10.1111/j.1540-6261.1973.tb01447.x

Song, Y., Lei, B., Tang, X., & Li, C. (2023). Volatility forecasting for stock market index based on complex network and hybrid deep learning model. *Journal of Forecasting*, *43*(3), 544–566. https://doi.org/10.1002/for.3049

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958. https://jmlr.csail.mit.edu/papers/volume15/srivastava14a/srivastava14a.pdf

Szakmary, A., Ors, E., Kim, J. K., & Davidson, W. N. (2003). The predictive power of implied volatility: Evidence from 35 futures markets. *Journal of Banking & Finance*, *27*(11), 2151–2175. https://doi.org/10.1016/s0378-4266(02)00323-0

Taylor, J. W. (2004). Volatility forecasting with smooth transition exponential smoothing. *International Journal of Forecasting*, *20*(2), 273–286. https://doi.org/10.1016/j.ijforecast.2003.09.010

Teller, A., Pigorsch, U., & Pigorsch, C. (2022). Short- to Long-Term Realized Volatility Forecasting using Extreme Gradient Boosting. *Social Science Research Network*. https://doi.org/10.2139/ssrn.4267541

Üstün, B., Melssen, W., Oudenhuijzen, M., & Buydens, L. (2005). Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization. *Analytica Chimica Acta*, *544*(1–2), 292–305. https://doi.org/10.1016/j.aca.2004.12.024

Vanguard. (2024, April 30). *BND – Vanguard Total Bond Market ETF*. Vanguard. https://investor.vanguard.com/investment-products/etfs/profile/bnd.

Vanguard. (2024, April 30). *VTI – Vanguard Total Stock Market ETF*. Vanguard. https://investor.vanguard.com/investment-products/etfs/profile/vti.

Vrontos, S. D., Galakis, J., & Vrontos, I. D. (2021). Implied volatility directional forecasting: a machine learning approach. *Quantitative Finance*, *21*(10), 1687–1706. https://doi.org/10.1080/14697688.2021.1905869

Wang, J., & Taaffe, M. R. (2015). Multivariate mixtures of normal distributions: properties, random vector generation, fitting, and as models of market daily changes. *INFORMS Journal on Computing*, *27*(2), 193–203. https://doi.org/10.1287/ijoc.2014.0616

Yang, D., & Zhang, Q. (2000). Drift independent volatility estimation based on high, low, open, and close prices. *The Journal of Business*, *73*(3), 477–492. https://doi.org/10.1086/209650

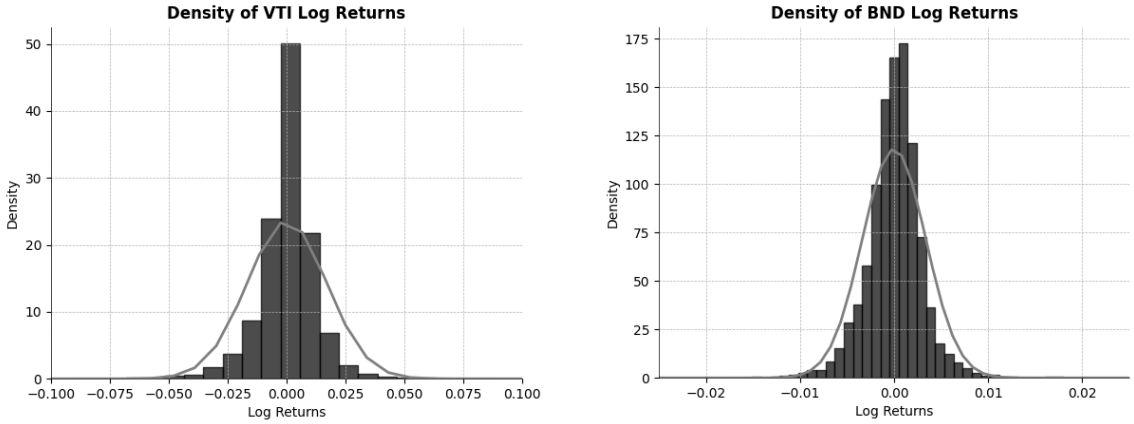
Zhang, C., Zhang, Y., Cucuringu, M., & Qian, Z. (2023). Volatility Forecasting with Machine Learning and Intraday Commonality. *Journal of Financial Econometrics*, *22*(2), 492–530. https://doi.org/10.1093/jjfinec/nbad005

Zhu, H., Bai, L., He, L., & Liu, Z. (2023). Forecasting realized volatility with machine learning: Panel data perspective. *Journal of Empirical Finance*, *73*, 251–271. https://doi.org/10.1016/j.jempfin.2023.07.003

APPENDIX A Distribution of returns

Figure 4

Distribution of returns for VTI and BND in the period July 2007 – December 2023.



Note: The graphs above represent the distributions of returns in the analysed sample, for the training and testing set. The x-axis was adjusted to the magnitude of returns of each asset class.