Master's Thesis - Quantitative Finance
IVS duality: A novel two-step approach to using IVS forecasts to model the underlying's daily volatility

Author: Virgjil Karaja
Supervisor: Gustavo Freire
Student ID: 566701
Second Assessor: Chen Zhou

Abstract

This paper introduces a new two-stage approach to daily volatility modelling where we first model the implied volatility surface (IVS) of the weekly options of the S&P 500 (SPWX) and then use the forecasts of the IVS to model the daily volatility of the S&P 500 itself. Using a rather large dataset of 853805 observations of weekly-expiring options on the S&P 500 across 1298 trading days (from 2^{nd} of January 2018 to 28^{th} of February 2023), we find our approach to improve the accuracy of daily volatility predictions for the S&P 500 greatly in terms of RMSE when compared to traditional benchmarks, such as T/GARCH. We test several models on IVS forecasting and find an ensemble, the equally-weighted combination of non-linear models (EW2) to be the best as judged by our performance criteria tests (SPA, StepM and MCS). Additionally, our TGARCH-V (1,1,1) model that makes use of the average IV across the forecasted IVS captures the daily volatility of the underlying statistically better than the rest. We find the IVS contains significant information for its underlying's volatility forecasts, but it may come at the cost of unstable estimation.

Date: 4th September 2024

Contents

1 Introduction

Options are financial instruments that give the holder a choice instead of an ultimatum. The holder decides on whether to exercise the contract, buy/sell an asset (the underlying) at a stated price within a stated time period, or not. Due to their convenience, options trading on exchanges worldwide have increased from $9.42 billion contracts in 2013 to $21.22 billion contracts in 2020 (Bali et al., 2023). A particular spectrum of option trading that has seen a rapid increase is related to short-term options. As an example, Andersen et al. (2017) show how S&P 500 option contracts with one week or less to maturity ("weeklies") have seen their share of trading at the Chicago Board of Options Exchange (CBOE) rise steadily from about 10% in early 2010 to almost 30% in mid-2015. While Wen (2020) finds no evidence of the introduction of weeklies leading to higher realized volatilities on their respective underlyings, they claim a significant change in investment and hedging opportunities for market participants. As such, modelling and understanding the dynamics of these short-term options is of importance to investors.

Most of the literature on S&P 500 index options focuses on pricing options of longer maturities (10 days +) (Almeida et al., 2022; Andersen et al., 2015; Gonçalves and Guidolin, 2006). The literature on the S&P 500 weekly is still relatively thin, despite their growing popularity and inherent differences to longer-dated options (Andersen et al., 2017). They elaborate on how the weeklies improve investors' ability to acquire or lay off exposure to diffusive and jump price risks, since the expected volatility and jump intensity do not vary much over the remaining life of the option, when the time to maturity is short. Further proof comes from the work of Almeida et al. (2024); Bandi et al. (2023) on zero days-to-expiration (0DTE) options, emphasizing the need of focusing on the weeklies explicitly, as the rationale behind longer maturity options does not tend to generalize.

It is a common approach in literature, to represent the option in terms of its implied volatility surface (IVS) (Almeida et al., 2022; Avellaneda et al., 2020; Mixon, 2002; Gonçalves and Guidolin, 2006; Freire and Kleen, 2024). The IVS plots the implied volatility (IV) of an option against moneyness[1] and time to maturity. The IV is derived as the value that equates the formula of Black and Scholes (1973) (BS) to the observed empirical price and is often referred to as the market's volatility forecast, due to a forward-looking nature Liu et al. (2021); Zeng and Klabjan (2017). When the other parameters of the BS formula belong to a known information set, there exists a one-to-one mapping between the option's price and its IV, allowing for easier comparison in the cross-section and over time Almeida et al. (2022). It is of noteable importance in the field of risk management (Muzzioli, 2010; Chen et al., 2023). Carr and Wu (2016); Hull and White (2017) also demonstrate the usefulness of IVs for managing open option positions, which is of major importance in option pricing models as well (Ait-Sahalia et al., 2016).

The standard BS model implies a constant IV across time, moneyness and maturity, which does not hold empirically. In fact, Rubinstein (1994); Canina and Figlewski (1993) show that the IVS resembles an asymmetric smile or smirk, which relates to the conditional non-normality

[1] denoted as the ratio between the stock price at the time and the strike followingf Freire and Kleen (2024)

of the underlying return risk-neutral distribution. In particular, a smile reflects fat tails in the return distribution whereas a skew indicates return distribution asymmetry. The existence and implications of the so-called "volatility smile" are phenomena studied extensively (Le and Zurbruegg, 2014; Han et al., 2015; Campa and Chang, 2023). The local properties of the IVS contain information about the underlying market dynamics (Kelly et al., 2023). For example, Breeden and Litzenberger (1978) state that variation along the moneyness dimension can be used to uncover the Arrow-Debreu state prices, while the surface slope along the maturity dimension contains information about volatility by the Dupire formula (Davis, 2011). Being aware of the severe misspecification of the Black–Scholes model, a large spectrum of option pricing models has been proposed to capture the observed implied volatility patterns, but, almost exclusively, for options of longer maturities (Almeida et al., 2022).

Studies have found predictable movements in the IVS, particularly in index options like the S&P 500 (Skiadopoulos et al., 2000), which is why we use data on the S&P 500 weeklies (SPXW). Dumas et al. (1998) propose a modified version of the original BS formulation (AHBS), thus providing a simple, parametric, benchmark approach to modelling the IVS (Berkowitz, 2009; Dixit and Singh, 2017). However, Christoffersen and Jacobs (2004) report the instability of the IVS coefficients when estimated on the weekly cross-section of S&P 500 options. Our focus is one ultra-short term options, as such, our data consists of daily cross-sections. We argue the larger density of our dataset lifts some of the concerns regarding the unstable IVS coefficients. Due to its simplicity and overall good predictive performance (Almeida et al., 2022), we use the AHBS model as our benchmark to beat in terms of IVS modelling.

Mixon (2002); Avellaneda et al. (2020) provide proof of better IVS predictability when accounting for other factors on top of moneyness and time to maturity. In their spirit we supplement our dataset with additional option-specific characteristic. Following the logic of Varian (2014); Gu et al. (2020), the convoluted underlying interactions, the large set of covarities, the nonlinear relations and the restricting assumptions needed by parametric IVS models pave the way for the usage of machine learning models (ML). In a similar manner to Gu et al. (2020) we apply a battery of ML methods to tackle the issue of IVS modelling on the daily cross-section of SPXW. We use elastic net (ENET) due to its inherent ability of handling multicolinearity and straigtforward interpretation (Zou and Hastie, 2005). Additionally, we utilise two tree-based method. The first is the random forest (RF) of Breiman (2001), documented as being robust to noise/outliers and providing accurate predictions in different financial applications (Gu et al., 2020). Secondly, we use the linear forest (LF) of Friedberg et al. (2018), which extends RF by conducting a linear regression inside the leaves, replacing taking the simple average. Differently from Freire and Kleen (2024), we apply LF directly to the IVS instead of on the AHBS coefficients. This leads to a loss in interpretability, but potential gains in capturing more complex interactions within the IVS and reducing estimation error across the different options on the same underlying. Kelly et al. (2023) argue that going a step further from ML into deep learning (DL) algorithms is beneficial. Concretely, the employ a convolutional neural network (CNN) (O'Shea and Nash, 2015) and rely on its ability to learn and identify patterns at various levels of

abstraction. They argue a CNN has the potential to recognize IVS patterns in the same manner a professional trader would, which is why we include CNN in our models used. We choose to also incorporate a simple recurrent neural network (RNN) to see the benefits of the additional complexity, CNN brings to an easier DL method.

Another lead to pursue in the context of IVS modelling is that of dimensionality reduction. Skiadopoulos et al. (2000); Mixon (2002); Andersen et al. (2015) demonstrate that most of the IVS information is captured largely by a few latent factors, leading to better predictability and simpler models. A common, yet effective approach, is to use the principal component analysis (PCA) of Pearson (1901) giving interpretable factors that explain most variation of the original data while being mutually orthogonal (Avellaneda et al., 2020) and use them in a principal component regression (PCR). Nevertheless, Andersen et al. (2015); Bali et al. (2013) raise concerns of nonlinear relations that PCA does not capture. A solution given by Gu et al. (2021) is the use of autoencoders (AE). An AE consists of two neural networks (NN) which are connected to each other through the latent factors, the encoder and the decoder. Thus, it allows for nonlinear relations between the original data and the factors and vice versa by construction Ning et al. (2023).

We employ a plethora of models to see which one predicts IVS of SPXW the best, but each model has its own merits. Clements and Vasnev (2021a); Dietterich (2000) provide proof of the benefits of ensembles against a single model. By combining several model predictions, which capture different aspects of the IVS, an ensemble exploits all their strengths while mitigating weaknesses. Rapach et al. (2009); Wang et al. (2017) among others report on the benefits of using an equally-weighted (EW) forecast combination and how it is difficult to beat the simple average (Timmerman, 2006). One ensemble method we use is the EW forecast combination (denoted simply by EW from now on) and the other is again an equally-weighted forecast, which only averages the predictions from the models that capture non-linear relations in the IVS (AHBS, RF, RNN, AE) that we refer to as EW2. We use several test for measuring predictive ability like the Superior Predictive Ability (SPA) (Hansen, 2005), Stepwise Multiple Testing (StepM) (Romano and Wolf, 2005) and the Model Confidence Set (MCS) (Hansen et al., 2011a). As a result, in terms of IVS modelling, we employ a total of 8 models and two ensembles: AHBS, ENET, RF, LF, PCR, RNN, AE, CNN and EW, EW2.

Böök and Sala (2020) document the predictive ability of short-term option IVs themselves. We aim to exploit this ability using our daily IVS forecasts to predict the underlying's (S&P 500) own daily return volatility (proxied by the squared daily return). The generalized autoregressive conditional heteroskedasticity (GARCH) of Hansen and Lunde (2001) is the benchmark to beat when it comes to excess return modelling and daily variance estimation (Hansen and Lunde, 2001). Due to its ability to capture the three stylized facts of returns (Granger and Ding, 2024) and subsequent performance, a lot of work has been done in extending and improving the GARCH (1,1) model (Hansen et al., 2011b). We hypothesize that the inclusion of information from the IVS forecasts leads to better forecasts for the daily volatility of the underlying itself. To that avail, we propose two extensions to the variance equation of the GARCH model, leading to

namely GARCH-V (1,1,1) and GARCH-V (1,1,3), which make use of the IVS forecasts from the best predictive IVS model. In GARCH-V (1,1,1) we add the average across the IVS forecasts for the given day of the best model as an explanatory variable, while for GARCH-V (1,1,3) we add the level, slope and curvature[2] of the IVS as explanatory variables. We argue that our estimated IVS over the cross-section of SPXW on each given day and its features are by construction more flexible to current market information and as such a better fit to supplement the GARCH (1,1) model than the more commonly used volatility measures such as VIX (Hansen et al., 2011b). We also add a threshold term to our novel models (Zakoian, 1994) to capture the more risk averse bias of the individual investor to be more mindful of losses than an equal-sized profit. We name these modified models TGARCH-V (1,1,1) and TGARCH-V (1,1,3), which make use of the same principle as previously explained. To the best of our knowledge, we are first to utilize this two-step approach to modelling daily returns and volatility by exploiting information from the corresponding IVS forecasts.

Through our comprehensive analysis we reach several conclusions. We document the dominance of non-linear models in IVS modelling, as claimed by Bali et al. (2013), out of which the RF demonstrates the best performance when excluding the ensemble methods. Overall, to not much surprise when being aware of the results of Timmerman (2006), the EW2, the combination of non-linear models is superior in terms of IVS modelling. Despite the horizon, it is always the only model included in the set of superior models implying that it is sufficient to obtain a very good forecast of the daily IVS. Furthermore, we see that almost all of our novel models, that utilize the information from the one-step ahead IVS forecasts, are significantly better than the two benchmarks (GARCH (1,1) and TGARCH (1,1,1)) in terms of modelling the daily variance of the underlying S&P 500 when using the squared daily returns as a proxy for the real value. In particular, our TGARCH-V (1,1,1) obtains the statistically best predictions in terms of RMSE, also supported by our statistical tests of superior predictive ability. Despite the clear benefits of using information extracted from the IVS, we see some instability in the coefficient estimates of our TGARCH-V (1,1,3) model, which utilizes the level, slope and curvature of the forecasted IVS. It is our worse performing model, potentially due to the small scales of the returns and IVS characteristics. Hence, we emphasize the necessity of developing a stable estimation procedure for this particular model. Nevertheless, our results provide the evidence needed to say that the "weeklies" of the S&P 500 (SPWX) contain valuable information in their IVS shape regarding the index itself and its price variability, especially the average IV across the surface.

Our contribution to the literature is three-fold. Firstly, we expand heavily on the gap on the IVS modelling and predictability of ultra-short term options, while showing that non-linear models dominate their linear counterparts in terms of IVS modelling even for very short-term securities. Furthermore, to the best of our knowledge, we are the first to propose an ensemble of non-linear models to tackle the issue of IVS modelling with it being dominantly the predictive superior across different horizons. Lastly, and arguably most importantly, we provide a novel two-step approach to volatility modelling, where we utilize the information in the IVS predictions in

[2]we follow Chen et al. (2023) to compute the IV features explained in more detail in Section 2.1

constructing four novel GARCH-type models to attempt to predict the S&P 500 daily volatility. Namely, we utilize the IVS forecasts we obtain from our IVS models in the S&P 500 daily volatility models to obtain better forecasts. We use the daily squared return as our estimate for the volatility and clearly prove the remarkable interconnection of the stock's daily volatility with its corresponding daily IVS with our models attaining significantly better predictions and improvements in terms of RMSE.

The rest of the paper is structured as followed. In Section 2 we go over the dataset that we use, introducing both the option related variables alongside the extra covariate data we collect. Furthemore, in Section 3, we explain our research methodology, starting with the IVS models and the GARCH-type models used, plus the evaluation metrics for comparison. In Section 4, we discuss our results, beginning with the IVS models' results and then the GARCH-type models. Lastly, we conclude in Section 5, also mentioning potential limitations in our research.

2 Data

We use European-style S&P 500 options traded at the Chicago Board Options Exchange (CBOE). We collect the SPX option data from OptionMetrics IvyDB in Wharton Research Data Services (WRDS)[3]. It includes various metrics ranging from implied volatility, strike price (which we divide by 1000 since WRDS has a multiplier of 1000 on it), days to maturity (we manually compute it as expiry date minus current date), best mid price (computed as the average between the best bid and ask prices on the given day across all exchanges), delta, gamma, theta, volume and open interest. The sample ranges from 2^{nd} of January 2018 until 28^{th} of February 2023, thus also including periods of financial instability (e.g. Covid-19). Due to our focus on ultra-short term weekly options, we filter to include only options expiring in seven days or less and that are of a weekly nature. We exclude options with null implied volatility, null best bid or best ask, less or equal to zero volume, bid greater than ask. Furthermore, we follow Freire and Kleen (2024) in computing moneyness as $m_{i,t} = \frac{S_t}{K_{i,t}}$ where S_t is the stock price of the underlying (S&P 500 index) at day t and $K_{i,t}$ is the strike price of option i at date t. Similarly, we restrict ourselves to options with a moneyness between 0.5 and 2.0, leaving us with a total 853805 observations across 1298 days with an average of around 658 observations per day. Moreover, due to our focus being shifted to IVs we do not distinguish between calls and puts. We provide summary statistics of the IVS when put against moneyness and time-to-maturity in Table 1.

We see the familiar volatility smile present itself where the volatility decreases when moving from OTM calls towards ITM and ATM options and rise again for OTM puts. All of the options in our dataset are ultra-short term, hence the U-shaped volatility smile is sharp, sharper than what we would expect from more medium/long-term options. Furthermore, we see a (mostly) decreasing trend within each moneyness category as the days to maturity increase showing that there does exist a difference between short-term and ultra-short term options motivating our choice to analyze them in more detail. The majority observations fall in the ATM category,

[3] https://wrds-www.wharton.upenn.edu/pages/get-data/optionmetrics/ivy-db-us/

Days to maturity	Moneyness				
	[0.5,0.9)	[0.9,0.97)	[0.97,1.03]	(1.03,1.1]	(1.1,2]
$\tau = 1$	1.2354	0.4386	0.2477	0.4668	0.9238
	0.5599	0.2147	0.1262	0.1469	0.3367
	1393	11181	33986	21934	11105
$\tau = 2$	0.8705	0.3341	0.2275	0.3796	0.7419
	0.3928	0.1620	0.1217	0.1231	0.2752
	1438	11636	31549	20897	15371
$\tau = 3$	0.7405	0.2804	0.1859	0.3170	0.6230
	0.4126	0.1391	0.1059	0.1096	0.2416
	1483	10986	30516	20464	14211
$\tau = 4$	0.6161	0.2521	0.1821	0.2951	0.5813
	0.2867	0.1225	0.1002	0.1047	0.2190
	1445	11688	31840	19860	14314
$\tau = 5$	0.6395	0.2297	0.1726	0.2706	0.5202
	0.3929	0.1186	0.0905	0.0957	0.1959
	1163	12092	38004	22546	13147
$\tau = 6$	0.5646	0.2249	0.1822	0.2705	0.5180
	0.2794	0.1091	0.0924	0.0930	0.2012
	1345	12584	40515	23581	13047
$\tau = 7$	0.4707	0.2144	0.1826	0.2647	0.4934
	0.2221	0.1067	0.0922	0.0894	0.1852
	2505	20109	51868	33015	26785

Table 1: Descriptive statistics of the IVs across days to maturity and moneyness from the 2^{nd} of January 2018 until 28^{th} of February 2023 for SPWX options on the S&P 500. Each cell corresponds to a certain days to expiry and moneness where [0.5,0.9) and (1,1,2] correspond to OTM options; [0.9,0.97) and (1.03,1.1] are ITM options and [0.97,1.03] are ATM options following Freire and Kleen (2024). Each cell contains the mean (top number), standard deviation (middle number) and total observations (bottom number) of the options in the corresponding block.

which is not surprising as the options are very short-lived making high deviations of the current price of the underlying from its corresponding strike price rarer.

We collect the S&P 500 index data from Yahoo Finance[4]. We also approximate the risk-free rate by the 3-month Treasury Bill of the U.S., taken from e St. Louis Federal Reserve Economic Data (FRED) database[5]. We deal with missing values by interpolation, namely we set them equal to the average of the last three days. As our proxy for the real variance of the S&P 500 daily returns we use the squared daily returns on the index due to unavailability of intra-day data.

In our novel GARCH-V (1,1,3) we use the three fundamental shape characteristics of the IVS: level, slope and curvature. We follow a similar approach to Chen et al. (2023), but utilize the

[4] https://finance.yahoo.com/quote/^SPX
[5] https://fred.stlouisfed.org/

daily IVS. The level is based on the IV corresponding to an ATM option ($IV_t^l = IV_t^{ATM}$); the slope is estimated by the IV of a spread strategy involving buying and OTM option and selling an ATM option ($IV_t^s = IV_t^{OTM} - IV_t^{ATM}$); the curvature is approximated by the IV of a butterfly spread strategy that involves buying OTM and ITM options while selling an ATM option ($IV_t^c = \frac{IV_t^{OTM}+IV_t^{ITM}}{2} - V_t^{ATM}$), for each day t. To avoid being as restrictive as Chen et al. (2023), we set IV_t^{ATM} equal to the average IV of options in day t with a delta between [-0.55,-0.45] (instead of only -0.5); IV_t^{OTM} to the average of options in day t with a delta between [-0.25,-0.15] and IV_t^{ITM} to average of options with deltas between [-0.85,-0.75]. In this manner we make use of more information and mitigate losses from our forecasting approach by taking averages.

3 Methodology

In this section we go through our methodology. We explain our IVS predictive models, the novel GARCH models for predicting excess returns and the volatility of the underlying asset (S&P 500) and the evaluation metrics we use for comparison.

3.1 IVS models

In this section we iterate over our models for the IVS starting with the AHBS benchmark. All the models are trained on each day t = 1,...,t-1 over the cross-section i = 1,...,n_t (we use the notation to show that the number of options in the cross-section changes each day) of SPWX options and are used for forecasting the following day t^* = 2,...,t. As such, most of the information is extracted from the cross-section of SPWX options while the time-varying regressors take the role of a constant in the regressions, since they do not change within the day. However, we see including them similar to a control variable as they capture the impact of differences from the estimated day to the forecasted day that do not stem from the cross-section of options themselves. In this manner our train and test set shift after each day and we use a randomized 20% selection of each day t that we train as the validation set for hyperparameter tuning (only for the models that need tuning). Following Almeida et al. (2022), we try different forecasts horizons, one week ahead (5 days) and one month ahead (22 days) as well to see if the IVS results generalise. We aim to predict the IVS and use the forecasts of our lowest RMSE model in the second step of our approach.

3.1.1 Benchmark: AHBS

The AHBS model modifies the constant IV assumption of BS as it allows for variability across the IVS. It specifies a linear regression of the IV on quadratic functions of moneyness and time to maturity. The AHBS model is as shown in (1:

$$\sigma_{i,t}^{AHBS} = \alpha_{0,t} + \beta_{1,t} m_{i,t} + \beta_{2,t} m_{i,t}^2 + \beta_{3,t} \tau_{i,t} + \beta_{4,t} \tau_{i,t}^2 + \beta_{5,t} m_{i,t} \tau_{i,t} + \epsilon_{i,t}, \tag{1}$$

where $\epsilon_{i,t} \sim \mathcal{N}(0,1)$. Next, $\sigma_{i,t}$, $m_{i,t}$ and $\tau_{i,t}$ are the observed IVs, moneyness and time to maturity of option i at day t, respectively. The parameters of (1) are estimated via ordinary

least squares (OLS), which is the same as minimizing the implied volatility mean squared error term (IVMSE):

$$IVMSE_t = \frac{1}{n}\sum_{i=1}^{n}[\sigma_{i,t} - \sigma_{i,t}^{AHBS}(\beta_t, m_{i,t}, \tau_{i,t})]^2,$$

where $\hat{\sigma}_{i,t}^{AHBS}(\beta_t, m_{i,t}, \tau_{i,t})$ are the fitted values of the AHBS model from (1) and β_t is the vector of parameters to be estimated. After estimation of (1) at day t, we keep the parameters fixed and use the explanatory variables of period t^* to predict the IVS of the next day ($\hat{\sigma}_{i,t^*}^{AHBS}$. The model also encapsulates the simple BS model by restricting all parameters, but the intercept, to zero, which we exclude from our analysis due to its poor performance in reality.

We use the sklearn package for python for conducting the linear regression. The advantages of the model rely on its simplicity, interpretability and practicality. It is flexible and can be calibrated to fit the IVS on any given day while being computationally efficient.

3.1.2 ENET

ENET performs a linear regression, with the penalty $\alpha||\beta||_1 + (1-\alpha)||\beta||_2^2$, meaning that it encapsulates both ridge (Hoerl and Kennard, 1970) and LASSO (Tibshirani, 1996). In this manner ENET performs both variable selection and shrinkage. The solution of the elastic net is found after performing the optimization problem in (2):

$$\min \ \frac{1}{2n}||\sigma_t - X_t\beta_t||_2^2 + \lambda[(\frac{1-\alpha}{2})||\beta_t||_2^2 + \alpha||\beta_t||_1], \tag{2}$$

where σ_t refers to the IVS of SPXW at day t and X_t is the matrix containing the cross-section of all of our regressors at day t. We estimate the model using the sklearn package for python. To numerically solve the problem we need to know the values of the exogenous parameters, α and λ. To that extent we emphasize the need of having a validation set for hyperparameter tuning. We explain our approach in detail in Appendix B alongside the parameter grid we use in Table 4.

The model performs feature selection and regularization making it a good choice to prevent overfitting and dealing with the high dimensional space of regressors. Additionally, it handles multicollinearity which is problematic and more often than not present within financial variables.

3.1.3 RF

The random forest is built upon several decision trees, where each new observation is categorized on the basis of its characteristics and it also provides forecasts of a dependent variable by taking the mean of the required characteristic over all the observations belonging to the same final node that the new observation is placed into. We supply an image of such a tree in Appendix A.1, adapted from Murphy (2012) and refer you to their work for a detailed explanation on the workings of a decision tree.

The RF reduces the high variance associated with a single decision tree (Hastie et al., 2009), as it uses bagging where a number of separate bootstrap samples are taken from the original data

to train the trees and finally an average is taken across the trees. Additionally, the random forest adds a randomness feature in variable selection lessening the correlation of the individual trees, which helps with reducing the variance of predictions leading to higher accuracy and robustness. We tune the parameters of RF over Table 5, as explained in Appendix B. The full RF algorithm of Breiman (2001) has the following structure:

Algorithm 1: Random Forest Prediction Procedure

Step 1: Draw B bootstrap samples from training data X_t.

Step 2: **for** $i = 1$ *to* B **do**

Grow a deep Decision Tree (DT) on b_i using the DT algorithm as described in A.1, with the modification that a random subsample H of feature variables is selected for splitting at each node, i.e., $H \subset J$ where J denote the sample of our all features.

Output: In each leaf node of the i-th tree compute the IV estimate for RF. Set the estimate to the mean value of the training samples that reach that node, as $\hat{\sigma}_{i,t}^{DT}$ (prediction of the decision tree for bootstrapped sample i).

Step 3 (across tree aggregation): The final RF prediction is the mean of the predictions obtained from all trees as

$$\hat{\sigma}_t^{RF} = \frac{1}{B} \sum_{b=1}^{B} \hat{\sigma}_{i,t}^{DT}$$

We estimate the RF using the sklearn package for python with the obtained parameters from the hyperparameter tuning. The RF also captures non-linear relationships, differently from the other models so far, while being robust to over-fitting due to averaging over many trees. It is stable with high dimensional data and diverse market conditions such as in our dataset.

3.1.4 LF

LF combines elements of RF with linear regressions. It constructs an ensemble of decision trees using bootstrapping and feature randomness, similar to traditional random forests. However, instead of using the raw predictions of individual trees, LF fits a linear regression model to a neighborhood of training data points around each test point and uses the linear model's prediction as the final output. It manages to capture more complex relations on each node by aggregating options in the cross-section based on similar characteristics. For example, ATM options of similar maturity (amongst other covariates) may be more closely related to one-another than an OTM option of a different maturity and the LF is able to exploit these relations. We tune the parameters of LF over Table 6, as explained in Appendix B. The full algorithm is as follows:

Algorithm 2: Local Linear Forest Prediction Procedure

Step 1: Draw B bootstrap samples from training data X_t.

Step 2: for $i = 1$ *to* B **do**

Grow a deep Decision Tree (DT) on b_i using the DT algorithm as descripted in A.1, with the modification that a random subsample H of feature variables is selected for splitting at each node, i.e., $H \subset J$ where J denote the sample of our all features

Output: On each node fit a linear model: $\hat{\sigma}_{i,t}^{DT} = \alpha_{0,i} + \sum_{j=1}^{k} \beta_{j,i} x_{j,t}$ for each covariate j = 1,...,k in our dataset (prediction of the decision tree for bootstrapped sample i).

Step 3 (across tree aggregation): Aggregate the predictions of each tree in several ways including, but not limited to: simple averaging, weighted averaging, considering the quality of the local linear regression fit and combining predictions using a more complex model.

***** Choose which prediction is relevant for the problem at hand.

We estimate the LF using the lineartree package for python with the obtained parameters from the hyperparameter tuning. The LF provides reduced overfitting such as RF with improved interpretability since linear models are fitted on the leaves. It combines linearity with flexibility and is useful when the data exhibits different linear relationships in different regions (ITM/OTM/ATM).

3.1.5 PCR

PCA of (Pearson, 1901) is a dimensionality reduction method, which assumes a linear and static relation between the data and the latent factors. It computes principal components (PCs) over the full set of original data. They are linear combinations of the data that capture as much variation as possible while being orthogonal to the rest of the PCs by construction. We construct our PCs by pre-standardizing the set of explanatory variables and they have the following form:

$$PC_{i,t} = \sum_{j=1}^{k} \beta_{i,j,t} x_{j,t} = \beta_{i,t}^{T} X_t \; i = 1, ..., k,$$

where $\beta_{i,t}^{T}$ is the vector of parameter estimates, denoted as the loading vector of the i^{th} PC and k is equal to the number of covariates we use. It corresponds to the eigenvector with the i^{th} highest eigenvalue (λ_i), where the ratio of λ_i over the sum of all eigenvalues denotes the percentage of variance explained by the i^{th} PC.

We first run a PCA using the sklearn package for python, selecting enough components to cover 99% of the total variance each day. Then, we perform a simple cross-sectional linear regression using the principal components as the regressors for each day t, as shown in 3:

$$\sigma_{i,t} = \alpha_{0,t} + \sum_{j=1}^{m} \beta_{j,t} PC_j \; i = 1, ..., n. \tag{3}$$

This model reduces dimensionality and noise, as such providing a simpler final model and a more straightforward insight on the market structure and what are the main driving factors of the IVS.

3.1.6 RNN

RNN is a type of NN where the output from the previous step is fed as input to the current step for predicting the next step, where the RNN allows of this by having a hidden layer. The main feature of RNN is its hidden state, which remembers some information about a sequence. The state is also referred to as memory state since it remembers the previous input to the network. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output, hence reducing the complexity of parameters. The structure of an RNN is as shown in Figure 1:

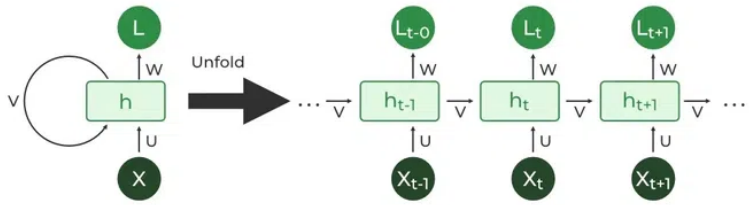


Figure 1: Recurrent Neural Network architecture

The Recurrent Neural Network consists of multiple fixed activation function units, one for each time step. Each unit has an internal state which is called the hidden state of the unit. This hidden state signifies the past knowledge that the network currently holds at a given time step. This hidden state is updated at every time step to signify the change in the knowledge of the network about the past. We use 'tanh' activation, where the formulas for calculating the hiddent state and output at each time step are given in Equations (4)-(5):

$$h_t = tanh(w_{hh}h_{t-1} + w_{xh}x_t), \tag{4}$$

$$y_t = w_{hy}h_t, \tag{5}$$

where h_t, x_t, y_t are the current state, input state and output state respectively while w_{hh}, w_{xh}, w_{hy} are the weight at recurrent neuron, at the input neuron and at the output neuron respectively. The parameters are updated through Bacpropagation through time (see Appendix A.2). The whole training process of a RNN model is as following:

Algorithm 3: Training RNN

1: We provide a single time step to the network as input.
2: We calculate the current state using a set of current input and previous state.
3: The current h_t becomes h_{t-1} for the next time step **4:** We go n_t time steps (equal to the cross-section of SPWX option at each day t) and join the information from all previous states
5: Once all time steps are done we use the final current state to calculate the output.
6: We compare the output to the target output, the validation IVS of the given day and errors are generated.
7: We backpropagate the error to the network to update the weights.
8: We use the tuned final model for predicting the next day.

We use a many-to-many, long-short-term-memeory (LSTM, explained in Appendix A.3.1) RNN structure, to include many regressors as input variables and to focus only in the information important for predicting the output, the IVS of the next day. We use our RNN as a replacement for the simple feedforward NN of Almeida et al. (2022), hence settle for a one LSTM layer with a vast 100 neurons and a one neuron Dense (see Appendix A.3.2) as our hidden layer to provide one forecast for each time step. We estimate the RNN using the versions 2.14.0 of keras and tensorflow packages for python. We compile the model using the 'adam' optimizer and fit over 100 epochs with a batch size of 1 with an early stop of patience 10 and minimum change of $1e^{-3}$.

RNN captures complex non-linear interactions between different factors over time and due to its memory of previous states it can capture the IVS and its evolution through time better. It is well suited for scenarios where the temporal dynamics (changes in the IVS through time) are of importance as it is in our research.

3.1.7 AE

An AE is a special type of unsupervised feedforward neural network. Differently from linear PCA, it captures complex nonlinear relations between the data and the factors. The AE consists of two parts: encoder and decoder. The decoder follows the encoder, and in the middle there is so called hidden layer. The encoder is a NN that constructs the factors using the original data, while the decoder is a NN that reconstructs the original data, given the factors which are estimated by the encoder. The architecture of a simple AE is depicted in Figure 2:

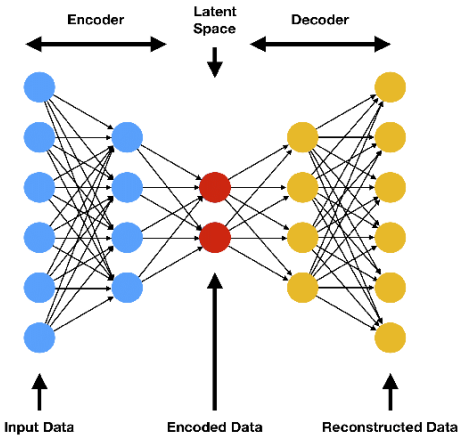


Figure 2: Autoencoder architecture

The encoder starts with the input layer which takes raw input data, followed by hidden layers which progessively reduce the dimensionality of the input capturing only important features. These layers make up the encoder and are followed by the final hidden layer, a bottle neck which represents the compressed encoding of the input data. It takes the encoded representation and expands it back to the original input. The hidden layers progressively increase the dimensionality and try to reconstruct the original input. The output layer produces the reconstructed output to resemble the initial input. We use the mean squared error (MSE) as our reconstruction loss function since our data is continuous (as opposed to binary cross-entropy) over the validation set of IVS of each day t. Having trained the model with the same fit parameters as the RNN, we use it to predict the IVS of the next day t^*.

We opt for a grid-based approach in our AE where the AE tries to find the factors that are best able to reconstruct the original data to retain the same structure between the input and output for IVS predictions. Running and forecasting over our 1297 trading days is computationally and time demanding, hence we adapt the results of Almeida et al. (2022), where they find a three layer NN with a one layer feedforward NN to achieve a good predictive performance. To that accord, we construct our encoder and decoders to have three LSTM layers each with 64, 32 and 8 neurons each and one Repeat Vector hidden layer (see Appendix A.3.3) and a Dense output layer with 1 neuron to obtain one prediction per time step. We use the 2.14.0 versions of tensorflow and keras packages for python.

The AE has most of the advantages of the PCR (feature extraction, dimensionality and noise reduction), but is not restricted to linear features, as it allows for non-linear relations. It provides a more compact represantion of the IVS and can also detect anomalies (differences from the more "typical" patterns the AE was trained on), which is useful in times of financial turmoil.

3.1.8 CNN

A CNN core building block consists of three operations: convolution, activation, and pooling. The core CNN building block is shown in Figure 3:

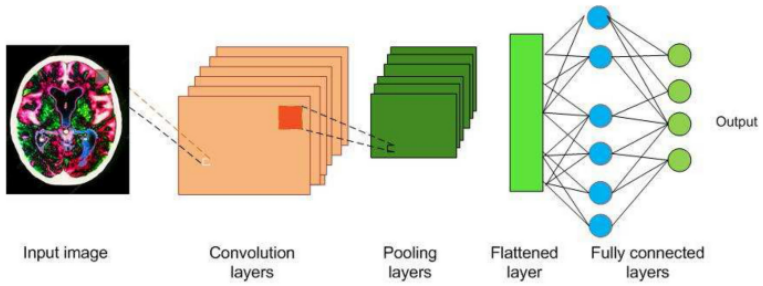


Figure 3: Convolutional Neural Network in image understanding

A convolutional layer convolves filters over the input data, extracting local spatial patterns and producing feature maps as its output. An activation function is a nonlinear function applied element-wise to the output of a layer. Pooling layers in CNNs are used to reduce the spatial dimensions of the feature maps produced by convolutional layers while retaining important information. The pooling operation involves dividing the input feature map into small regions and performing a summary operation, such as taking the maximum value (max pooling) or averaging the values (average pooling), within each region or computes the average value across the entire feature map (global average pooling). The fully-connected layers are employed towards the end of the network architecture to perform high-level reasoning based on the features extracted by earlier convolutional and pooling layers to provide an output.

CNNs are designed to capture spatial relationships between pixels in images, thanks to their use of convolutional layers. These layers apply filters to local regions of the input image, allowing the network to learn and identify patterns at various levels of abstraction. Furthermore, CNNs inherently possess translation and rotational invariance, meaning that they can recognize patterns and features regardless of their position in the image.

We apply a CNN to our IVS modelling problem by transforming our daily dataset into a three-dimensional tensor appropriate for a CNN (samples, timesteps, features). We have the samples (IV values across the surface on a given day) and the features (our set of regressors for the IVS), but are missing the timesteps. We obtain the latter by sacrificing 10 of our observation (a value which is less than 20% of the total observations for the day in our dataset with the fewest IV samples) and using them as timesteps. We use a Conv1D (see Appendix A.3.4) as our convolution layer with a MaxPooling1D (refer to Appendix A.3.5) for pooling. To fit the model to timeseries data we employ a vast LSTM layer with 100 neurons and conclude with a Dense layer that provides a single output for each sample. We estimate the model using the keras and tensor packages for python.

The CNN extracts spatial features and can identify patterns such as volatility smiles and skews across different regions since the IVS can be seen as a 2D grid of maturities and stike prices. It handles high dimensional data with lots of inputs, as ours, and possesses translation invariance, it recognizes patterns regardless of their position in the IVS.

3.1.9 EW

One of the ensemble methods we consider, is the simple $\frac{1}{N}$ average, following Clements and Vasnev (2021b) where they show improved accuracy from simple equally-weighted averaging. We combine the forecasts obtained across our models and generate the equally-weighted forecasts (EW) using the formula in Equation 6:

$$\hat{\sigma}_{i,t}^{EW} = \frac{1}{N} \sum_{i=1}^{N} \hat{\sigma}_{i,t}^{j}, \tag{6}$$

where $\hat{\sigma}_{i,t}^{j}$ is the forecasted value of the i^{th} option's IV at day t by model j.

This combination leverages the strengths of each individual model. It improves robustness and generalizations while reducing the risk of the errors associated with using a single model. The EW ensemble approach provides a well-rounded forecast and is ideal for broad coverage and stability across different market conditions.

3.1.10 EW2

Bali et al. (2023); Andersen et al. (2015); Gonçalves and Guidolin (2006) among others stress the presence of underlying non-linear relationships in IVS, which linear based models may miss and lead to poor predictions. The poorness of linear models impacts the EW forecast as well, to which we propose another simple equally-weighted ensemble model, EW2. EW2 utilizes the same concept as EW, but combines only forecasts from models that capture non-linear relations (AHBS, RF, RNN and AE). Since the numbers of models averaged over is smaller, the forecasts are more volatile, but they do not suffer from missing non-linear relations in the regressors.

This ensemble comprehensively captures more complex relationships due to its focus on non-linear models and as a consequence has enhanced pattern recognition. It adapts to the market quicker, especially in periods of high volatility where linear models may struggle. The EW2 ensemble approach is optimal in markets where non-linear relations dominate and it is crucial to adapt to these changes as quickly as possible and would be preferred from a high-volume intraday trading perspective.

3.2 GARCH-type models

In this section we go over the novel GARCH-type models we propose and the GARCH-type benchmarks we compare against for the daily S&P 500 returns' volatility. First, we explain the specifics of our set-up with regards to the GARCH-type models.

We calculate daily returns as the log difference, as they they offer a more accurate and insightful perspective on investment returns while being easier to continuously compound. The formula we use is as in Equation (7):

$$r_t = ln(\frac{P_t}{P_{t-1}}), \tag{7}$$

where P_t denotes the price of the S&P 500 on day t. We use the first 70% of the observations as our training set and conduct one step ahead forecasts over the remainder 30% test set, where the model is estimated up to each day t and used to forecast each day t+1 in our test set. We evaluate the performance of the models using the forecasted variance and use the squared daily returns as a proxy for the real variance of each day.

In our novel models we also use four external regressors pertaining to the IVS curve. To avoid look-ahead bias, we construct these variables by using the actual IVS values over the training set and using the IVS forecasts from our best performing model (which are also computed one day ahead) for the test set. E.g. the time series of the average across the IVS through time contains the average of the actual IVS for the days that are part of the training set and the average of the forecasted IVS for days in the test set.

It is noteworthy to emphasize the difference in train-test nature between the GARCH-type models and the IVS models, where the latter are trained and tested recursively throughout our whole dataset. In our paper, for the IVS models, we only report their performance over the entire dataset. However, when selecting the 'best' model/s to use for the IVS characteristics, we pick the model/s whose performance is/are better over the first 70% observations (representing the training set of the GARCH-type models). We do that to avoid any residual data-snooping effects and maintain representative results.

3.2.1 Benchmark 1: GARCH (1,1)

The first benchmark we use is the famous GARCH (1,1) of Bollerslev (1986). The full model is given in Euqation (8). The first line corresponds to the equation of the returns where they are regressed on a constant and a random error term ϵ scaled by the corresponding day's volatility. We assume normally distributed standard errors as it is the common approach in literature, Bollerslev (1986). Furthermore, the second line represents the variance equation, which is driven by the unexpected returns squared, that works as a proxy for the real return variance and a lagged term for the previous period's volatility to make the variance more dynamic, around a mean ω. The model captures the three stylized facts of returns.

$$\begin{aligned} r_t &= \mu + \sigma_t \epsilon_t, \\ \sigma_{t+1}^2 &= \omega + \alpha(r_t - \mu)^2 + \beta\sigma_t^2, \\ \epsilon_t &\sim \mathcal{N}(0,1). \end{aligned} \tag{8}$$

We estimate the model using the rugarch package for R, which maximizes the log-likelihood given in Equation (9) with the parameters estimated recursively starting from t = 1:

$$L(\theta|r) = \prod_{t=1}^{T} (2\pi\sigma_t^2)^{-\frac{1}{2}} exp(-\frac{1}{2\sigma_t^2}(r_t - \mu)^2). \tag{9}$$

3.2.2 Benchmark 2: TGARCH (1,1,1)

The second benchmark we use is the threshold GARCH model (TGARCH (1,1,1)) of Zakoian (1994). The model improves upon the GARCH model by incorporating different impacts from negative shocks compared to positive shocks to capture the risk-averse behavior of investors where losses are more detrimental than profits. We depict the full model in Equation (10):

$$\begin{aligned} r_t &= \mu + \sigma_t \epsilon_t, \\ \sigma_{t+1}^2 &= \omega + \alpha_1 (r_t - \mu)^2 \mathbb{1}_{\epsilon_t > 0} + \alpha_2 (r_t - \mu)^2 \mathbb{1}_{\epsilon_t \leq 0} + \beta \sigma_t^2, \\ \epsilon_t &\sim \mathcal{N}(0,1). \end{aligned} \tag{10}$$

We estimate the model using the rugarch package for R, which maximizes a similar log-likelihood as for the GARCH (1,1).

3.2.3 Novel 1: GARCH-V (1,1,1)

GARCH-V (1,1,1) is the first novel we propose and is a type of GARCH-X model (GARCH with exogeneous regressors). We aim to exploit the information contained by the IVS each day with regards to the underlying asset, S&P 500 daily returns. To that extent, we add a new variable to the variance equation of a regular GARCH (1,1) model corresponding to the average value of the IVS of a given day. The full model is given in Eqaution (11):

$$\begin{aligned} r_t &= \mu + \sigma_t \epsilon_t, \\ \sigma_{t+1}^2 &= \omega + \alpha (r_t - \mu)^2 + \beta \sigma_t^2 + \beta_t^{avg} IVS_t^{avg}, \\ \epsilon_t &\sim \mathcal{N}(0,1), \end{aligned} \tag{11}$$

where IVS_t^{avg} is the average across the IVS of the SPWX options on day t. We estimate the model using the rugarch package for R, but the maximization procedure becomes exponentially more intricate due to the presence of external regressors, leading to potential not invertible Hessians for the standard errors. For a more detailed explanation of how the optimization procedure works, we refer you to the official site of rugarch.[6]

3.2.4 Novel 1.2: TGARCH-V (1,1,1)

TGARCH-V (1,1,1) utilizes the same concept as our first novel model, however, it also incorporates the asymmetric shock effect, same as the TGARCH (1,1,1). The full model is given in Equation (12):

$$\begin{aligned} r_t &= \mu + \sigma_t \epsilon_t, \\ \sigma_{t+1}^2 &= \omega + \alpha_1 (r_t - \mu)^2 \mathbb{1}_{\epsilon_t > 0} + \alpha_2 (r_t - \mu)^2 \mathbb{1}_{\epsilon_t \leq 0} + \beta \sigma_t^2 + \beta_t^{avg} IVS_t^{avg}, \\ \epsilon_t &\sim \mathcal{N}(0,1). \end{aligned} \tag{12}$$

[6] https://cran.r-project.org/web/packages/rugarch/rugarch.pdf

The model is estimated in a similar manner as GARCH-V (1,1,1), but allowing for different effects from negative to positive shocks.

3.2.5 Novel 2: GARCH-V (1,1,3)

GARCH-V (1,1,3) is the second novel model we propose. In a similar manner to GARCH-V (1,1,1), we aim to extract information regarding the underlying from its IVS. The difference comes in how we extract that information. For GARCH-V (1,1,3), we do not restrict ourselves to the simple average across the IVS, but utilize the level, slope and curvature of the IVS, which are enough to manage characterizing a surface well (Afonso and Martins, 2010; Clarke, 2022). We hypothesize that having the defining characteristics of the whole IVS covered in our GARCH-X type model leads to better predictions of the underlying. The full model is shown in Equation (13):

$$\begin{aligned} r_t &= \mu + \sigma_t \epsilon_t, \\ \sigma_{t+1}^2 &= \omega + \alpha(r_t - \mu)^2 + \beta\sigma_t^2 + \beta_t^{lvl} IVS_t^{lvl} + \beta_t^{sl} IVS_t^{sl} + \beta_t^{cur} IVS_t^{cur}, \\ \epsilon_t &\sim \mathcal{N}(0,1), \end{aligned} \tag{13}$$

where $\beta_t^{lvl} IVS_t^{lvl}$, $\beta_t^{sl} IVS_t^{sl}$ and $\beta_t^{cur} IVS_t^{cur}$ are the level, slope and curvature of the IVS on day t respectively. The estimation of the model is the same as for GARCH-V (1,1,1), but with an even more difficult to compute Hessian due to the set of external regressors being multidimensional.

3.2.6 Novel 2.1: TGARCH-V (1,1,3)

TGARCH-V (1,1,3), similarly to TGARCH-V (1,1,1), is just an adaption of the corresponding novel model that allows for asymmetric shock effects. The full model is given in Equation (14):

$$\begin{aligned} r_t &= \mu + \sigma_t \epsilon_t, \\ \sigma_{t+1}^2 &= \omega + \alpha_1(r_t - \mu)^2 \mathbb{1}_{\epsilon_t > 0} + \alpha_2(r_t - \mu)^2 \mathbb{1}_{\epsilon_t \leq 0} + \beta\sigma_t^2 + \beta_t^{lvl} IVS_t^{lvl} + \beta_t^{sl} IVS_t^{sl} + \beta_t^{cur} IVS_t^{cur}, \\ \epsilon_t &\sim \mathcal{N}(0,1). \end{aligned} \tag{14}$$

We estimate the model in the same manner as TGARCH-V (1,1,1).

3.3 Evaluation Metrics

In this section we go over the evaluation metrics we use, containing the loss function we utilize and the tests we use. We employ the metrics over both, IVS forecasts and S&P 500 variance forecasts, over their respective test sets. Understandably, the metrics are the same, but the time frame, the forecasts and actuals used differ. Here, we change the notation of the IVs from $\sigma_{i,t}$ to $IV_{i,t}$ to differentiate between the IV forecasts and the S&P 500 volatility forecasts.

3.3.1 RMSE

We use the RMSE as our preferred loss metric, because it provides a measure of error that is in the same unit as the target variable, making it easier to interpret and compare. Furthermore,

RMSE penalizes larger errors more, making it more sensitive to outliers and more appropriate on a financial setting where we try to mitigate large losses. We provide the RMSE formula for all IVS horizons (h $\in$ {1,5,22}) in Equation (15) and for the S&P 500 volatility in Equation 16:

$$RMSE_{IV,h} = \sum_{t=h+1}^{T} RMSE_t = \sum_{t=h+1}^{T} \sqrt{\frac{\sum^n i = 1(IV_{i,t} - \hat{IV}_{i,t})^2}{n}}, \tag{15}$$

$$RMSE_\sigma = \sqrt{\frac{\sum_{t=0.7*T+1}^{T}(r_t^2 - \hat{\sigma_t}^2)^2}{0.3 * T}}, \tag{16}$$

where $IV_{i,t}$, r_t^2 and $\hat{IV}_{i,t}$, $\hat{\sigma}^2$ denote the actual value of the IV/S&P 500 volatility and the forecasted value of IV/S&P 500 volatility (for option i) on day t respectively. The IVS consists of panel data, in other words, a cross-section of IVs for each day t. To that avail, we compute the RMSE over the cross-section first to get an RMSE for each day in the corresponding out-sample (all but day t=1) and then sum up the daily RMSE's to get a one number final value. Meanwhile, for the volatility estimates, we just compute the RMSE over the corresponding out-of-sample period (the last 30% of the days in our dataset).

3.3.2 Superior Predictive Ability (SPA) test

The SPA test from Hansen (2005) tests whether a particular forecasting procedure is outperformed by alternative forecasting procedures. It does so by taking into account a specific loss function and testing whether any of the models are better than a specific benchmark in terms of expected loss. The null hypothesis of the test is that the benchmark is not inferior to any of the alternative models. More specifically, it defines the variable $d_{k,t}$ as the relative performance of model k compared to the benchmark at time t, formally:

$$d_{k,t} = L_{0,t} - L_{k,t},$$

for k=1,2,..,m, where $L_{k,t}$ is the value of the loss function for model k (k=0,1,2,..,m with 0 the benchmark model) at time t. When comparing the IVS models we use the RMSE computed over the cross-section on each day t as our $L_{k,t}$ for each model k. While for comparing the GARCH-type models we use simple residuals of the forecasts as our $L_{k,t}$ (we avoid using squared errors as the returns and volatility estimates are very small even before squaring them). The values of $d_{k,t}$ for each model 1 until m are put in a vector $\mathbf{d}_t$. The null hypothesis is then formulated as :

$$H_0 : E(\mathbf{d}_t) \leq 0.$$

We conduct the SPA test using the arch package for python comparing against the specified benchmarks. There are some disadvantages to this method. First of all, the rejection of the null only concludes one or more models as significantly better than the benchmark, but does not specify which models this concerns. To overcome that disadvantage we look at another test, namely the Stepwise Multiple Testing (StepM).

3.3.3 Stepwise Multiple Testing (StepM)

The StepM of Romano and Wolf (2005) is a procedure designed to control the family-wise error rate (FWER) in multiple hypothesis testing (probability of making one or more false discoveries or type I errors when testing multiple hypothesis). The method is useful in high-dimensional settings with many models to be tested, like ours, as it also provides which models explicitly outperform a certain benchmark. The premise of StepM is that one has several null hypothesis, one for each model besides the benchmark/s (H_1, ..., H_m), each with an associated p-value, p_k. The full algorithm of the StepM method is given below:

Algorithm 4: Conducting StepM

1: Order the p-values in ascending order, $p_{(1)} \leq p_{(2)} \leq ... \leq p_{(m)}$.

2: Set the significance level, $\alpha = 5\%$ and test the smallest p-value: if $p_{(1)} \leq \frac{\alpha}{m}$, reject $H_{(1)}$.

3: For k=2,..,m test the k-th smallest p-value: if $p_{(k)} \leq \frac{\alpha}{m-k+1}$, reject $H_{(k)}$.

4: Continue the third step until you reach k such that $p_{(k)} \geq \frac{\alpha}{m-k+1}$. At this point stop and do not reject any more hypothesis.

The theory relies on principles from order statistics and the properties of p-values under the null hypothesis. StepM is be particularly powerful when dealing with positively dependent test statistics, a common scenario in high-dimensional testing, especially when considering our equally-weighted combinations. It provides balance between controlling the error rate and maintaining statistical power, We use the 'arch' package for python fof the StepM test, using the same losses and benchmarks as for the SPA test. However, while StepM does extend on the SPA by giving the models that beat the benchmark, it does not say anything in terms of how the models rank. Hence, we conclude our tests with the Model Confidence Set (MCS) to know the best forecasting model/s.

3.3.4 Model Confidence Set (MCS)

The MCS is an approach with the advantage that it can compare the statistical significance of several forecasting methods at once. It creates a set of models that are the best from the total set of model with a certain probability. The steps of MCS are complex and explained in detail in Hansen et al. (2011a), however we give a brief overview of the general idea and algorithm. The MCS starts by setting a set of superior models as in (17):

$$\mathcal{M}^* := \{i \in M_0 : \mathbb{E}(d_{ij,t}) \leq 0, \forall j \in M_0\}, \tag{17}$$

where $d_{ij,t}$ is the loss differential between model i and j, at time t and M_0 is the set containing all models. The algorithm for the method is a series of consecutive tests where the null hypothesis of (18) is considered:

$$H_{0,M} : \mathbb{E}(d_{ij,t}) = 0 \quad \forall i, j \in M. \tag{18}$$

The algorithm starts with the full set M_0 and tests hypothesis 18 recursively to determine which models are kept and which are discarded from the superior set at a confidence level of α. MCS

also provides a p-value associated with each model in the set where the higher the p-value, the more likely the model is part of the superior set. For the MCS test, we use the 'arch' package for python, using the same losses as for the other two tests.

4 Results

In this section we go over our results in order, starting with the IVS models and then moving on to the GARCH type models.

4.1 IVS results

We first present the cumulative RMSE of all our IVS models through time in Figure 4, excluding the CNN model from the graph[7] due to its poor performance and keep the rest of the results in scale. It depicts the cumulative RMSE of the one day ahead (top), one week ahead (middle) and one month ahead (bottom) forecasts. We see that our results stay constant despite the horizon, with the linear models (ENET particularly) performing visibly worse. Meanwhile the benchmark, NN models and RF perform fairly well do to their ability to capture non-linear relations. However, our ensemble of non-linear models, EW2 attains the lowest cumulative RMSE across horizons.

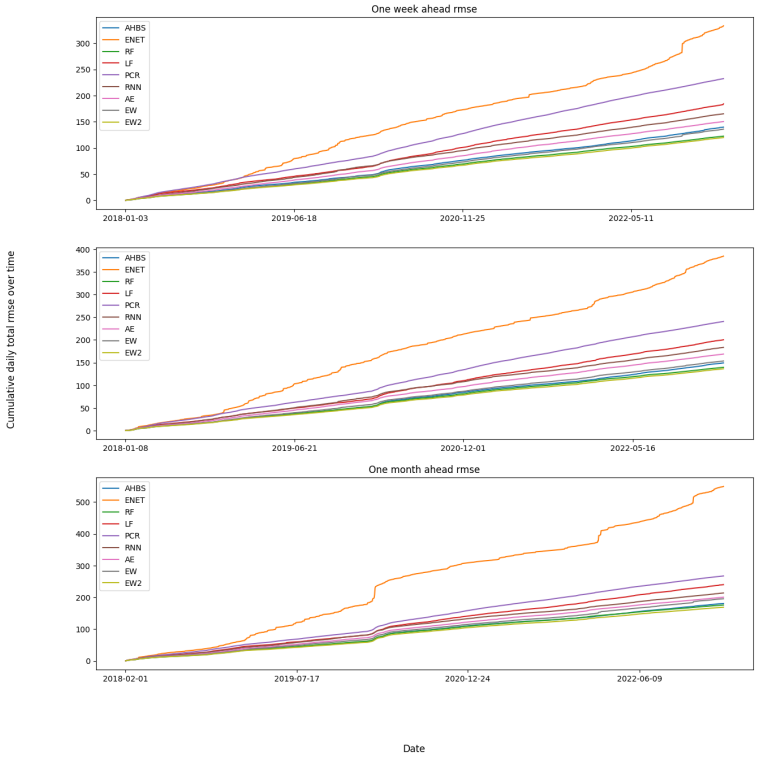


Figure 4: The cumulative RMSE of our IVS models through time for each horizon (without CNN)

[7]We provide the same graphs and further discussion simply for the CNN model in Appendix C.2 in Figure 10.

To accompany the graph with precise numbers, we provide in Table 2 the average RMSE of each IVS model throughout the days and stock cross-sections. We reaffirm our conclusion drawn from the graphs that linear models attain visibly worse RMSE's, while our proposed combination of non-linear methods consistently achieves the lowest RMSE's. A new conclusion we reach is the persistent drop in performance when increasing the horizon, no matter the horizon. The matter is especially of interest when considering the one week ahead and one month ahead forecasts have 5 days and 22 days less to sum over respectively.

	$RMSE_1$	$RMSE_5$	$RMSE_{22}$
AHBS	0.1078	0.1154	0.1415
ENET	0.2570	0.2976	0.4303
RF	0.0944	0.1079	0.1378
LF	0.1420	0.1549	0.1881
PCR	0.1792	0.1861	0.2095
RNN	0.1274	0.1419	0.1675
AE	0.1160	0.1305	0.1571
CNN	0.3422	0.3440	0.3490
EW	0.1048	0.1187	0.1532
EW2	0.0924	0.1054	0.1323

Table 2: Total RMSE of each IVS averaged over the cross-section of SPWX options on each day, then summed over the days (rounded to 4 decimal points).

Our results resembles a common belief regarding the stock market (Bollerslev (1986) base the GARCH model on this belief), that tomorrow's volatility is heavily impacted by the volatility today. Hence, the market on stocks, in particular SPWX 'weeklies', follows a similar pattern and today's IVS holds a lot of information regarding tomorrow's IVS. By definition SPWX are short-term options, hence it is as expected that their predictive power one month ahead is low, but should we also disregard the information they hold for one week ahead? Figure 5 shows how that is not the case. It depicts the RMSEs of all models besides ENET (we remove it to provide more interpretable graphs, the full graph including ENET is given in Appendix 9.

We see from the Figure 5 that the one week ahead RMSE ($RMSE_{22}$) has the least peaks even when compared to the one day ahead RMSE ($RMSE_1$). We conclude that the IVS contains useful information concerning its weekly term structure and there are benefits to predicting one week ahead. While the erorrs are slightly higher than when predicting one day ahead, the errors are less severe, making it useful from the perspective of a risk-averse investor. In our paper, we focus on attaining the smallest overall error possible, hence we utilize the one day ahead forecasts for our GARCH models, but it could be beneficial to explore a weekly GARCH volatility model in further research.

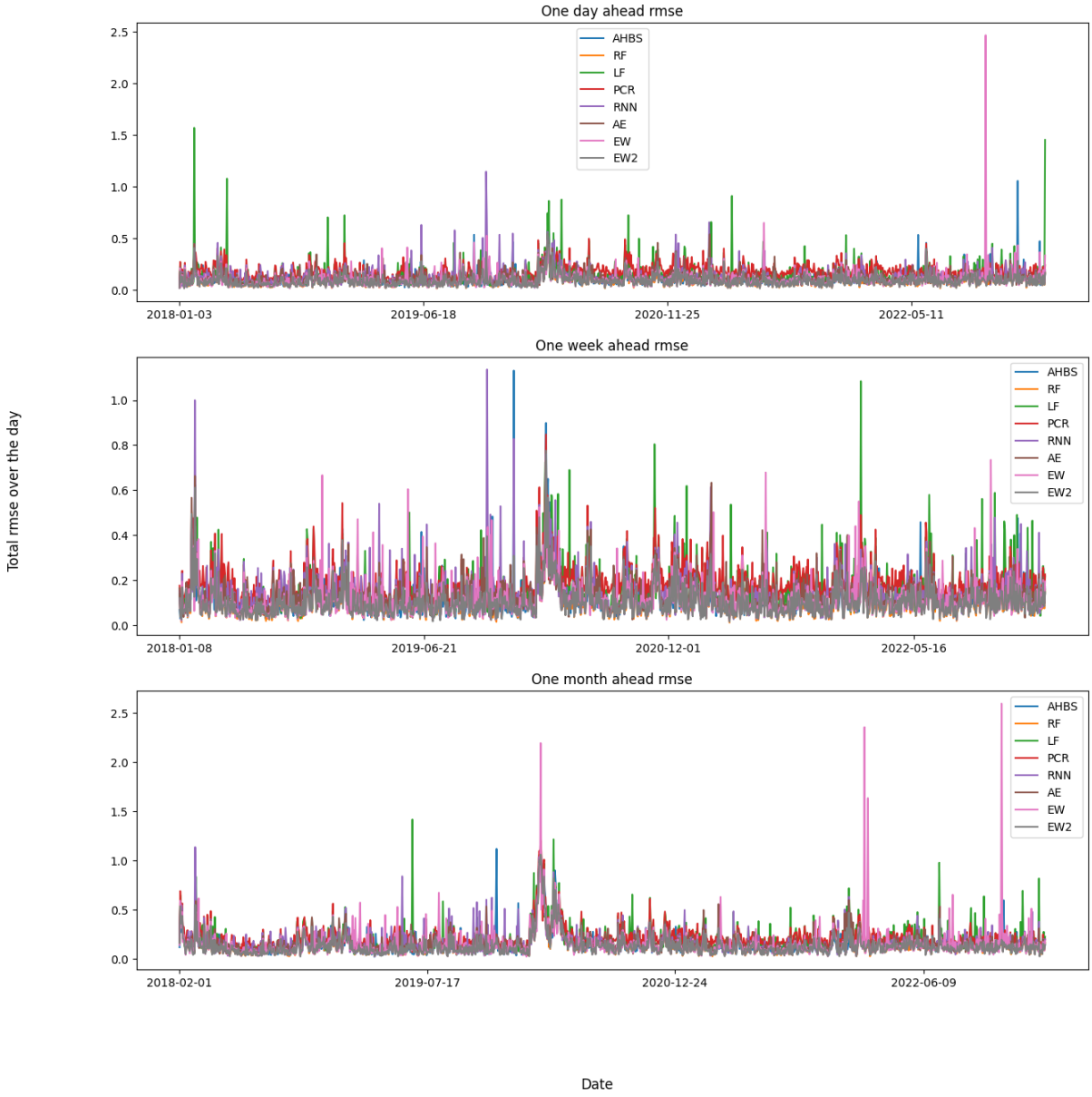


Figure 5: The RMSE of the cros-section of SPWX for all models on each day (without ENET and CNN) for each horizon.

In terms of statistical significance, our SPA tests return a p-value of 0.0, 0.001 and 0.002 for the different forecasting horizons in order, indicating sufficient evidence to say that at least one of our models beat the benchmark at the 5% significance level for each horizon. We apply the StepM test to extract the specific models and find that the benchmark (AHBS) is beaten by RF and EW2 for one day ahead forecasts, same for one week ahead and only by EW2 for one month ahead. Furthermore, the MCS test only includes EW2 in the set of superior predictive models. Considering that EW2 is also the bottom line in all six graphs at all times (Figures 4 and 5), we select its forecasts as our external regressor in our novel GARCH-type models. Also when using the average IVS across the cross-section on each day, the results indicate the superiority of the EW2 model, with a slight difference in p-values.

4.2 S%P 500 volatility results

In this section we show the results from the GARCH-type models. Table 3 contains the RMSE of each GARCH-type model over the corresponding out-of-sample period (last 30% observations). We do not present any regression outputs due to all models being re-estimated over the out-of-sample period and we are more interested in the predictive performance rather than the estimates of the regressors over time.

	GARCH	TGARCH	GARCH-V (1,1,1)	TGARCH-V (1,1,1)	GARCH-V (1,1,3)	TGARCH-V (1,1,3)
RMSE	0.009212	0.009242	0.008948	0.008629	0.008741	0.013163

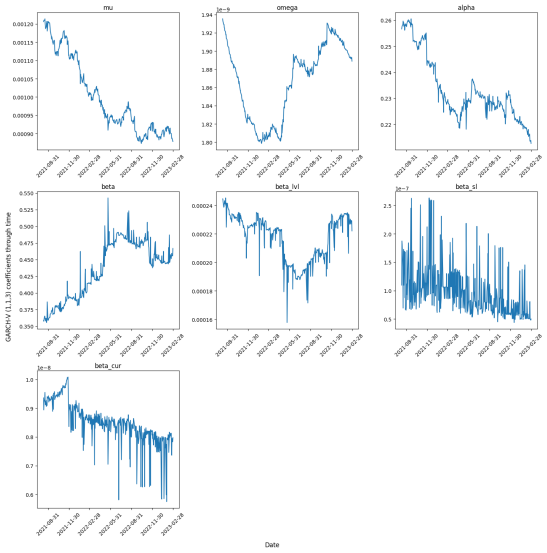
Table 3: The RMSE of each GARCH-type model over the out-of-sample period of the last 30% of the total number of days (rounded to 4 decimal points).

We see that almost all of our novel models beat the benchmarks in terms of RMSE with the exception of the TGARCH-V (1,1,3). Additionally, despite TGARCH-V (1,1,1) being our best performing model, we infer that the models incorporating the skew returns effect do not necessarily outperform their more 'basic' counterpart (e.g. GARCH (1,1) has a better performance than TGARCH (1,1,1)). The result indicates that the asymmetric effect of the returns does not make up for the added complexity from the extra term for our dataset. Nevertheless, a combination of the asymmetric term with the average forecasted IV (the TGARCH-V (1,1,1)) model manages to capture the nature of the returns the best. Since the improvement within the TGARCH models comes only when including the average IV, we conclude that the average IV term improves upon the omitted variable bias of a simple TGARCH (1,1,1) model and makes for a more parsimonious model. As such, we state that the IVS of an underlying index on any given day contains significant information regarding the return volatility of said index. The result speaks volumes to the S&P 500 return volatility to the IVS of its weekly options. Furthermore, we see that not all of the IVS information is useful in terms of predictive performance as the inclusion of slope and curvature reduces the model's effectiveness and can lead to overfitting as seen from the RMSE of TGARCH-V (1,1,3).

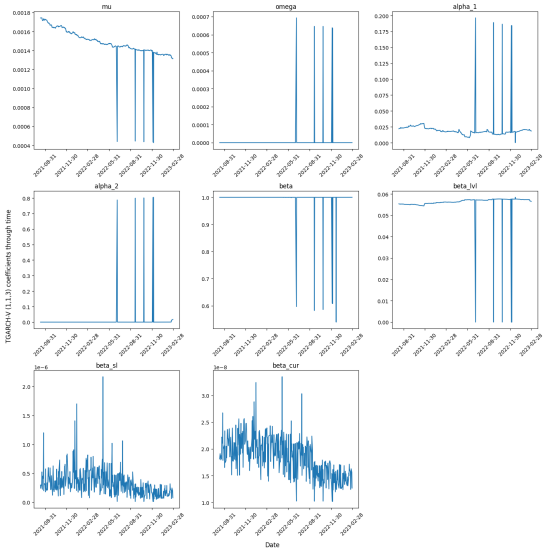
Another matter of interest is the large difference in performance between most of our novel models and TGARCH-V (1,1,3), where we conclude that the latter is overly complex and fails to model the nature of the returns properly despite its inherent ability to capture the so-called volatility smile. We argue that the volatility smile is often called the smirk/skew when it is asymmeytric, and as such, the asymmetric term associated with the TGARCH structure, not only captures the investor's bias to be impacted by negative shocks more, but also part of the volatility skew. Due to this overlapping, we claim that the estimation procedure of the TGARCH-V (1,1,3) is more prone to failure and overfitting. Our claim is supported by Figures 8 and 6 where we depict the coefficient estimates of the GARCH-V (1,1,3) and TGARCH-V (1,1,3) models over our out-of-sample set[8]. We see the TGARCH-V (1,1,3) estimates are, for lack of a better word, all over the place and highly unstable with consistent large, unprovoked spikes, especially

[8]The graphs of our other novel models T/GARCH-V (1,1,1) are provided in Appendix C.3, Figure 11, but not of the benchmarks since they are already prevalent in literature

when compared to its seemingly more robust counterpart GARCH-V (1,1,3). We find sufficient evidence in this figure say that the asymmetric term of a TGARCH model overlaps with the IVS characteristics and makes the estimation procedure unstable amd unreliable.



(a) Estimated coefficients of the GARCH-V (1,1,3) model through time



(b) Estimated coefficients of the TGARCH-V (1,1,3) model through time

Figure 6: Plots of coefficient estimates of the novel models containing IV_{lvl}, IV_{sl} and IV_{cur}.

We do not find any historical evidence to support these values, hence we attribute the observations as outliers caused by not proper convergence of the estimation procedure. We pose this occurrence as being in need of further research. The slope and curvature are the only stable estimates at very low values, as such, one could argue regarding their usefulness/significance and blame the variables for the few spurious estimates. To answer to that remark we estimate the TGARCH-V (1,1,3) model once more by setting β_{sl} and β_{cur} to 0. While we get more stable coefficient estimates[9], the results improve drastically with an out-of-sample RMSE of 0.009023). Due to this finding, we emphasize the collision of the asymmetyric term of the TGARCH models with the slope and curvature IVS chracteristics and urge further research on a more stable estimation procedure for its coefficients. To test whether the results do generalize, we repeat the GARCH-type analysis using the forecasts from another model, AHBS (depicted in Appendix C.4).

In terms of statistical significance we conduct our progressive battery of tests. The results for both benchmarks (GARCH (1,1) and TGARCH (1,1,1)) are very similar, with the p-value of the SPA test being 0.0 in both cases. Thus, we know that at least one of our models beats both benchmarks and according to the StepM test, three of our novel models beat the GARCH (1,1) benchmark, namely GARCH-V (1,1,1), TGARCH-V (1,1,1) and GARCH-V (1,1,3), while only two beat the TGARCH (1,1,1) benchmark, TGARCH-V (1,1,1) and GARCH-V (1,1,3). Lastly, the MCS test results in only the TGARCH-V (1,1,1) being included in the set of superior mod-

[9]We do not report the results/graphs in our paper since the model does not offer any new information in itself

els, leading to us concluding that our model is a better approach to modelling daily returns than some common approaches used in literature (Bollerslev, 1986; Zakoian, 1994; Hansen and Lunde, 2001) and that the IVS characteristics contain useful information regarding the return movements of their underlying, especially the average IV across the surface on each given day.

We also provide in Figure 7 the actual volatility, as proxied by the daily returns alongside the prediction from our various GARCH-type models. We see a very clear result the explains the under-performance of the TGARCH-V (1,1,3) model, being that the scale of those forecasts is a lot smaller than the actual values than the other models, which can be attributed to the very small values of its regressors (the values of the slope and curvature of the IVS are around $10e^{-4}$). Structurally, as expected, the GARCH-type models are smoother than the squared daily returns, but follow a similar pattern with spikes in the same dates where the returns also spike to indicate periods of high turmoil. We see that the TGARCH model often captures the larger spikes in the actual volatility more appropriately, but also overestimates the cases of lower volatility just as often. Overall, the TGARCh-V (1,1,1) offers the best trade-off between under and overestimating the real volatility values. Hence, we are set on our novel model being a better alternative to volatility forecasting and the relation between the underlying's volatility and its IVS structure.

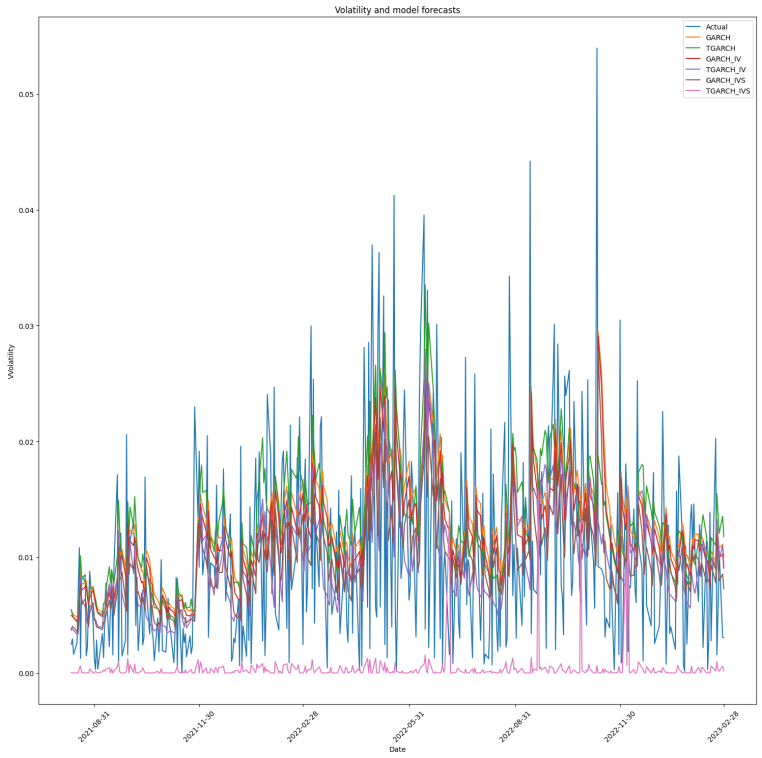


Figure 7: S&P 500 volatility and forecasts over the out-of-sample period

5 Conclusions

In this paper we introduce and use a new two-step approach to a stock's daily volatility modelling, where we first model/forecast its daily IVS and then utilize the information in said IVS to improve the prediction of the underlying's daily volatility. We estimate a total of ten models for the IVS using a day-to-day estimation procedure where we train on each day t and forecast day t+h, for three horizons (h$\in\{1,5,22\}$ for our whole sample of 853805 observations partaining to 1298 trading days. We estimate another six daily volatility models, two benchmark and four novel models proposed by us, over the first 70% of total days and conduct one step-ahead forecasts and re-estimations for the remainder 30% test set. We use three different tests, SPA, StepM and MCS to asses the predictive power and statistical performance of our models. We find evidence to support the benefits of using our novel two-stage approach to volatility modelling.

We find the equally-weighted average of the non-linear models (EW2) attains the best forecast by far and we hence use its forecasts in our subsequent analysis. Thre of our four extensions of famous volatility models, GARCH (1,1) and TGARCH (1,1,1), that use the information from IVS forecasts of the EW2 model, all beat traditional benchmarks in daily volatility modelling. The only under-performing model is the TGARCH-V (1,1,3) model, which we examine detail and attribute its poor predictive power to high estimation instability arising from an overlapping of the asymmetric returns effect with the IVS characteristics. Our statistically best performing model is the TGARCH-V (1,1,1), to which we claim the importance of considering the average of the IVS curve and the individual investor's risk averse nature when modelling a stock's daily volatility, but alsoraise awareness to the potential instability that stems in coefficient estimates when considering the full IVS shape features.

Our paper can be extended by first finding a more stable estimation procedure for the coefficients of TGARCH-V (1,1,3). There are other models to try both in terms of IVS modelling and volatility modelling. A CNN application to the images of the IVS, in the spirit of Kelly et al. (2023) may provide useful insights and improve our two-step approach. Also, a mixed-data sampling (MIDAS) model (Ghysels et al., 2004) including data at different frequencies where the volatility data or additional regressors for the IVS models (indeces, FX rates etc.) are daily, but the IVS data is cross-sectional on the day. Another lead to follow, probably the most interesting one, is two condense our approach into a singular step by incorporating different stochastic differential equations (SDEs) or other non-linear equations for the IVS (e.g. AHBS) into the estimation procedure of the GARCH-type models and solve the complicated problem in one step to avoid increased variance.

5.1 Limitations and acknowledgments

We acknowledge that our paper can be seen as an optimistic higher bound on the performance of our novel approach due to the look-ahead bias contained in the IVS explanatory variables.

We model IVS at day t using variables of dat t which may not be known in advance (e.g. option-specific variables). While one can incorporate individual or pooled models for said variables, it increases the variance of the final volatility estimates and impacts performance. Additionally, our approach is time-intensive and to that extent we have not tested its generalization to other asset classes or even stocks of a different volatility profile to the S&P 500 nor different return horizons.

References

A. Afonso and M. M. F. Martins. level, slope, curvature of the sovereign yield curve, and fiscal behaviour. *ECB working paper*, 2010.

Y. Ait-Sahalia, C. Li, and X. Li, C. Implied stochastic volatility models. *The Review of Finanical Studies*, 120(1):1–20, 2016.

C. Almeida, J. Fan, G. Freire, and F. Tang. Can a Machine Correct Option Pricing Models? *Journal of Business & Economic Statistics*, 41(3):995–1009, 2022.

C. Almeida, G. Freire, and R. Hizmeri. 0dte asset pricing. *SSRN Working Paper*, 2024.

T. Andersen, N. Fusari, and V. Todorov. Short-Term Market Risks Implied by Weekly Options. *The Journal of Finance*, 72(3):1335–1386, 2017.

T. G. Andersen, N. Fusari, and V. Todorov. The risk premia embedded in index options. *Journal of Financial Economics*, 117(3):558–584, 2015.

M. Avellaneda, B. Healy, A. Papanicolaou, and G. Papanicolaou. Pca for implied volatility surfaces. *The Journal of Financial Data Science*, 2(2):85–109, 2020.

T. Bali, J. Hu, and M. Scott. Option implied volatility, skewness, and kurtosis and the cross-section of expected stock returns. *Social Science Research Network*, 2013.

T. G. Bali, H. Beckmeyer, M. Mörke, and F. Weigert. Option Return Predictability with Machine Learning and Big Data. *The Review of Financial Studies*, 36(9):3548–3602, 2023.

F. M. Bandi, N. Fusari, , and R. Ren'o. 0dte option pricing. *SSRN Working Paper*, 2023.

J. Berkowitz. On justifications for the ad hoc black-scholes method of option pricing. *Studies in nonlinear dynamics and econometrics*, 14(1), 2009.

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

T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.

D. T. Breeden and R. H. Litzenberger. Prices of state-contingent claims implicit in option prices. *Ssrn.com*, 1978.

L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

A. Böök and C. Sala. The forecasting power of short-term options. *Social Science Research Network*, 2020.

J. M. Campa and Chang. Testing the expectations hypothesis on the term structure of volatilities in foreign exchange options, 2023.

L. Canina and S. Figlewski. The informational content of implied volatility. *The Review of financial studies*, 6(3):659–681, 1993.

P. Carr and L. Wu. Analyzing volatility risk and risk premium in option contracts: A new theory. *Journal of Financial Economics*, 120(1):1–20, 2016.

D. Chen, B. Guo, and G. Zhou. Firm fundamentals and the cross-section of implied volatility shapes. *Journal of financial markets*, 63, 2023.

P. Christoffersen and K. Jacobs. The importance of the loss function in option valuation. *Journal of Financial Economics*, 72(2):291–318, 2004.

C. Clarke. The level, slope, and curve factor model for stocks. *Journal of Financial Economics*, 143(1):158–187, 2022.

A. Clements and A. Vasnev. Forecast combination puzzle in the har model. *Social Science Research Network*, 2021a.

A. Clements and A. L. Vasnev. Forecast combination puzzle in the har model. *SSRN working paper*, 2021b.

M. Davis. The dupire formula. *Imperial College London, Finite Difference Methods Course material*, 2011.

T. G. Dietterich. Ensemble methods in machine learning. *Lecture Notes in Computer Science*, page 1–15, 2000.

A. Dixit and S. Singh. Ad-hoc black–scholes vis-à-vis tsrv-based black–scholes: Evidence from indian options market. *Journal of Quantitative Economics*, 16(1):57–88, 2017.

B. Dumas, J. Fleming, and R. E. Whaley. Implied volatility functions: Empirical tests. *The journal of finance*, 53(6):2059–2106, 1998.

G. Freire and O. Kleen. Equity option prices and firm characteristics. *SSRN Working Paper*, 2024.

R. Friedberg, J. Tibshirani, S. Athey, and S. Wager. Local linear forests. *Journal of Computational and Graphical Statistics*, 30(2):503–517, 2018.

E. Ghysels, P. Santa-Clara, and R. Valkanov. The midas touch: Mixed data sampling regression models. *UCLA: Finance*, 2004.

S. Gonçalves and M. Guidolin. Predictable dynamics in the s&p 500 index options implied volatility surface. *The Journal of Business*, 79(3):1591–1635, 2006.

C. Granger and Z. Ding. Some properties of absolute return: An alternative measure of risk. *Annales d'Économie et de Statistique*, 40:67–91, 2024.

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

S. Gu, B. Kelly, and D. Xiu. Autoencoder asset pricing models. *Journal of Econometrics*, 222 (1):429–450, 2021.

Q. Han, J. Liang, and B. Wu. Cross economic determinants of implied volatility smile dynamics: Three major european currency options. *European Financial Management*, 22(5):817–852, 2015.

P. Hansen, A. Lunde, and J. Nason. The model confidence set. *Econometrica*, 79(2):453–497, 2011a.

P. R. Hansen. A Test for Superior Predictive Ability. *Journal of Business & Economic Statistics*, 23(4):365–380, 2005.

P. R. Hansen and A. Lunde. A forecast comparison of volatility models: Does anything beat a garch(1,1)? *Social Science Research Network*, 2001.

P. R. Hansen, Z. Huang, and H. H. Shek. Realized GARCH: a joint model for returns and realized measures of volatility. *Journal of Applied Econometrics*, 27:877–906, 2011b.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. ELSEVIER, 2009.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

A. Hoerl and R. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

J. Hull and A. White. Optimal delta hedging for options. *Journal of Banking and Finance*, 82: 180–190, 2017.

B. T. Kelly, B. Kuznetsov, S. Malamud, and T. A. Xu. Deep learning from implied volatility surfaces. *Swiss Finance Institute Research Paper*, 2023.

V. Le and R. Zurbruegg. Forecasting option smile dynamics. *International Review of Financial Analysis*, 35:32–45, 2014.

D. Liu, Y. Liang, L. Zhang, P. Lung, and R. Ullah. Implied volatility forecast and option trading strategy. *International Review of Economics & Finance*, 71:943–954, 2021.

S. Mixon. Factors explaining movements in the implied volatility surface. *The journal of futures markets*, 22(10):915–937, 2002.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

S. Muzzioli. option-based forecasts of volatility: an empirical study in the dax-index options market, 2010.

B. Ning, S. Jaimungal, X. Zhang, and M. Bergeron. Arbitrage-free implied volatility surface generation with variational autoencoders. *Siam Journal on Financial Mathematics*, 14(4): 1004–1027, 2023.

K. O'Shea and R. Nash. An introduction to convolutional neural networks, 2015.

K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

D. Rapach, J. Strauss, and G. Zhou. Out-of-sample equity premium prediction: Combination forecasts and links to the real economy. *The Review of Financial Studies*, 23(2):821–862, 2009.

J. P. Romano and M. Wolf. Stepwise multiple testing as formalized data snooping. *ECONOMETRICA*, 73(4):1237–1282, 2005.

M. Rubinstein. Implied binomial trees. *The Journal of finance (New York. Print)*, 49(3):771–818, 1994.

L. S. Shapley. *The Shapley Value: Essays in Honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

G. Skiadopoulos, S. Hodges, and L. Clewlow. he dynamics of the s&p 500 implied volatility surface. *Review of Derivatives Research*, 3(3):263–282, 2000.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58, 1996.

A. Timmerman. *Handbook of Economic Forecasting*, volume 1. ELSEVIER, 2006.

H. R. Varian. Big data: New tricks for econometrics. *Journal of Economic Perspectives*, 28(2): 3–28, 2014.

Y. Wang, L. Liu, and C. Wu. Forecasting the real prices of crude oil using forecast combinations over time-varying parameter models. *Energy Economics*, 66:337–348, 2017.

Y. Wen. How Do Options Affect the Volatility of the Underlying Equity Market? Evidence from the Introduction of Weekly Options. *Journal of Derivatives*, 27(4):89–107, 2020.

J. M. Zakoian. Threshold heteroskedastic models. *Journal of Economic Dynamics and Control*, 18(5):931–955, 1994.

Y. Zeng and D. Klabjan. Online adaptive machine learning based algorithm for implied volatility surface modeling. *arXiv Working Paper*, 2017.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net, 2005.

A Additional Concepts

A.1 Decision Trees

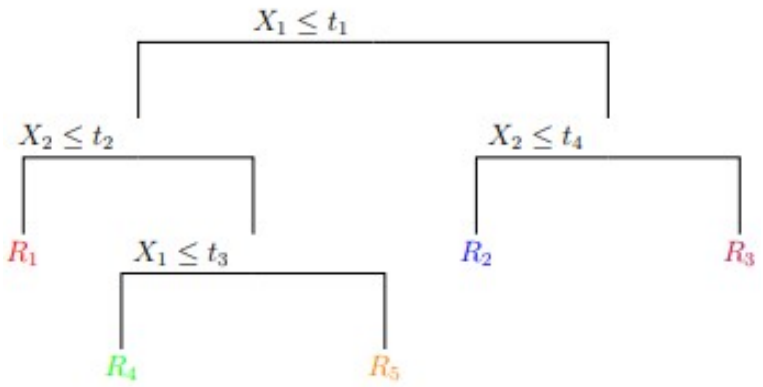


Figure 8: Example of a simple decision tree

The idea of Decision Trees (DT) is to partition the feature space into smaller regions until a final region is reached. The goal in training the Decision Tree is to find splitting rules of the feature space that provide the most consistent leafs for the predictions. This is done by minimizing the respective loss function (criterion). A criterion is used to evaluate the quality of splits during the tree-building process. For classification purposes, the following two criteria are most often used:

- Gini Index

$$c_{\text{Gini}}(\mathcal{T}) = |I_{\mathcal{T}}| \sum_{k} p_{\mathcal{T}}(k)\big(1 - p_{\mathcal{T}}(k)\big)$$

  where $\mathcal{T}$ denotes the leaf (region), $|I_{\mathcal{T}}|$ is the set of features in leaf $\mathcal{T}$ and $p_{\mathcal{T}}$ represent the average probability for class k in a region $\mathcal{T}$. This measure thus describes how often a randomly chosen training point would be misclassified if it was randomly labeled based on $p_{\mathcal{T}}$.

- Entropy Criterion defined as

$$c_{\text{entropy}}(\mathcal{T}) = -|I_{\mathcal{T}}| \sum_{\substack{k \\ p_{\mathcal{T}}(k) \neq 0}} p_{\mathcal{T}}(k) \log p_{\mathcal{T}}(k)$$

  measures the uncertainty of training points in leaf $\mathcal{T}$ with respect to their class labels.

The training algorithm of DT as introduced in Breiman et al. (1984), in their famous book Classification and Regression Tree (CART), runs in a following way:

1. At the beginning, all training data are in a single node, called the root node. Here starts the node splitting, it is done in a so called greedy way, i.e., the algorithm for loop over all features (J) and their unique values (S) to find such a splitting rule that minimizes

the criterion value. The goal is to obtain new nodes such that after the split the overall criterion is the lowest possible, this corresponds to minimizing the difference of after split and before split criterion value over all possible splits, i.e., $\min c_{\mathcal{T}_L} + c_{\mathcal{T}_R} - c_{\mathcal{T}}$, where $c_{\mathcal{T}_L} + c_{\mathcal{T}_R}$ describes the overall criterion value after split and $c_{\mathcal{T}}$ is the before split criterion.

2. Once the optimal splitting point is established based on the criterion value, two new child nodes emerge. The same procedure as in step 1. is repeated on the newly created nodes.

3. The splitting of new nodes is done until a pre-defined constraint is reached: the tree reaches maximum depth, minimum number of training points on a node to split on is matched or a maximum number of leaf nodes is attained.

4. In order to prevent overfiting, prunning based on cost complexity is usually preformed

$$C_\alpha(\mathcal{T}) = \sum_{m=1}^{|\mathcal{T}|} \sum_{x_j \in \mathcal{T}_m} c_m + \alpha|\mathcal{T}|$$

where m denotes the respective nodes and c_m is the criterion used for splitting the tree. This approach prevent overfiting by penalizing for a large trees, the constant α determines the strength of the penalty on tree size. Note that α needs to be tuned to obtain its value.

To sum up, DTs are a versatile method for modelling non-linear relationships and as a consequence are used in a wide range of application. What is more, DTs tend to be popular due to its good relative interpretability compared to other machine learning methods, e.g. by using Shapley values (Shapley, 1988). A disadvantage that can rise is in relation to the high variance across trees, to which avail bagging (Breiman, 1996) and random variable selection (Breiman, 2001) are incorporated within the trees to reduce the correlation present.

A.2 Backpropagation through time

In RNN the neural network is in an ordered fashion and since in the ordered network each variable is computed one at a time in a specified order like first h_1 then h_2 then h_3 and so on. Hence we apply backpropagation throughout all these hidden time states sequentially. We first define:

1. L(θ) - the loss function (in our case MSE) depending on h3
2. h_3 - the third state, depending on h2 and weights, w
3. h_2 - the second state, depending on h1 and w
4. h_1 - the first state, depending on h0 and 2
5. h_0 - a constant starting state
6. w - the weight matrix which remains the same across the network, a characteristic of RNN

We apply backpropagation on only one row for simplicity, leaving us with $\frac{\partial L(\theta)}{\partial w} = \frac{\partial L(\theta)}{\partial h_3}\frac{\partial h_3}{\partial w}$. The first term of the equation, $\frac{\partial L(\theta)}{\partial h_3}$ is easy to compute as it is the same as any simple deep NN backpropagation. The difficulty arises with the second term, $\frac{\partial h_3}{\partial w}$, where we need to consider both the explicit (treating all other inputs as constant) and implicit (summing over all indirect paths from h3 to W). If we set the explicit part as $\frac{\partial h_3^+}{\partial w}$, we get the expression in Equation 19:

$$\frac{\partial h3}{\partial w} = \frac{\partial h_3^+}{\partial w} + \frac{\partial h_3}{\partial h_2} \cdot \frac{\partial h_2}{\partial w} = ... = \sum_{k=1}^{3} \frac{\partial h_3}{\partial h_k} \cdot \frac{\partial h_k}{\partial w}. \tag{19}$$

This algorithm consitutes the Backpropagation through time as we go over all previous time steps, but can encounter gradient problems, such as an exploding or vanishing slope making the training of a RNN a not easy task.

A.3 NN layers

In this section we give a short explanation of the different types of layers used in our NN methods.

A.3.1 LSTM

An LSTM layer is an RNN layer that learns long-term dependencies between time steps in time-series and sequence data aimed at dealing with the vanishing gradient problem thus helping the training of a RNN. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. Forget gates decide what information to discard from a previous state by assigning a previous state, compared to a current input, a value between 0 and 1. A (rounded) value of 1 means to keep the information, and a value of 0 means to discard it. Input gates decide which pieces of new information to store in the current state, using the same system as forget gates. Output gates control which pieces of information in the current state to output by assigning a value from 0 to 1 to the information, considering the previous and current states. Selectively outputting relevant information from the current state allows the LSTM network to maintain useful, long-term dependencies to make predictions, both in current and future time-steps. For a detailed explanation we refer you to the work of Hochreiter and Schmidhuber (1997).

A.3.2 Dense

A Dense layer is a fully/densely-connected neural network. The primary advantage of dense layers is that they are able to capture complex patterns in data by allowing each neuron to interact with all the neurons in the previous layer. On the other hand they are omputationally expensive to train and, in addition, can suffer from overfitting. To lessen the disadvantages from Dense layers, we use them with a low number of neurons to mitigate the overfitting and computational time.

A.3.3 Repeat Vector

A Repeat Vector layer repeats the input a specified number of times. We use the layer with an argument of one as a bridge between the encoder and decoder to reduce the dimensionality of the input regressors to one which is the required shape for the output of each time step.

A.3.4 Conv1D

The 1D convolution layer creates a convolution kernel that is convolved with the layer input over a single spatial (or temporal) dimension to produce a tensor of outputs and adds a bias vector to the results.

A.3.5 MaxPooling1d

The layer conducts the max pooling operation for 1D temporal data. It downsamples the input representation by taking the maximum value over a spatial window of a given size, where the window is shifted by the same amount as the given size.

B Model validation

Three of our models (ENET, RF and LF) have parameters that need to be pre-determined before estimation. To ensure we utilize our models to their full extent we set a validation set to pick the parameter values that would lead to the best performance. During each day when we estimate our models, we randomize 20% of our IVS observations for the day as a validation set over which we test a grid of potential values for the parameters, using the negative mean absolute error as the performance criteria.

For the ENET model, we specify a range of values for α and λ and utilize over our validation set on each given day the hyperparameter grid depicted in Table 4:

Parameter	Values
λ	{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}
α	{$1e^{-5}$, $1e^{-4}$, $1e^{-3}$, $1e^{-2}$, $1e^{-1}$, 0, 1, 10, 100}

Table 4: Grid of hyperparameters ENET This table displays all the hyperparameters used in the Grid Search for ENET over our validation set

For the RF model, we specify a grid containing the number of trees used (n_estimators); the maximum depth of the tree (max_depth) and the number of features to consider when looking for the best split (max_features). The hyperparameter grid is depicted in Table 5:

For the LF model, we specify a grid containing the maximum depth of the tree (max_depth); the minimum number of samples required to split an internal node (min_samples_split) and the

Parameter	Values
n_estimators $\{10, 100, 500, 1000\}$ max_depth	{None, 1, 10, 50}
max_features	{auto, sqrt}

Table 5: Grid of hyperparameters RF This table displays all the hyperparameters used in the Grid Search for RF over our validation set

minimum number of samples required to be at a leaf node (min_samples_leaf). The hyperparameter grid is depicted in Table 6:

Parameter	Values
max_depth	{None, 5, 10, 20}
min_samples_split	{1, 3, 6, 9}
min_samples_leaf	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}

Table 6: Grid of hyperparameters LF This table displays all the hyperparameters used in the Grid Search for LF over our validation set

C Additional Results

C.1 RMSE of all models across horizons

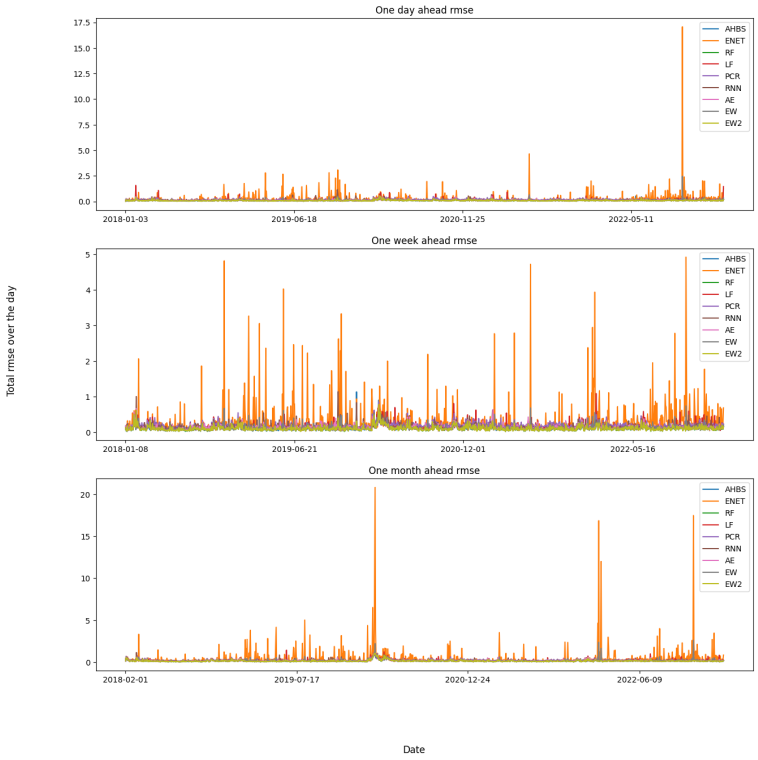
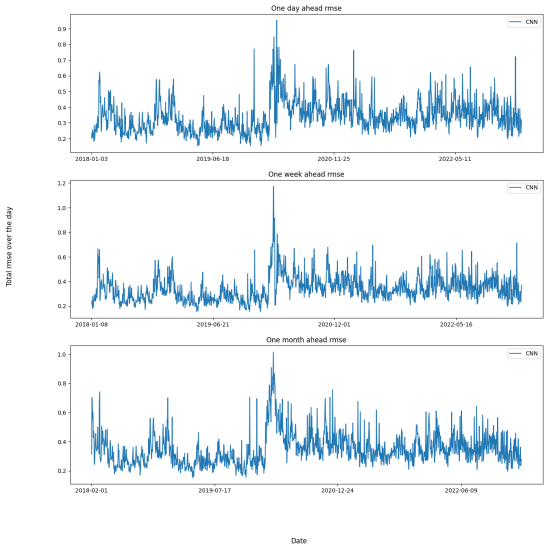


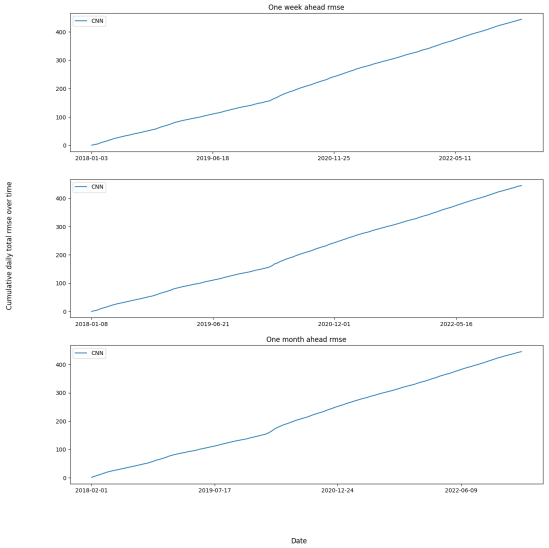
Figure 9: The RMSE of the cros-section of SPWX for all models on each day

C.2 Results for CNN

We know the model offers by far the worst predictions. We take a closer look to determine the reason of this under-performance. We observe that the CNN model severely underestimates the actual IVS values and offers forecasts that are almost exclusively near zero. Figure 10a shows that the RMSEs of each day are relatively low, however Figure 10b shows how the cumulative RMSE increases at a sharper slope than for all other models. Hence, we state that the CNN model makes very small predictions, hence not obtaining that large of an RMSE on a daily basis, but it is consistently wrong in doing so, hence it attains the highest RMSE across all horizons. We conclude that the CNN model should not be applied on a day to day estimation procedure as we do, however it is a useful tool when applied to a train set consisting of IVS graphs of different days as done in Kelly et al. (2023).



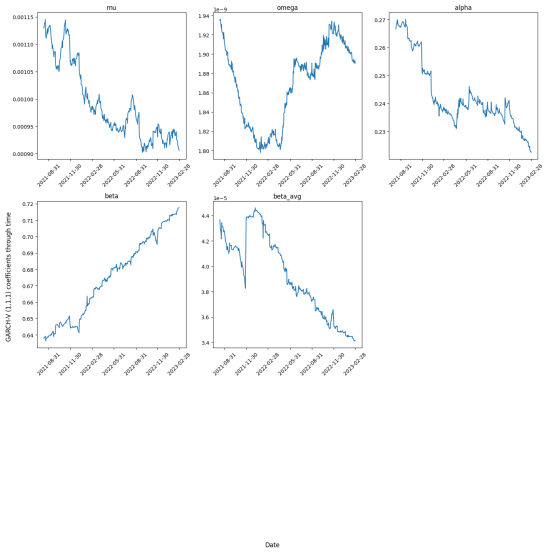
(a) The RMSE of the cross-section of SPWX for the CNN model on each day for each horizon.



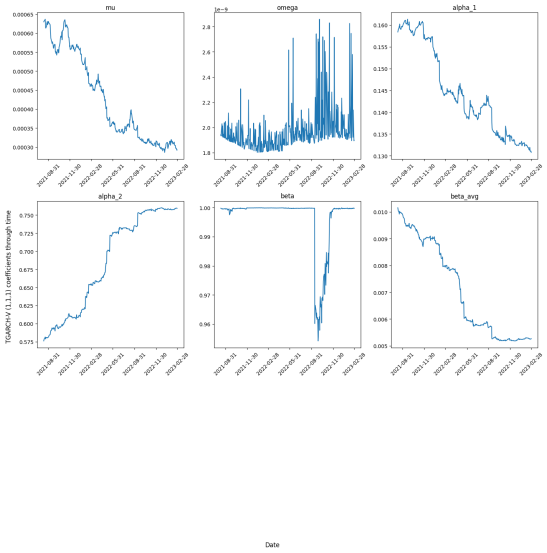
(b) The cumulative RMSE of our IVS models through time for each horizon for the CNN model.

Figure 10: RMSE graphs for the CNN model

C.3 GARCH-type models' coefficients



(a) Estimated coefficients of the GARCH-V (1,1,1) model through time



(b) Estimated coefficients of the TGARCH-V (1,1,1) model through time

Figure 11: Plots of coefficient estimates of the novel models containing IV_{avg}

C.4 RMSE results when using AHBS IVS forecasts in the GARCH-type models

Table 7 contains the RMSE of our different GARCH-type models when using the IVS forecasts from the AHBS model (instead of EW2). We see the same pattern, with slightly different numbers, where the TGARCH-V (1,1,1) is superior to the other models. Hence, we firmly state that the performance of the model is not spurious to the model which is used for the IVS forecasts. Additionally, we see a slightly worse performance from all of our novel models that make use of IVS forecasts, emphasizing the importance of having as good a model as possible for IVS modelling in itself.

	GARCH	TGARCH	GARCH-V (1,1,1)	TGARCH-V (1,1,1)	GARCH-V (1,1,3)	TGARCH-V (1,1,3)
RMSE	0.009212	0.009242	0.009107	0.008728	0.008809	0.013562

Table 7: The RMSE of each GARCH-type model over the out-of-sample period of the last 30% of the total number of days (rounded to 6 decimal points)

D Common abbreviations

1. **IV** - implied volatility
2. **IVS** - implied volatility surface
3. **SPWX** - weekly options on the S&P 500
4. **AHBS** - ad-hoc Black and Scholes model

5. **ML** - machine learning
6. **ENET** - elastic net
7. **RF** - random forest
8. **LF** - linear forest
9. **DL** - deep learning
10. **CNN** - convolutional neural network
11. **NN** - neural network
12. **PCR** - principal component regression
13. **AE** - autoencoder
14. **EW** - equally weighted combination
15. **EW2** - equally weighted combination of non-linear models
16. **SPA** - Superior Predictive Ability
17. **StepM** - Stepwise Multiple Testing
18. **MCS** - Model Confidence Set
19. **(T)GARCH-V (1,1,1)** - our first (threshold) GARCH-type extension where we include the average IV across the IVS as an explanatory variable in the variance equation of a standard (T)GARCH model
20. **(T)GARCH-V (1,1,3)** - our second (threshold) GARCH-type extension where we include the level, slope and curvature of the IVS as explanatory variables in the variance equation of a standard (T)GARCH model
21. **OTM** - out the money
22. **ITM** - in the money
23. **ATM** - at the money