Lisbon School of Economics & Management
Universidade de Lisboa

MASTER
Finance

MASTER´S FINAL WORK
DISSERTATION

VOLATILITY FORECASTING WITH MACHINE LEARNING AND APPLICATION TO VOLATILITY SWAPS

MIGUEL MONTEIRO FERREIRA DE CARVALHO DINIS

JUNE – 2025

Lisbon School of Economics & Management
Universidade de Lisboa

MASTER
FINANCE

MASTER´S FINAL WORK
DISSERTATION

VOLATILITY FORECASTING WITH MACHINE LEARNING AND APPLICATION TO VOLATILITY SWAPS

MIGUEL MONTEIRO FERREIRA DE CARVALHO DINIS

SUPERVISION:
JOÃO AFONSO BASTOS
RAQUEL M. GASPAR

JUNE – 2025

ABSTRACT

This dissertation provides novel insights on the use of Machine Learning models, particularly the XGBoost algorithm, for realized volatility forecasting and its employment in investment strategies. The study focuses on the SP500 Index, from 1990 to 2024. A trading strategy using volatility swaps is simulated from 2018 to 2024 using the forecast results.

We perform an analysis to test which combination of machine learning model and inputs perform better at the task at hand. The models studied are the XGBoost, Random Forests and Neural Networks, and the variable inputs the Intrinsic Mode Functions (IMFs) resulting from CEEMDAN decomposition, and market variables from the United States. The XGboost algorithm with both the IMFs and market variables is the best performing ML model out of the study made, while also outperforming two different naïve models used as benchmarks.

We demonstrate the applicability of the forecasts through simulating a volatility swap investment that under realistic circumstances and various test scenarios is able to produce encouraging payouts, although with extremely high payoff volatility.

KEYWORDS: Volatility Forecasting; XGBoost; Random Forests; Neural Networks; Volatility Swaps.

JEL CODES: C63, G15, G17, G11.

RESUMO

Esta dissertação oferece novos contributos obre o uso de modelos de Machine Learning, especificamente o algoritmo XGBoost, para previsão de volatilidade realizada e sua aplicação em estratégias de investimento. O estudo concentra-se no índice SP500, de 1990 a 2024, e simula uma estratégia de investimento com swaps de volatilidade entre 2018 e 2024 com base nas previsões obtidas.

Realiza-se uma análise para testar qual combinação de modelo de Machine Learning e variáveis apresenta melhor desempenho na tarefa. Os modelos avaliados incluem XGBoost, Florestas Aleatórias (Random Forests) e Redes Neuronais (Artificial Neural Networks), enquanto as variáveis testadas são as Funções Modais Intrínsecas (IMFs - Intrissic Mode Functions) resultantes da decomposição CEEMDAN e variáveis de mercado dos Estados Unidos. O algoritmo XGBoost, combinando IMFs e variáveis de mercado, destaca-se como o modelo de ML mais eficiente no estudo, tendo também melhor performance que dois "naïve models" diferentes usados como referência.

A aplicabilidade das previsões é testada por meio da simulação de um investimento em swaps de volatilidade, que, com suposições realistas e diversos cenários, demonstra resultados encorajadores em termos de retornos, no entanto apresentado uma volatilidade extremamente alta.

PALAVRAS-CHAVE: Previsão de Volatilidade; XGBoost; Florestas Aleatórias; Redes Neurais; Swaps de Volatilidade.

CÓDIGOS JEL: C63, G15, G17, G11.

GLOSSARY

AN – Annualized Return.

ANN – Artificial Neural Networks.

ATM – At-the-money.

BSM – Black-Scholes Model.

CART - Classification and Regression Trees.

CEEMDAN - Complete Ensemble Empirical Mode Decomposition with Adaptive Noise.

EMD - Empirical Mode Decomposition.

EEMD - Ensemble Empirical Mode Decomposition.

IMF – Intrinsic Mode Function.

IV – Implied Volatility.

MAE – Mean Absolute Error.

MAPE – Mean Absolute Percentage Error.

ML – Machine Learning.

MLP - Multi-Layer Perceptron.

MSE – Mean Squared Error.

RF – Random Forest.

RMSE – Root Mean Squared Error.

RV – Realized Volatility.

SP500 - Standard & Poor's 500 Index.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

ACKNOWLEDGEMENTS

Writing a thesis while also starting a new job position was never going to be easy and I am only writing these words now, at the culmination of 6 months of hard work, thanks to the people present in my life that give me the strength and motivation to move ahead.

I would first like to thank my parents, Luísa and Paulo, for all their support in all shapes and sizes and through all my life. From taking me to swimming practice at 6am when I was younger, through all the experiences you've allowed me to have, the lunch boxes you've prepared for me, to allowing me to take the master that I am now about to finish, but mainly your constant presence and care and availableness, your touch is a constant in my life and it gives me the stability that I so much needed during these past months.

To my brother and sister, Vasco and Mimi, your youth and energy gave me some very needed laughs and moments where there was no thesis nor job to be done.

To my girlfriend, Beatriz, you are the light in a world that sometimes seems a bit too dim. You keep me grounded while lifting me higher than anyone could ever do and I seriously doubt any acknowledgments would have been written if I did not have you in my life.

To my friends, thank you for your company and for giving me some very needed times of being irresponsible and ignoring all that I needed to do.

To my team in my workplace, thank you for always being so understanding of my situation and for allowing me to balance studies and work so well.

Lastly, to my supervisors, Professor Raquel and Professor João, thank you for your guidance, your very needed corrections and observations. Your ideas, knowledge and recommendations were very much needed to make the path of this dissertation straighter and clearer. And I promise I will try to use less passive sentences during my life.

1. INTRODUCTION

Financial markets moved more than a quadrillion dollars in 2024 (London Stock Exchange, 2024), derivatives and structured products included. A derivative is a financial product whose value depends on the value of an underlying security (Bouzoubaa & Osseiran, 2010) such as a stock or a bond for example. To price them and even more complicated products such as structured products, which combine two or more derivatives, there are multiple variables that one needs to consider. The risk-free rate and the spot price of the underlying are common inputs in most pricing models. Volatility of the underlying is another variable indispensable in pricing a structured product. But there is a big difference between these three variables. At any given time, the risk-free rate across different maturities and markets can be closely inferred from government bonds and instruments alike, and the spot price of the underlying is usually known, as well as the other inputs like time to maturity and the strike price. But the volatility of the underlying asset across the maturity of the product is an incognita until we reach the maturity date of the product. As such, a key element to price structured products is impossible to know at the moment of pricing, giving great importance to the study of the best and most efficient ways to find proxies to this unknowable input.

There are different ways to measure volatility, but it is usually presented as an annualized standard deviation of log returns measured on the closing prices of the underlying. It is also of extreme importance to mention the two different volatilities that are used in option pricing theory and that will appear in this dissertation. Realized volatility (RV) is the observed volatility over a certain period of time and is the variable which this study intends to forecast. Implied volatility (IV), on the other hand, is the volatility implied by the price of vanilla options being traded in the market. That is, given a certain price for an option, there is only one volatility according to the Black Scholes Model (BSM) (Black and Scholes, 1973) that corresponds to that price. The BSM result for calls is shown in Equation 1.

$$C_t = N(d_1)S_t + N(d_2)Ke^{-rt}\,, \tag{1}$$

where:

$$d_1 = \frac{\ln\left(\frac{S_t}{k}\right) + \left(r + \frac{\sigma^2}{2}\right)t}{\sigma\sqrt{t}}\,, \tag{2}$$

$$d_2 = d_1 - \sigma\sqrt{t}\,, \tag{3}$$

and:

- C_t is the price of the call at time t
- S_t is the price of the underlying at time t
- K is the strike of the option
- r is the theoretical risk-free rate
- σ is the volatility of the underlying
- N is the Normal Distribution

Since vanilla option prices for liquid assets are determined by the offer and demand dynamics in the market, the BSM can be used to extract Implied Volatilities (IVs) using reserve engineering from those prices. It is commonly said that for a vanilla option, its value can be either expressed as its price or by the volatility implied by that price. These IVs are then used to price more complex financial instruments. IVs can then be understood as forward guess of the market for future realized volatility valid up to the maturity of the product.

As a forward guess of the market, IVs do not perfectly represent reality and the future RVs. As such, there is still the paradox of needing an unknowable variable to get the correct prices for non-liquid products. To deal with the volatility paradox, multiple approaches have been used over the years and across multiple models. The most common has been the use of the IV, even with its shortcomings. Despite its empirical success, this approach exhibits limitations beyond the Black-Scholes-Merton framework. A critical constraint arises from the absence of liquid options markets for certain underlying assets, which impedes reliable implied volatility derivation.

This parallel between volatilities and option prices, as well as the importance of volatility to the pricing of highly traded products makes it so that having an upper hand in forecasting realized volatility can help to provide the market with an alternative. In addition, one could try to "beat" the market by entering in long positions when forecasted realized volatilities are above the implied volatility and short positions when the opposite happens. Accurate volatility forecasting also improves pricing accuracy, which in turn improves market efficiency.

As such, techniques to forecast the RVs have been employed over the years. From using Implied Volatilities, to the first statistical models such as ARCH (Engle, 1982) and GARCH (Bollerslev, 1986) and then to Machine Learning models, a plenitude of approaches have been used to forecast volatility. In this work, we propose studying the

performance of three widely used Machine Learning models (Random Forests, XGBoost and Neural Networks) in combination with a signal decomposition method, CEEMDAN (Torres et al., 2011) which is employed for volatility forecasting by Zhu & Zhong (2024).

The objective of this dissertation is not only to expand on the work by Zhu & Zhong (2024), by introducing new variables and changing the model structure, but to also test the model forecasts in a trading scenario.

2. Literature Review

This section is divided into three subsections to give a clear picture of the developments made in the field of volatility forecasting, from the early pre-machine learning models to the more state-of-the-art algorithms which have become prevalent with the increase in computer power.

2.1.Early Volatility Forecasting Methods

Engle (1982) introduced the Autoregressive Conditional Heteroskedasticity (ARCH) models that capture time-varying volatility and volatility clustering. However, they can be overly responsive to outliers and may require many parameters for longer lag structures. Bollerslev (1986) developed the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models which improve on ARCH by incorporating both past volatility and past variance forecasts. They are more parsimonious and better capture long-memory effects in volatility. However, they assume symmetric responses to positive and negative shocks while it has been shown that negative shocks have more impact than positive ones (Bouzoubaa & Osseiran, 2010).

In the 90s, asymmetric GARCH models such as EGARCH (Nelson, 1991), GJR-GARCH (Glosten, Jagannathan & Runkle, 1993) were introduced to deal with the afore-mentioned problem but were at times too complex to estimate. In the early 2000s the first RV forecasting models using high-frequency intraday data were used but the data are usually not widely available to the public.

Proposed by Corsi in 2009, the Heterogeneous Autoregressive model of Realized Volatility (HAR-RV) model captures long-memory properties and heterogeneous

market participants. It is simple to estimate and provides good forecasting performance but may not fully capture extreme market events (Corsi, 2009).

2.2.Machine Learning in Volatility Forecasting

In recent years, due to the rise in processing power and the scientific community effort in bringing machine learning (ML) into a plenitude of areas of study, ML models have been applied to RV forecasting with good results. These models can capture complex, non-linear patterns in the data. However, they often require large amounts of data and can be less interpretable than traditional statistical models.

There have been multiple studies on the viability of using ML models to forecast volatility with favorable results in out-of-sample accuracy when compared to traditional linear models (see Rosa, Maciel, Gomide & Ballini, 2014, Miura , Pichl & Kaizoji, 2019, Zhu et al., 2023, Zhang et al., 2024).

In this study, Random Forests (RF) (Breiman, 2001), XGBoost (Chen & Guestrin, 2016) and Artificial Neural Networks (ANN) (see McCulloch and Pitts, 1943 and Rosenblatt, 1958) are used, all of these models being of simple implementation and widely used in financial markets problems. RFs have been employed in both option pricing (Lin et al., 2021) and for volatility forecasting (Zhu et al., 2023). Its simple and intuitive structure, as well as its resistance to overfitting, are commonly cited as the strengths of these decision-based algorithms. XGBoost is an extension of the Random Forest framework, with improvements to decrease variance and bias of the results. ANNs are the most commonly used ML model for volatility forecasting due to their non-parametric structure, not needing to assume any underlying function to the data (see Bucci, 2020, Christensen et al., 2023, Zhang et al., 2024, Zhu et al., 2023).

2.3.CEEMDAN Decomposition

Forecasting financial time series such as realized volatility presents unique challenges due to their inherently non-linear, non-stationary, and noisy nature. To improve the predictive accuracy of machine learning models in such conditions, it is beneficial to preprocess the data using signal decomposition methods. One such method is the Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN), as first proposed by Torres et al. (2011) and later employed in volatility

forecasting by Zhu & Zhong (2024), a powerful extension of the original Empirical Mode Decomposition (EMD) technique (Huang et al., 1998).

3. Data Description

3.1.First Database

This study focuses on the US market, specifically on the Standard & Poor's 500 Index (BBG ticker: SPX, referred as SP500 moving forward in this paper). As such, the data used are US and SP500 related metrics.

The data has daily observations from January 2nd 1990 until December 31st 2024 across several variables. Those variables can be split into two categories: SP500 descriptive variables and US market variables.

The first group is composed of the SP500 closing value and of the VIX (CBOE Volatility Index) value. It is worth defining this VIX Index. As per the official CBOE website:

> The VIX Index is a calculation designed to produce a measure of constant, 30-day expected volatility of the U.S. stock market, derived from real-time, mid-quote prices of S&P 500 Index (SPX) call and put options. On a global basis, it is one of the most recognized measures of volatility -- widely reported by financial media and closely followed by a variety of market participants as a daily market indicator.
>
> In: CBOE

The second group of variables consists of the daily values of the following variables: 10 years US Bonds Yield, 3 months US T-Bill Yield, Gold Futures prices (BBG ticker: GCM5), Brent Oil Futures prices (BBG ticker: LCOM5), USD/EUR exchange rate. In Table I we can see descriptive statistics of the variables, while in Figure 1 we have a visualization of the SP500 Closing Values and the VIX Index over time. In this graph it is easy to identify some well-known periods of high volatility which correspond to global crisis (2008 subprime crisis, 2020 Covid Crash, start of the war in Ukraine in 2022).

TABLE I

DESCRIPTIVE STATISTICS OF ORIGINAL VARIABLES

	SP500 Close ($)	VIX	10 Years Bond Yield	3-Month T-Bill Yield	Gold Future Prices ($)	Brent Oil Future Prices (%)	USD/EUR Exchange Rate
Count	9,131	9131	9131	9131	9131	9131	9131
Mean	1,695.28	19.46%	4.25%	2.67%	913.48	53.10	0.85
Standard Deviation	1,252.01	7.82%	1.97%	2.23%	612.71	32.73	0.11
Min	295.50	9.1%	0.52%	-0.05%	253.90	9.64	0.63
25%	909.90	13.8%	2.57%	0.20%	365.60	20.81	0.77
Median	1284.40	17.6%	4.15%	2.37%	679.70	49.61	0.84
75%	2101.00	22.8%	5.76%	4.90%	1333.40	76.81	0.91
Max	6090.27	82.7%	9.09%	7.99%	2800.80	146.08	1.21

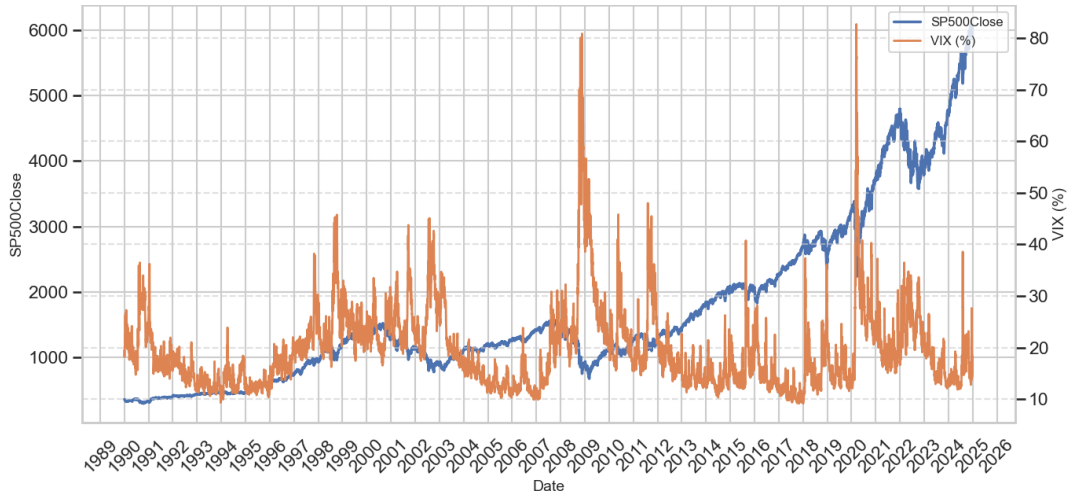

FIGURE 1 - SP500 Close ($) and VIX plotted over time

3.2.Final Database

To improve the variables and to better characterize it in preparation for it being used to train the ML models, several new variables are computed.

We compute the daily log returns of the SP500 and of the market variables, and calculate the standard deviation of those log returns for the SP500 over different time frames. This standard deviation is in fact the RV used in this work. For consistency

with past works and field practices, the RVs are annualized. The 7-day (5 trading days), 20-day (15 trading days) and 30-day (22 trading days) RVs are added to the database. The behavior of volatility over different timespans is visualized in Figure 2, where we can see the higher volatility of the 7-day RV compared to the 30-day.

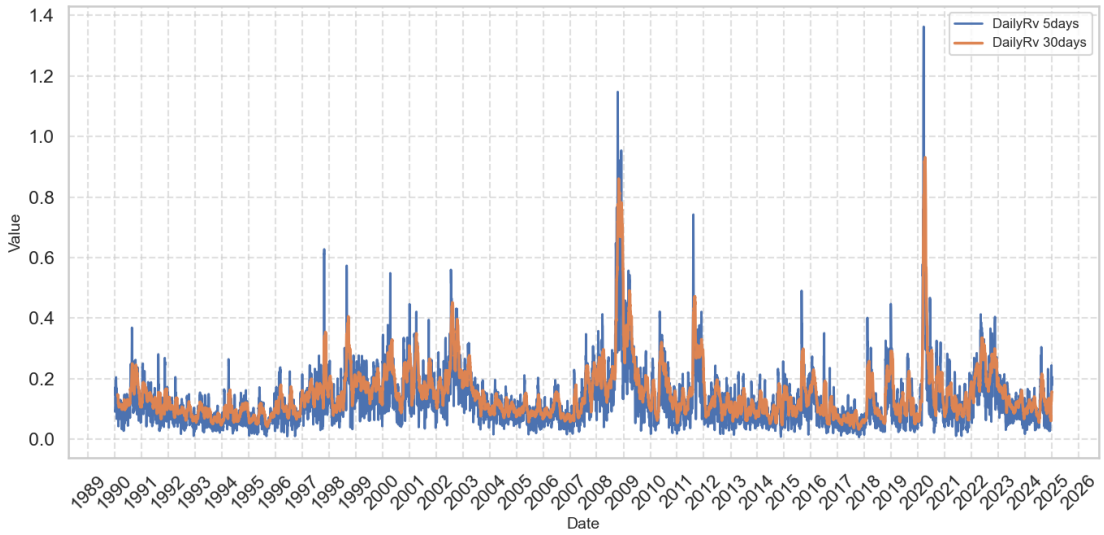

FIGURE 2 - 7 and 30-day RVs plotted over time

To better understand the behavior of RV and to serve as inputs to the ML models ahead, the Sharpe Ratio of the SP500, as well as the Negative Realized Semi Variance, Realized Semi Jump (as used by Zhu et al. (2023) with positive results in improving forecast accuracy) and lagged variables are computed. The Sharpe Ratio, defined by William F. Sharpe in 1966, is a widely used metric for assessing the risk-adjusted return of an investment. It is defined as the difference between the portfolio's return and the risk-free rate, divided by the standard deviation of the portfolio's excess return:

$$Sharpe\ Ratio = \frac{\bar{R}_p - R_f}{\sigma_p}, \tag{4}$$

where $\bar{R}_p$ is the expected return of the portfolio (in our study the daily log return of SP500), R_f the risk free rate (in our study the 3-Month T-Bill yield converted from yearly to a daily yield) and σ_p the volatility of the portfolio (in our study the 7-day past realized volatility). The Negative Realized Semi Variance is a measure of downside risk, measuring the variance of negative returns of the SP500 as:

$$RS^- = \frac{1}{N^-} \sum_{t=1}^{N} R_t^2 \times \mathbb{I}(R_t < 0), \tag{5}$$

where N^- is the number of days with negative returns on the studied interval, R_t the log return between *t* and *t-1* and $\mathbb{I}$ and operator that assumes the value of 1 when $R_t <$

0 is true and 0 otherwise. In this study *N* was set as 30. Lastly, the Realized Semi Jump is an extension of Realized Semi Variance which focuses on observations where the returns are above or below a certain threshold. To focus on the downside again, we defined a jump as an observation with its log return observing the following inequality:

$$R_t < \bar{R}_t - 1.96\sigma_t \,, \tag{6}$$

where $\bar{R}_t$ is the average of the past 30 days log returns and σ_t the realized volatility during the same interval. These three metrics and the 7 and 30-day RVs are further described in Table II, as they are the most relevant for the study ahead.

TABLE II

FURTHER DESCRIPTIVE STATISTICS

	5-day Realized Volatility	30-day Realized Volatility	Sharpe Ratio	Negative Realized Semi Variance	Realized Semi Jump
Count	9126	9101	9,130	9101	9101
Mean	12.98%	14.89%	0.36%	-0.74%	-12.62%
Standard Deviation	9.89%	9.29%	7.36%	0.58%	10.99%
Min	0.74%	3.36%	-31.42%	-5.46%	-64.98%
25%	7.00%	9.28%	-4.67%	-0.91%	-17.78%
Median	10.61%	12.57%	0.48%	-0.59%	-13.70%
75%	15.94%	17.83%	5.70%	-0.38%	0.00%
Max	136.23%	93.10%	38.96%	-0.01%	0.00%

4. METHODOLOGY

4.1.Defining the Target Variable and overview of methodology

We define the target variable, that is, the variable intended to forecast, as the Future Realized Volatility. The RVs currently present in the database describe the volatility observed on the SP500 over the past *X* trading days (*X* being 5, 15 and 22). The objective of the study is to forecast the observed volatility over the next 30 regular days, so that the forecast is comparable with that of the market, using VIX as a proxy for market sentiment. That being said, the target variable for an observation on any given

day is the RV of the observation located 22 trading days ahead, which on average corresponds to 30 regular days, the timespan of the VIX.

During the study, we first analyze the forecast of the 30-day RV across three different ML models to choose the optimal model. This optimal model results from a study on which model performs better over four performance metrics (MAE, RMSE, R^2 Score and MAPE). During this same step, it is also tested the effects of adding a CEMMDAN decomposition on our RV series as proposed by Zhu & Zhong (2024) . The effects of using market variables as inputs are also analyzed by training the models with and without them. We also compare the ML models to two naïve models that serve as benchmarks. The first model is based on the VIX, where each forecasted future RV is the current value of the VIX Index on the day of forecasting. This model assumes that the implied volatility given by the market is a perfect predictor of the realized volatility. The second naïve model is based on the past 30-day RV. We take the volatility realized over the past 30 days as the forecast for the RV over the next 30 days.

We use the forecast of the best model found to simulate a trading strategy based on volatility swaps, with the goal of testing the practical usefulness of these forecasts. We measure the performance of the strategy across several metrics and the assumptions made for the strategy are tested over different sensitivity analysis tests.

4.2.CEEMDAN Decomposition

CEEMDAN is designed to address two critical shortcomings of its predecessors, EMD (Huang et al., 1998) and Ensemble Empirical Mode Decomposition (EEMD, see Wu & Huang, 2009): mode mixing and reconstruction error. Mode mixing refers to the issue where signals of vastly different scales are either spread across multiple components or entangled within a single one. While EEMD mitigates this by adding white noise during decomposition, it introduces new challenges, such as incomplete reconstruction and residual noise in the final signal. CEEMDAN effectively solves both problems by introducing a systematic, adaptive noise-assisted strategy that allows for accurate reconstruction of the original signal and cleaner separation into meaningful components (Torres et al., 2011, Zhu & Zhong, 2024).

The CEEMDAN algorithm works by adding white noise to the original time series and performing multiple decompositions. Through ensemble averaging of the decomposed modes across noise-added replications, it yields a series of Intrinsic Mode

Functions (IMFs), each representing a distinct frequency component from high to low, and a final residual trend. Mathematically, the original series can be represented as the sum of these IMFs and the residual, $r(t)$:

$$x(t) = \sum_{i=1}^{n} IMF_i(t) + r(t)\,. \quad (1)$$

Each IMF captures a specific oscillatory mode of the signal, while the residual captures the underlying long-term trend.

In the context of volatility forecasting, CEEMDAN helps transform a highly volatile and chaotic series into multiple smoother and more stationary sub-series. These decomposed components can then be used as inputs to machine learning.

In Zhu & Zhong (2024) each model is trained on individual IMFs, and the final prediction is obtained by aggregating the predictions from all components. This strategy tailored to each IMF allows models to better learn both short-term fluctuations and long-term trends, improving forecast accuracy. However, in this paper, each IMF computed from the target variable series will be used as an input to the models, and the forecast will be made on the RV itself instead of on each IMF series. This approach is chosen with the goal of simplifying the approach by reducing the number of forecasts made, while testing if the usefulness of using the CEEMDAN decomposition still stands.

The effectiveness of CEEMDAN has been empirically demonstrated by Zhu & Zhong (2024) across multiple financial indices, including the S&P 500, where it outperformed traditional and hybrid models in terms of mean squared error (MSE) and mean absolute error (MAE). Its integration into volatility modelling frameworks presents a robust preprocessing technique for enhancing model performance in noisy and non-linear financial environments.

4.3.ML Models to be used and Performance Metrics

4.3.1. Performance Metrics

To get the best possible results, three ML models are tested based upon four different performance metrics (MAE, RMSE, R^2 Score and MAPE).

The MAE measures the average of the absolute errors of each prediction made by the model. It is calculated as:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}|}{n}\,, \tag{2}$$

where y_i are actual values and $\hat{y}_i$ are predicted values.

MSE measures the average of the squares of the errors — that is, the average squared difference between the predicted values and the actual values. It penalizes larger errors more heavily, making it useful for models where large deviations are especially undesirable. Mathematically, it is calculated as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\,. \tag{3}$$

Root Mean Squared Error (RMSE) is simply the square root of MSE. It brings the unit of the error back to the original unit of the target variable, making it easier to interpret in a financial context.

The R^2 Score, an extension of the classical coefficient of determination, is adapted for use in non-linear and non-parametric machine learning models. It measures the proportion of the variance in the target variable that is captured by the model's predictions, regardless of how the model is estimated. A score of 1 indicates perfect prediction, while a score of 0 implies that the model performs no better than predicting the mean of the observed data.

The R^2 Score evaluates how well the predicted values replicate the true values relative to a baseline model (here the mean of the target). So:

$$R^2\ Score = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y_i})^2}\,. \tag{4}$$

The mean absolute percentage error (MAPE) expresses accuracy as a percentage, making it easier to communicate performance across different datasets. It is defined as the difference between actual and predicted values as a proportion of the actual values:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|\,. \tag{5}$$

This metric, although helpful due to allowing comparison across different models forecasting different variables, can be problematic when actual values are close to zero, which happens while studying volatility. However, it is still a helpful extra measure of comparison between models.

These metrics together provide a comprehensive assessment, from error magnitude (MSE, RMSE), to explanatory power (R^2 Score), and practical interpretability (MAPE).

4.3.2. Random Forest

The Random Forest model was developed by Breiman (2001) based on the idea that a committee of slightly different trained algorithms that give different opinions comes to a better conclusion than a single more specialized algorithm. This approach has been shown to enhance stability and accuracy of predictions. The final output of the Random Forest model is determined as the average of predictions made by an ensemble of decision trees, each trained on a different subset of the original training data. These subsets are generated using the Bootstrap Aggregating (or Bagging) technique, which involves randomly sampling the training data with replacement to create diverse datasets for each tree (Breiman, 1996). This approach reduces variance and helps prevent overfitting by ensuring that the individual trees are less correlated, despite being trained on the same underlying problem. Ho (1995) had already demonstrated that growing trees in random feature subspaces further enhances generalization, as it introduces additional randomness and diversity into the ensemble.

Each tree in a Random Forest commences with a root node that contains a bootstrap sample of the original dataset. For a given node, the decision regarding the split is made by employing a variant of the CART (Classification and Regression Trees) (Breiman et al., 1984), which finds the optimal split using the Gini impurity criterion (adapted to decision tree algorithms by Breiman et al. (1984) based on the work of Gini (1912)) for classification tasks and MSE for regression tasks. As our forecasting problem is a regression, the following MSE formula is employed:

$$MSE = min_{j,t}\left(\frac{N_{left}}{N} MSE_{left} + \frac{N_{right}}{N} MSE_{right}\right), \quad (6)$$

where j is a feature, t is the threshold, N is the total number of samples at the node, and N_{left} and N_{right} are the numbers of samples in the left and right subsets formed by the split. The Impurity in each node is then given by:

$$MSE_{node} = \frac{1}{N_{node}} \sum_{(i \in node)} (y_i - \bar{y}_{node})^2 , \quad (7)$$

where $\bar{y}_{node}$ is the average of the target values in the node.

Each tree ends in leaves, where the final value is determined by the average of the target variable values of the observation in the leaf, when there is no further efficient split possible. The final output is the average of each trees output. A visualization of the algorithm can be seen in Figure 3, while we show the hyperparameters used are in Table III.

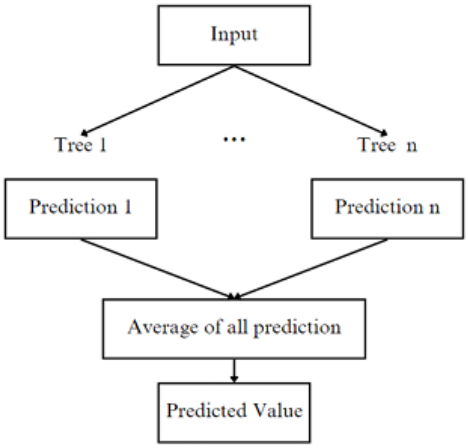

FIGURE 3 – Random Forest Algorithm

TABLE III

RANDOM FOREST HYPERPARAMETERS

Hyperparameter	Value
Maximum tree depth	10
Number of decision trees	300
Minimum number of samples required to split an internal node	5

4.3.3. XGBoost

XGBoost (Extreme Gradient Boosting), developed by Chen & Guestrin (2016), is a high-performance implementation of gradient boosted decision trees, designed for speed and performance. It has gained popularity in various machine learning tasks, including structured data classification and regression, due to its scalability, regularization, and robustness to overfitting (Chen & Guestrin, 2016). These characteristics make it a suitable model for volatility forecasting.

XGBoost constructs an ensemble of decision trees in a sequential manner, where each tree attempts to correct the errors made by the previous ones. It minimizes a regularized objective function that balances model complexity and training error, defined as:

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k), \tag{8}$$

Where l is a differentiable loss function (e.g., logistic or squared error), and $\Omega(f_k)$ is a regularization term that penalizes model complexity to prevent overfitting:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \tag{9}$$

with T representing the number of leaves and w_j the score on each leaf.

Each prediction at time *t* is also adjusted via a learning rate ratio to avoid overfitting by slowing learning and improving generalization.

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i), \tag{10}$$

where η is the learning rate ($0 < \eta < 1$) and $f_t(x_i)$ the output of the newly added tree at iteration t.

Some of the key innovations that contribute to XGBoost's effectiveness in advanced machine learning tasks relate to its handling of missing data, memory efficiency, and optimization strategy. The algorithm is designed to automatically manage missing values by learning, during training, the optimal direction for handling such instances at each split. This allows the model to make accurate predictions even when some input features are incomplete, without requiring prior imputation.

In addition, XGBoost employs a columnar (or column block) data structure, which improves computational efficiency by enhancing memory access. This format reduces memory overhead and speeds up tree construction. XGBoost also performs second-order optimization, meaning it utilizes both the first and second derivatives (gradients and Hessians) of the loss function. This enables more precise and stable updates to tree splits compared to traditional boosting methods, which typically only rely on first-order gradients. These architectural and algorithmic advances make XGBoost highly scalable and robust across a wide range of structured data problems (Chen & Guestrin, 2016).

The model is applied with some fixed hyperparameters that influence aspects such as maximum tree dept and learning rate. Table IV presents the hyperparameters used.

TABLE IV

XGBOOST HYPERPARAMETERS

Hyperparameter	Value
Maximum tree depth	6
Learning Rate (η)	0.05
Maximum number of boosting rounds (trees)	300
Early stopping rounds	50
Loss function	squared error
Fraction of features used to train each tree	0.8
Fraction of training data used per tree	0.8
Initial prediction score for all instances	0.5

4.3.4. Neural Networks

Artificial Neural Networks (ANNs) (see McCulloch and Pitts, 1943 and Rosenblatt, 1958), are a class of non-parametric machine learning models inspired by the interconnected structure of biological neurons. Their principal strength lies in their ability to model complex, non-linear relationships between inputs and outputs, making them well-suited to domains like financial time series forecasting, where such relationships are rarely linear or stationary (Gaspar et al., 2020).

In this study, a feedforward neural network is implemented using a Multi-Layer Perceptron (MLP) architecture (Rumelhart et al., 1986. The network architecture consists of three hidden layers with sizes (100, 100, 50). The activation function used in the hidden layers is the Rectified Linear Unit (ReLU), defined as:

$$\Phi(z) = \max(0, z), \tag{11}$$

which introduces non-linearity while mitigating vanishing gradient issues (Glorot et al., 2011).

The forward pass through each neuron in the network is governed by the equation:

$$a_j^{(l)} = \Phi\left(\sum_{i=1}^{n=l-1} w_{j,i}^l a_i^{l-1} + b_j^l\right), \tag{12}$$

where:

- $a_j^{(l)}$ is the activation of neuron *j* in layer *l*,
- $w_{j,i}^{l}$ is the weight connecting neuron *i* in layer *l*-1 to neuron *j* in layer *l,*
- b_j^{l} is the bias term.

The final output layer computes the predicted value $\hat{y}$,which is compared to the actual value y using the Mean Squared Error (MSE) loss function. This loss is minimized during training using the Adam optimizer (Kingma and Ba, 2015), an adaptive stochastic gradient descent algorithm that combines momentum and adaptive learning rate techniques:

$$\theta_t = \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \tag{13}$$

where $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected first and second moment estimates (moving averages of past gradients and past squared gradients, respectively), and η is the learning rate. ϵ is a term added to ensure numerical stability even when $\hat{v}_t$ is close to 0.

The network training process includes early stopping, a regularization technique that halts training when performance on a validation set no longer improves. This helps mitigate overfitting, which is a common risk with deep architectures on small or noisy financial datasets.

From a theoretical perspective, the Universal Approximation Theorem asserts that a feedforward neural network with at least one hidden layer and a sufficient number of neurons can approximate any Borel-measurable function to any desired accuracy, under mild assumptions (Hornik et al., 1989). This provides strong justification for applying MLPs to complex regression problems such as volatility prediction.

4.4.Trading Strategy

The trading strategy on which the forecasts are tested is based on volatility swaps as described by Bouzoubaa & Osseiran, 2010.

4.4.1. Volatility Swaps

A volatility swap is a product that allows investors to directly trade volatility without being exposed to other factors. They trade over the counter (OTC) and are

illiquid instruments that are usually only available in intra-bank trades or to institutional clients.

However, given that they are the most suitable product to trade volatility, we use them in our strategy.

Volatility swaps pay the difference between the realized volatility of a financial asset over the life of the swap and some prespecified strike volatility. The strike is chosen between the counterparties in order for the present value of the swap to be zero. This strike is "determined by the implied volatility skew", depending on the at-the-money (ATM) implied volatility and the convexity of the skew. These swaps "allow one to take a clean view on the levels of implied volatility being high or low relative to the expected realized volatility" (Bouzoubaa & Osseiran, 2010). It is also worth noting that volatility swaps usually only have one cashflow exchange date at maturity.

The payoff of the swap to the buyer is:

$$Swap_{payoff} = (RV(T) - K) \times N\,, \tag{20}$$

where *RV(T)* is the observed realized volatility of the underlying during the life of the swap, *K* is the strike volatility and *N* is the notional. Note that like in regular swaps, the notional *N* never exchanges hands and is used only to compute the payoff at maturity.

As a convention, the buyer of the swap is defined as the party that has a bullish view on volatility, i.e. the party receiving the payoff described in (20), and the seller the party with the bearish view, receiving the symmetric payoff.

4.4.2. The Strike of the Volatility Swap

Since the VIX is defined as "a measure of constant, 30-day expected volatility of the U.S. stock market, derived from real-time, mid-quote prices of S&P 500 Index (SPX) call and put options" (official CBOE website), here we take it as the strike for the volatility swaps, which in the strategy have 30 days maturity to align with the VIX. This way, using a simple financial product, one can directly bet on volatility based on the value of the forecast and that of a public available Index, whose live values are always available. This is of course an assumption as in real trades, the strike of volatility swaps is not pre-defined nor needs to be the VIX, being instead any volatility level agreed by both parties. This assumption is later tested by performing sensitivity analysis to variations in the trading costs which in turn can be also interpreted as changes in the strike.

4.4.3. Trading Strategy Implementation

With volatility swaps already defined, the actual trading strategy can be defined as follows. The three variables of interest are:

- RV_i is the realized volatility of the SP500 over the 22 trading days (around 30 days) after the day i
- F_i is the forecast for that realized volatility, that is, F_i is the output of the best model on any day i
- VIX_i is the VIX index value on any day i and is the strike of each volatility swap entered.

The strategy consists of entering a volatility swap with 30 days maturity (22 trading days) on every trading day, with the direction of the swap (the investor being the buyer or seller) defined by the signal of $F_i - VIX_i$. If this difference is positive, the model signals that the VIX is lower than what the forecast of future RV is, and as such the investor would be the buyer of the swap. The opposite would happen if the signal were negative.

As such, here are the payoffs for any swap entered on day i, for each scenario:

- If $F_i > VIX_i$, we buy a swap and 30 days later the payoff is:

$$Swap_{payoff} = (RV_i - VIX_i) \times N\ , \tag{21}$$

- If $F_i < VIX_i$, we sell a swap and 30 days later the payoff is:

$$Swap_{payoff} = -(RV_i - VIX_i) \times N\ . \tag{22}$$

The strategy then consists of 22 constant open positions on 30-day maturity volatility swaps as described above. These 22 open positions come from the fact that we approximate 30 regular days by 22 trading days, and the swaps that mature each day are continually replaced by new ones.

4.4.4. Trading Costs, Spread and Initial Capital

As mentioned before, these are illiquid products, and as such, trading and transactions costs need to be taken into account. Let us denote by c the trading cost (for simplification of language, from here on the trading and transaction costs will be bundled together and studied as if merged) in percentage of the notional for each swap. Then, the payoff in case the investor is the buyer can be rewritten as:

$$Swap_{payoff} = (RV_i - VIX_i) \times N - c \times N = (RV_i - VIX_i - c) \times N\ . \tag{23}$$

Even as the seller of the swap, the investor still incurs on these trading costs as here buyer and seller are only terms used to define our view on volatility, but the investor is always the one who is entering a trade with a market maker who charges these fees whichever the direction of the swap. As such, the payoff for being the sellers of the swap and taking into account trading costs is:

$$Swap_{payoff} = -(RV_i - VIX_i) \times N - c \times N = -(RV_i - VIX_i + c) \times N\,. \qquad (24)$$

In theory, an investor would not need any initial capital to enter in this strategy, however, if the swap ends up giving a negative payoff, the investor needs to cover the loss. As such, in the simulation, the strategy starts with a loan of $X, which pays quarterly interest based on the 3-month T-bill yield plus a spread (s), as it is unrealistic to assume that the interest paid would merely be a proxy to the risk-free rate.

In the simulation, RV_i and VIX_i are known values, and F_i is the output of the model. However, the trading costs (c), the notional amount (N), the initial loan amount (X) and the spread (s) are inputs to the simulation that need to be given values. The notional and loan amount are closely related and can be seen as a relation, so what is studied is the loan amounts needed for a fixed notional, which is $100. As such, we do a sensitivity analysis on each one of the three variables (c, s and X) to see how the strategy behaves in different possible scenarios and assumptions.

5. Results

5.1. CEEMDAN Decomposition

The first results we present are that of the CEEMDAN decomposition of the 30-day Realized Volatility series. Using the *PyEMD* library from Python, the CEEMDAN function is applied to our time series, with the decomposition being shown in Figure 4. The 11 IMFs and the Residuals can be seen. It is easily observable how the decomposition starts from extracting very short-term patterns in the data in the first IMFs, and progressively finding longer trends in the time series until IMF 11 which is the least varying sub-series. It is also worth noting the scale of the residual series, whose values are on a 1 x 10^{-16} scale, showing how complete the CEEMDAN decomposition is, with most information on the original series being present on the 11 IMFs and barely any left as residual.

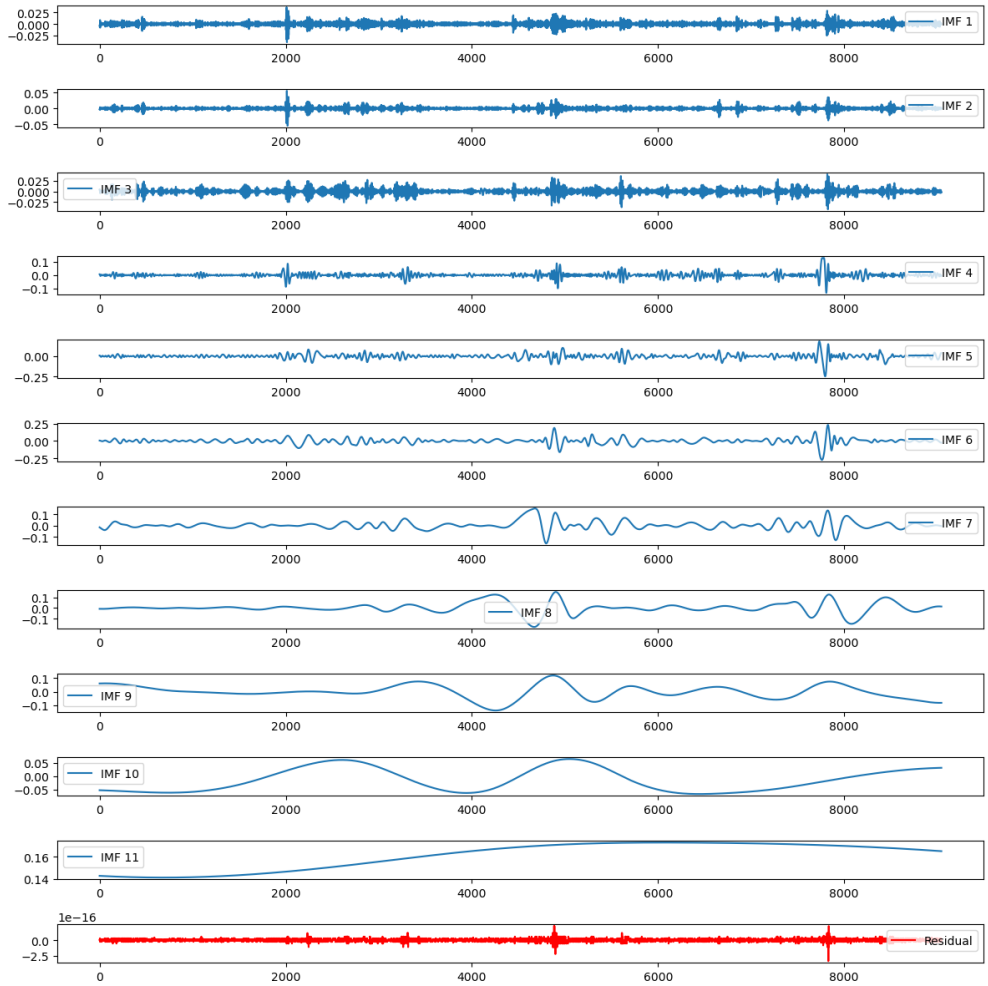

FIGURE 4 – CEEMDAN decomposition of the 30-day Realized Volatility

5.2.Best Model and Forecast – XGBoost

Next, the three ML models are trained on a train – test split of 82/18. We present the performance metrics of each model with the different inputs in Table V. There, we also present the same metrics applied to out two naïve models, the naïve with VIX and naïve with RV. As it can be observed, the XGBoost model using IMFs and market variables as variables was overall the best performing model. The effects of adding the IMFs as inputs are also noticeable, with significant increase in performance in all metrics. The effects of the market variables are more nuanced, with the changes being positive although almost negligible for XGBoost and Random Forest, but negative for the ANN.

TABLE V

PERFORMANCE METRICS OF ML MODELS

	Without IMFs and Without Market Variables			
	MAE	RMSE	R^2 Score	MAPE
Random Forest	0.0320	0.0445	-0.45	36.74%
XGBoost	0.0300	0.0383	-0.07	34.46%
Neural Network	0.0316	0.0394	-0.13	35.72%
	With IMFs and Without Market Variables			
	MAE	RMSE	R^2 Score	MAPE
Random Forest	0.0225	0.0274	0.45	24.82%
XGBoost	0.0202	0.0245	0.56	22.83%
Neural Network	0.0197	0.0271	0.47	**19.79%**
	With IMFs and With Market Variables			
	MAE	RMSE	R^2 Score	MAPE
Random Forest	0.0225	0.0273	0.46	24.82%
XGBoost	**0.0195**	**0.0240**	**0.58**	21.53%
Neural Network	0.0251	0.0350	0.10	24.84%
	Benchmark Models			
	MAE	RMSE	R^2 Score	MAPE
Naïve with VIX	0.0554	0.0634	-1.93	61.90%
Naïve with RV	0.0263	0.0337	0.17	26.38%

In Table V we also highlight in bold the best model and input combination for each metric. The XGBoost model with both the IMFs and the market variables performs better than the rest in all but one metric, that being the MAPE, on which the ANN with only IMFs is the best model. Going more in depth on each one of the tested inputs, there is no doubt that the inclusion of the outputs from the CEEMDAN decomposition improves performance, with all metrics getting better, namely the R^2 Score. The market variables have a slightly positive impact on the forecast performance of XGBoost and

Random Forest models but considerably make the ANN perform worse across all metrics. This model is worse at dealing with the increase in complexity due to extra variables and cannot extract any useful new information from the market variables without losing interpretability of the IMFs and original inputs.

It is also important to compare the ML models with the two naïve models. First comparing them with each other, we see that the model based on the past RVs (Naïve with RV) shows better metrics than the one based on the VIX, suggesting that using past realized volatilities as proxies to future ones might be more accurate than using implied volatilities. Then, comparing our best Naïve model with the ML models, we start by noting that it outperforms the three ML models when we don't use the IMFs as inputs. However, the addition of the CEEMDAN decomposition does improve the ML models performance significantly, making all three ML models perform better than the naïve with RVs model across all metrics. This is an important result as the extra complexity of using ML for forecasting volatility needs to be justified by an increase in accuracy over simple models that are easier to implement.

With these results, we get the final forecast of the 30-day RV using the best model found, the XGBoost with both IMFs and market variables. We recall the Train-Test split, which is the following:

- Train Data: 02/01/1990 – 31/12/2017 (around 82% of the data)
- Test Data: 02/01/2018 – 31/12/2024 (around 18% of the data)

We show the forecasting results in Figure 5, where the model predictions can be seen next to the real values. It is seen that the forecast follows most changes in the observable RV, albeit with some lag at times. Note as well how the model is not able to fully predict the huge spike in volatility from COVID (beginning of 2020). In Figure 6 the VIX index is added as well, providing a clear picture of both estimators (our model and the VIX) compared to the actual values. It is seen that the VIX shows the same problem of sometimes lagging real RV. However, the VIX is a more volatile estimator than our forecast, showing higher and more frequent spikes in volatility.

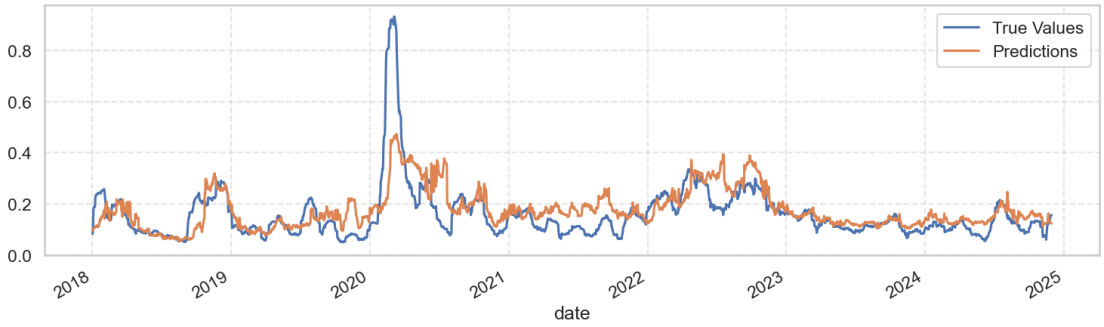

FIGURE 5 –30-day observed RV and XGBoost Predictions plotted

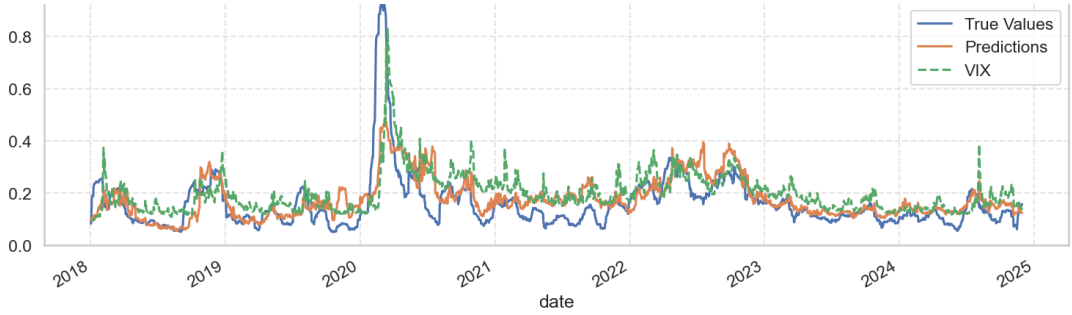

FIGURE 6 –30-day observed RV, XGBoost Predictions and VIX Index Plotted

5.3.Trading Strategy Results

The trading strategy is simulated during the test period. As mentioned in the Methodology Section, it consists of a portfolio of 30-day maturity volatility swaps, whose direction depends on our forecast. The notional of each swap is set to $100. For the first simulation, whose results can be seen in Figure 7 and Table VI, the trading costs were fixed at 2%, the spread at 1% and the initial loan amount to $100.

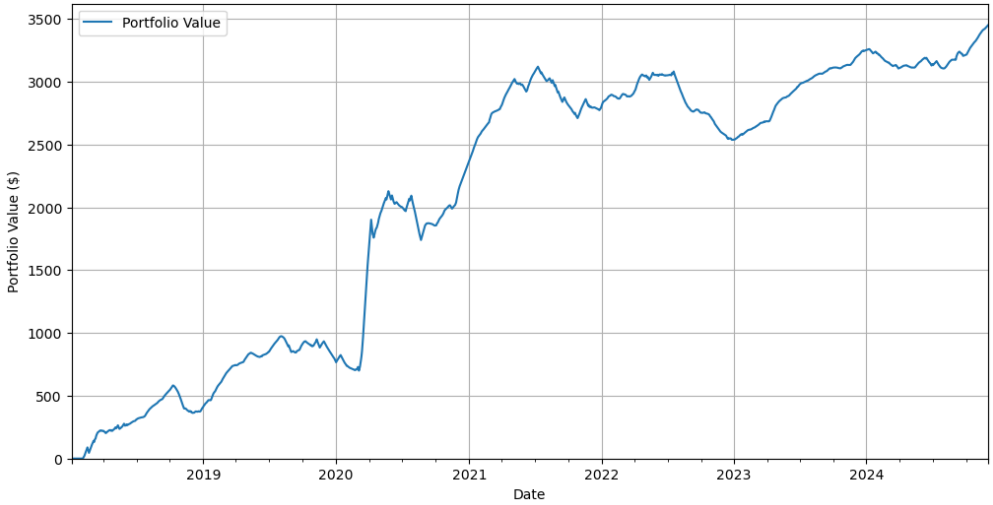

FIGURE 7 –Accumulated profits of the Strategy (c=2%, s=1% and X=$100)

TABLE VI

TRADING STRATEGY RESULTS (C=2%, S=1% AND X=$1M)

Parameter	Value
Notional per Swap	$100
Initial Loan Amount	$100
Trading Costs	2%
Spread	1%
Final Portfolio Value	$3,548.79
Annual Return on Notional	68.30%
Percentage of Swaps with Positive Payoffs	65.47%
Average Positive Payoff	$6.13
Average Negative Payoff	$5.87

Before starting the interpretation of the results, it is relevant to define how they will be measured. As this particular investment strategy does not require any initial investment, popular measures such as returns on investment are not applicable due to the lack of an initial investment. As such, different ways to measure the performance of the strategy are employed. The annualized return (AR) on notional is defined in Equation 24. Rolling returns are also used, defined as the return on the accumulated value of the strategy at a certain date. We also employ the use of absolute returns, taking the returns at face value, without comparing them to any reference amount. Lastly, we analyse the Sharpe ratio of the strategy. The return on notional is written as:

$$AR = \left(\frac{FV - N}{N} + 1\right)^{\frac{1}{n}} - 1\,, \tag{24}$$

with FV being the final value of the portfolio, N the notional of each swap, and n the number of years the strategy was ran.

The positive performance of the strategy is easily seen in Figure 7. It is worth noting the returns during the COVID period, where the accumulated value of the strategy went from around $800 to $2,000 in around two months. The AR on notional, defined in Equation 24, is 68.30%. As comparison, the SP500 had a performance of around 14% on this same metric in the same time frame. However, these results only have meaning if we do an adequate sensitivity analysis, as they fully depend on the values set for our variables (c as the trading costs, s as the spread and X as the initial loan amount).

FIGURE 8 – Cashflows over 2018, 2020, 2022 and 2024

For better clarity on how the strategy performs, Figure 8 shows the daily cashflows of the strategy. Each green bar represents a trading day where the corresponding maturing swap has a positive payoff, while red bars represent negative payoffs. We present the years of 2018, 2020, 2022 and 2024 in Figure 8 to give an overview of the cashflows behavior, while the remaining 3 years of the strategy are in the Appendices. The choice of the years presented in this section is arbitrary. Over the four years presented, it is worth noticing the balance between positive and negative flows in 2018 and 2024, while 2020 not only shows higher values as well as a majority of positive days. 2022 shows values in the same scale as 2018, however this year is characterized by a majority of negative payoffs, namely in the second half of the year.

After, we present two plots with rolling returns. Figure 9 has the rolling return on the accumulated profit from the last trading day, i.e. the payoff of each day expressed as a percentage of the accumulated portfolio value from the day before. Figure 10 shows the same but compares the gain in one month (22 trading days) of the strategy, with the accumulated value at the beginning of those trading days. As expected, both series converge to 0 as the portfolio value grows while the returns keep the same scale, as they are calculated on the constant notional. It is interesting however to note that the daily returns seem to have an average close to 0 while the monthly returns already show the clear tendency for the strategy to have strong positive performance.

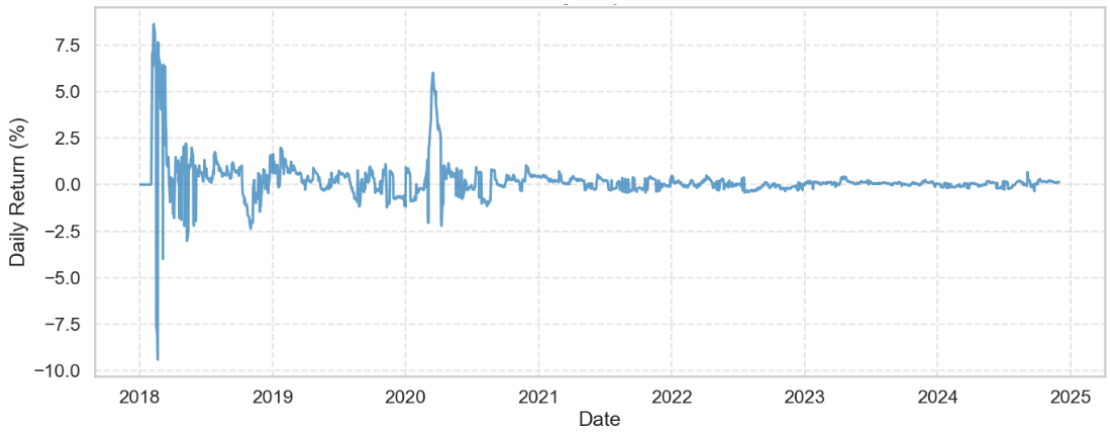

Figure 9 – Daily Rolling Returns

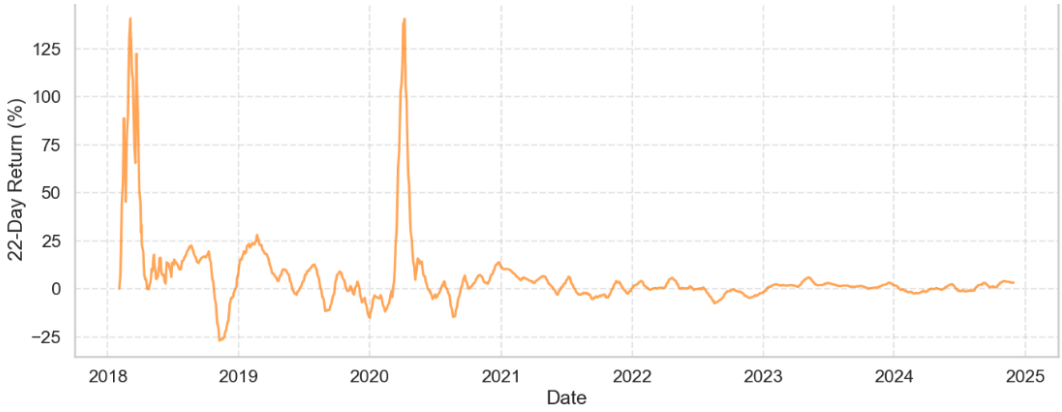

Figure 10 – Monthly Rolling Returns

To conclude we look into the risk-return profile of the strategy using the Sharpe ratio. As mentioned in the Data Section, the Sharpe ratio is defined as the expected excess return of the strategy in comparison to the risk-free rate, by units of risk measured by the volatility. This measure is interesting as it puts the returns of an investment in a risk management perspective, in the sense that it gives how much the investment yields by unit of volatility. However, this volatility does not distinguish upside and downside volatility. This means that two strategies with symmetric returns, one positive and the other negative, since having same volatility, will be equally penalised by their volatilities in the Sharpe ratio. Certain strategies can yield only positive returns but if they wildly vary, i.e. highly volatile, the Sharpe ratio will still be low, potentially leading one to believe the strategy to be riskier than it is.

The inputs to calculate the volatility swap strategy Sharpe ratio are the mean return of the strategy, its volatility, measured as the standard deviation of the returns and the mean risk-free rate. The mean return of the strategy is calculated as the average of the swaps payoffs in percentage of the nominal while the mean risk-free rate is taken as the average of taking the daily 3-month T-Bill yield and making it a 1-day yield. The mean excess return is the difference between the two aforementioned means.

Table VII presents the Sharpe ratio analysis for the whole period of the simulation and for each individual year. It is here that the caveat of the trading strategy is found. While the high returns cannot be ignored, when put in comparison to the strategy volatility, one can see that these payoffs are highly volatile, causing the Sharpe ratio to be low, around 0.22, i.e. there is not a lot of return per unit of volatility. It is usually said that a "good" Sharpe ratio is one above 1, as one would get more returns than the volatility being exposed to. As mentioned above, this ratio does not take into account the difference between upside and downside volatility, so a low Sharpe ratio does not

make the strategy a bad investment by itself, but it is a good indicator of the type of investment this would be – one where payoffs can vary a lot and very fast, almost as easily providing extreme positive returns one day as giving great losses the next.

Taking also a look at each individual year, it is worth noting the year of 2023, being the standout performer with a Sharpe ratio above 1. While the average daily return of this year is not the highest observed, that title belonging to the year of 2020, the low volatility of returns in 2023 increases the ratio by a substantial amount. We also highlight the year of 2022 as it is the only year with a negative Sharpe ratio, due to the negative expected excess return.

TABLE VII

SHARPE RATIO ANALYSIS

	2018-2024	2018	2019	2020	2021	2022	2023	2024
Sharpe Ratio	0.22	0.29	0.27	0.33	0.22	-0.17	1.16	0.19
Mean Daily Excess Return	2.0%	1.6%	1.5%	6.1%	1.7%	-1%	2.8%	1%
Volatility of Returns	8.9%	5.6%	5.4%	18.4%	7.9%	5.8%	2.4%	4.5%

5.3.1. Trading Strategy - Sensitivity Analysis

In this subsection we first present the sensitivity analysis for the trading costs and for the spread, *ceteris paribus*. For the initial loan amount, the minimum value that makes the strategy never go negative is computed as a function of the trading costs. The performance of the strategy starting at different points in time is also tested by analysing rolling one-year performances.

Figures 11 and 12 show the sensitivity of the AR on notional to trading costs and spread, respectively. First looking at Figure 11, it is noticeable how the strategy stops yielding profits at 4% of trading costs, showing an average sensitivity of around -20% of AR for every +1% of trading costs. One can look at the trading costs not only as a fee charged by the counterparty, but also as a penalization of taking the VIX as strike and of other assumptions made for our model and strategy. It is seen that our forecast can take a 4% combination of fees and penalizations under these conditions.

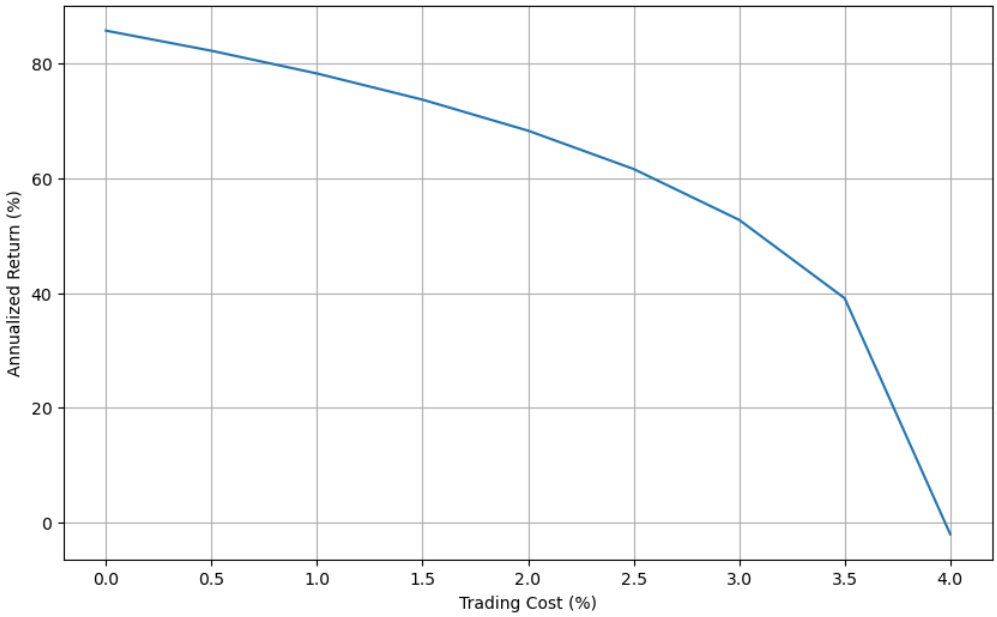

FIGURE 11 –Sensitivity Analysis to Trading Costs (s=1% and X=$100)

On the other hand, we show in Figure 12 that the spread has little effect on the returns of the strategy, which is expected, as the interest payments are only quarterly (by our own assumption) while the trading costs come into effect every trading day. Simulating the strategy with the spread varying from 0 to 10% shows an impact of less that 2% on the AR on Notional. It is also important to note that the spread is also very dependent on the loan amount, meaning that if he had chosen a higher loan amount when performing this analysis, the impact of the spread could have been higher. However, since the strategy never has any days where the accumulated wealth is negative for the values that we assumed as trading costs and initial loan amount, it can be inferred that this is a conservative analysis as the initial loan amount could be lower.

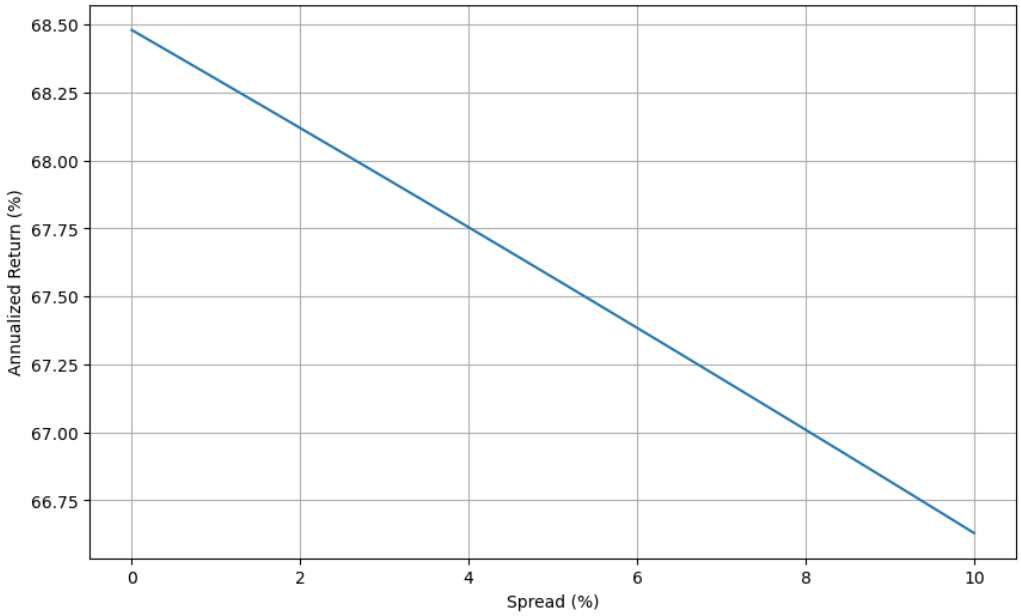

FIGURE 12 –Sensitivity Analysis to Spread (c=2% and X=$100)

Next, we perform the study on the minimum initial loan amount. This value is defined as the one that makes the strategy never have a day were the accumulated wealth is negative. This is an important analysis as big loans are not accessible to every investor, and as such, it is important to test how much this strategy is free of initial capital. Also, it is in the interest of the investor to minimize the loan taken. In Figure 13 we see that this minimum amount grows exponentially with the trading costs. It is once again shown that the limit of the effectiveness of the strategy lies at the 4% trading costs mark. Until the 3% level, however, the strategy is indeed free of any initial capital in the world of the assumptions made.

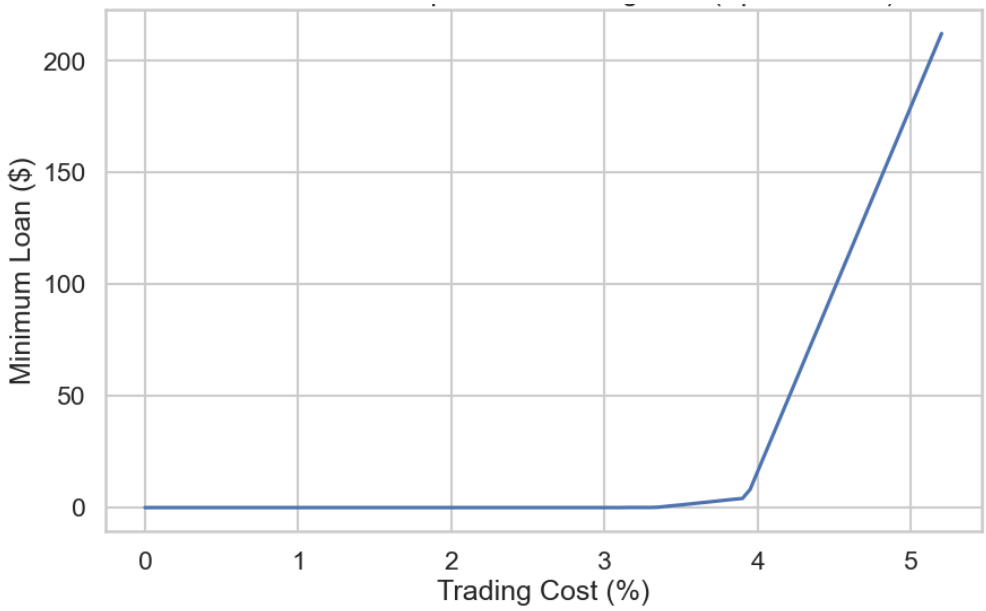

FIGURE 13 –Minimum Loan Amount in function of Trading Costs (s=1%)

The final test performed is the rolling one-year performance of the strategy, where we compute the performance over one-year periods on different starting points. We do this analysis to check for certain period dependency of the strategy, as well as to analyse which conditions favour the strategy the most. In the upper panel of Figure 14 we show the analysis with trading costs fixed at 2% and it is seen that on only three starting dates (shown as the blue dots) the following one-year performance was negative, those negative performances coming on the mid-2021 to mid-2023 period. The best performing period, as noticed before, is the one including the beginning of the COVID outbreak, a period characterized by extremely high volatility and bearish markets. Figure 14 also shows the same but for 3% trading costs, in the lower panel. In these conditions, the last two simulations show negative returns as well, once again proving the strategy sensitivity to trading costs.

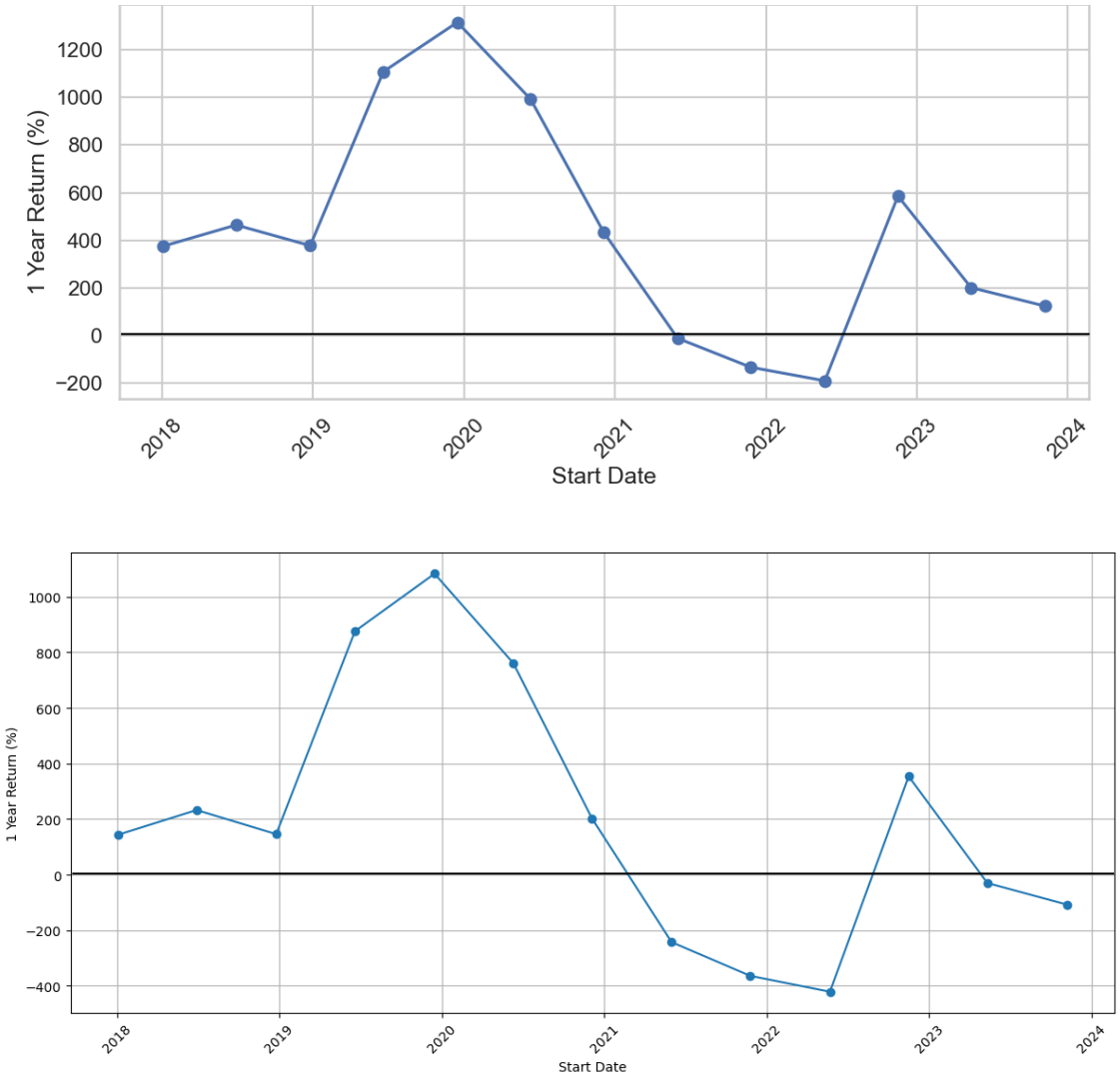

FIGURE 14 – Performance of the strategy with different starting points, above c=2%, below c=3% (s=1%, X=\$100)

6. CONCLUSION

The objective of this work is to test the adequacy of using machine learning to forecast volatility. In the last chapter we show that the forecast made by the XGBoost model, when applied in a trading strategy and depending on the assumptions made, can bring returns above the SP500 itself.

The previous literature in the field of volatility forecasting is mostly focused on the models and inputs to be used and that analysis is also made here, as it is essential and the driving force of theoretical innovation. We show that the XGBoost model is the most efficient model between the three (XGBoost, Random Forests and Artificial Neural Networks) proposed, while the CEEMDAN decomposition has shown to definitely improve performance when applied as input to the models. This approach simplifies the previous method employed by Zhu & Zhong (2024) while keeping the improvements of adding the decomposition to the forecast. The combination of the ML model with the CEEMDAN decomposition outperforms the two chosen benchmark naïve models. As suggested by Zhu & Zhong (2024) in their own study, the impact of adding market variables is also tested and is shown to also improve performance in the tree-based models, albeit in a much smaller scale while worsening the performance of the ANN.

Where we go further than previous literature is by putting the performance of the forecasts to the test in a trading strategy using volatility swaps. Although the strategy performs well in terms of returns in the tested sample and with realistic assumptions, it is found that the trading costs are the main driver of the performance of the strategy, while the volatility of the strategy returns are high, leading to a poor Sharpe ratio.

In future studies, the models applied could be further developed, not only by using more state-of-the-art models but also by performing a more in depth hyperparameter search. The assumptions for the trading strategy can also be further studied and compared to those observed on traded financial products observed on the market.

REFERENCES

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166. https://doi.org/10.1109/72.279181

Black, F. and Scholes, M., 1973. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3), pp.637–654. Available at: https://www.jstor.org/stable/1831029

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

Bouzoubaa, M. & Osseiran, A. (2010). Exotic Options and Hybrids – A Guide to Structuring, Pricing and Trading, 1st Ed. United Kingdom: John Wiley & Sons, Ltd.

Breiman, L., 1996. Bagging predictors. *Machine Learning*, 24(2), pp.123–140. https://doi.org/10.1007/BF00058655

Breiman, L., 2001. Random forests. *Machine Learning*, 45(1), pp.5–32. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J., 1984. Classification and regression trees. Belmont, CA: Wadsworth International Group. https://link.springer.com/book/10.1007/978-1-4899-3242-6

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

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Christensen, K., Siggaard, M.V. & Veliyev, B., 2023. *A Machine Learning Approach to Volatility Forecasting*. Journal of Financial Econometrics, 21(5), pp.1680–1727. https://doi.org/10.1093/jjfinec/nbac020

Corsi, F (2009). A Simple Approximate Long-Memory Model of Realized Volatility. *Journal of Financial Econometrics, 2009, Vol. 7, No. 2,* 174–196. https://academic.oup.com/jfec/article/7/2/174/856522

Engle, R.F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica 50,* 987–1007. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation on JSTOR

Gaspar, R.M., Lopes, S.D., Sequeira, B., 2020. Neural Network Pricing of American Put Options. *Risks* 8, 73. https://doi.org/10.3390/risks8030073

Gini, C., 1912. *Variabilità e mutabilità*. Bologna: Tipografia di P. Cuppini.

Glorot, X., Bordes, A. and Bengio, Y., 2011. Deep sparse rectifier neural networks. In: *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*. https://proceedings.mlr.press/v15/glorot11a.html

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), pp.1779–1801. https://www.jstor.org/stable/2329067

Ho, T.K., 1995. Random decision forests. In: *Proceedings of the 3rd International Conference on Document Analysis and Recognition (ICDAR 1995)*, Montreal, Canada, 14–16 August 1995. Vol. 1. IEEE, pp.278–282. https://doi.org/10.1109/ICDAR.1995.598994

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

Hornik, K., Stinchcombe, M. and White, H., 1989. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), pp.359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.C., Tung, C.C. and Liu, H.H., 1998. *The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis*. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 454(1971), pp.903–995. https://doi.org/10.1098/rspa.1998.0193

Ioffe, S. & Szegedy, C., 2015. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. In: *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. Lille, France: PMLR, pp.448–456. https://proceedings.mlr.press/v37/ioffe15.html

Kingma, D.P. and Ba, J., 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1412.6980

Lin, L., Dong, Y. and Ye, V., 2021. *Comparison of Chinese 50 ETF put option pricing based on four algorithms*. In: Proceedings of the 2021 12th International Conference on E-business, Management and Economics (ICEME 2021), Beijing, China, 17–19 July 2021. New York: ACM, pp.230–238. https://doi.org/10.1145/3481127.3481246

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 5(4), 115–133.

Miura, R., Pichl, L., Kaizoji, T. (2019). Artificial neural networks for realized volatlity prediction in cryptocurrency time series. *Springer International Publishing, Cham,* pp. 165–172. Artificial Neural Networks for Realized Volatility Prediction in Cryptocurrency Time Series | SpringerLink

Nelson, D.B., 1991. *Conditional heteroskedasticity in asset returns: A new approach*. Econometrica, 59(2), pp.347–370. https://www.jstor.org/stable/2938260

Rosa, R., Maciel, L., Gomide, F., Ballini, R. (2014). Evolving hybrid neural fussy network for realized volatility forecasting with jumps. *IEEE, London,* pp. 481–488. https://ieeexplore.ieee.org/document/6924112

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386–408.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. Journal of Machine Learning Research, 15, 1929–1958. https://jmlr.org/papers/v15/srivastava14a.html

Torres, M.E., Colominas, M.A., Schlotthauer, G. and Flandrin, P. (2011). A complete ensemble empirical mode decomposition with adaptive noise. In: *Proceedings of the 36th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2011)*, Prague, Czech Republic, 22–27 May 2011. IEEE, pp. 4144–4147. https://doi.org/10.1109/ICASSP.2011.5947265

Wu, Z. and Huang, N.E., 2009. *Ensemble empirical mode decomposition: a noise-assisted data analysis method*. Advances in Adaptive Data Analysis, 1(1), pp.1–41. https://doi.org/10.1142/S1793536909000047

Zhang, C., Zhang, Y., Cucuringu, M., & Qian, Z. (2024). 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

Zhu, Y., & Zhong, Z. (2024). Stock index volatility forecasting based on novel CEEMDAN-LSTM-BN model. *Proceedings of the International Conference on Digital Economy, Blockchain and Artificial Intelligence (DEBAI 2024)*, August 23–25, 2024, Guangzhou, China. ACM. https://doi.org/10.1145/3700058.3700078

Databases

Daily 10-year US Bond Yield - https://fred.stlouisfed.org/series/DGS10 (accessed on the 27/01/2025)

Daily 3-month T-Bill rate - https://fred.stlouisfed.org/series/DTB3 (accessed on the 27/01/2025)

Daily Gold Future prices - https://www.investing.com/commodities/gold-historical-data (accessed on the 27/01/2025)

Daily Brent Oil Future prices - https://www.investing.com/commodities/brent-oil-historical-data (accessed on the 27/01/2025)

Daily USD/EUR exchange rate - https://www.investing.com/currencies/usd-eur-historical-data (accessed on the 27/01/2025)

Others

CBOE Official Website - https://www.cboe.com/tradable_products/vix/ (accessed on the 06/04/2025)

LSEG Official Report - https://www.lseg.com/content/dam/data-analytics/en_us/documents/charts/lseg-size-of-global-market-2024-in-charts.pdf (accessed on the 30/04/2025)

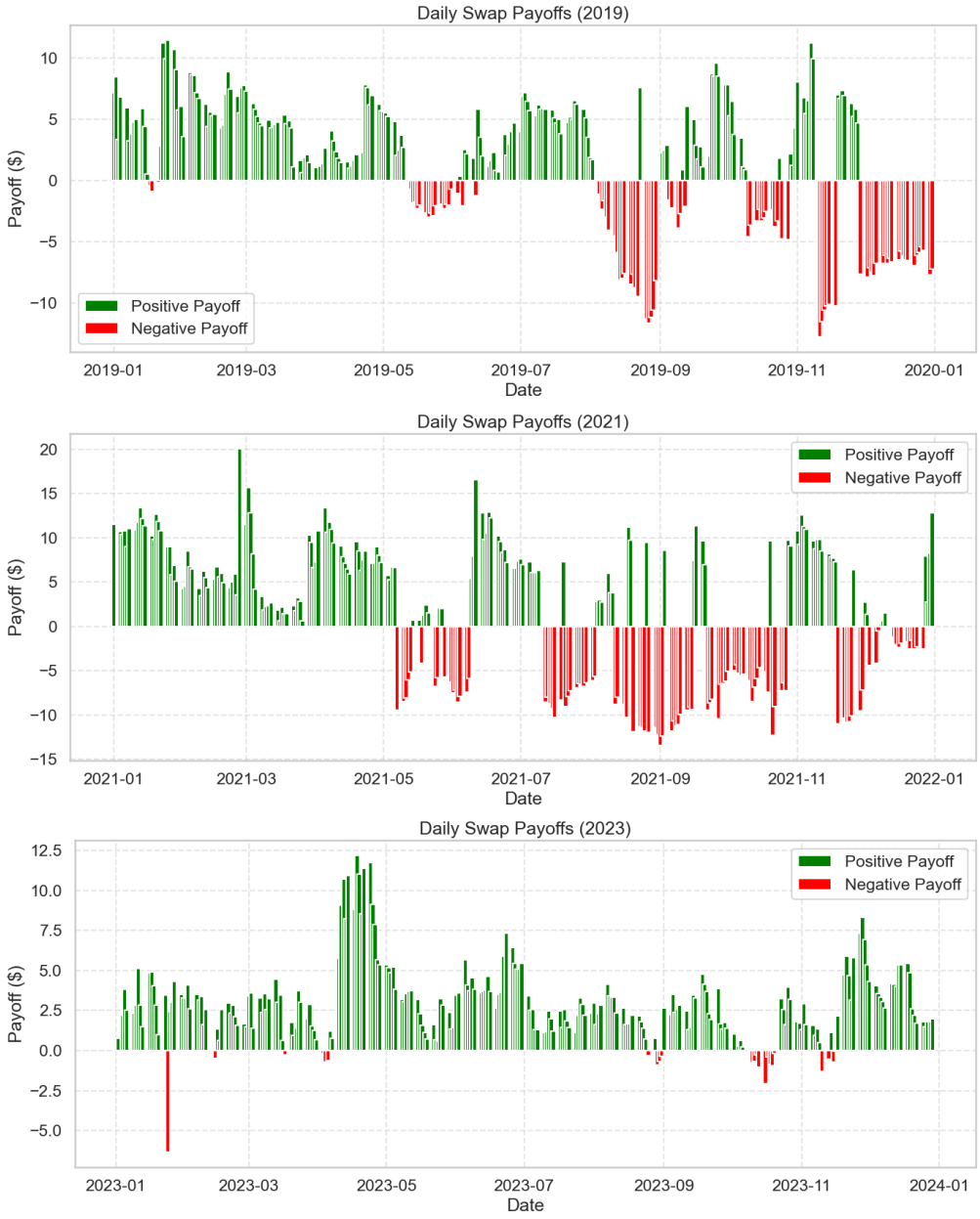

Figure 15 – Cashflows over 2019, 2021 and 2023

DISCLAIMER

This master thesis was developed with strict adherence to the academic integrity policies and guidelines set forth by ISEG, Universidade de Lisboa. The work presented herein is the result of my own research, analysis, and writing, unless otherwise cited. In the interest of transparency, I provide the following disclosure regarding the use of artificial intelligence (AI) tools in the creation of this thesis: I disclose that AI tools were employed during the development of this thesis as follows:

- AI-based research tools were used to assist in literature review.
- Generative AI tools were used to debug and improve the code behind the models used in this work.

However, all final writing, synthesis, and critical analysis are my own work. Instances where AI contributions were significant are clearly cited and acknowledged.

Nonetheless, I have ensured that the use of AI tools did not compromise the originality and integrity of my work. All sources of information, whether traditional or AI-assisted, have been appropriately cited in accordance with academic standards. The ethical use of AI in research and writing has been a guiding principle throughout the preparation of this thesis. I understand the importance of maintaining academic integrity and take full responsibility for the content and originality of this work.

Miguel Dinis, 30-06-2025