Erasmus University Rotterdam
Erasmus School of Economics
Master thesis Quantitative Finance

Forecasting the Characteristics of the Implied Volatility Surface for Weekly Options: How do Machine Learning Methods Perform?

Tim van de Noort (570623)

Supervisor:	dr. Freire, G.
Second assessor:	dr. Vladimirov, E.
Date final version:	27th July 2024

Abstract

The implied volatility surface (IVS) is important for many different types of investors. Accurately forecasting the IVS may lead to substantial profits. Through this paper, we aim to expand the literature on weekly options IVS forecasting on the S&P 500 index by using machine learning (ML) methods. A comparison is made between five different methods, of which three ML methods: Elastic Net (ENet), Random Forest (RF), and the Neural Network (NN). We find that the non-linear ML method Random Forest (RF) consistently performs best for the level, slope, and curvature characteristics of the IVS. Additionally, to be able to interpret certain models, we make use of the cumulative sum of squared error difference (CSSED) and the permutation variable importance (VI) metrics.

Keywords: Option pricing, S&P 500 index, European Options, Modeling, Forecasting, Implied Volatility Surface Characteristics, Machine Learning, Random Forest regression, Neural Networks, Long Short-Term Memory (LSTM), Ordinary Least Squares (OLS), Autoregressive model (AR), Elastic Net (ENet), Hyperparameter Tuning, Out-of-Sample R-squared, Root mean squared error (RMSE), Cumulative Sum of Squared Error Difference (CSSED), Permuted Variable Importance (VI)

Contents

1 Introduction

Financial investors have been interested in forecasting the stock market for many years. Accurate forecasts on stock prices can aid investors in making knowledgeable choices and potentially increase the returns on their investments. With regard to forecasting these returns, the use of machine learning (ML) techniques has grown in popularity due to advances in technology and Big Data. In addition, considerable research has focused on applying ML methods to predict stock market trends. However, only a few papers dive into forecasting option prices of 'weeklys' (options that mature in one week or five trading days). Nowadays, these options may be considered one of the most important options as their trading volume covers over 45% of the trading market. Hence, accurate pricing or forecasting weeklys could be highly beneficial to various types of investors. Therefore, the main goal of the research is to fill in this gap in the current literature regarding pricing options.

The approach to the problem is directly via the implied volatilities (IV), which is more practical than using option prices. This is due to the fact that we can easily compare implied volatilities in the cross section, which is harder if we were to be using option prices. As the price of an option has a one-to-one mapping with its implied volatility, we can relate the results of the implied volatility surface (IVS) directly to the options' prices. Many investors use volatility surfaces to develop trading strategies that can lead to more informed decisions and potentially higher returns if applied correctly. The options used in this paper are weeklys on the Standard and Poor index (S&P 500).

There are numerous methods to model implied volatility, and in this study, we explore five of them. The first method is the Ordinary Least Squares (OLS) method. On top of that, we also look at an autoregressive model of order one (AR(1)). These two methods can be considered as reference methods as the more sophisticated models are expected to outperform these relatively simple models. As the title of this paper already suggests, our primary focus is on exploring the performance of several ML methods. We take linear and non-linear ML methods into account to look at whether we can make a distinction in performance for these two different types of ML methods. The linear ML method that we consider in this paper is the Elastic Net method (ENet). Vrontos et al. (2021) strongly advise this method as "it can be used to narrow down the number and identify the important predictors". Finally, the two non-linear ML meth-

ods that are discussed in this paper are the Random Forest (RF) method and an artificial neural network (NN) method. The RF method seems to be performing quite well with large datasets, and is also suggested by many papers, for instance, by Gu et al. (2020). The NN method also performs well in this context and is thus incorporated in the study.

The forecasting procedure will contain three steps: the first step is to calculate the three characteristics of the IVS (level, slope, and curvature). As we have many different options traded for different prices and thus with different IVs, we compute weighted averages on them for each given day based on their moneyness levels and trading volumes. This is discussed more closely in Section 3. Step two is validating the machine learning models. We tune the hyperparameters of the ML models in the predefined period (January 2021 to December 2021). Then finally, in the forecasting period (January 2022 to December 2022), using the trained and tuned models, forecasts are made for the IVS characteristics. We split the data into three different samples, in which the three steps are subsequently executed. For a more comprehensive explanation on this, refer to Section 3.

The main research question that this paper answers is:
How do machine learning methods perform in forecasting the characteristics of the implied volatility surface for weekly type options on the S&P 500 index?

To be able to answer this research question, we construct various sub-questions. Our first sub-question reads: How well do the ML methods perform compared to the non-learning-based models? Secondly, we ask ourselves: Are the findings for the weeklys in line with prior research on long-term IV predictive performance? Additionally, we look at whether the non-linear ML methods (RF and NN) outperform the linear ML method (ENet) or not. Finally, it is also interesting to study the variable selection of the models. These aspects encapsulate the topics that are addressed in this paper.

The results in this paper show that for all three characteristics of the IVS, the Random Forest model outperforms all other models. For the level and curvature characteristics it even outperforms all other models significantly. The non-learning-based models perform relatively well when forecasting the slope characteristic. Moreover, it is observed that the previous values of the

characteristics are the most important for making predictions (lagged variables). Furthermore, we find that the Neural Network machine learning model underperforms as a result of overfitting issues. The findings in this paper are interesting for anyone who wishes to forecast the IVS.

The remainder of this paper is organized as follows. First, in Section 2 we give a brief overview of the literature related to this topic. In Section 3 the data that is used and adjusted for this research is described. In addition, multiple variables are constructed in this section. Subsequently, the methodology is explained in Section 4. After we discuss the methodology, the results can be found in Section 5. Subsequently, we make our final conclusions in Section 6. Finally, the appendices can be found in Appendices A, B, C, and D.

2 Literature

The IVS is of great importance for several groups of investors. Options traders can make use of the IVS by comparing multiple IVs of similar options and thus may find options that are mispriced. Differences in IVs can signal arbitrage opportunities. There are also other types of investors, such as portfolio managers, who can use the IVS to manage the risk associated with other options in their portfolio. Furthermore, the IVS is interesting to market makers who want to ensure that liquidity is provided while minimizing their own risk. Recently, we have observed a change in trading preferences. Namely, nowadays, the short-term options, also known as 'weeklys', account for over 45% of the trading volume (see Almeida et al. (2024)). Hence, accurately forecasting their IVS's can result in large profits. Due to this recent shift in trading preferences, there has not yet been much research on this topic, even though many researchers do write about IVS forecasting for many different options. Most of these researches focus on options that mature relatively far ahead in the future, often 21 trading days (1 month) or more (Almeida et al., 2023; Almeida and Freire, 2022; Christoffersen et al., 2013; Liu et al., 2019). Some articles do examine options that mature within a week (Almeida et al., 2024; Andersen et al., 2017), but not so much on their characteristics. This research aims at expanding the literature on the 'weeklys' IVS modeling.

The IVS can be studied by looking at its characteristics. Chen et al. (2023) splits up the IVS by looking its level, slope, and curvature characteristics. For forecasting these characteristics of the IVS, many methods and models can be used. Nowadays, machine learning methods

are often utilized to make forecasts as they are capable of solving complex problems. Papers as Gu et al. (2020); Medvedev and Wang (2022); Vrontos et al. (2021) make use of different ML models which are also used in this paper. It is also interesting to implement simple models to make a comparison between the non-learning-based models and the ML models. Gu et al. (2020) also uses the simple linear model (OLS), and due to the fact that implied volatilities are highly autocorrelated (Cont and Da Fonseca, 2002), using an AR(1) model is likely to capture most of the persistence. These two models are usually used in this literature for reference purposes.

3 Data

Before being able to analyze the data, we first have to introduce some data-specific variables, such as moneyness, which can be found in the first subsection below. In the second subsection, we dive deeper into a data-specific analysis.

3.1 Variable construction

The Implied Volatility Surface can be considered as a mapping of time t, the strike price of the option K, and the expiration date T plotted against implied volatility. One may also plot the implied volatility against moneyness and time-to-maturity to obtain the IVS. Using these variables is advantageous as they facilitate the comparison of options with varying prices and expiration dates. It is also possible to opt for a stronger measure of moneyness, such as the log-moneyness or even the standardized log-moneyness, as is used in the paper of Almeida et al. (2024). However, in this paper, where the emphasis is primarily on the comparison of various forecasting models, using the basic definition of moneyness is sufficient. We define the basic definition of moneyness as stated in Equation 1.

$$m_t = \frac{S_t}{K} \tag{1}$$

In Figure 1a an example of an IVS is plotted. Firstly, note that in the paper of Cont and Da Fonseca (2002), from where this figure has been captured, moneyness*[1] is defined as K/S_t. Therefore, in our case the IVS can be regarded to as an approximated mirrored plot of this figure. As can be observed in Figure 1a, we conclude that the implied volatility is relatively high when the moneyness* of the option is low, and that it is at its lowest point when the moneyness*

[1]The moneyness that is used in the paper of Cont and Da Fonseca (2002) is referred to as *moneyness** to avoid confusion. Therefore, the conclusions made might not seem consistent but in fact, they are.

of the option is a little bit greater than 1.0.

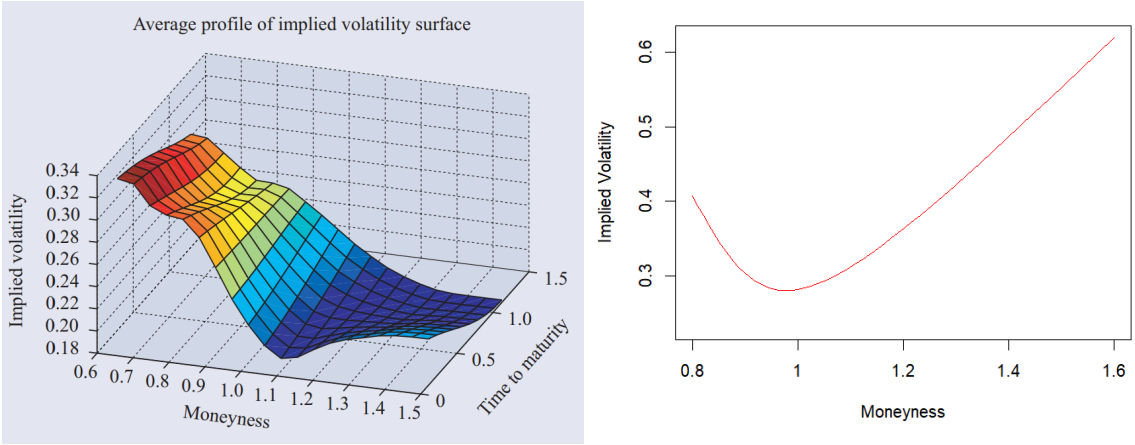
In this paper, we make a distinction between certain values of moneyness. This is mainly done to be able to obtain the characteristics of the IVS. The levels of moneyness that are used are: $m_t \in [0.80, 0.90); [0.90, 0.97); [0.97, 1.03); [1.03, 1.10); [1.10, 1.60]$ for Deep-Out-of-the-Money-Call (DOTMC), Out-of-the-Money-Call (OTMC), At-the-Money (ATM), Out-of-the-Money-Put (OTMP), and Deep-Out-of-the-Money-Put (DOTMP), respectively. In Figure 3, a histogram is plotted to show how often a contract is traded in relation to their value of moneyness. As can be seen, options that are close to ATM option moneyness levels are traded relatively more regularly than, for instance, DOTM call options. Moreover, DOTM put options are traded with a higher frequency compared to the DOTM call options.

As already described in the introduction, the main purpose of this paper is to close the gap in the literature regarding the weekly contract IVS forecasting. Many researchers have done research on longer-term maturity contracts, but, as we can observe in Figure 1a, the IVs for short-term time-to-maturity contracts lie somewhat higher than those of long-term time-to-maturity contracts. As a result of the recent increase in demand for these short-term options, the disparity between IVs for short- and long-term contracts has increased even further. Figure 1b plots the average IVS of the S&P 500 weeklys for the data period of January 2020 to December 2022. Note that this does not look like a surface such as Figure 1a. This is due to the fact that we only incorporate weekly options instead of including all options (with a higher number of days to expiration). Therefore, this third dimension disappears for our analysis. If you compare both Figure 1a and 1b, you do see a likewise 'volatility smile' when only looking at a time-to-maturity of close to 0 years (in Figure 1a). In addition, it can be noticed that IVs are higher for particular moneyness levels than others. Now, just like in the paper of Cont and Da Fonseca (2002), we see that for high levels of moneyness, the IVs are higher than for low levels of moneyness (when using the definition of Equation 1). The main reason for this phenomenon is that OTM put options are higher in demand on short-term notice as they can be used for protection against downside risk. These types of options are often called protective puts or married puts (Merton et al., 1982).

For the calculation of the IV on a specific date for a certain moneyness level, we make use of

an aggregation method. In our dataset, we have many different options which take on different strike prices, trading volumes, etc. Therefore, to be able to obtain a time series of observations which we can work with, we take a weighted average over all IVs for similar[2] contracts that are traded on a specific date with a maturity of at most one week or 5 trading days (weighted on the trading volume; see Equation 5). This results in a time series of weeklys' IVs per level of moneyness. Subsequently, this can be used to construct the IVS, or, as plotted in Figure 1b, an average IVS, where we take the average IVS over the entire sample.

Figure 1: Visualizations of the IVS



(a) Example of an IVS

(b) Average IVS (weeklys)

Note: sub-figure (a) corresponds with the average IVS of the S&P 500 stock at March 1999, where they take the average over the IVS's in that particular month. Also note that the moneyness in this sub-figure is defined as the strike price of the option divided by its current stock price, instead of the other way around (as is done in my paper). Moreover, the *time-to-maturity* shown in the sub-figure is measured in years. This figure is obtained from the paper of Cont and Da Fonseca (2002). Sub-figure (b) corresponds with the average IVS, where we take the average IVS over time (January 2020 to December 2022) for the weekly options.

For the calculation of IV, the well-known Black-Scholes[3] formula for a call and a put options can be used. The BS formula for a Call option is as follows:

$$C(S_t, K, \tau, \sigma) = S_t\Phi(d_1) - e^{-rT}K\Phi(d_2), \tag{2}$$

where

$$d_1 = \frac{\log(\frac{S_t}{K}) + (r + \frac{\sigma^2}{2})\tau}{\sigma\sqrt{\tau}}, \text{ and } d_1 - d_2 = \sigma\sqrt{\tau}, \text{ with } \tau = T - t, \tag{3}$$

and Φ denotes the cumulative distribution function of the standard normal distribution. Conversely, as also pointed out by Wenyong Zhang and Zhang (2023), it is widely recognized that

[2]This is based on the moneyness level of the contracts. If one contract falls within the same level of moneyness as the other, we include it in the moneyness-specific averaging procedure.

[3]also see Black and Scholes (1973) for a more comprehensive explanation on their obtained formula

the Black-Scholes model is incorrectly specified. To obtain the implied volatility, one should solve for σ from Equation 4.

$$C(S_t, K, \tau, \sigma) = C_{market}, \tag{4}$$

where C_{market} corresponds to the observed market price of the call option.

3.2 Data analysis

We perform our analysis on one of the most reliable indicators of overall health and direction of the US stock market: the Standard and Poor index (S&P 500 or SPX). We obtain data of the option metrics on the S&P 500 via OptionMetrics, which is available in the Wharton Research Data Services (WRDS) database[4]. In this paper, only European options are considered. We filter out all options that do not have an implied volatility available, options that have a volume of 0, and options that are too deep out of the money, as is done in Almeida et al. (2023); contracts with a moneyness of $m_t \in (0, 0.80)$ for a call or $m_t \in (1.60, +\infty)$ for a put are left out of the analysis. We only make use of OTM options, as these are relatively more liquid and reliable than their counterparts: the ITM options. For computing the moneyness, we require data on the stock price of the S&P 500. Historical stock price data is available from the Center for Research in Security Prices (CRSP), which is also available in WRDS.

The IVS can be studied by looking at its characteristics. In this paper, we examine three characteristics, namely its level, slope, and curvature, as is done in the paper of Chen et al. (2023). We capture the characteristics by using several different measures. For the level characteristic, we simply look at the average IV of all option contracts traded on a specific date (see Equation 6). For the second characteristic, the slope of the IVS, we consider the measure which is made by taking the difference between the IV of OTM put options and the IV of OTM call options (see Equation 7). Finally, for the third characteristic, the curvature of the IVS, we make use of a measure inspired by Chen et al. (2023) (see Equation 8). These measures represent the characteristics of the IVS for weeklys. This results in a daily dataset of the characteristics of the IVS across the entire dataset, which correspond to our dependent variables.

$$WIV_{t,J_t} = \frac{1}{V_{J_t}} \sum^{j \in J_t} IV_j \cdot V_j, \tag{5}$$

[4] https://wrds-www.wharton.upenn.edu/

$$IV_t^l = WIV_{t,ALL}, \tag{6}$$

$$IV_t^s = WIV_{t,OTMP} - WIV_{t,OTMC}, \tag{7}$$

$$IV_t^c = \frac{WIV_{t,OTMP} + WIV_{t,OTMC}}{2} - WIV_{t,ATM}, \tag{8}$$

where WIV_{t,J_t} represents the weighted average of all options that fall within the moneyness level J_t, where the weights are based on the trading volume per option. V_j represents the volume of option $j \in J_t$. J_t corresponds with the set of all options that have moneyness level J at day t. V_{J_t} is the total volume of the options that fall within a certain moneyness level J on day t.

In Figure 2 the three time series of the characteristics of the IVS are plotted. We can observe a large peak in the time series of the level and the slope characteristic during the COVID-19 recession. In Table 1 the descriptive statistics of the three characteristics of the IVS are shown. We observe that the standard deviation for the level characteristic is the highest, while for the curvature characteristic it is the lowest. Furthermore, we observe that all characteristics exhibit a right-skewed distribution. For a standard Normal distribution, the skewness should be approximately equal to 0 and the kurtosis should be approximately equal to 3. We find that for the level and slope characteristics of the IVS, the skewness is relatively high. This suggests the presence of large outliers, which is also confirmed by the maximum values stated in the table. For the curvature characteristic, we only see a relatively small positive skewness compared to the other two characteristics, which can also be seen when looking at Figure 2. On top of that, the curvature characteristic does not contain many outliers in the time series, resulting in a negative kurtosis. Thus, it can be concluded with confidence that none of the characteristics adhere to a Normal distribution. On the other hand, after performing the Augmented Dickey-Fuller (ADF) test, all three characteristics are found to be stationary based on a 5% significance level for the ADF test. Hence, no adjustments are necessary on the time series to ensure reliable results.

In Equation 9 the variance-covariance matrix ($\widehat{\mathbf{V}}$) and the correlation matrix ($\widehat{\boldsymbol{\rho}}$) are shown, where the first, second, and third row and column correspond with the level, slope, and curvature characteristic of the IVS, respectively. We see that the correlations between the three characteristics are quite low, especially for the slope and curvature characteristic. We also show the partial autocorrelations in Figure 13 in Appendix A. These plots are used in our decision to take lagged values of the IVS characteristics into account.

Table 1: Descriptive statistics of the three characteristics of the IVS

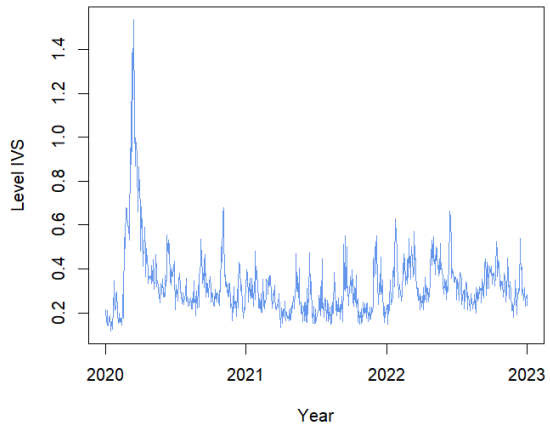
	Mean	Std. Dev.	Max	Min	Skewness	Kurtosis
Level	0.325	0.152	1.538	0.115	16.0	3.13
Slope	0.119	0.057	0.657	-0.031	16.7	2.47
Curvature	0.059	0.028	0.159	-0.055	1.1	-0.11

Note: we display the mean, standard deviation, maximum value, minimum value, skewness, and kurtosis in this table. The three characteristics are calculated as stated in Equation 6, 7, and 8.

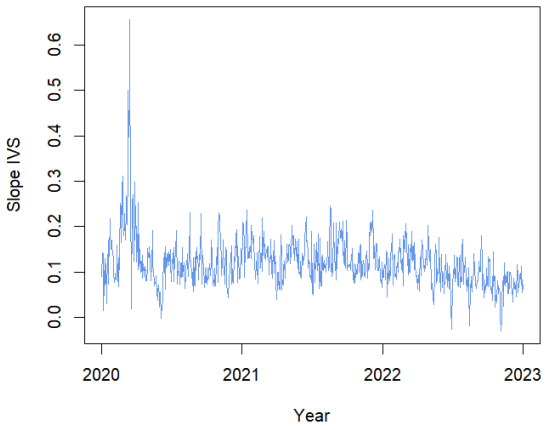
$$\widehat{\mathbf{V}} = \begin{pmatrix} 0.023 & 0.005 & -0.002 \\ 0.005 & 0.003 & 0.000 \\ -0.002 & 0.000 & 0.001 \end{pmatrix} \quad ; \quad \widehat{\boldsymbol{\rho}} = \begin{pmatrix} 1.00 & 0.54 & -0.45 \\ 0.54 & 1.00 & 0.02 \\ -0.45 & 0.02 & 1.00 \end{pmatrix} \tag{9}$$

Figure 2: The time series of the characteristics of the IVS

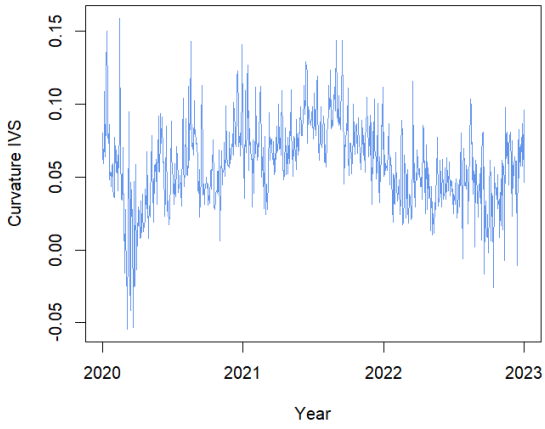
(a) Time series level IVS



(b) Time series slope IVS



(c) Time series Curvature IVS



Note: The level, slope, and curvature characteristics of the IVS are calculated following Equations 6, 7, and 8 respectively.

To assess the predictability of the weeklys' IVS, many variables are used. In this paper, we consider four different types of variables, namely: Option specific variables, macroeconomic variables, S&P 500 related variables, and finally, a cryptocurrency price (Bitcoin). The specific variables used are stated in Table 7 in Appendix A. The macroeconomic variables (including *VIX*) and the Bitcoin variable are obtained from the Federal Reserve Bank of St. Louis[5]. Although *Bitcoin* may appear as an unconventional choice among our variables, its significant insights into market volatility suggest that it could help predict the characteristics of the IVS. While there are also other cryptocurrencies, we only consider Bitcoin as it is the largest by market capitalization. When looking at the properties of the characteristics of the IVS, we find that each one of them experiences high partial autocorrelations (refer to Figure 13 in Appendix C). This implies that the variables are highly persistent and thus it is recommended to take lagged IVS characteristics as explanatory variables into account for the modeling and forecasting procedure.

The data set runs from January 2, 2020, to December 30, 2022. As the stock market is only open on business days, this also results in the data set being based on business days; there are 756 observation dates and due to including lagged variables (1-day-lag), we work with 755 observations in total. This data is split up into three parts, where every split consists of one year (252/252/251 days respectively) of data. The first split is used for estimation, the second split for validation, and the third split is solely used for forecasting evaluation purposes. It is important to note that the second period, after having served as a validation set, is also included in the training set for forecast purposes. As a result, we generate 251 forecasts spanning from January 3, 2022, to December 30, 2022.

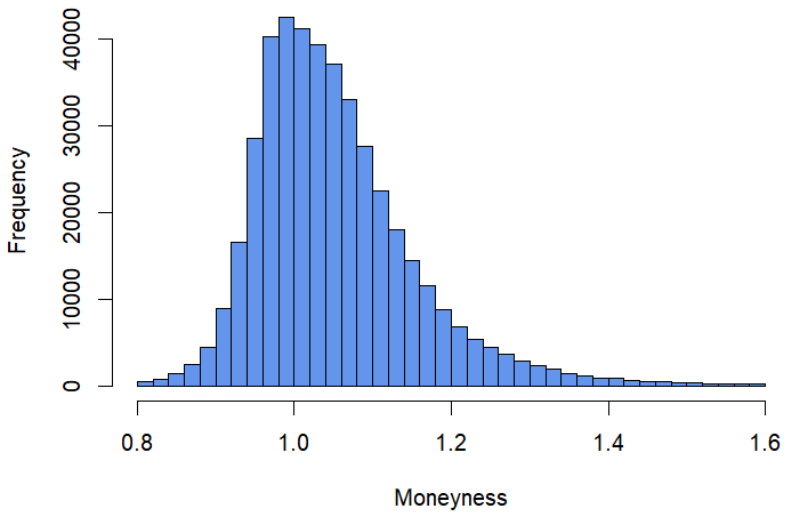
In our dataset, we have a couple of variables for which some data is missing. This is the case for some of the variables that have been collected from the Federal Reserve Bank of St. Louis database. The five variables for which this was the case are shown in bold in Table 7 in Appendix A. To obtain a complete dataset, linear interpolation has been applied to these variables.

As part of this study, several data transformations were necessary to enhance robustness. It is well known that when an explanatory variable is not stable or stationary, this can lead to

[5] https://fred.stlouisfed.org/

overfitting in predictive modeling, where the model fits the noise in the data instead of the underlying patterns. Subsequently, this may reduce the predictive performance of our models. We use the *Augmented Dickey-Fuller* (ADF) test to find out which variables are non-stationary (refer to Section 4 for more information on this test). In Table 7 in Appendix A, the variables that have underwent a transformation are indicated with an 'l', 's', and/or 'd'. 'l' indicates a logarithmic-transformation, 's' indicates a shift-transformation (we shift the variable by the minimal value (+1) recorded in the time series such that we can take the log over that variable), and 'd' indicates a simple first differencing procedure which removes the trend in the time series, making it stationary.

Figure 3: Distribution of the types of option contracts based on their moneyness



Note: this histogram is split up into 40 separate bars; every bar corresponds with an interval of 0.02. Moneyness is defined as in Equation 1.

4 Methodology

In this section, we dive deeper into the methods used to model and predict the IVS. We explore five different methods that are chosen due to their comparatively strong results in previous studies.

4.1 Non-learning-based models

First, we discuss the non-learning-based methods, which are regularly used as reference models. The models that we discuss are the Ordinary Least Squares, and an autoregressive model.

4.1.1 Ordinary Least Squares (OLS)

The first method we examine is the well-known ordinary least squares (OLS) method. We expect this method to perform the worst when it comes to forecasting the IVS as it is only able to capture linear relations between the characteristics and the explanatory variables. In addition, using many variables in the model often leads to overfitting of the in-sample data. As a result, this often leads to inaccurate forecasts. Its mathematical form is as follows:

$$y_t = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + ... + \beta_k x_{kt} + \epsilon_t, \tag{10}$$

where y_t denotes either one of the three IVS characteristics, $x_t = (1, x_{1t}, x_{2t}, ..., x_{kt})$ represents the explanatory variables as listed in Table 7 in Appendix A, $\beta = (\beta_0, \beta_1, \beta_2, ..., \beta_k)$ is the coefficient vector associated with these explanatory variables (and a constant term), and ϵ_t represents the idiosyncratic error component.

4.1.2 Autoregressive model (AR)

The second model that is used as a benchmark, is the autoregressive model of order one, or also called the AR(1) model. This model is also straightforward, yet effective. As implied volatilities often exhibit high serial correlation, this model is very suitable in the context of this paper. In Figure 13 in Appendix C this feature of the IVS characteristics is confirmed. In Equation 11 the mathematical description of the AR(1) model is shown.

$$y_t = c + \phi y_{t-1} + \epsilon_t \tag{11}$$

Generally, this model performs relatively well, but again, as this model uses only linear relations, it is expected that this model does not perform as well as some of the ML methods described in the following subsection.

Before using this model, we first have to check whether the stationarity assumption holds. This is done by using the ADF test, further explained in Section 4.3. The AR(1) model assumes

a constant mean over time (indicated with c in Equation 11) so it cannot adequately model a changing mean, which would lead to poor predictions. If the variance is not constant, this could also lead to unreliable estimates. The main consequence of a non-stationary time series is that it could either lead to spurious results and/or unreliable predictions. Therefore, it is important to ensure that the time series is stationary.

4.2 Machine Learning methods

Moving onward to the most interesting part of this research, the ML methods. Why they are interesting to look at, is due to the fact that machine learning methods are able to capture non-linearity in the data which may not be captured by less complex methods. Hence, using ML methods offers another angle to address the issue at hand. There are two types of ML methods, namely linear- and non-linear ML methods. As this study aims to determine which method provides the most precise forecasts, both types of ML methods are examined. The two non-linear ML methods are the Random Forest method (RF) and the artificial neural network (NN).

4.2.1 Elastic Net (ENet)

First, the Elastic Net method (ENet) is considered. This method is the only linear ML method that we discuss in this paper. Prior research has shown that this method works relatively well when forecasting IVs, see Vrontos et al. (2021). The main reason for this is due to the fact that the ENet method is able to narrow down the number of predictors, while also identifying the most important predictors. This can also be seen in Equation 12. The $|\beta_i|$ penalization term sets certain unimportant predictors exactly equal to zero, which leads to the reduction of predictors. ENet is a combination of LASSO penalization and Ridge Regression (RR) penalization. If α were set equal to zero, the minimization problem would be reduced to the RR model. On the other hand, if α is set equal to one, the minimization problem reduces to the LASSO problem. The α is optimally tuned in the validation set, to obtain the most accurate results. The main advantage of the RR compared to LASSO is that it is less prone to overfitting and that it handles highly correlated predictors better. The main advantage from LASSO is that it shrinks some coefficient to exactly zero, removing them from the estimation process.

$$\hat{\beta} := \arg\min_{\beta} \sum_{t=1}^{T} (y_{t+1} - X_t\beta) + \lambda \sum_{i=1}^{k} (\alpha|\beta_i| + (1-\alpha)\beta_i^2) \tag{12}$$

4.2.2 Random Forest (RF)

The second ML method that we consider is the Random Forest method. This method generally performs relatively well in practice for forecasting time series such as implied volatilities. RF is a robust and flexible ensemble learning technique that utilizes multiple decision trees to make predictions. It is mainly advantageous to use due to its ability to manage different types of data structures and handle them to capture relations within the data.

As an ensemble method, Random Forest constructs multiple decision trees and combines their predictions to improve accuracy and robustness. This results in a reduction of overfitting and simultaneously improves the overall performance of the method. Decision trees are very sensitive to the specific data on which they are trained. Therefore, the RF method uses a technique known as bootstrap aggregation, or bagging for short. What bagging does is that it uses multiple decision trees to make a final estimation/forecast, which leads to robust results. Each decision tree is trained on a different bootstrap sample of the data, where each bootstrap sample is created by randomly sampling from the training data. Furthermore, RF introduces randomness in feature selection by considering a random subset of features at each split in a tree[6]. This helps create diverse trees that are less correlated with each other. Finally, the prediction of the RF is made by taking the average over all predictions made by the individual trees; see Equation 13. This aggregation process reduces variance and enhances the model's accuracy.

$$\hat{y}_t = \frac{1}{N} \sum_{i=1}^{N} \hat{y}_{i,t} \tag{13}$$

In the equation above, $\hat{y}_{i,t}$ represents the forecast made by 'tree i', and N represents the number of trees. As can be seen, this results in an aggregated predicting of all individual tree predictions where every prediction is weighted equally. In Figure 4 the procedure is visualized. Note that the number of trees[7] is tuned and differs from the 600 trees as shown in the figure.

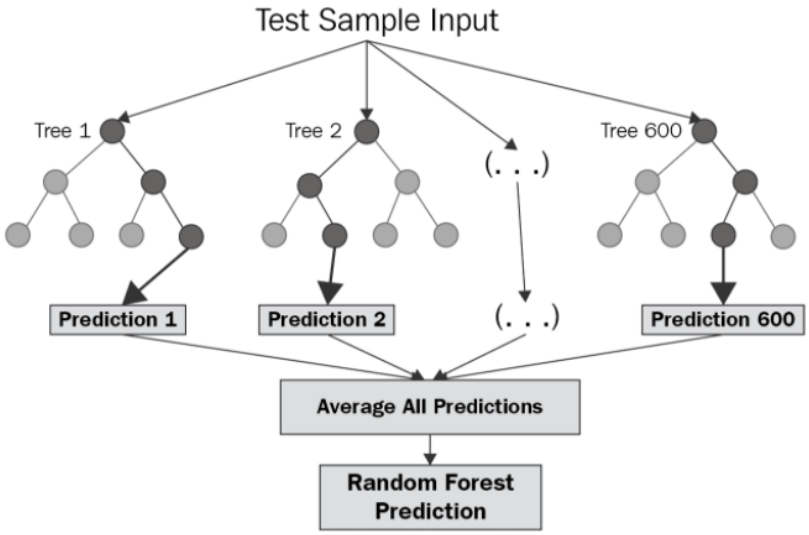
An important aspect of forecasting using the Random Forest method is tuning the hyperparameters. For this research, for the RF method, we tune two parameters, namely the number of features considered for splitting ('mtry'), and the number of trees ('ntree'). In Table 8 in

[6]The number of variables that are considered at each split, which is also tuned in this analysis, is called the 'mtry'.

[7]The number of trees used to perform the forecasting procedure is tuned. The corresponding parameter is 'ntree' (see Table 8 in Appendix A).

Appendix B the specific tuning grid for these two variables can be found.

Figure 4: Visualization of the Random Forest regression method



4.2.3 Neural Networks (NN)

The NN method is the final ML method discussed in this paper. This method is arguably one of the most powerful ML methods, as it is able to capture non-linear relations between variables which may not be captured by other methods. Neural networks have an input layer, one or more hidden layers, and finally, an output layer which generates predictions. There are various forms of neural networks. The simplest form of a neural network is the Feedforward Neural Network (FNN). The FNN method is characterized by the direction of the flow of information between its layers. As the name of the method already suggests, the flow of information is only from the input layer straight towards the output layer. This method is widely used for regression and classification tasks. The functional form of a standard neural network method is stated in Equation 14:

$$\hat{y}_k(X,\beta) = \sigma\left(\sum_j \beta_{kj}^{(l)} h\left(\sum_s \beta_{js}^{(l-1)} h\left(...h\left(\sum_i \beta_{ji}^{(1)} X_i\right)\right)\right)\right), \tag{14}$$

where l is the number of hidden layers, σ and h are the activation functions, β corresponds with the coefficient connected with one layer to another. In this paper we choose to make use of the rectified linear unit (ReLU) activation function. Equation 15 shows the formulation for this activation function. In Figure 5a the parameterization of these variables as well as the FNN

itself is visualized.

$$f(x) = max\{0, x\} \tag{15}$$

An other type of neural networks is the recurring neural network (RNN). RNNs are designed for sequential data and have certain loops which allow the model to maintain information from the past. In this paper, we use the RNN type of neural network to make forecasts. This is because the RNN is more qualified for time series forecasting purposes than the FNN in our case. As we experience high autocorrelations for each one of the IVS characteristics, keeping the time dependence intact is most likely a good idea.

A disadvantage of the RNN method is that it encounters the well-known 'vanishing gradient problem'. This problem is encountered when training the neural network model, and it occurs as the sequence length increases. As this sequence length increases, the gradient magnitude is expected to decrease. When having a long enough sequence length, this may imply that the method cannot be trained. Hochreiter (1998) further analyses this problem and makes some suggestions on how to overcome the problem of vanishing gradients. A method that performed well in his analysis is the advanced NN method called the Long Short-Term Memory (LSTM) method. This method is a type of RNN that is particularly aimed at dealing with the vanishing gradient problem. What it does compared to a normal RNN is that it provides a short-term memory for RNN that helps in training the model. Greff et al. (2016) define the complex construction of different LSTM models, and we refer to this paper for a more comprehensive description on the LSTM method. Given our relatively long sequence length, employing the LSTM method is advisable and thus is also used in this paper. To prevent confusion, note that we indicate the LSTM model by NN from now on.

Although NNs perform relatively well in forecasting, one of the major disadvantages of them is that they are arguably one of the least interpretable machine learning methods due to their complexity. In sub-Figures 5a and 5b a single-hidden-layer FNN and a LSTM block are shown respectively. This also shows the complex interpretability of the NN as we do not know exactly what happens in the hidden layers.

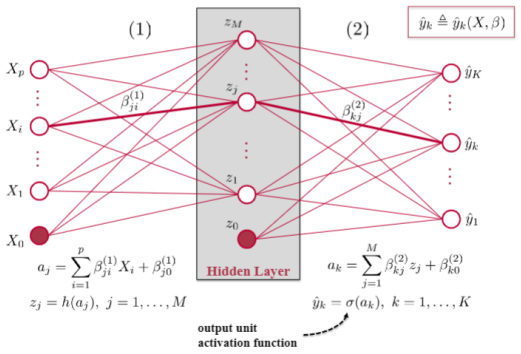
Another disadvantage of neural networks is that they tend to be heavily parameterized. As a result, overfitting is a serious danger. It can be prevented by having a dropout rate, using reg-

ularization methods, early stopping (stop training if the performance drops), or a simplification of the model (by reducing the amount of hidden layers). In this paper, we use the dropout rate of 0.2, L1 and L2 regularization, and take neural networks with at most 5 hidden layers into consideration (also see Table 8 in Appendix B).

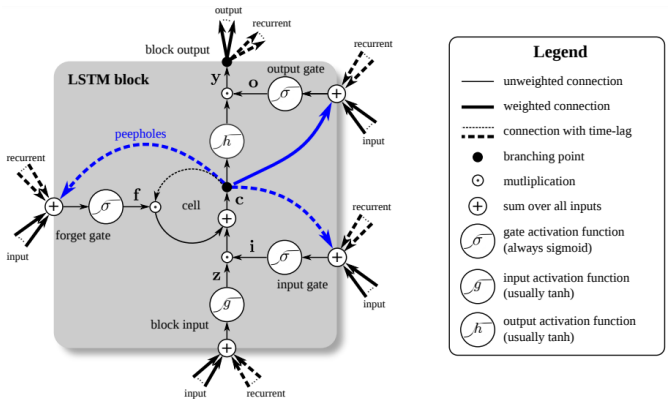
Previous research has shown that the non-linear ML methods RF and NN perform well in this context (Gu et al., 2020). Therefore, the main focus of this thesis lies on these models.

Figure 5: Visualization of the different neural network models

(a) Feedforward Neural Network (FNN)



(b) Long Short-Term Memory (LSTM)



Note: in these sub-figures we visualize and parameterize the feedforward neural network (a) and the LSTM (b). Refer to Greff et al. (2016) for a more comprehensive explanation on the visualization of the LSTM model. Also note that we use the ReLU input activation function (see Table 8 in Appendix B) instead of the suggested tanh activation function as stated in the legend.

4.3 Stationarity

To test whether a variable is stationary or not, we make use of the augmented Dickey-Fuller (ADF) test (Dickey and Fuller, 1979). The ADF test tests the null-hypothesis that there is a unit root in the time series in question. Rejecting this null-hypothesis implies that the time series in question is stationary. The functional form of the ADF test is as follows.

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + ... + \delta_{p-1} \Delta y_{t-p+1} + \epsilon_t, \tag{16}$$

where α is a constant, β the coefficient of a time trend, and p the lag order of the autoregressive process, but most importantly γ is the coefficient which is used to obtain the test statistic (see Equation 17).

$$T = \frac{\hat{\gamma}}{SE(\hat{\gamma})} \tag{17}$$

If the Dickey-Fuller T-statistic is significant, this implies that we should reject the null-hypothesis and thus conclude that the time series is stationary.

4.4 Evaluation measures

In this paper we make use of two different forecasting performance evaluation metrics. The percentage *Out-of-Sample R-squared* (R^2_{oos}) and the percentage *Root Mean Squared Error* (RMSE) are used. For parameter tuning, we also mainly make use of the RMSE to assess which parameter performs best in the validation set. These two measures represent how well our models are performing and are easy to compare. The higher the R^2_{oos}, the better the performance of the model. For RMSE, lower values indicate better performance. The main difference between the two models is that one (R^2_{oos}) compares the errors of the model with those of a benchmark model, while the other measure (RMSE) compares only the forecasts with the actual values. The two measures are calculated as formulated in Equation 18 and 19.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i}(y_i - \hat{y}_i)^2} \tag{18}$$

$$R^2_{oos} = 100\% \times \left(1 - \frac{MSE_{model}}{MSE_{bench}}\right) \tag{19}$$

In this paper, we make use of a historical moving average as the benchmark for the R^2_{oos}. We construct the forecasts of this benchmark model as stated in Equation 20.

$$\hat{y}_t^{j,bench} = \frac{1}{252}\sum_{i=t-253}^{t-1} IV_i^j, \tag{20}$$

where j indicates the type of characteristic: level, slope, or curvature. As also can be noticed in this equation, we use a historical moving average with a range of one year (252 trading days). This is mainly due to the fact that during the Covid-19 crisis, the values for the characteristics of the IVS (mainly the level and slope characteristics) were not representable compared to the more recent years. This can be observed in Figure 2.

The Diebold-Mariano statistic (DM) can be used to find out whether a certain model '1' significantly outperforms another model '2'. In Equation 21, we start by computing the squared errors for model 1 ($e^2_{t,1}$) and model 2 ($e^2_{t,2}$). Next, we find the difference between these squared errors, resulting in the error difference measure (d_t). A negative value of d_t indicates that model 1 out-

performs model 2, as a small forecast error is obviously desirable. In contrast, a positive value d_t implies that model 2 is superior. The DM-statistic is further calculated by using Equation 22.

$$d_t = e_{t,1}^2 - e_{t,2}^2. \tag{21}$$

$$DM = \frac{\bar{d}}{\sqrt{V(d)}}. \tag{22}$$

Here, $\bar{d}$ represents the average and the V(d) corresponds with the variance of the error differences time series (d).

As an additional layer to this research, it is interesting to find out why some models outperform others. For machine learning models, it can sometimes be difficult to interpret why a certain model outperforms another model. Therefore, to give some sort of interpretation behind the results in this paper, we introduce the relative variable importance measure. The measure that is used to obtain the relative variable importance is the so-called permutation Variable Importance (VI). This measure is obtained by measuring the difference between prediction errors before and after one variable is permuted. The higher the difference, the more important the variable, as a high difference implies that leaving the variable out of the equation results in a higher prediction error than before. In this paper, we calculate the measure in a relative way (all VI's add up to 1) so that they are easier to interpret. For the Random Forest model specifically, the importance measured via 'IncNodePurity', which is also a measure of how much the model error increases when a particular variable is randomly permuted or shuffled (see Tohry et al. (2020)). Hence, since we use the same measure of VI for each model, we can also compare them with each other.

Finally, the last measure we use is the *Cumulative Sum of Squared Error Difference* (CSSED). This measure can help with interpreting forecast results as well as comparing them with other models. For example, using this metric allows us to contrast the model of interest with a basic model. This comparison can reveal specific times at which one model outperforms or underperforms relative to the other, as highlighted by the measure. This measure is mainly interesting if one model performs unexpectedly poorly or exceptionally well. We calculate the metric as

described in Equation 23.

$$CSSED_t = \sum_{i=505}^{t} \left((y_i - \hat{y}_i^A)^2 - (y_i - \hat{y}_i^B)^2 \right), \tag{23}$$

where $\hat{y}_i^X$ stands for the forecast made by model X for the time-index i. Note that we use $i = 505$ as a starting point as this index corresponds with the first iteration of the forecasting period for which we calculate the CSSED. Thus, a negative value for CSSED implies that the squared errors obtained by the forecasts made by model A are smaller than those of model B. In contrast, if the CSSED is positive, it means that model B's squared errors are smaller than those of model A.

4.5 Tuning Parameters

Hyperparameter tuning is beneficial as it enhances overall predictive performance: it ensures that the model is not too simplistic, which could result in a high bias, or that the model is too complex, leading to a high variance. In addition, the tuning of the hyperparameters contributes to the robustness of the model, ensuring that the predictions remain reliable under different conditions. In Table 8 in Appendix A, the tuning grid for the elastic net, random forest, and neural network models is reported.

We must also note that we can not use the usual K-fold Cross Validation (k-CV) for the tuning procedure, as it randomly splits the data into k subsets (folds) and shuffles them, breaking the temporal order of the data. When the data has trends, the temporal order of the data is crucial for making accurate forecasts. Instead, we use the so-called walk-forward validation. This method does maintain the temporal structure of the data.

5 Results

This section is split up into two parts. In the first part, the main forecast results are discussed and evaluated. We make use of the models stated in Section 4, tune them where required, and subsequently make forecasts on each one of the three characteristics of the IVS. In the second part, we dive deeper into why certain methods perform better or worse than others. We look at the variable selection of the models via the permuted variable importance (VI) measure and also take the cumulative sum of squared error difference (CSSED) into account.

5.1 Forecasting Results

Before we dive deeper into the evaluation of the forecast results, it is important to note that the ML methods for which the results are shown represent the models with optimally tuned hyperparameters. Note that for the ENet method, the second most optimal α is also included, where ENet (1st) and ENet (2nd) indicate the ENet model with the most optimal and the second most optimal α during the tuning part. The hyperparameter tuning grid that is used for this research can be found in Table 8 in Appendix A.

In Table 2 and Table 3 the main forecast results of this research are shown. In these tables, we show the forecast accuracy per model per characteristic. In Table 2 the forecast accuracy is reported while using the percentage Out-of-Sample R-squared measure. For all values, we make the comparison with the model stated in the first column with the historical moving average as the benchmark model (refer to Equation 20). Hence, by using this measure, we can also directly see whether the methods outperform the benchmark model in forecasting. If the R^2_{oos} gives a negative value, this indicates that the *MSE* of the benchmark model is smaller than the *MSE* of the model in question (refer to Equation 19). There is a single instance where this occurs; specifically, for the slope characteristic, the NN method performs quite poorly, even worse than the historical moving average forecasts. All other methods manage to outperform the benchmark model. In Table 3 we show the percentage RMSE. In Table 4, 5, and 6 the corresponding DM statistics are reported.

Table 2: Forecasting accuracy in terms of Out-of-Sample R-squared (%)

	Level	Slope	Curvature
OLS	68.2%	27.2%	21.3%
AR(1)	61.5%	28.8%	39.9%
ENet(1st)	62.1%	19.0%	33.3%
ENet(2nd)	64.6%	10.1%	33.2%
RF	75.2%	29.4%	56.4%
NN	69.1%	-6.4%	23.5%

Note: The values shown above are the percentage Out-of-Sample R^2's where we compare each model's performance with the historical moving average as a benchmark (refer to Equation 19 and 20). ENet(1st) and ENet(2nd) correspond with an elastic net model with the optimal α and the second-best α during the tuning procedure.

Table 3: Forecast accuracy in terms of RMSE (%)

	Level	Slope	Curvature
OLS	5.91%	3.66%	2.57%
AR(1)	6.50%	3.62%	2.25%
ENet(1st)	6.45%	3.86%	2.37%
ENet(2nd)	6.23%	4.07%	2.37%
RF	5.22%	3.61%	1.91%
NN	5.82%	4.43%	2.53%

Note: ENet(1st) and ENet(2nd) correspond with an elastic net model with the optimal α and the second-best α during the tuning procedure.

Now, let us take a closer look at the values reported in Table 2 and 3. Firstly, the level characteristic. The model that performs worst is the AR(1) model. This model obtains a R^2_{oos} of 61.5% and a RMSE of 6.50%. Although this result might be surprising, it highlights the other models to perform really well. It was expected for the AR(1) model to perform quite well due to the fact that the level characteristic of the IVS exhibits high autocorrelations (see Figure 13a in Appendix C). Thus, the fact that all other models outperform the AR(1) model is surprising but also interesting. When looking at the variable importance measure in the next subsection (Figure 9), it is also confirmed that for all models the lagged level variable is of great importance, indicating that these models do well by forecasting with the lagged level characteristic.

The first two models that outperform the AR(1) model are the two elastic nets. The optimally tuned elastic net (ENet(1st)) obtains a R^2_{oos} of 62.1% and a RMSE of 6.45% while the second-best elastic net obtains an even higher R^2_{oos}, namely 64.6%, and thus also a lower RMSE, 6.23%. It is interesting that the model that performs second-best for the tuning part outperforms the optimal model in the tuning part. The optimally tuned α corresponds with an α of 0.2 for the level characteristic, followed by an α of 0.0 (based on RMSE). For the calibration of the optimal α, we did not only use the RMSE metric, but also looked at how often each elastic net model outperformed all others. For the latter metric we found that the ENet(0.2)[8] outperformed all other models only 18 times out of the 252 observations, while the ENet(0.0) outperformed all other models 103 times out of the 252 observations. This suggests that the ENet(0.0) is con-

[8]We indicate an elastic net model with a certain α by ENet(α) from now on.

sistently outperforming the ENet(0.2) model, but that the ENet(0.0) model's forecasts during the tuning procedure has had some extreme outliers. By using the CSSED metric, we find that this is indeed the case. In Figure 8 the corresponding CSSED time series are shown, where ENet(0.2) corresponds to model A and ENet(0.0) to model B in Equation 23. In sub-Figure 8a we see that just before the end of November 2021 the ENet(0.0) model fails to make accurate forecasts compared to the ENet(0.2). This confirms that the ENet(0.0) model might perform better in the long run, and therefore, we also take this second-best ENet model into account for the forecast comparison. In the end, we see this indeed being the case, ENet(0.0) (insignificantly) outperforming ENet(0.2). In sub-Figure 8b the forecasting performance is shown in terms of CSSED where it can be observed that the second-best elastic net is more accurate over time.

The next best model is the OLS model which obtains an R^2_{oos} of 68.2% and a RMSE of 5.91%. As expected, the two most accurate models are the ML models, where the RF model, with an R^2_{oos} of 75.2% outperforms the NN model (with one hidden layer), with an R^2_{oos} of 69.1%. Thus, the RF model is the model with the most accurate forecasts for the level characteristic . When looking at the differences in RMSE, we also see that this difference is relatively large. It is confirmed by the DM statistics that the RF model significantly outperforms the NN based on a 5% significance level, with a t-statistic of 2.30.

Table 4: Diebold-Mariano test statistics for the level characteristic

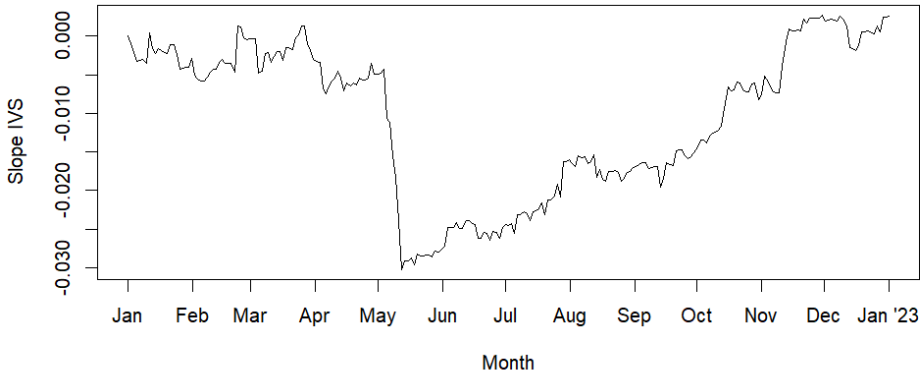
	OLS	AR(1)	ENet(1st)	ENet(2nd)	RF	NN
OLS	-	1.60	1.76*	1.42	-2.30**	-0.43
AR(1)	-1.60	-	-0.21	-0.70	-4.06***	-1.84*
ENet(1st)	-1.76*	0.21	-	-0.74	-3.86***	-2.07**
ENet(2nd)	-1.42	0.70	0.74	-	-3.14***	-1.54
RF	2.30**	4.06***	3.86***	3.14***	-	2.34**
NN	0.43	1.84*	2.07**	1.54	-2.34**	-

Note: We compare all models with each other. In this case, the models stated in the header row of the table are used as 'model 1', and the corresponding models stated in the first column are used as 'model 2' (see Equation 21). Thus, this implies that whenever the DM statistic is negative, the corresponding model in the first column performs worse than the corresponding model in the header row, and vice versa. *,**, and *** represent with the significance levels 10%, 5%, and 1% respectively. Thus, * indicates $p \in (0.05, 0.10]$, **: $p \in (0.01, 0.05]$, and finally, *** a p-value smaller than 0.01. On top of that, we make use of the two-sided DM test.

For the slope characteristic, we observe some different outcomes compared to the performances for predicting the level characteristic. The model that performed the worst for the level characteristic, the AR(1) model, now outperforms all other models, except for the RF model, with a R^2_{oos} of 28.8% and a RMSE of 3.62%. This is interesting to see as the first autocorrelation for the slope characteristic showcased in sub-Figure 13b in Appendix C is smaller than the value for the first autocorrelation for the level characteristic. When using this result combining it with the variable importance measure in the next subsection (Figure 10), we see that even though for all models the lagged variable is the most important, the AR(1) model still manages to outperform four out of the five other models. This indicates that these four models suffer from overfitting.

The one model that does manage to outperform the AR(1) model is, again, the RF model. The RF model namely obtains an R^2_{oos} of 29.4% and a RMSE of 3.61% resulting in a difference of only 0.01% in RMSE. When looking more closely at this result via the CSSED, plotted in Figure 6, we observe a very poor performance for the RF model around the beginning of the month May (2022). The same holds for the OLS model (see Figure 14 in Appendix D). From then on, one can observe a positive trend of the RF outperforming the AR(1) model, leading up to a small outperformance in accuracy based on their final RMSEs. As a result, the RF only outperforms the AR(1) model insignificantly for this characteristic.

Figure 6: CSSED of the AR(1) model compared to the RF model for the level characteristic



Note: In this figure the dates run from January 2022 to the end of December 2022. We compare the forecasts of the AR(1) model with those of the RF model. Thus, we use AR(1) as model 'A' and RF as model 'B' in Equation 23.

The worst performing model is the NN model. Its value for the R^2_{oos} reported in Table 2 is even negative (-6.4%), which indicates that the benchmark model, the historical moving average model, outperforms the NN model. This is likely to be due to the variable selection of the model. In the following subsection, we will discuss more on this topic. The elastic net models

outperform the NN significantly, but all other models are also significantly outperforming the elastic nets. Finally, we again see that the OLS model performs relatively well, ranking as the third most accurate model behind the AR(1) and RF models.

Table 5: Diebold-Mariano test statistics for the slope characteristic

	OLS	AR(1)	ENet(1st)	ENet(2nd)	RF	NN
OLS	-	-0.31	1.66*	3.52***	-0.61	4.46***
AR(1)	0.31	-	2.18**	3.53***	-0.12	5.14***
ENet(1st)	-1.66*	-2.18**	-	3.72***	-2.66***	6.72***
ENet(2nd)	-3.52***	-3.53***	-3.73***	-	-4.48***	3.46***
RF	0.61	0.12	2.66***	4.48***	-	5.55***
NN	-4.46***	-5.14***	-6.73***	-3.46***	-5.55***	-

Note: We compare all models with each other. In this case, the models stated in the header row of the table are used as 'model 1', and the corresponding models stated in the first column are used as 'model 2' (see Equation 21). Thus, this implies that whenever the DM statistic is negative, the corresponding model in the first column performs worse than the corresponding model in the header row, and vice versa. *,**, and *** represent with the significance levels 10%, 5%, and 1% respectively. Thus, * indicates $p \in (0.05, 0.10]$, **: $p \in (0.01, 0.05]$, and finally, *** a p-value smaller than 0.01. On top of that, we make use of the two-sided DM test.

Finally, for the third characteristic of the IVS, curvature, the results are again interesting. We see that the OLS model, which performed relatively well for the level and slope characteristics, now gives the least accurate forecasts, with an R^2_{oos} of 21.3% and a RMSE of 2.57%. The main reason for this to be the case is that the model encounters a large forecasting error. This is discussed more closely in the next paragraph. The model that is again performing very poorly, is the NN model with an R^2_{oos} of 23.5% and a RMSE of 2.53%. Next up are the elastic nets with an R^2_{oos} equal to 33.3% and 33.2%, and an RMSE equal to 2.37% for ENet(1st) and ENet(2nd) respectively. Hence, these models do not differ in performance as much as they did for forecasting the level and slope characteristic. The second most accurate model is the AR(1) model with an R^2_{oos} of 39.9% and a RMSE of 2.25%. Finally, the model that performs the best is the RF model. The RF model significantly outperforms all other models for the curvature characteristic forecasting procedure.

When examining the DM statistics in Table 6, we observe something interesting, namely, the statistic for RF versus OLS is relatively low (1.73) in comparison to the statistics for RF against all the other models (for instance, 4.26 for the AR(1) model). This might seem out of the order

as the OLS model exhibits the lowest accuracy metrics R^2_{oos} and RMSE for the curvature characteristic as it is often the case that when these metrics are higher for a model 'X' compared to a model 'Y', then the DM statistic between a model 'Z' versus 'X' is greater than that of the model 'Z' versus 'Y' (given that model 'Z' outperforms models 'X' and 'Y' both)[9]. However, this instance does not follow that pattern. Therefore, we inspect the CSSED of the OLS model compared to the RF. In Figure 7a the CSSED is plotted[10] in which we observe that the OLS model is mainly performing poorly due to a couple observations. We see that the value of the CSSED suddenly surges at June 13th. As a result, we find a large difference in forecasting accuracy, while having a relatively small value for the DM statistic. This implies that the OLS model does not generally produce poor forecasts but suffers from a few poor forecasts. When looking at the CSSED of the NN compared to the OLS (see Figure 7b), we also see that the OLS model is consistently forecasting more accurately than the NN model, except for the forecasts around June 13th 2022. Therefore, only looking at the accuracy measures R^2_{oos} and RMSE gives a distorted picture of what is really happening. In the next subsection this result is also related to the variable importance measure to give some further insights.

Table 6: Diebold-Mariano test statistics for the curvature characteristic

	OLS	AR(1)	ENet(1st)	ENet(2nd)	RF	NN
OLS	-	-0.92	-0.90	-0.89	-1.73*	-0.11
AR(1)	0.92	-	0.85	0.86	-4.26***	3.35***
ENet(1st)	0.90	-0.85	-	0.80	-2.92***	1.24
ENet(2nd)	0.89	-0.86	-0.80	-	-2.94***	1.23
RF	1.73*	4.26***	2.92***	2.94***	-	7.58***
NN	0.11	-3.35***	-1.24	-1.23	-7.58***	-

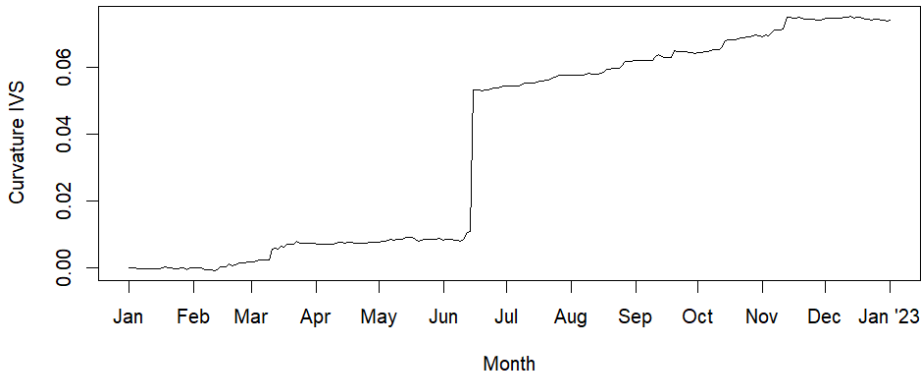
Note: We compare all models with each other. In this case, the models stated in the header row of the table are used as 'model 1', and the corresponding models stated in the first column are used as 'model 2' (see Equation 21). Thus, this implies that whenever the DM statistic is negative, the corresponding model in the first column performs worse than the corresponding model in the header row, and vice versa. *,**, and *** represent with the significance levels 10%, 5%, and 1% respectively. Thus, * indicates $p \in (0.05, 0.10]$, **: $p \in (0.01, 0.05]$, and finally, *** a p-value smaller than 0.01. On top of that, we make use of the two-sided DM test.

[9] In this case, model 'X' can be seen as the OLS model, 'Y' as some other model, let us say the AR(1) model, and 'Z' as the RF model.

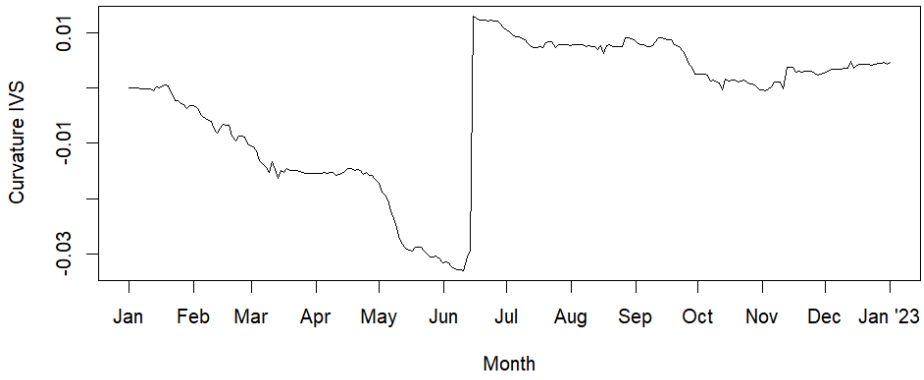
[10] where the OLS model corresponds with model 'A' and the RF model with model 'B' in Equation 23.

Figure 7: CSSED of the OLS model compared to the RF and the NN models for the curvature characteristic

(a) CSSED OLS-RF curvature



(b) CSSED OLS-NN curvature



Note: In these sub-figures the dates run from January 2022 to the end of December 2022. We compare the forecasts of the OLS model with those of the RF (a) / NN (b) model. Thus, we use OLS as model 'A' and RF/NN as model 'B' in Equation 23.

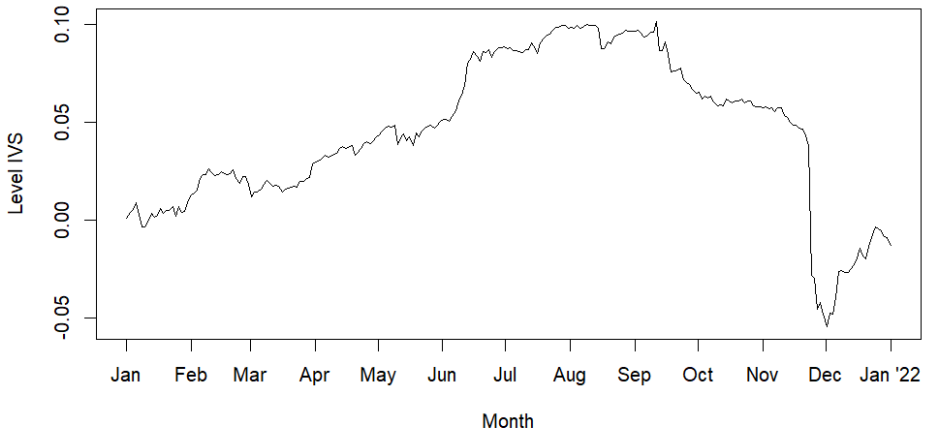
It is also interesting to mention the differences in RMSEs. As previously mentioned, the curvature characteristic exhibits low volatility (see Table 1), which is also evident in Table 3. Specifically, it is expected for the RMSE to be small if the time series in question exhibits low volatility. In our case the level characteristic shows the highest volatility, the slope the second highest, and the curvature the lowest. We observe this result returning in our findings for the RMSE; the RMSEs are highest for the level characteristic, the second highest for the slope characteristic, and the lowest for the curvature characteristic. This result is not observed for the R^2_{oos} as this measure compares the models to a benchmark model, which apparently performs relatively well for the slope characteristic, resulting in relatively low R^2_{oos}'s compared to the other characteristics.

By examining at the Diebold-Mariano statistics we find some more interesting results. The

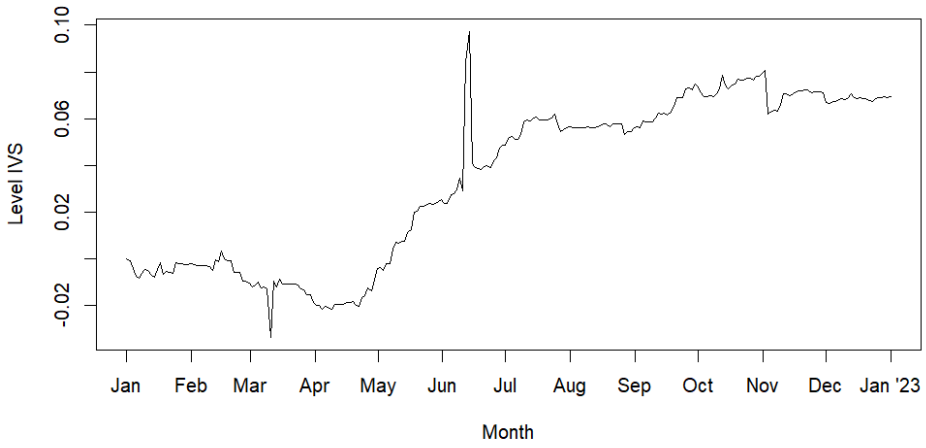
main result we find in these tables is that the RF model significantly outperforms all models with a significance level of at most 10% except when comparing the accuracy of the OLS and the AR(1) model for forecasting the slope characteristic. It still manages to outperform them, but not significantly. We also find that the NN ML model performs worse than expected for the slope and curvature characteristic. Only for the level characteristic it manages to make relatively accurate forecasts. It even significantly outperforms the AR(1) and the ENet(0.0) model for this characteristic of the IVS (or indicated as ENet(1st) in Table 4).

Figure 8: CSSED of the two ENet models for the level characteristic

(a) CSSED ENets tuning



(b) CSSED ENets forecasting



Note: In these sub-figures the dates run from January 2021 (2022) to the end of December 2021 (2022) for sub-figure a (b). We compare the two elastic nets with each other: ENet(0.2) and ENet(0.0), where ENet(0.2) corresponds with model 'A' and ENet(0.0) with model 'B' in Equation 23.

5.2 Variable Importance

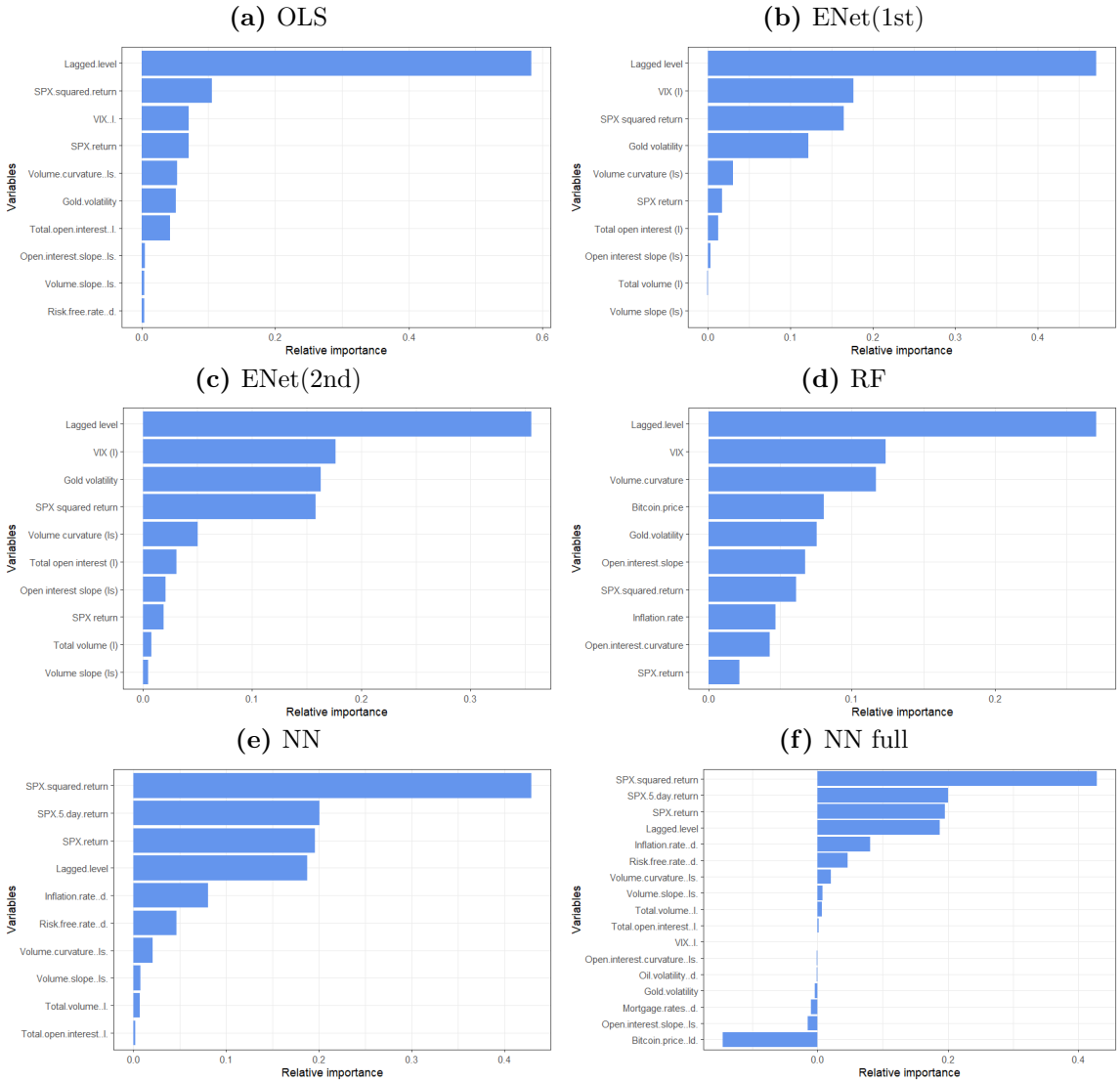
For the NN model, it is hard to interpret the results by only looking at the R^2_{oos} and the RMSE. Therefore, we also study the variable selection made by the models used in this paper. We do this by calculating the permutation variable importance (VI) measure for all models, except for the AR(1) model. This is due to the fact that the AR(1) model only takes the first lagged characteristic into account, and thus the VI measure would not make much sense. It should also be noted that for the VI of the NN model the total VI shown in the sub-figures with the top ten most important variables is larger than 1. Although this may seem counter intuitive, we encounter certain variables that obtain a VI smaller than 0. A variable obtains a negative value for the relative VI if it not only fails to contribute but also harms the model's accuracy. In other words, if the variables for which negative values of VI are obtained would be removed from the dataset, the model would give more accurate forecasts. As this is only the case for the NN models, this implies that the NN models suffer from overfitting, fitting the errors instead of the underlying time series.

We plot the top ten most important variables for each model in Figures 9, 10, and 11. Alternatively, in sub-Figures 9f, 10f, and 11f we include all variables to show the VI of each variable, also illustrating the negative values for certain variables. For the level and the curvature characteristic, the largest negative value of VI is obtained by the variable *Bitcoin price*. While there are also some other variables for which this is the case, the difference between the *Bitcoin price* variable and the others is significant. For the slope characteristic, this is primarily the *SPX return* variable. We see that this variable obtains a very large negative value. This implies that the variable is 'hurtful' for the forecasts and that the model performs better when not including it. For all other models, we do not find any negative values.

When analyzing the results for the level characteristic, we find some interesting results. Namely, for the NN model, the four most important variables are the three SPX-related variables and the *Lagged level*. All models agree on the latter variable, but not so much on the SPX-related variables. We see for the RF model, which significantly outperforms the NN model, that *SPX squared return* obtains only the seventh largest value of VI. However, for all other models, the corresponding VI measure is relatively high. This indicates that this variable is definitely useful, but not the best one to rely on. For the *SPX 5-day return* variable, we even see that it is not

in the top ten most important variables for all other models than the NN. Despite this, the NN model performs comparatively well, being more accurate all other models, with some even being outperformed significantly.

Figure 9: In these sub-figures the relative importance per variable are shown for forecasting the level characteristic. Note that we only plot ten variables that have the highest importance.

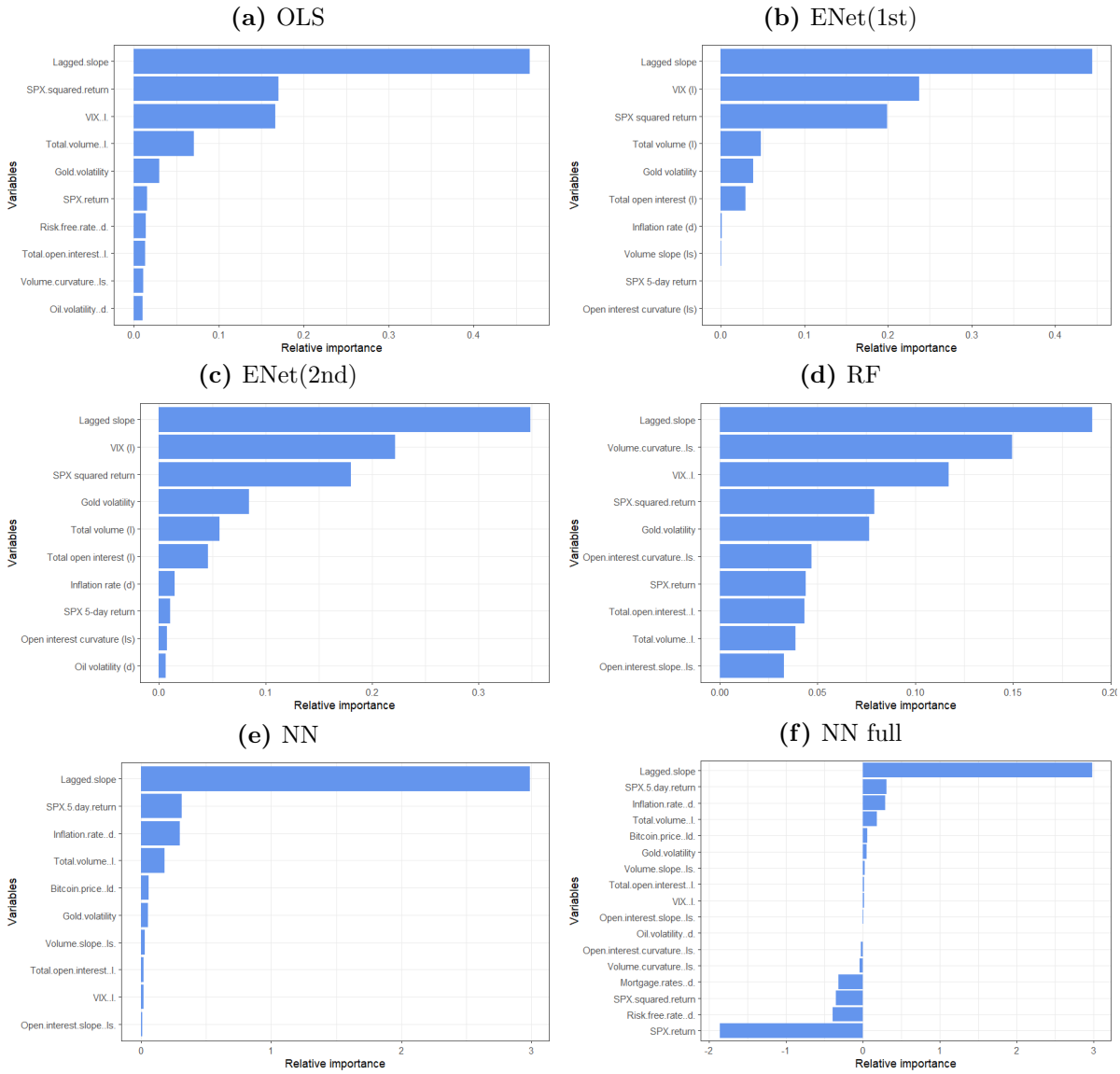


Note: we show the ten variables which have obtained the highest values, except for sub-Figure (f) in which all variables' VIs are reported.

When comparing models based on their accuracy and the importance of the variables used in their forecasts, some other interesting results are found. Firstly, for the NN model, we find very poor results for the slope and curvature characteristics. When looking at which variables are important for this model for forecasting the slope characteristic, the model is heavily hurt by the variables *SPX return*, *Risk-free rate*, *SPX squared return*, and *Mortgage rates*. On the other hand, when looking at what the well-performing models, such as RF or OLS, make use

of, we observe that the variable *SPX squared return* is of relatively great importance for their accurate forecasts. This indicates that the NN model is not able to successfully make use of this variable variable. On top of that, the NN model its second most important variable is the *SPX 5-day return*. For the OLS, ENet(1st), and RF model, this variable is either of very small importance to no importance for the model. For the best elastic net model (ENet(1st)) this variable obtains a coefficient equal to zero, so that this variable is not even taken into account. Only the second-best elastic net model takes this variable into account, but as it is outperformed by all other models (except for NN), this suggests that the variable is not very helpful for the final forecasts.

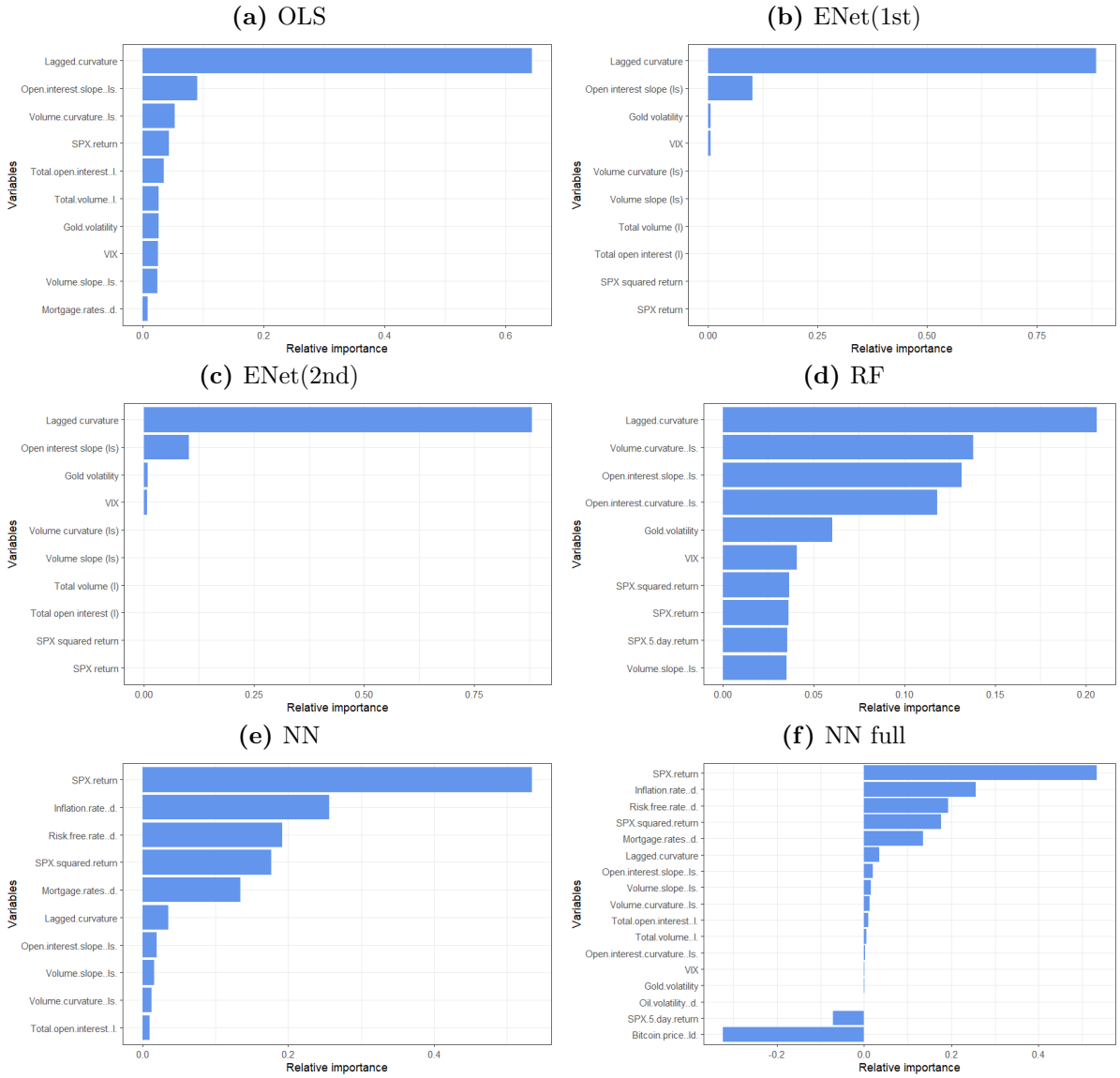
Figure 10: In these sub-figures the relative importance per variable are shown for forecasting the slope characteristic



Note: we show the ten variables which have obtained the highest values, except for sub-Figure (f) in which all variables' VIs are reported.

Looking more closely at the variable importance for the NN model, we see that the *SPX return* variable obtains a very high negative value (-1.86). This indicates that the variable has been extremely detrimental to the forecasts. Compared to the other negative values that the NN obtains for the level and curvature characteristics, we see that the magnitude of these values is very high. This implies that the the *SPX return* variable introduces relatively more noise to the forecasts than for the other forecasts for this model. Hence, we can conclude with certainty that the model suffers from overfitting. This is also the case for the other characteristics. We can see this result reflected in the poor accuracy metrics.

Figure 11: In these sub-figures the relative importance per variable are shown for forecasting the curvature characteristic

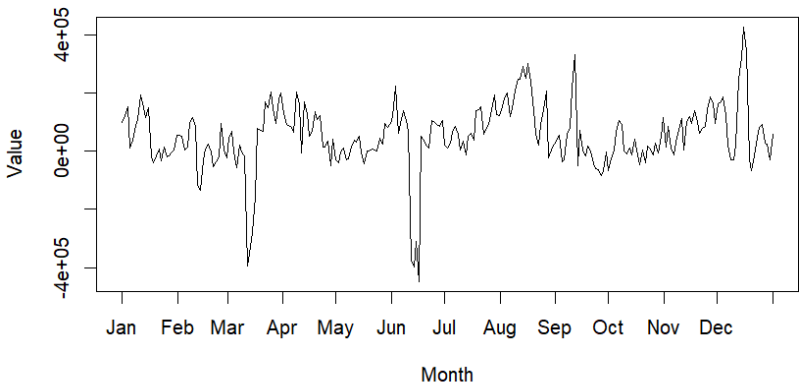


Note: we show the ten variables which have obtained the highest values, except for sub-Figure (f) in which all variables' VIs are reported.

When analyzing the results for the curvature characteristic, we immediately observe that one particular variable is of great importance for all models, except for the NN model, namely, *Lagged curvature*. Almost all models are more accurate than the NN model, except the OLS model. However, as mentioned in the previous subsection, this model shows low accuracy due to a few outliers in forecast errors. These large forecast errors are mainly caused by the *Open interest slope* variable (in the next paragraph we discuss more on this). Hence, the reason why the OLS model does not perform as well as the others is not due to the inclusion of the *Lagged curvature* variable. Therefore, as the *Lagged curvature* variable is of great importance for all models, and they all perform relatively well by using this variable, we can conclude that it is recommended to take this lagged variable into account.

As just mentioned, the *Open interest slope* variable causes the forecast metrics corresponding to the OLS model to be relatively poor. We see that this is the case when combining the CSSED for the OLS model against the RF model (Figure 7a) with the time series of the *Open interest slope* variable (Figure 12). We see a large peek in the CSSED on June 13th (2022), indicating that the OLS model gives a very poor forecast at that date compared to the RF model. We also see a large drop in the time series of the variable *Open interest slope*. When relating these results to the variable selection of the OLS model, we indeed see that this variable is of great importance to the OLS model. As the variable exhibits a sudden jump in value, this leads to the model making a very poor forecast for June 13th 2022. As a result, the accuracy measures give a distorted picture of the OLS overall performance for the curvature characteristic. In reality, the OLS model performs quite well, except for that single observation.

Figure 12: Time series of *Open interest slope*



Note: this figure shows the time series of the *Open interest slope* variable. The date range spans from January 3rd to December 30th, and includes only trading days.

When analyzing the variables that are most important for the best performing model (RF in all cases), there are some variables that are consistently of great importance. The most important variable is the lagged characteristic variable. We see this result returning for the level, slope, and curvature characteristic of the IVS. This is in fact not surprising as the first partial autocorrelations for each characteristic shown in sub-Figures 13a, 13b, and 13c in Appendix C are significant. In addition, we see that the volume and open interest 'characteristics' are often of high relative importance as well. Specifically, the *Volume curvature (ls)* variable is among the top most important variables for the RF models. Chen et al. (2023) also find that "the common fundamentals that drive the IV curves are related to the market liquidity", which is in line with our results.

6 Conclusion

The main goal of this research was to find out how machine learning methods perform in forecasting the characteristics of the implied volatility surface for weekly options on the S&P 500. We used three different machine learning methods and two non-learning-based methods, where we found that the non-learning-based methods perform comparatively well with the machine learning methods. On the other hand, there is one model that consistently outperforms all other models, which is the Random Forest model. An unexpected result which is found in this paper is that the Neural Network models do not manage to make accurate forecasts for the slope and curvature characteristic. Conversely, the Neural Network model does make relatively accurate forecasts for the level characteristic. The linear ML method Elastic Net is consistently outperformed by the non-linear ML method Random Forest, as well as the Neural Network model for the level characteristic.

For the second sub-research question, we see that the results are somewhat in line with prior research on long-term IV predictive performance. For example, we observe that the variable selection of the models is in line with other papers, such as Chen et al. (2023). On the other hand, while the Random Forest machine learning method performs well, the other machine learning methods fail to fulfill their predictive potential. In many papers it is found that the NN model performs well regarding long-term IV forecasting. The fact that it does not work for the short-term IVS characteristic forecasting in this paper is most likely not due to the difference in contract maturity. We namely do see that the NN model does give relatively accurate

forecasts for the level characteristic of the IVS. The poorly performing NN models for the slope and curvature characteristics are harmed by overfitting problems.

A reason for the NN model to encounter overfitting problems could be due to the fact that we have used a relatively small data set. NN models perform exceptionally well in a data rich environment, but if this is not the case, forecasts made by NN models tend to be easily affected by overfitting problems. In this study we have used only three years of data which may have lead to the model being affected by overfitting problems. Hence, for future research, it could be interesting to perform the same research with the same models while using a larger data set. On top of that, note that we did not use the 'standard' FNN method, but we used the LSTM variant of a RNN. Although this choice is well substantiated (see Additional remarks), there may be a model which is even more fit for forecasting the characteristics of the IVS. According to Medvedev and Wang (2022), their findings specifically indicate that the convolutional LSTM model (ConvLSTM) significantly outperforms the LSTM model. Therefore, taking this model into consideration might give even more accurate results than the Random Forest model does.

For future research, one could also look more to make an extension on the number of variables that are selected for this research. Bernales and Guidolin (2014) describe that "there are strong cross-sectional and dynamic relationships between the IVS of equity options and the IVS of index options". Hence, one can take some of those variables into account in combination with the well-performing RF model to acquire an even more accurate forecasting model. Another adjustment that can be made is transforming the dependent variable. We saw that the three characteristics did not follow a standard Normal distribution. Applying logarithmic transformations for these characteristics might result in a better performance of the NN compared to the results in this study.

The main takeaway from this study is that the Random Forest model is a very useful machine learning model in predicting the characteristics of the implied volatility surface. On top of that, the variables that are of great importance to forecast the IVS are either its lagged values or are variables that are related to the market liquidity. This study serves as a stepping stone to future research, expanding the literature on the relatively understudied ultra short-term IVS literature domain.

Additional remarks

Before concluding this paper, I make some additional remarks on the progress during this thesis process regarding the Neural Networks models performances. First, I have performed the forecasting procedure for a standard feedforward neural network (FNN), but due to a very poor performance I have decided to use a more suited neural network: the Long Short-Term Memory neural network. Using the FNN model with one hidden layer resulted in a RMSE of 15.2% compared to a RMSE of approximately 6.0% for the other models for the level characteristic. Changing the number of hidden layers ranging from 1 hidden layer to 5, resulted in very slight improvements or reductions of the values corresponding with the models accuracy. Using the LSTM neural network made this difference significantly smaller; it even managed to outperform some of the other models for the level characteristic. Hence, the LSTM model represents the NN method in this paper.

Acknowledgements

I would like to thank dr. G. Freire for his helpful guidance and supervision during this project. On top of that, doing research on option pricing has always interested me and thus having worked on this paper has helped me to gain knowledge on a part of the option pricing of which I was not yet educated on. For this I would also like to thank dr. G. Freire as he helped me with this thesis idea.

References

Almeida, C., Fan, J., Freire, G., and Tang, F. (2023). Can a machine correct option pricing models? *Journal of Business & Economic Statistics*, 41(3):995–1009.

Almeida, C. and Freire, G. (2022). Pricing of index options in incomplete markets. *Journal of Financial Economics*, 144(1):174–205.

Almeida, C., Freire, G., and Hizmeri, R. (2024). 0dte asset pricing. *Available at SSRN*.

Andersen, T. G., Fusari, N., and Todorov, V. (2017). Short-term market risks implied by weekly options. *The Journal of Finance*, 72(3):1335–1386.

Bernales, A. and Guidolin, M. (2014). Can we forecast the implied volatility surface dynamics of equity options? predictability and economic value tests. *Journal of Banking & Finance*, 46:326–342.

Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654.

Chen, D., Guo, B., and Zhou, G. (2023). Firm fundamentals and the cross-section of implied volatility shapes. *Journal of Financial Markets*, 63:100771.

Christoffersen, P., Jacobs, K., and Chang, B. Y. (2013). Forecasting with option-implied information. *Handbook of economic forecasting*, 2:581–656.

Cont, R. and Da Fonseca, J. (2002). Dynamics ofimplied volatility surfaces. *Quantitative finance*, 2(1):45.

Dickey, D. A. and Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association*, 74(366a):427–431.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2016). Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273.

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116.

Liu, S., Oosterlee, C. W., and Bohte, S. M. (2019). Pricing options and computing implied volatilities using neural networks. *Risks*, 7(1):16.

Medvedev, N. and Wang, Z. (2022). Multistep forecast of the implied volatility surface using deep learning. *Journal of Futures Markets*, 42(4):645–667.

Merton, R. C., Scholes, M. S., and Gladstein, M. L. (1982). The returns and risks of alternative put-option portfolio investment strategies. *Journal of Business*, pages 1–55.

Tohry, A., Chelgani, S. C., Matin, S., and Noormohammadi, M. (2020). Power-draw prediction by random forest based on operating parameters for an industrial ball mill. *Advanced Powder Technology*, 31(3):967–972.

Vrontos, S. D., Galakis, J., and Vrontos, I. D. (2021). Implied volatility directional forecasting: a machine learning approach. *Quantitative Finance*, 21(10):1687–1706.

Wenyong Zhang, L. L. and Zhang, G. (2023). A two-step framework for arbitrage-free prediction of the implied volatility surface. *Quantitative Finance*, 23(1):21–34.

Appendix

A Feature list

Table 7: Brief description of variables used

Variable	Type
Lagged level of the IVS	Option
Lagged slope of the IVS	Option
Lagged curvature of the IVS	Option
Total Open Interest (l)	Option
Open Interest 'slope' (ls)	Option
Open Interest 'curvature' (ls)	Option
Total Volume (l)	Option
Volume 'slope' (ls)	Option
Volume 'curvature' (ls)	Option
Mortgage index (30Y fixed rate) (d)	Macroeconomic
Risk-free rate (10Y T-Bill) (d)	Macroeconomic
Inflation rate (5Y break-even) (d)	Macroeconomic
Gold ETF Volatility Index	Macroeconomic
Crude Oil ETF Volatility Index (d)	Macroeconomic
VIX	SPX / Macroeconomic
SPX return	SPX
SPX return squared	SPX
SPX 5-(trading)day return	SPX
Coinbase Bitcoin (CBBTCUSD) (ld)	Cryptocurrency

Note: For the Open Interest and the Volume variables, this involves a level, slope, and curvature measure just as is done for the IVs. The one difference is that we now do not use a weighted average on the observations but simply look at the total amounts (as averaging them does not make much sense). The variables shown in **bold** are variables for which some data points were missing. For these missing data points we have applied linear interpolation. Note that the letters in the brackets represent additional adjustments made to the data; l corresponds with a log transformation, d with first differencing, and s with a shift. More information on this can be found in Section 3.

B Hyperparameter Tuning Grid

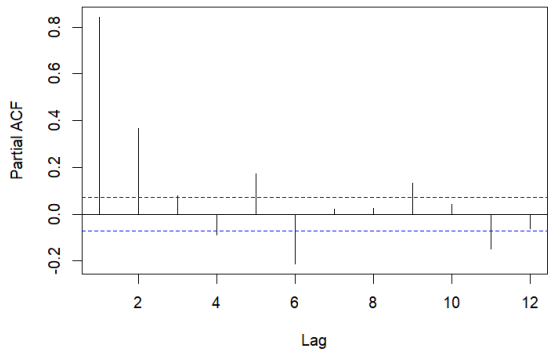
Table 8: Hyperparameter tuning grid

Elastic Net	Random Forest	Neural Networks
$\alpha \in \{0.1, 0.2, ..., 0.9\}$	$ntree \in \{1, 2, ..., 500\}$	hidden layers $\in \{1, 2, ..., 5\}$
	$mtry \in \{1, 2, ..., 10\}$	dropout rate = 0.2
		activation function = ReLu
		units per layer $\sim$ geometric pyramid rule
		batch size = 64
		epochs = 100

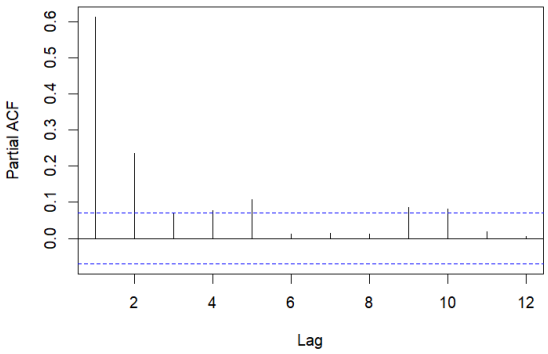
C Partial Autocorrelation Plots

Figure 13: Partial Autocorrelation Functions (PACF) for the three characteristics of the IVS

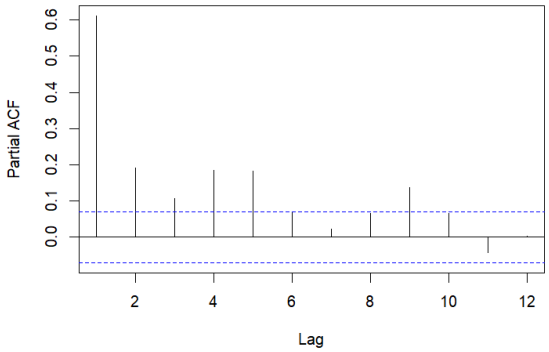
(a) PACF level



(b) PACF slope

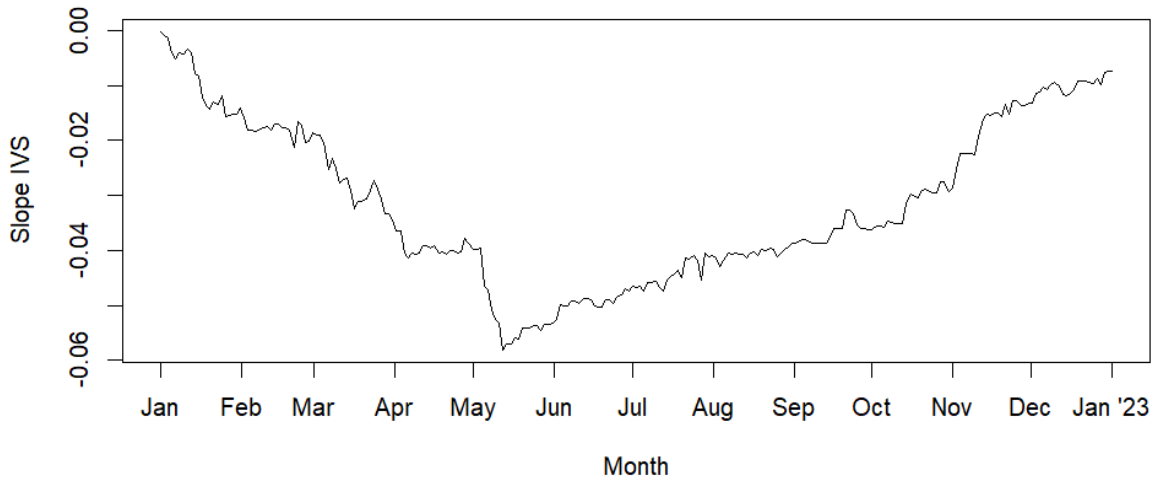


(c) PACF curvature



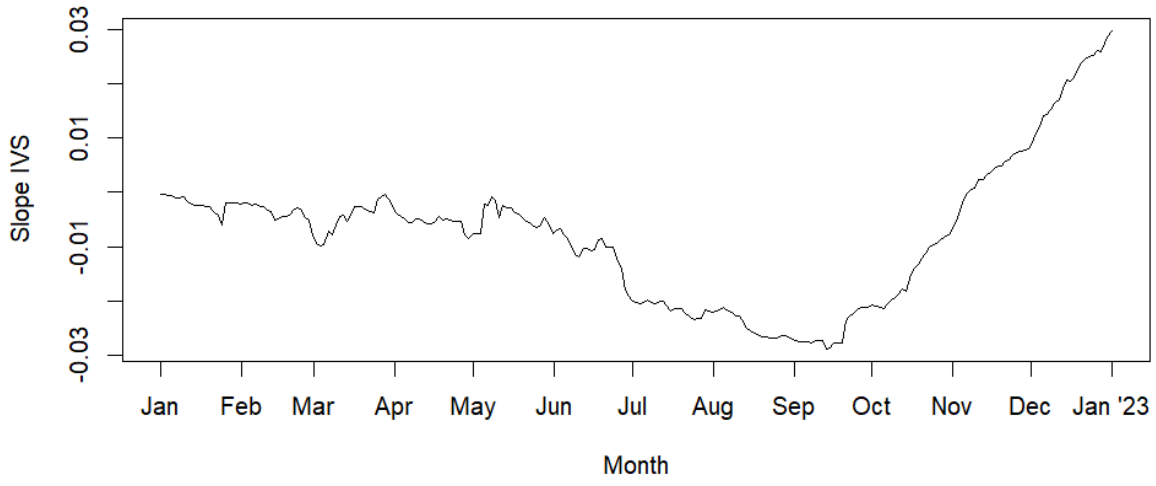
D Additional figures

Figure 14: CSSED of the AR(1) model compared to the OLS model for the slope characteristic



Note: In this figure the dates run from January 2022 to the end of December 2022. We compare the forecasts of the AR(1) model with those of the OLS model. Thus, we use AR(1) as model A and OLS as model B in Equation 23.

Figure 15: CSSED of the NN model compared to the benchmark model for the slope characteristic



Note: In this figure the dates run from January 2022 to the end of December 2022. We compare the forecasts of the NN model with those of the benchmark model. Thus, we use NN as model A and the benchmark as model B in Equation 23.