ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Master Thesis Quantitative Finance

July 19, 2023

Two-step Approach to Forecast the Implied Volatility of Individual Equity Options

---

Name student:
Laura van Slooten
Student ID number:
480889

Supervisor:
dr. (Gustavo) G. Freire
Second Assessor:
dr. (Maria) M. Grith

Abstract

This paper presents a novel two-step framework that combines parametric option pricing models with machine learning techniques to enhance the accuracy of predicting the implied volatility surface of individual equity options. Using a large dataset of options from the 50 most liquid underlying stocks in the U.S. between 2014 and 2019, augmented with additional covariate data, we demonstrate the effectiveness of our approach. Our empirical results show that the machine-corrected regression tree models, specifically XGBoost, outperform the parametric models in terms of IVRMSE and other performance metrics. The XGBoost combined with the Black-Scholes parametric option pricing model performs best regarding the IVRMSE. The feature importance and interpretability analysis reveal the significance of option-specific features, firm fundamentals, macroeconomic variables, and volatility indices in predicting the implied volatility surface.

Contents

1 Introduction

Options trading is a popular financial activity that involves buying and selling contracts that give the holder the right to buy or sell an underlying asset at a specified price within a specific time frame. One of the key factors impacting options prices is the volatility of the underlying asset. Implied volatility (IV), derived from option prices, reflects the market's expectation of the underlying asset's future volatility. The Black and Scholes model (1973) assumes constant volatility. This assumption is, however, contradicted by the shape of the implied volatility surface (IVS), which shows that the volatilities implicit in option contracts differ across strike prices and time-to-maturity.

Understanding the dynamics of the IVS is crucial for option market participants, traders, and investors, as institutional investors often use IV to manage their option positions (Carr & Wu, 2016). In addition, the IVS offers insights into the market's expectation for future price movements of the underlying stock. Trading desks may be interested in estimating the dynamic process of the IVS of individual equity options to hedge existing portfolios or other over-the-counter derivatives. Next to this, investigating the IVS dynamics of equity options is crucial not only for investors who need to hedge equity option positions but also for investment decisions in other fields, where options are often used to obtain forward-looking information on the market.

Many studies have found predictable movements in the IVS, particularly in index options such as the S&P 500 (Skiadopoulos et al., 2000). However, the dynamics of the IVS in individual equity options have yet to be thoroughly researched. Unfortunately, the equity options market suffers from illiquidity and incompleteness. This illiquidity problem arises as each trading day averages approximately only 21 equity option observations, even though the S&P 500 index has over a thousand daily options (Freire & Kleen, 2023). As a result, most individual equities cannot effectively utilize advanced option pricing techniques that require extensive options data.

Due to the illiquidity problem, previous research has mainly focused on pricing index options such as the S&P 500. As the findings on these index options show favorable results, it is of interest to direct similar research on individual equities. In this study, we consider daily options data on underlying stocks of the top 50 most liquid firms in the US market that have been traded for at least one calendar year. The data ranges from January 2014 to December 2019.

There has been lots of research on parametric option pricing models. However, they still allow for mispricing, which reduces the prediction accuracy. Almeida et al. (2022) suggested a two-step approach to overcome this issue, which involves training ML models on model-

implied pricing errors to correct mispricing by various fitted parametric models. Directly fitting a nonparametric model is equivalent to the nonparametric correction of the Black-Scholes model, which does not yield information on the shape of the IVS (Almeida et al., 2022). Hence, it makes sense to correct a parametric model that can capture specific patterns in IVs because it will have a more straightforward flattened pricing errors surface, making nonparametric estimation easier. This method introduces an easily adaptable architecture based on improving parametric option pricing models' accuracy. First, the observations are fitted to any given parametric model. The difference in observed and estimated IV gives the model implied pricing error. Second, Almeida et al. (2022) use feedforward neural networks to predict these errors so that the models can be corrected. This paper follows this two-step approach by first fitting a parametric model to each stock. Then, we train different machine learning models on the combined pricing errors. Instead of fitting an ML model for each stock or index as in Almeida et al. (2022), we jointly fit the machine learning models across all underlying assets to leverage the information in the cross-section. To evaluate the effectiveness of this approach, we conduct a comparative analysis using multiple ML models, following the framework outlined by Gu et al. (2020). These ML models include Elastic Net (ENet), Random Forest (RF), eXtreme Gradient Boosting (XGBoost), and multiple feedforward Neural Networks (NNs). The parametric models included in this study are the Black-Scholes (1973) model, the ad-hock Black-Scholes by Dumas et al. (1998), the Heston (1993) structural stochastic volatility model, and the Carr & Wu (2016) model, utilizing a parametric definition for the characteristics of the IVS.

The main objective of this research is to accurately predict the IV of options. Previous research has shown that firm fundamentals have strong predictive power on the dynamics of the IVS (D. Chen et al., 2023; David & Veronesi, 2000; Freire & Kleen, 2023; Rathgeber et al., 2021). Hence, our machine learning models incorporate monthly firm fundamentals, retrieved from D. Chen et al. (2023) as explanatory variables. Additionally, we want to include more high-frequency data. Daily data allows for more accurate and timely estimates of important variables, such as market prices and macroeconomic indicators. These features capture short-term fluctuations and changes in market conditions that may be missed by using longer-term data. In addition, using daily data can provide more information on the distribution and volatility of the forecasted variables, which can improve the accuracy of forecasting models. Hence, we consider daily data of variables that could have explanatory power on the option prices. We incorporate features such as macroeconomic variables and volatility indicators to investigate the possible correlation between the dynamics of individual stocks' IVS and various time series factors. Furthermore, we include data on the dynamics of the S&P 500 options surface, as prior research has demonstrated the effectiveness of ML

techniques in predicting the IVS of this index. More details about these variables are in the data section (2). To prevent overfitting, we apply the Fuzzy Forests algorithm of Conn et al. (2019) to reduce the number of features used in the ML models.

The overarching research question that we seek to answer is:

'Can Machine Learning techniques successfully improve fitted parametric option pricing models to predict the implied volatility surface of equity options?'

Lastly, to analyze the outcomes of the most effective ML model, this paper applies the model fingerprint of Li et al. (2020). This method allows for an investigation of the independent fluences of the features, aiding in identifying the most significant factors in predicting the IV. Therefore, we will also provide an answer to the sub-question:

'What aspects of the nonlinearity effects in our best performing Machine Learning model improve the predictive performance of cross-sectional implied volatilities of individual equity options?'

Our empirical findings strongly support the machine-correcting approach, outperforming traditional parametric models. XGBoost performs best in correcting option pricing models, especially when combined with the Black-Scholes model. However, the Heston and Carr-Wu models exhibit poor performance as parametric models for individual equity options, particularly for extreme time-to-maturities. The neural network architectures and elastic net models did not offer significant improvements. Although the Carr-Wu model corrected by the hybrid neural network with the option-specific adjustment cause a very high outperforms rate compared to the uncorrected Carr-Wu model. Lastly, random forests performs better than XGBoost in handling extreme values for the Heston model.

The option-specific variables show to be the most important features for nonparametric correction. The importance of firm fundamentals, macroeconomic variables, and volatility indices in predicting the IVS varies across different parametric and machine learning models, indicating the diverse information captured by these models. Additionally, our interpretability analysis highlights the nonlinear effects of features such as moneyness, time-to-maturity, and option type on the predictions, while firm fundamental covariates exhibit a more linear relationship with the model errors.

The paper is organized as follows: Section 2 describes the data used in the research, while Section 3.1 explains the parametric models employed in the two-step approach. The Fuzzy Forests algorithm for feature reduction is discussed in Section 3.2. Section 3.3 elaborates on the machine learning models used to predict pricing errors. Empirical implications are

presented in Section 3.4, followed by an evaluation of model performance in Section 3.5. The importance of features over time is analyzed in Section 4, and the model fingerprint method is explained in Section 5. Results are provided in Section 6, and the paper concludes with the answers to the research questions in Section 7.

1.1 Related Literature

Our research is closely linked to various aspects of the option pricing literature. In this article, we will discuss four main strands that are most relevant to our study. Firstly, parametric models have been widely used for decades. With its assumption of constant volatility, the classic Black and Scholes (1973) model allows for a closed-form solution to the implied volatility of options. This model has been extensively researched and extended to other underlying assets, as well as generalized specifications in the structural option pricing model. The Black model (1976) remains a common choice for pricing bonds, futures, and interest rate caps and floors. Dumas et al. (1998) presented an ad-hoc correction to the Black-Scholes model, which effectively smoothed implied volatilities and outperformed the local volatility models of Dupire et al. (1994) and Rubinstein (1994).

On the other hand, a large stream of literature focuses on structural continuous-time option pricing models that fall under the family of the general affine jump-diffusion (AJD) processes that incorporate additional sources of risk. Heston (1993) proposed a stochastic volatility diffusion process with an analytical derivation for the option pricing formula, while Duffie et al. (2000) priced fixed income and equity derivatives using the general family of AJD models. Empirical findings on the affine class show mixed evidence (Chernov et al., 2003; Dai & Singleton, 2000; Ghysels & Ng, 1998), leading to several augmentations of the single volatility factor diffusions, including the addition of jumps (Andersen et al., 2001; Bates, 1996; Chernov et al., 2003; Mehrdoust et al., 2017). Another significant classification of models employs approximation theory to represent implied volatilities based on the parameters of stochastic volatility models. Notable contributions in this category include the works of Medvedev & Scaillet (2007), Gatheral & Jacquier (2014), and Carr & Wu (2016).

Despite the numerous advancements, parametric models still show drawbacks and limitations due to statistical or economic assumptions, which can lead to pricing errors. This gives rise to the second strand of research, which explores the use of machine learning models as an alternative to parametric models for pricing options and predicting implied volatilities. Liu et al. (2019) applied neural networks to numerically solve option pricing models, aiming to reduce computational costs. Buhler et al. (2019) and Ruf & Wang (2022) utilized neural networks for option hedging. Zheng et al. (2019) developed a deep neural network prediction

model tailored to the implied volatilities of S&P 500 index options.

Additionally, Ivaşcu (2021) constructed nonparametric machine learning models using the XGBoost algorithm for option pricing and demonstrated improved prediction performance. Luo et al. (2022) compared the prediction accuracy of parametric Heston and Bi-Heston models with the XGBoost model for ETF options. Their in-sample results favored the parametric models, while out-of-sample performance leaned more towards the machine learning models, albeit with less stability. These findings show promising results in combining both types of models. Our paper offers a flexible methodology to enhance the predictive accuracy of any given parametric option pricing model by combining parametric and nonparametric models, making it easily adaptable.

This leads us to the strand of literature that explores a similar two-step approach to correct option pricing models. Das & Padhy (2017) employed parametric option pricing models such as Black-Scholes, Monte Carlo, and finite difference methods, along with machine learning methods like support vector regression and extreme learning machine-based regressions, for pricing European-style index options. More recently, Almeida et al. (2022) explored the use of neural networks to estimate the errors implied by calibrated structural models when predicting implied volatilities of S%P 500 index options. They covered the main variants and extensions of structural models and adopted the neural network architectures proposed by Gu et al. (2020) as adjustment models. Their work serves as the foundation for our two-step framework. Rather than applying a machine learning model to each stock or index, as in Almeida et al. (2022), we simultaneously train machine learning models across all underlying assets, leveraging on the information present in the cross-section. Additionally, we extend the neural network models by incorporating several regularization methods and different activation functions. Furthermore, we expand the framework to include other machine learning techniques that have demonstrated strong performance, such as Random Forests, XGBoost, and Elastic Net (Gu et al., 2020).

Although these models have shown promising results, their empirical evidence is primarily based on index options. We aim to expand this strand of literature by providing an empirical analysis of individual equity options. Therefore, we also review the literature exploring pricing models for individual equity options. Freire & Kleen (2023) developed an option pricing model using local linear random forests on groups of stocks, which are selected endogenously based on firm characteristics. D. Chen et al. (2023) combined machine learning tools with firm fundamentals following Green et al. (2017) to explain the shape of the option implied volatility curve. Their results show economic and statistical significance, which motivated us to use the relationship between firm fundamentals and option prices as predictor variables in our models. Bernales & Guidolin (2014) examined the association between macroeconomic

factors and the dynamics of the implied volatility surface of individual equity options. They found that information from the dynamics of the S&P 500 options surface provided influential information on the implied volatility surface of individual options. Therefore, in addition to firm characteristics, we expanded our set of predictor variables to include macroeconomic data, volatility indicators, the dynamics of the S&P 500 options, and other covariates with potential predictive power on the implied volatility surface.

In summary, our paper adopts the two-step framework proposed by Almeida et al. (2022), where we employ the same parametric models in the first step. In the second step, we jointly fit machine learning models to the pricing errors across all stocks. We increase the number of machine learning models to conduct a comparative analysis across different machine learning approaches, following Gu et al. (2020). Within these machine learning models, we utilize the predictive power of firm characteristics and macroeconomic variables, as suggested by D. Chen et al. (2023), Freire & Kleen (2023), and Bernales & Guidolin (2014). Our analysis focuses on individual equity options, which differentiates us from most existing papers that primarily examine index options.

2 Data

The primary data of this research incorporates the top 50 firms in the U.S. market with the highest average number of options per day traded between January 1, 2014, and December 31, 2019[1]. We will first cover the options' data and, second, the covariates data.

2.1 Option Price Data

The dataset consists of observed IVs of options on a daily frequency. Options data is sourced from OptionMetrics through the Wharton Research Data Services (WRDS) and includes various parameters such as end-of-day bid and ask quotes, implied volatility, volume, strike price, open interest, and expiration date. The individual equity options are American-style options; therefore, their price can be affected by early exercise premiums. OptionMetrics uses the Cox-Ross-Rubinstein (CRR) binomial tree model, a commonly accepted method in the industry, to estimate the IV of options with an American-style exercise feature. Hence, we can treat the options in our analysis as European style (Christoffersen et al., 2018). The option prices are calculated as the midpoint between the bid and ask quotes. The data cleaning process involves removing zero-volume observations, null bid or ask prices, null IV, zero bid price, violations of the no-arbitrage assumption, and option prices below 1/8.

[1]See Appendix A.1 for further details on the tickers.

Second, we exclude options with an expiration period of less than seven trading days as these contracts typically provide limited insights into the IVS (Dumas et al., 1998).

The stock prices and annualized dividend data of the options' underlying are retrieved from The Center for Research in Security Prices, LLC (CRSP). Lastly, we follow Freire & Kleen (2023) and only consider options with moneyness between 0.5 and 2.0, where *moneyness* is defined as the ratio between the stock and strike price, $m_{i,j,t} = S_{i,t}/K_{i,j,t}$ for stock i, option j, and day t.

The primary focus of this paper centers around the implied volatility of individual equity options rather than their prices. This choice stems from the ability to make meaningful comparisons across the options' cross-section and the one-to-one mapping between option prices and implied volatilities. In Table 1, we present the summary statistics of the options grouped by time-to-maturity and moneyness. An intriguing volatility pattern, known as the "smile," emerges within the short-term equity options. This distinctive pattern demonstrates a consistent trend where average implied volatilities decrease from out-of-the-money (OTM) calls (or in-the-money (ITM) puts) towards at-the-money (ATM) options, only to rise once again for OTM puts (or ITM calls). A similar U-shaped pattern is also observed among medium-term options, albeit with a flatter shape. Additionally, the average implied volatilities exhibit a declining trend as the time-to-maturity increases within each specific moneyness category. It is noteworthy that the majority of observations pertain to ATM short-term options ($m_{i,j,t} \in [0.97 - 1.03)$), while medium and long-term options primarily fall into the deep OTM categories ($m_{i,j,t} \in [0.5 - 0.9) \cup [1.1 - 2]$).

Table 1: Summary statistics of equity options' implied volatility

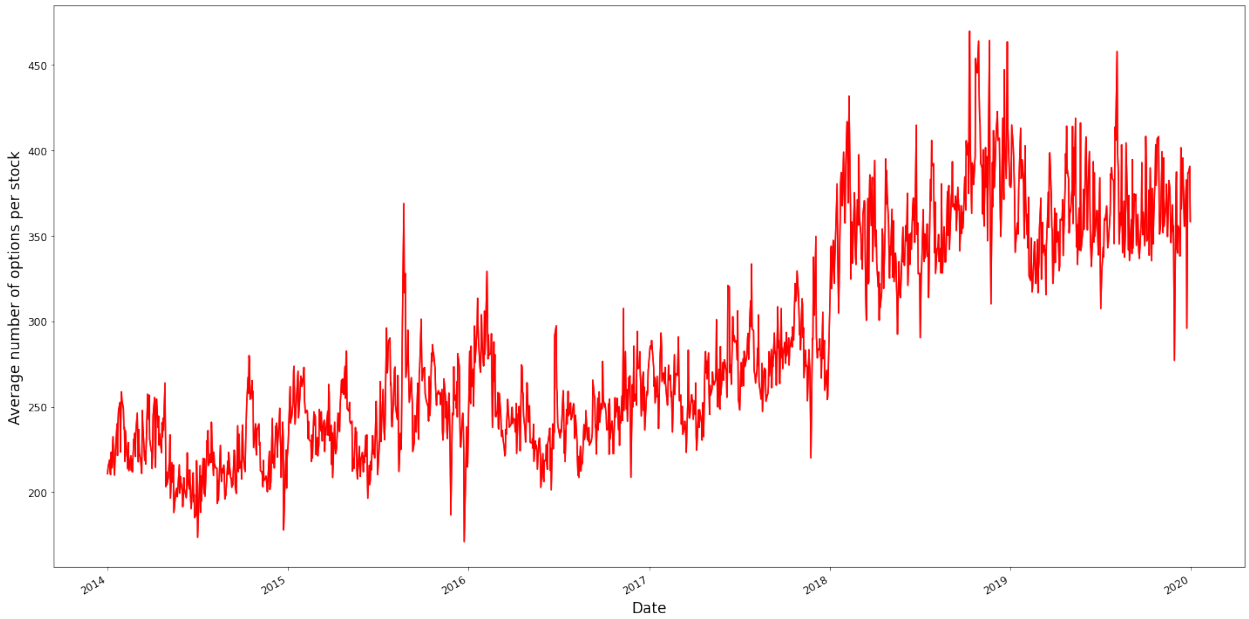
	Moneyness				
	[0.5-0.9)	[0.9-0.97)	[0.97-1.03)	[1.03-1.1)	[1.1-2]
Time to Maturity					
	0.50	0.34	0.28	0.32	0.44
$\tau \leq 90$	[0.29, 0.76]	[0.20, 0.52]	[0.16, 0.43]	[0.18, 0.49]	[0.24, 0.67]
	1,956,741	3,302,722	5,033,996	2,669,274	1,374,514
	0.36	0.29	0.28	0.28	0.31
$90 < \tau \leq 365$	[0.25, 0.51]	[0.20, 0.42]	[0.19, 0.41]	[0.18, 0.41]	[0.19, 0.47]
	1,637,017	738,200	721,641	638,431	1,244,029
	0.34	0.29	0.29	0.28	0.29
$\tau > 365$	[0.24, 0.46]	[0.21, 0.41]	[0.20, 0.40]	[0.20, 0.40]	[0.20, 0.43]
	738,864	216,341	213,493	176,673	564,925

Note. This table presents summary statistics of our equity options data. Each day, we group all options into buckets by moneyness and time-to-maturity (see the row and column names). Per bucket and day, we calculate (1) the average implied volatility in %, (2) the 10% and 90% quantiles of implied volatilities (indicated within square brackets), and (3) the number of observations. The table displays the time-series averages for these measurements. The sample ranges from January 2, 2014, to December 17, 2019.

Figure 2.1 shows how the average number of options per day changes over time. The plot reflects a mainly increasing trend from 2014 to 2020, indicating the growth in liquidity that

the equity options market experienced during those years. Starting at around 200-250, the average number of options gradually rises over six years, reaching approximately 400.

Figure 1: Average number of options per stock over time



2.2 Covariates Data

To benefit from additional predictive power on the IVS, we include different types of variables, such as firm fundamentals, macroeconomic variables, volatility indices, and other covariates that could affect the shape of the IVS. This comprehensive approach allows us to make accurate predictions on implied volatility errors. Previous studies have demonstrated the predictive power of firm fundamentals on the implied volatility surface (An et al., 2014; D. Chen et al., 2023; Christoffersen et al., 2018; Dennis & Mayhew, 2002; Freire & Kleen, 2023). However, these fundamental variables are typically available at a monthly or even quarterly frequency. Since our objective involves daily predictions, we aim to augment our analysis by incorporating higher-frequency data, specifically daily data. This inclusion of higher-frequency data enhances the precision of our predictions, enabling a more granular understanding of the IV dynamics.

First, monthly firm fundamentals are retrieved from D. Chen et al. (2023), which studied the explanatory power of firm characteristics on the shape of the IV curve. Their dataset includes firm characteristics of all firms in the U.S. market traded between 1996 and 2019. They follow Green et al. (2017) and Han et al. (2022) to construct 94 firm fundamentals. The observations are standardized by the cross-sectional mean and standard deviation for

each month's fundamental. In cases where data is missing, we replace it with a standardized zero value according to the fundamental's cross-sectional mean.

Next, we leverage additional data that may have predictive power on the shape of the IVS in addition to the options data and its firm characteristics. To achieve this, we incorporate daily features, such as the CBOE Volatility Index (VIX) based on the S&P 500 Index, which provides information about stock market volatility over the following 30 days (Bernales & Guidolin, 2014), the CBOE Crude Oil ETF Volatility Index (OVX) that measures crude oil price fluctuations (Tian et al., 2021), the CBOE Gold ETF Volatility Index (GVZ) which estimates the expected 30-day volatility of gold prices, the CBOE NASDAQ 100 Volatility Index (VXN) representing the 30-day volatility expectation of ATM options from the NASDAQ-100, the FED Funds Rate of the U.S. Federal Reserve, Equity Market-related Economic Uncertainty Index, the Economic Policy Uncertainty Index for the U.S., the 5-Year Breakeven Inflation Rate, the risk-free rate[2], the credit and term spread[3], and the U.S. Dollars to Euro Spot Exchange Rate. We obtain these features from the St. Louis Federal Reserve Economic Data (FRED) database.[4] Additionally, we include the shape characteristics (level, slope, and curvature) of the IVS of SPX options (following Aït-Sahalia et al. 2021), and lastly, historical data on the S&P 500 Index. For these predictor variables, we handle missing data using forward filling. Appendix A.2 provides a comprehensive overview of all covariates in our dataset.

3 Modeling and prediction IVS

In this section, we provide a detailed explanation of the methodology used to predict the IVS. Our approach consists of several steps, which are outlined below.

1. **Pricing models**: We daily calibrate option pricing models to each stock using parametric models and obtain the model-implied errors.

2. **Feature reduction**: We apply the fuzzy forests algorithm to reduce the number of features used in our machine learning models, thereby improving their efficiency.

3. **Machine learning models**: We use machine learning models to forecast the implied errors and reconstruct the estimated implied volatilities.

[2]We use the 3-Month Treasury Constant Maturity U.S. Treasury Securities data to represent the risk-free rate following Almeida et al. (2022).

[3]We follow Almeida et al. (2022) and define the credit spread as the Moody's Seasoned Baa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity and the term spread as the 10-Year Treasury Constant Maturity Minus 3-Month Treasury Constant Maturity.

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

4. **Cross-sectional Prediction**: We use the empirical data to make daily cross-sectional out-of-sample predictions.

5. **Evaluation**: We assess the performance of our approach using the implied volatility root mean squared error metric, which measures the accuracy of our forecasts relative to the actual values.

3.1 Pricing Models

The first step in our two-step approach includes fitting the observed IVs to the parametric models such that each parametric model is calibrated daily per stock. We have implied volatilities $\sigma_{i,j,t}$ data of options $j = 1, ..., J_i^{(t)}$ across different moneyness $m_{i,j,t}$ and time-to-maturities $\tau_{i,j,t}$ in the cross-section for each stock $i = 1, ..., N$ at time $t = 1, ..., T$. The subscript t is added as the number of options J_i per stock varies over time due to the unbalanced panel data. However, we will ignore this notation for a clearer overview and use J_i.

We use the same parametric models to compare our results to those of Almeida et al. (2022), giving us a diverse selection of option pricing models over time. The parametric models include the Black-Scholes (1973), the ad-hock Black-Scholes by Dumas et al. (1998), the Heston (1993) structural stochastic volatility model, and the Carr & Wu (2016) model, utilizing a parametric specification for the characteristics of the IVS.

3.1.1 Black-Scholes Model

The Black-Scholes (BS) model of Black & Scholes (1973) assumes that stock prices follow a Geometric Brownian Motion:

$$\frac{dS_T}{S_t} = \mu dt + \sigma dW_t. \tag{1}$$

The value of the call and put options of the Black-Scholes model are defined by the formulae:

$$C_{BS}(S_t, K, \tau, r, \sigma) = S_t\Phi(d_1) - K\mathrm{e}^{-r\tau}\Phi(d_2), \tag{2}$$

$$P_{BS}(S_t, K, \tau, r, \sigma) = K\mathrm{e}^{-r\tau}\Phi(-d_2) - S_0\Phi(-d_1), \tag{3}$$

where

$$d_1 = \frac{\log(S_t/K) + (r + \sigma^2/2)\tau}{\sigma\sqrt{\tau}},$$
$$d_2 = \frac{\log(S_t/K) + (r - \sigma^2/2)\tau}{\sigma\sqrt{\tau}} = d_1 - \sigma\sqrt{\tau}.$$

Here, $\Phi(\cdot)$ is the standard normal cumulative distribution function, and r is the risk-free rate. The volatility implied by the BS model is equal to solving the inverse of the pricing formula $\sigma_{i,j,t} = C_{BS}^{-1}(C_i; S_t, K_i, \tau_i, r)$ for a call option and $\sigma_{i,j,t} = P_{BS}^{-1}(P_i; S_t, K_i, \tau_i, r)$ for a put option with C_i and P_i as the observed option price, respectively. Using these implied volatilities, the IV predictions of the BS are equal to a simple ordinary least squares regression of the IVs of that day on a constant:

$$\sigma_{i,j,t} = \alpha_{i,t}^{(0)} + \epsilon_{i,j,t}, \qquad j = 1, ..., J_i. \tag{4}$$

The estimated IV is constant across all maturities and moneyness, namely $\hat{\sigma}_{i,t}^{BS} = \hat{\alpha}_{i,t}^{(0)}$. This equates to calculating the average IV of underlying asset i observed on day t. Hence, the shape of the IVS predictions is flat.

3.1.2 AHBS Model

As a result of the constant volatility assumption of BS, the method fails to capture the dynamics of the IVS. The ad-hoc Black-Scholes (AHBS) model, first introduced by Dumas et al. (1998), is used to fit the IVS such that this can be used in the BS model. The AHBS violates the constant volatility assumption of BS and is, therefore, inconsistent with this model. The IVS is modeled by regressing time-to-maturity $\tau_{i,j,t}$ and moneyness $m_{i,j,t}$ on the implied volatility $\sigma_{i,j,t}$ including second-order and interaction terms:

$$\sigma_{i,j,t} = a_{i,t}^{(0)} + a_{i,t}^{(m)} m_{i,j,t} + a_{i,t}^{(m^2)} m_{i,j,t}^2 + a_{i,t}^{(\tau)} \tau_{i,j,t} + a_{i,t}^{(\tau^2)} \tau_{i,j,t}^2 + a_{i,t}^{(m\tau)} m_{i,j,t}\tau_{i,j,t} + \epsilon_{i,j,t}, \quad j = 1, ..., J_i, \tag{5}$$

where $\epsilon_{i,j,t}$ is the error term for option j with underlying asset i at time t. The model is trained using ordinary least squares regression to minimize the mean squared error (MSE) between the observed IV and the predicted IV. This process yields a fitted vector parameter, denoted as $\hat{\boldsymbol{a}}_{i,t} = (a_{i,t}^{(0)}, a_{i,t}^{(m)}, a_{i,t}^{(m^2)}, a_{i,t}^{(\tau)}, a_{i,t}^{(\tau^2)}, a_{i,t}^{(m\tau)})$. Note that by restricting all parameters to zero except the intercept $a_{i,t}^{(0)}$, we get the same estimate as for the BS model.

3.1.3 Heston Model

In contrast to the BS model, the model of Heston (1993) assumes the price volatility of the underlying stock to be a random process. Under the risk-neutral measure, the dynamics of the underlying asset follow:

$$\frac{dS_T}{S_t} = rdt + \sigma dW_{1,t}, \tag{6}$$

$$dV_t = \kappa(\overline{v} - V_t)dt + \sigma_v\sqrt{V_t}dW_{2,t}, \tag{7}$$

where κ is the mean-reversal rate of the long-run variance $\overline{v}$, V_t equals the spot variance, σ_v is the volatility of the volatility process and $W_{1,t}$ and $W_{2,t}$ represent Wiener processes with correlation equal to ρ. The Heston model belongs to the affine class of parametric option pricing models and provides a quasi-closed form solution for the price of a European option (Appendix A.3). To derive option prices under the Heston model, we use the numerical Fourier-cosine series expansion method introduced by Fang & Oosterlee (2009).

To achieve our objective of predicting the IVS, we utilize the Heston model and estimate its parameters $\boldsymbol{\xi_t} = (V_t, \overline{v}, \kappa, \sigma_v, \rho)$ by minimizing pricing errors in terms of the IV. This method involves calculating the model-implied option prices, translating them to implied volatilities[5], and then minimizing the loss function given by:

$$\frac{1}{J_i}\sum_{j=1}^{J_i}\left[\sigma(m_{i,j,t}, \tau_{i,j,t}) - \hat{\sigma}_H^i(\hat{\boldsymbol{\xi}}_t^i, S_{i,t}, K_{i,j,t}, \tau_{i,j,t}, r_t)\right]^2, \qquad i = 1, ..., N. \tag{8}$$

Nonlinear least squares (NLS) is used for the minimization procedure each day, with the fitted values of the Heston model being equal to $\hat{\sigma}_H^i(\hat{\boldsymbol{\xi}}_t^i, S_{i,t}, K_{i,j,t}, \tau_{i,j,t}, r_t)$ for underlying i, option j at time t. We follow Almeida et al. (2022) and do not impose the Feller condition during estimation, as the main goal is to make predictions.

When calibrating the Heston model on the first day, we apply the differential evolution (DE) optimization algorithm. This algorithm enables us to find the global optimum of the model parameters without requiring initial estimates. For subsequent iterations, we take the fitted parameters from the previous day as the initial guess for the NLS optimization, as the DE algorithm is computationally expensive to run daily.

[5]We use the implementation of Jäckel (2015) available in the Python wrapper for the `Vollib` library

3.1.4 Carr-Wu Model

The model of Carr and Wu (2016) differentiates from the other models as it incorporates the volatility smile: options with different strike prices but with the same time-to-maturity have different IVs. The model assumes that the underlying asset's volatility follows a deterministic function of the stock price, distance from the strike price, and time-to-maturity, allowing for a more accurate representation of the volatility smile.

Under the Carr-Wu (CW) model, the risk-neutral dynamics of the stock price S_t and the IV of the option $\sigma_t(K,\tau)$ can be denoted as:

$$\frac{dS_T}{S_t} = \sqrt{v_t}dW_t, \tag{9}$$

$$\frac{d\sigma_t(K,\tau)}{\sigma_t(K,\tau)} = \mathrm{e}^{-\eta_t\tau}(m_t dt + w_t dZ_t), \tag{10}$$

with v_t as the time-t instantaneous variance rate of the stock price process, the exponential dampening parameter $e^{-\eta_t\tau}$ accounts for the empirical observation that the IV for options with high time-to-maturity tends to exhibit less movement, and m_t and w_t are the drift and volatility process of the IV respectively. Additionally, η_t, m_t and w_t are stochastic processes which are independent of K, τ and $\sigma_t(K,\tau)$. The Wiener processes W_t and Z_t have a correlation equal to a stochastic process ρ_t, which has values on the [-1, 1] interval.

After imposing no-arbitrage constraints, they demonstrate that $\sigma_t^2(k,\tau)$ is contingent on the underlying asset price in relation to the relative strike $k = \ln(K/S_t)$. This relationship is formulated as the following quadratic equation:

$$\begin{aligned}\frac{1}{4}\mathrm{e}^{-2\eta_t\tau}w_t^2\tau^2\sigma_t^4 + (1 - 2\mathrm{e}^{-\eta_t\tau}m_t\tau - \mathrm{e}^{-\eta_t\tau}w_t\rho_t\sqrt{v_t}\tau)\sigma_t^2 \\ -(v_t + 2\mathrm{e}^{-\eta_t\tau}w_t\rho_t\sqrt{v_t}k + \mathrm{e}^{-2\eta_t\tau}w_t^2k^2) = 0.\end{aligned} \tag{11}$$

An interesting and significant characteristic of the solution to Equation (11) is the dependence of the no-arbitrage restriction on the present values of the stochastic processes $(v_t, m_t, w_t, \eta_t, \rho_t)$, rather than the exact dynamics of the processes. As a result, one can treat the values of the processes at time t as parameters when fitting the IVS on a specific day, resulting in a parametric family $\sigma_{CW}^2(\boldsymbol{\theta}_t, k, \tau)$, with $\boldsymbol{\theta}_t = (v_t, m_t, w_t, \eta_t, \rho_t)$. By minimizing Equation (12) below via NLS, we can estimate the parameter $\boldsymbol{\theta}_t$:

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{t}} = \underset{\boldsymbol{\theta_t}}{\operatorname{argmin}} \sum_{i=1}^{n} \Bigg[\frac{1}{4} \mathrm{e}^{-2\eta_t \tau_{i,t}} w_t^2 \tau_{i,t}^2 \sigma_{i,t}^4 + \left(1 - 2\mathrm{e}^{-\eta_t \tau_{i,t}} m_t \tau_{i,t} - \mathrm{e}^{-\eta_t \tau_{i,t}} w_t \rho_t \sqrt{v_t} \tau_{i,t}\right) \sigma_{i,t}^2 \\ - \left(v_t + 2\mathrm{e}^{-\eta_t \tau_{i,t}} w_t \rho_t \sqrt{v_t} k_{i,t} + \mathrm{e}^{-2\eta_t \tau_{i,t}} w_t^2 k_{i,t}^2\right) \Bigg]^2, \tag{12}$$

with $\sigma_{i,t}$ as the observed implied volatility, and $k_{i,t}$ and $\tau_{i,t}$ as the respective relative strike price and time-to-maturity of the option. Given an initial parameter estimate of $\hat{\boldsymbol{\theta}}_t$, we iteratively apply NLS to solve for the optimal fitted parameter set. Similar to the approach in Subsection 3.1.3, we apply DE optimization at the first iteration and use the optimal parameters of the previous day as an initial guess in the NLS optimization for subsequent days. The CW model's IV predictions for a given option are obtained by solving Equation (11) utilizing the calibrated parameters $\hat{\boldsymbol{\theta}}_t$ as inputs, along with the options' k and τ. The implied volatility of the CW then equals the square root of the positive solution to the quadratic equation.

3.2 Fuzzy Forests

As noted by the findings of Green et al. (2017), numerous variables in our dataset are highly correlated. In this paper, we implement the Fuzzy Forests (FF) algorithm of Conn et al. (2019) to control for the multicollinearity across the set of predictor variables and prevent overfitting by reducing the number of used predictors in our ML models. Furthermore, by decreasing the number of variables, we limit the spillover effect of noise from specific features, such as measurement errors, into our models. This method will enable us to generate relatively impartial rankings of variable importance measures (VIMs) based on the explanation of cross-sectional variance (Conn et al., 2019). In our context, the objective of applying FF is to select a reasonable amount of features that provide the most explanatory power on the model-implied pricing errors. As we have four parametric models, we will apply the FF algorithm four times and have four sets of chosen features. The dependent variable used in the FF algorithm is the model-implied error of the parametric models across all underlying stocks. To address computational constraints and account for features updated at a lower frequency, we convert the daily dataset to a monthly resolution by selecting the last Wednesday of each month. This approach balances computational efficiency and ensures sufficient variation in the reduced dataset for the less frequently updated features.

The first step splits all features into clusters using the Weighted Gene Coexpression Network Analysis (WGCNA) framework. This framework identifies clusters of closely in-

terconnected features, such that the correlation within a cluster is high and between the clusters is low. The partitioned feature set consists of P clusters such that $P = \{P_1, ..., P_m\}$ with $\sum_{l=1}^{m} p_l = p$ if $p_l = |P_l|$.

Next, we use the partitioned clusters from WGCNA and implement the FF algorithm in two essential steps: the screening and selection steps. In the screening step, we apply a Recursive Feature Elimination Random Forest (RFE-RF) to each cluster P_l to eliminate irrelevant features. The selection step then executes a final RFE-RF to allow for inter-module interactions, thus resulting in a ranking of VIMs across all clusters. For a more comprehensive understanding of the RFE-RF applied in the WGCNA framework, the screening, and the selection step, please refer to Appendix A.4.1. Furthermore, Appendix A.4.2 provides details on the grid used for hyperparameter tuning, as well as an overview of other implementation details.

3.3 Machine Learning Correction Models

Despite significant progress in the literature toward developing more comprehensive parametric models that can incorporate stylized facts in option data, they still fall short of accurately replicating the IVS. This means that even with a model, denoted as p, there will still be a pricing error surface represented by $\epsilon_p(m, \tau) = \sigma(m, \tau) - \sigma_p(m, \tau)$, where $\sigma(m, \tau)$ is the true implied volatility and $\sigma_p(m, \tau)$ is the implied volatility calculated by the pricing model.

To begin analyzing a collection of $j = 1, ..., J_i$ options on day t, we start by defining a parametric model denoted as p to fit the observed IVS represented by $\sigma(m_{i,j,t}, \tau_{i,j,t})$. This process allows us to obtain fitted values represented by $\hat{\sigma}_p(m_{i,j,t}, \tau_{i,j,t})$, as well as model-implied pricing errors represented by $\hat{\epsilon}_p(m_{i,j,t}, \tau_{i,j,t}) = \sigma(m_{i,j,t}, \tau_{i,j,t}) - \hat{\sigma}_p(m_{i,j,t}, \tau_{i,j,t})$. Next, the surface of the pricing errors denoted as $\epsilon_p(\mathbf{m}, \boldsymbol{\tau})$ is estimated through minimization of the following objective function:

$$\frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{1}{J_i} \sum_{j=1}^{J_i} \left[\hat{\epsilon_p}(m_{i,j,t}, \tau_{i,j,t}) - f(\boldsymbol{x}_{i,j,t}^{(p)}) \right]^2, \tag{13}$$

for several fitted nonparametric models $f(\cdot)$, where $\boldsymbol{x}_{i,j,t}^{(p)}$ is a vector of the M chosen predictor variables by the FF for parametric model p. We aim to determine the function $\hat{f}(\boldsymbol{x}^{(p)})$ that provides the best possible approximation of the pricing error surface. Using this approach, we calculate the IVS by combining the fitted value of the model with its nonparametric corrections. Specifically, for parametric model p and ML model s this can be represented as the sum: $\hat{\sigma}_p(m_{i,j,t}, \tau_{i,j,t}) + \hat{f}_s(\boldsymbol{x}_{i,j,t}^{(p)})$.

The ML models include Elastic Net, Random Forest, eXtreme Gradient Boosting, and

multiple feedforward Neural Networks.

3.3.1 Elastic Net

The Elastic Net method is a regularization technique used in regression analysis to introduce sparsity in the predictor variables of an Ordinary Least Squares (OLS) regression. It combines the characteristics of Lasso (L1) and Ridge (L2) regularization, resulting in a hybrid approach. By adjusting the model parameters, we can control the extent to which the penalty term emphasizes a specific type of regularization. The objective in Equation 13 for the ENet model becomes:

$$\frac{1}{TN}\sum_{t=1}^{T}\sum_{i=1}^{N}\frac{1}{J_i}\sum_{j=1}^{J_i}\left[\hat{\epsilon_p}(m_{i,j,t},\tau_{i,j,t})-\theta_0-\boldsymbol{\theta}^\top \boldsymbol{x}_{i,j,t}^{(p)}\right]^2+\lambda(1-\rho)\sum_{m=1}^{M}|\boldsymbol{\theta}_m|+\frac{1}{2}\lambda\rho\sum_{m=1}^{M}\boldsymbol{\theta}_m^2, \quad (14)$$

where $f(\boldsymbol{x}_{i,j,t}^{(p)})$ is replaced by an OLS regression with a constant term θ_0 and a coefficient vector $\boldsymbol{\theta}$. The model includes two hyperparameters, namely $\lambda \geq 0$ and $\rho \geq 0$, which require tuning and serve two special cases. When $\rho = 0$, the penalization corresponds to the Lasso method, which uses an absolute value penalization on the parameters. This approach serves as a variable selection method, effectively identifying the most important predictors for the model. On the other hand, when $\rho = 1$, the penalization resembles ridge regression, using an l_2 norm on the parameters. Ridge regression shrinks the estimated coefficients toward zero but not exactly to zero, preventing them from becoming excessively large. It can be considered as a shrinkage method. The ENet model, with values of ρ between 0 and 1, combines the advantages of both variable selection and shrinkage. For a more detailed description of the model and its hyperparameters, please refer to Appendix A.4.3.

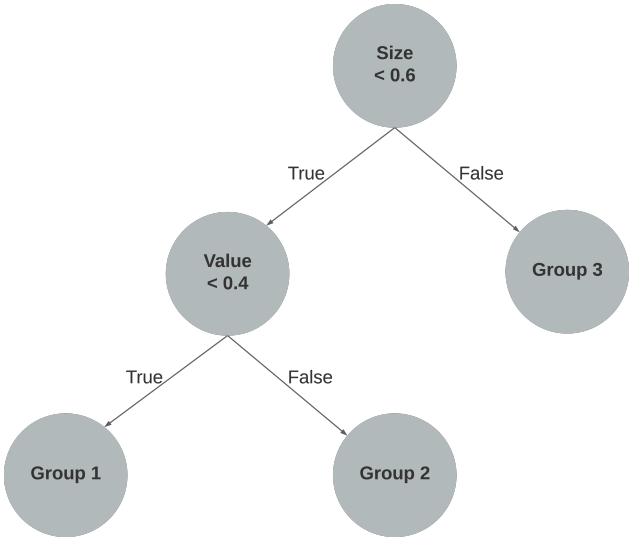
Using the ENet regularization technique, we can effectively balance the sparsity of predictor variables while controlling the impact of different regularization approaches. This flexibility makes the ENet method a valuable tool in regression analysis.

3.3.2 Random Forests and eXtreme Gradient Boosting

Regression trees serve as a nonparametric ML approach used to model interactions between multiple predictor variables. Their objective is to identify clusters of observations with similar behavior. Trees are built through a step-by-step construction process, each step creating a new "branch" to partition the remaining data from the step before based on the value of one specific predictor variable. The prediction generated by a tree, as defined by Gu et al. (2020), equals the average outcome within each partition.

Figure 2 provides a straightforward illustration of the process. It shows a simple example involving two predictor variables: "size" and "b/m". The observations are initially sorted based on their size, dividing them into two groups. Those with a size greater than 0.6 are categorized into group 3. Subsequently, the remaining data are further sorted based on the "b/m" variable, using a breakpoint of 0.4. Observations with a "b/m" value below 0.4 and a small size are assigned to Group 1, while those above the breakpoint are allocated to Group 2. The predictions for each observation within these groups are determined by taking the average outcome variable value among the observations belonging to that particular group.

Figure 2: Graphical depiction of a regression tree



To define the predictions formally, we write the prediction of tree T with K partitions and depth L as:

$$g(\hat{\epsilon}_{i,j,t}^{(p)}; \theta, K, L) = \sum_{k=1}^{K} \theta_k \mathbb{1}_{\{\hat{\epsilon}_{i,j,t}^{(p)} \in C_k(L)\}}, \tag{15}$$

here, $C_k(L)$ represents the product of up to L indicator functions of the predictor variables. The θ_k is the sample average outcome within partition k. The process of growing trees involves carefully selecting predictors and their corresponding values to minimize forecast error. In order to approximate optimal trees, the classification and regression algorithm (CART), as discussed by Loh (2011), is employed. As for the example in Figure 2, the prediction formula becomes:

$$g(\hat{\epsilon}_{i,j,t}^{(p)}; \theta, 3, 2) = \theta_1 \mathbb{1}_{\{\text{size}_{i,j,t} < 0.6\}} \mathbb{1}_{\{\text{value}_{i,j,t} < 0.4\}} + \theta_2 \mathbb{1}_{\{\text{size}_{i,j,t} < 0.6\}} \mathbb{1}_{\{\text{value}_{i,j,t} \geq 0.4\}} + \theta_3 \mathbb{1}_{\{\text{size}_{i,j,t} \geq 0.6\}}.$$

To enhance the out-of-sample prediction accuracy, we implement two tree structures well-known in machine learning literature. The first is an ensemble learning method, Random Forests (Breiman, 2001), suitable for classification and regression tasks. RF generates numerous decision trees through random subset selection of the training data and combines their predictions to generate a final output. This process effectively prevents overfitting and enhances the overall accuracy of the model. For a detailed overview of the parameter grid used in the hyperparameter tuning, please refer to Appendix A.4.4.

Another technique heavily relying on regression trees can be observed in gradient-boosted regression trees. In this approach, the primary distinction from RF arises from the sequential nature of boosted trees, where regression trees are no longer trained simultaneously on the same dependent variable. Instead, weak decision trees are trained sequentially to correct the residuals of the previous model, with the first tree being trained on the original dependent variable. Additionally, incorporating randomness while fitting an individual decision tree is now optional. The concept behind this technique is that by recursively combining predictions made by relatively simple and shallow trees, referred to as weak learners, an overall robust learner is achieved, leading to improved stability and a favorable impact on the bias-variance trade-off.

The approach uses gradient descent to minimize the loss function of the model. It iteratively adjusts the parameters of each subsequent tree to give more weight to the observations that were previously misclassified. Multiple boosting algorithms are available to optimize the objective function, such as Adaptive Boosting (AdaBoost) and eXtreme Gradient Boosting, each with its unique loss function, approximation method for optimization, and the possibility of incorporating regularization terms (e.g., the regularization term of XGBoost is $\Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2$). This research paper will focus on XGBoost due to its increasing popularity and remarkable computational efficiency (T. Chen & Guestrin, 2016). An overview of the hyperparameter grid used for tuning XGBoost is given in Appendix A.4.5.

3.3.3 Neural Networks

We consider five feedforward neural networks as predictive models, adopting a similar architecture proposed by Gu et al. (2020). The NNs consist of three main components: an input layer that receives the raw predictors, one or more hidden layers that use nonlinear interactions to transform the predictors, and an output layer that aggregates the transformed information from the last hidden layer into a final prediction. The number of predictors matches the number of units in the input layer. Forecasts are generated by combining the weighted signals in the output layer. The architectures of our NNs follow a pyramid scheme of nodes by Masters (1993), which can be described as follows:

- NN1: One hidden layer with 32 nodes.
- NN2: Two hidden layers with 32 and 16 nodes, respectively.
- NN3: Three hidden layers with 32, 16, and 8 nodes, respectively.
- NN4: Four hidden layers with 32, 16, 8, and 4 nodes, respectively.
- NN5: Five hidden layers with 32, 16, 8, 4, and 2 nodes, respectively.

Each neuron within a layer receives input from multiple neurons in the previous layer. These inputs are weighted according to their importance, and the neuron applies a nonlinear activation function to the weighted sum of the inputs. This activation function introduces nonlinearity into the network, allowing it to capture complex relationships in the data. Here, we use the Rectified Linear Unit (ReLU) activation function (Sussillo & Abbott, 2014) for faster derivations. We use the He normal weight initialization method to initialize the weights, which has proven to give robust results when combined with the ReLU activation function (He et al., 2015). The transformed outputs from each neuron in a layer are then passed as inputs to the neurons in the subsequent layer, and this process continues until the output layer is reached.

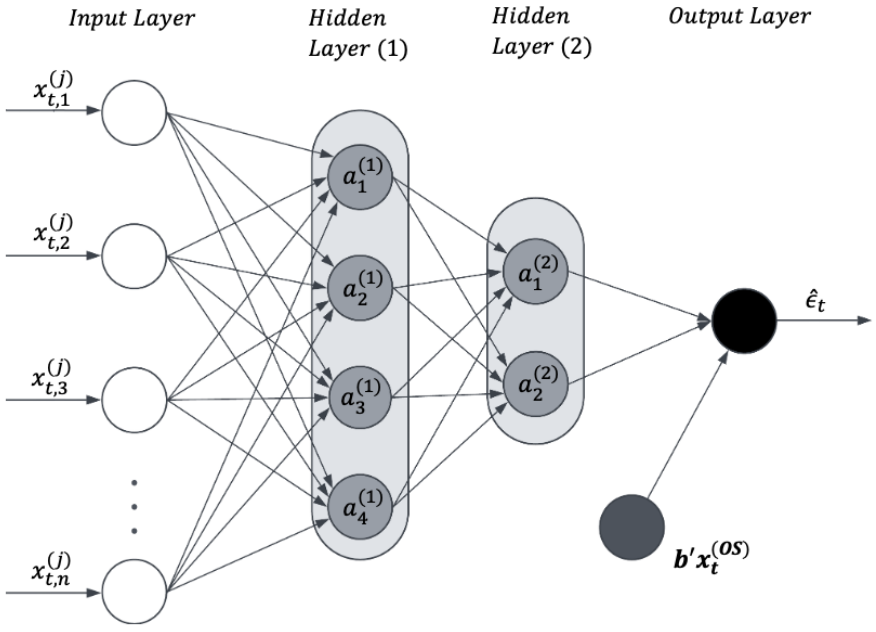
During the training phase, neural networks learn to optimize their parameters through backpropagation, an iterative process that updates the weights between the neurons to minimize the loss error. This optimization is typically achieved using gradient descent algorithms that traverse the network in the opposite direction of the data flow, updating the weights based on the calculated gradients. Here, to reduce computational heaviness, we use the Stochastic Gradient Descent (SGD) that computes the gradient on a small, randomly selected subset of the data. This approach allows for more efficient optimization without the need to compute gradients on the entire dataset. In this study, we employ an altered version of gradient descent, incorporating Nesterov momentum (Nesterov, 1983). Unlike regular SGD, prone to oscillations amidst local minima, Nesterov momentum (or Nesterov accelerated gradient) enhances SGD's acceleration in the appropriate direction.

In addition to the standard architecture of the NNs, we propose a modification to the output layer by adding a linear combination of input variables directly into the output layer (following the hybrid structure of Bianchi et al. 2021). We do this to exploit possible linear relations between the covariates and the output while still using the nonparametric architecture for the other features. The variables that are directly added to the output layer will not be included as initial input variables in the neural network. We split the input variables into option-specific and fuzzy forests covariates. This hybrid approach will

be performed twice, once where the option-specific features will be reserved to be directly inserted in the model (the exogenous regressors), and the input variables only consist of the fuzzy forests covariates, and visa versa. The number of layers and nodes, along with the other hyperparameters, will follow the same rules as the previous NN structure. We will refer to this type of architecture as a *hybrid* neural network.

In Figure 3, we present a graphical illustration of a hybrid NN model with two hidden layers, four nodes in the first hidden layer and two nodes in the second hidden layer, where the option-specific variables are included at the output layer as exogenous predictors. Here, the variables $x_{t,i}^{(j)}$, for $i = 1, ..., n$, indicate the input features selected by the FF algorithm at time t for option j. In our example, the number of features n equals 20. Moving to the first hidden layer, $a_j^{(1)} = f\left(\sum_{i=1}^{n} w_{i,j} x_{t,i} + b_j\right)$ represents a node with activation function $f(\cdot)$, weights $w_{i,j}$, and bias b_j. Each arrow contains a weight that indicates the connection between the input feature and that node. As mentioned earlier, we consider the ReLU activation function at each hidden layer, which is given by $f(x) = \max(0, x)$. After the second layer, we include a linear combination of the option-specific features in the output layer, and we end up with one node in the output layer, which is our prediction of the implied error for option j at time t. This process is done for the whole test set.

Figure 3: Graphical illustration of a hybrid neural network with two hidden layers



Note. This figure provides a graphical illustration of the hybrid neural network with a direct linear combination of the option-specific variables, $\boldsymbol{b'x_t^{(OS)}}$, which is included as exogenous regressors.

Next, we have adapted the architectures (including the regular NN) to implement four regularization techniques to prevent overfitting the training data. Firstly, after each layer, except for the output layer, we apply dropout, which randomly sets neurons to zero during training, where the dropout rate controls the exclusion probability. Secondly, after each hidden layer, we employ batch normalization to normalize neuron activations, improving training stability and speeding up convergence. Thirdly, we add ridge regularization (L2) as penalty term in the objective loss function for each hidden layer. Lastly, we apply the early stopping procedure during fitting, which halts the training if there is no significant loss decrease after a defined number of successive iterations on the validation set. We set the two hyperparameter values of this regularization equal to the ones used in Gu et al. (2020). For further information on the hyperparameter tuning grids, please refer to Appendix A.4.6. For the NNs, all features are scaled to the [-1, +1] interval over time, following Kelly et al. (2019).

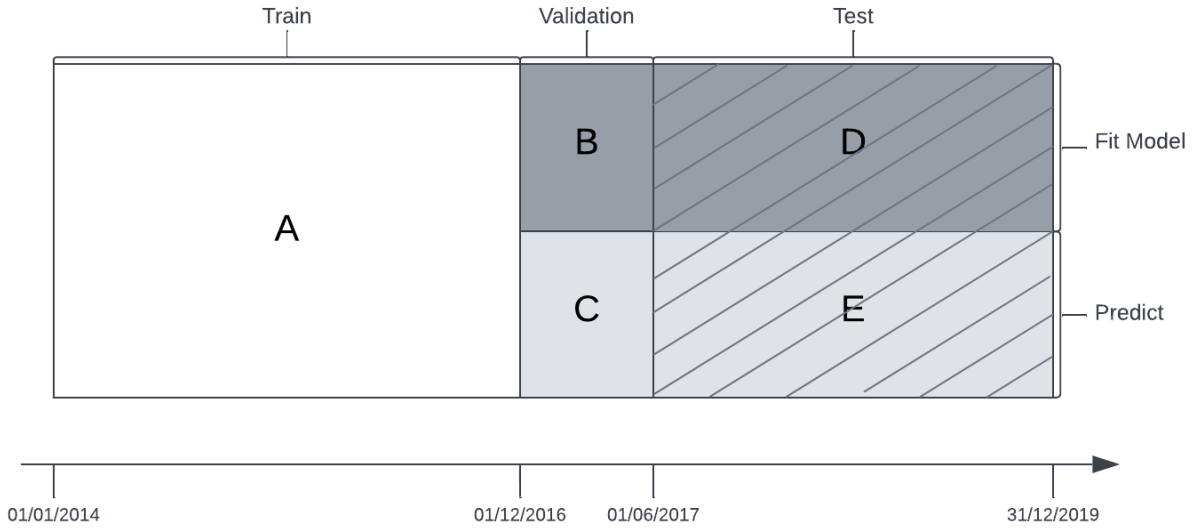
3.4 Cross-sectional Prediction

For our prediction exercise, we split our data into a train, validation, and test set. Our machine learning models need a large training set to learn the misspecification tendencies across the whole cross-section of options over time. Hence, the train and validation set cover 60% (in-sample), while the test set covers the last 40% (out-of-sample). We split the in-sample data into train and validation sets according to an 80% - 20% splitting rule, respectively. The training set is used to tune the ML models' hyperparameters, where each hyperparameter set's performance is assessed in the validation set. We split the validation and test set each day randomly, such that half of the options for a stock on a given day are used to fit the parametric models, and the other half is used to make same-day cross-sectional predictions.

Figure 4 gives an overview of how the data is split and used during the cross-sectional predictions. First, the parametric models are calibrated daily per stock on the entire training data (section A), and the combined cross-sectional residuals of the fitted models give the dependent variables to train the fuzzy forests and the ML models. Next, we use the first half of the option data in the validation set (section B) to calibrate the parametric models again and make predictions on the second half of the option data on that day (section C). We use the predictor variables in section C to make predictions on the model-implied errors and combine these with the predictions of the parametric models to construct the total estimated implied volatilities. Using a specific metric, we select the best hyperparameters based on the performance of the total predictions in section C. Next, we use the whole in-sample dataset

to jointly fit the ML models (section A + B + C). Lastly, the test set is again split into two parts, half of the options data of a stock is used to fit the parametric models (section D), and the other half is used to make out-of-sample predictions on the same day (section E).

Figure 4: Train-Validation-Test split for cross-sectional predictions



Note. Figure 4 presents the data splits where the hyperparameters of the FF and ML models are tuned on the training set, using the validation set as a performance measure. The models are then fitted on the training + validation set, and the test set is used to make out-of-sample predictions. For the validation and test set, the option data is split in half, where the first half is used to fit the parametric models daily, and the second half is used as a holdout to make out-of-sample predictions.

3.5 Evaluation

We follow Almeida et al. (2022) and use the implied volatility root mean squared error (IVRMSE) to evaluate the accuracy of the daily out-of-sample predictions for each individual stock, corresponding to the loss function minimized by the considered models during estimation. To aggregate performance across stocks, we summarize the IVRMSEs by taking the average IVRMSE across the stocks. However, only taking the average would overlook the varying IV levels in the cross-section. Therefore, we also compare the results of our machine-corrected models to their respective parametric models to provide context. This approach allows us to determine whether the machine learning models effectively correct the option pricing models.

First, we average the frequency of instances where a given model outperforms the benchmark model based on the IVRSME metric. For this, we use the outperformance rate (OR), which can be computed using the following formula:

$$\mathrm{OR}_t^j = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[\mathrm{IVRMSE}_{i,t}^j < \mathrm{IVRMSE}_{i,t}^{\mathrm{P}_j}\right], \tag{16}$$

where $\mathrm{IVRMSE}_{i,t}^j$ indicates the IVRMSE of stock i on day t using model j and $\mathrm{IVRMSE}_{i,t}^{\mathrm{P}_j}$ is the IVRMSE of the stock-specific parametric model P corresponding to model j model for stock i at day t. Next to this, we compare the models based on the median and mean loss ratio:

$$\mathrm{MedLR}_t^j = \underset{i=1,\ldots,N}{\mathrm{median}} \frac{\mathrm{IVRMSE}_{i,t}^j}{\mathrm{IVRMSE}_{i,t}^{\mathrm{P}_j}}, \tag{17}$$

$$\mathrm{MeanLR}_t^j = \frac{1}{N}\sum_{i=1}^{N} \frac{\mathrm{IVRMSE}_{i,t}^j}{\mathrm{IVRMSE}_{i,t}^{\mathrm{P}_j}}. \tag{18}$$

Finally, instead of evaluating the IVRMSE of each model against their respective parametric model, we employ the same three metrics to assess the models compared to the uncorrected stock-specific AHBS model, our primary benchmark model. This allows us to compare the various models in the cross-section. We replace $\mathrm{IVRMSE}_{i,t}^{\mathrm{P}_j}$ with the IVRMSE of the stock-specific AHBS model for stock i at day t, $\mathrm{IVRMSE}_{i,t}^{\mathrm{AHBS}}$.

4 Feature Importance over Time

To better understand the contribution of features in our prediction exercise, we adopt the framework of Daul et al. (2022). They use the reduction in MSE as a performance metric to determine feature importance over time. An advantage of this method is that it is not dependent on the model type, in contrast to, for example, the mean decrease in impurity for trees of Hastie et al. (2009), which is dependent on the given machine learning model. First, the model is fit on the training data. Then iteratively, the values of one feature in the whole sample are set to its mean, keeping the other feature values unchanged; the MSE is then calculated again using the new dataset. We do this each day to compare the MSE using the actual data with the MSE of the adjusted data over time. The feature importance is proportional to the increase in MSE. We do this for each parametric model in combination with its best-performing correction model.

An increase in MSE indicates that the model relied on variations from that specific feature to make accurate predictions. When the feature was adjusted to its average value,

the accuracy of the model's predictions decreased. On the other hand, if the MSE decreases upon fixing the feature to its mean, it suggests that the feature is not as crucial for the model's prediction performance. In such cases, we assign a feature importance of zero to that particular feature. We normalize all feature importances to add up to 1 to ensure a consistent scale. Lastly, we have categorized features with similar characteristics into groups to comprehend the results better. Details on the definition of these groups are provided in Appendix A.5.

5 Interpretability

To evaluate each feature's independent influence in the model, we perform the model fingerprint method of Li et al. (2020). They provide an extension to the partial dependence, which was introduced by Friedman (2001). The fingerprint method enables us to open the 'black box' known as ML models and decompose the prediction power into linear and nonlinear components. We measure the change in the predicted values caused by changes in each input variable while keeping all other features constant. The prediction function of a model is defined as the model fitted to the training data:

$$\hat{\gamma} = \hat{f}(x_1, x_2, ..., x_m). \tag{19}$$

The prediction relies on all M input features, but the partial dependence function $\hat{\gamma}_k$ only relies on one input variable, x_k. The partial dependence function in Equation (20) estimates the predicted outcome for a specific value of x_k by taking into account all potential values of the other predictors, which are denoted as x_{k-}:

$$\hat{\gamma}_k = \hat{f}_k(x_k) = E_{x_{k-}}[\hat{f}(x_1, x_2, ..., x_m)]. \tag{20}$$

Below, we outline the steps to compute the partial dependence function in practice using empirical data:

1. Select a permissible value x_k for feature k.
2. Produce a new prediction with x_k and actual input vector x_{k-} using Equation 19.
3. Iterate step 2 for each input vector of x_{k-} while keeping x_k constant, and save the predictions.
4. Attain the partial prediction $\hat{\gamma}_k$ of the fixed x_k by taking the average of all the predictions for that value.

5. Iterate steps 1 to 4 for any possible value of x_k, and graph the resulting partial dependence function.

The partial dependence function will show large deviations if a feature significantly influences the model's predictions. To reduce the computational expensiveness of iterating over all possible values of x_k in step 5, we adopt the approach proposed by Daul et al. (2022), which involves limiting the range of values to 50 observations that are equally spaced between the minimum and maximum of x_k.

Next, we define the linear effect of the features as the average absolute deviation between the linear predictions and the mean marginal dependence:

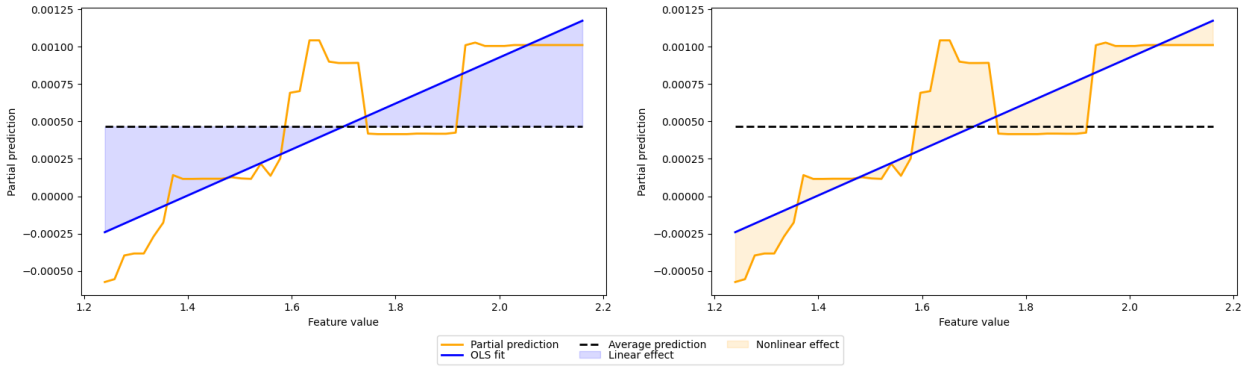
$$\textit{Linear effect } (x_k) = \frac{1}{N}\sum_{i=1}^{N} \text{abs}\left(\hat{l}_k[x_{k,i}] - \frac{1}{N}\sum_{j=1}^{N}\hat{f}_k[x_{k,j}]\right). \tag{21}$$

The linear fit $\hat{l}_k[x_{k,i}]$ is the least squares fit of x_k on the partial predictions $\hat{\gamma}_k$, the subscript i denotes the ith value in the data. The nonlinear effect can be seen as the mean absolute difference between the partial and linear predictions:

$$\textit{Nonlinear effect } (x_k) = \frac{1}{N}\sum_{i=1}^{N} \text{abs}\left(\hat{f}_k[x_{k,i}] - \hat{l}_k[x_{k,i}]\right). \tag{22}$$

Note that this value equals zero if the model used to calculate the marginal dependence is an ordinary linear regression model, as it should be. Figure 5 presents the contrasting effects of linear and nonlinear predictions. The shaded areas represent their magnitudes, revealing that the linear effect predominantly explains the prediction, with the nonlinear effect playing a lesser role in the partial prediction in relation to the IR feature.

Figure 5: Decomposition of the partial prediction for the feature IR



6 Empirical Results

In this section, we first cover the results of the FF algorithm, which determines the features used in our ML models. Second, we discuss the results of the cross-sectional predictions exercise and conduct a comparative analysis of our models. Thirdly, we review the feature importance of the best machine-corrected models over time per parametric model. Lastly, we review the interpretability results of our best-performing model.

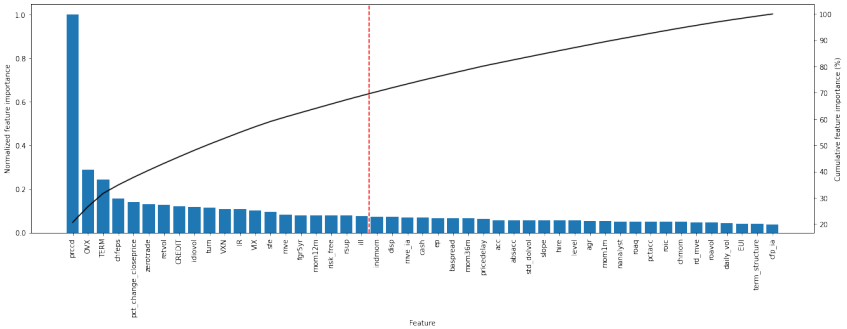
6.1 Fuzzy Forests

We have implemented the FF algorithm to our in-sample model-implied error dataset to address feature correlation and overfitting. Figure 6 displays the normalized output of this algorithm for each parametric model, with the vertical red line indicating the feature cut-off point and the black line representing the cumulative percentage of feature importance. Notably, we have excluded option-specific variables such as time-to-maturity, moneyness, strike price, and the put/call dummy from this graph, as they are included in the feature set regardless. Since no general rule dictates the number of features to be included in the reduced model, we opt to include enough features to account for at least 60% of the cumulative feature importance across all four data sets. In our specific case, this entailed selecting 20 features.

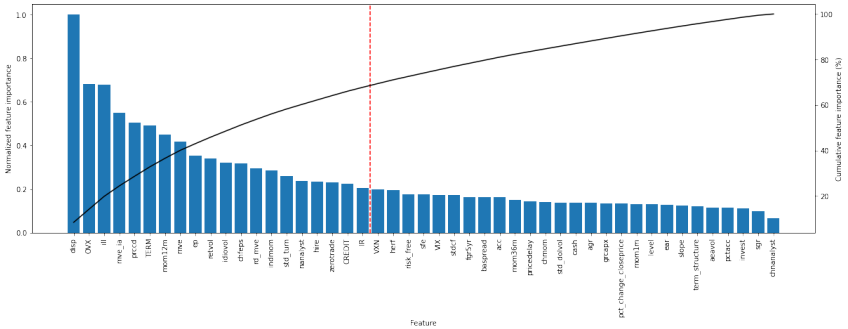
The distribution of feature importance varies across the different parametric models. In the BS model, the closing price of the underlying equity stands out as the most significant, followed by the OVX and the term spread. Next, the AHBS model displays a more balanced feature importance. Its top five features include dispersion in forecasted earnings per share (disp), OVX, illiquidity (ill), industry-adjusted size (mve_ia), and the underlying's closing price. Moving on to the FF result of the CW model, it is noteworthy that the most important feature, the federal funds rate (DFF), is not among the selected variables in the other models. The percentage change in the underlying's closing price, OVX, return volatility (retvol), and industry momentum (indmom) are also influential.

Lastly, the five features with the highest VIM for the errors of the Heston model are the risk-free rate, the S&P 500 index closing price, OVX, the closing price of the underlying, and the term spread. Among the selected features, 26 firm fundamental variables are chosen across all four parametric models, with twelve overlapping features, of which retvol and idiovol are in the chosen feature set of all four models. Interestingly, the shape characteristics of the S&P 500 index options are not selected, however, the S&P 500 index closing price is included in the CW and Heston model's chosen feature set.

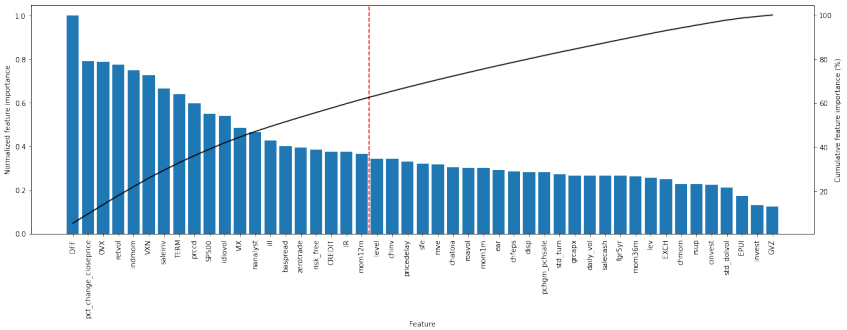
Figure 6: Ranked feature importance of the fuzzy forests algorithm



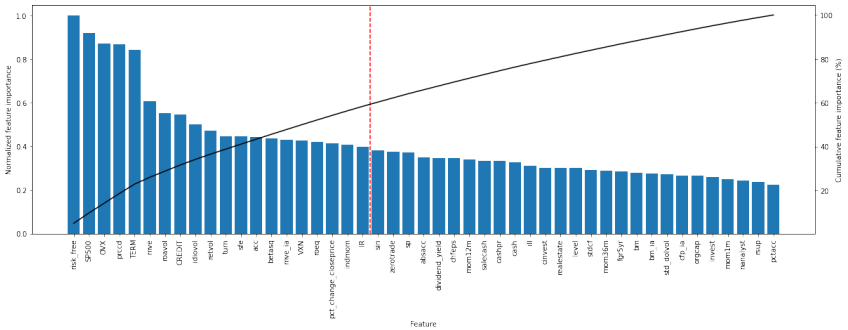
(a) BS



(b) AHBS



(c) CW



(d) Heston

Note. Figure 6 presents the VIMs values generated by the FF algorithm. The solid line represents the normalized cumulative VIM for the features plotted in our graph, while the red-dotted line represents the threshold for selecting variables for further analysis.

6.2 Cross-sectional Prediction

The cross-sectional prediction exercise involved calibrating the parametric models each day per underlying asset and fitting the secondary models on the model-implied errors over the whole set of underlying assets. The results of the IVRMSE for each model combination are shown in Table 2. The exact hyperparameter set of all models chosen during calibration can be found in Appendix A.4.

Table 2: Out-of-sample IVRMSE (%) of the cross-sectional prediction exercise

	No ML	ENet	RF	XGB	NN	NN-OS	NN-FF
BS	6.913	6.262	4.077	**3.973**	6.848	6.740	6.817
AHBS	5.360	5.360	4.230	**4.209**	5.360	5.357	5.357
Heston	8.158	7.677	5.999	**5.953**	8.060	7.933	8.002
CW	21.122	18.949	**18.009**	18.354	20.409	20.078	19.981

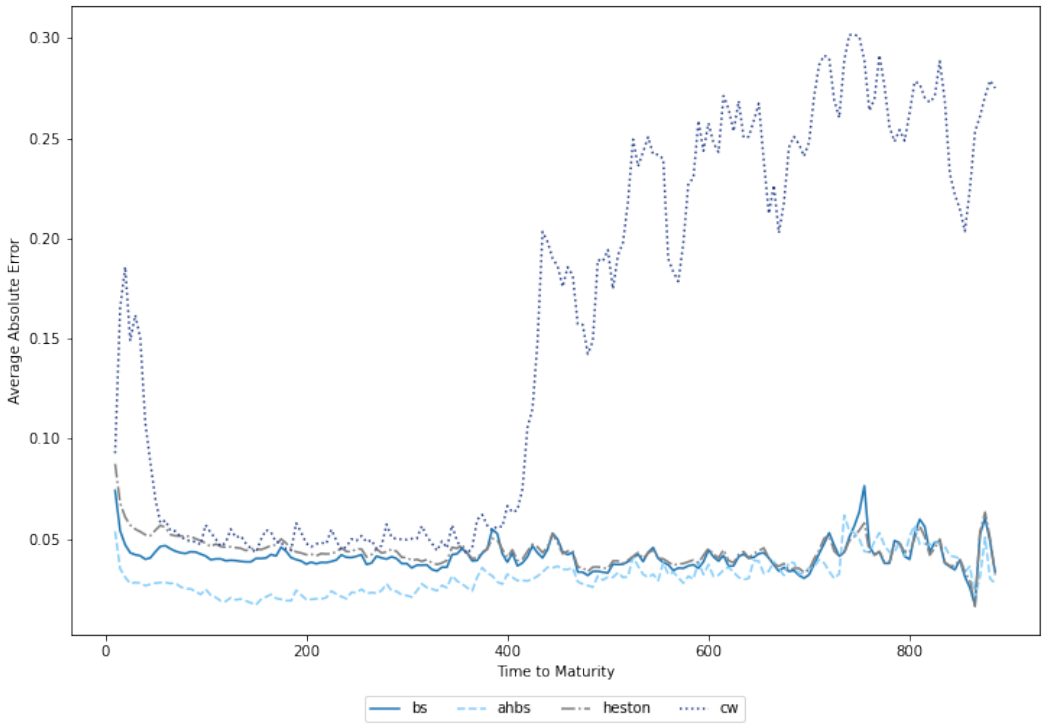
Note. This table presents the average IVRMSE (in %) across the stocks of each day for the same-day predictions of each model. Each row represents a parametric model, and the columns indicate the applied machine learning correction model (Elastic Net (ENet), Random Forests (RF), XGBoost (XGB), and NN, NN-OS, and NN-FF refer to the neural network architectures without, with option-specific and with fuzzy forests direct input layer, respectively) or, in the case of no correction model (No ML). The bold values indicate the best-performing correction model, while the underscored value represents the best-performing model based on the IVRMSE. The test sample ranges from June 1st, 2017, to December 18th, 2019.

The results presented in Table 2 highlight the dominance of regression trees over the other considered models. XGBoost emerges as the best-performing secondary model for the BS, AHBS, and Heston models, while RF performs best for the CW model. Among all the models, the combination of the Black-Scholes option pricing model and XGBoost yields the most favorable outcomes, achieving an average IVRMSE of 3.973%. Interestingly, the neural network architectures (NN, NN-OS, and NN-FF) and the elastic net models do not exhibit significant improvements compared to the other models. Moreover, there appears to be minimal variation in performance among the different neural network architectures. These findings suggest that the neural networks do not provide substantial advantages over the other models examined for this dataset. Notably, the parametric model BS shows the highest degree of correctability. When left uncorrected, the AHBS model performs the best among the parametric models. Meanwhile, the CW model seems to be a poor fit for the data, as is evident from its relatively higher IVRMSE than the other models.

To analyze why the CW model, and to a lesser extent the Heston model, perform worse than the AHBS, we examine the average absolute error per time-to-maturity (see Figure 7). Initially, all models exhibit higher errors for shorter time-to-maturities, which decrease as time-to-maturity increases. However, beyond approximately 400 days to maturity, the CW model's average absolute error sharply increases and remains high for longer maturities, while the other models exhibit more convergence. It is worth noting that including extreme time-

to-maturities in the model fitting process can introduce noise to predictions for the other maturities. However, eliminating these extreme cases could lead to sample illiquidity, which mainly affects the Heston and CW models as they rely on a larger dataset for parameter estimation due to their model complexity.

Figure 7: Average out-of-sample absolute error of the parametric models per time-to-maturity



As mentioned before, by only considering the average IVRMSE, we will overlook the varying IV levels across options. Hence, we consider three additional metrics: the outperformance rate, the median loss ratio, and the mean loss ratio. Compared to their respective parametric model, the results of these metrics are presented in Table 3 to assess the effectiveness of machine-corrected models. Consistent with previous findings, the regression tree models outperform the other models across all parametric combinations. The performance levels of the RF and XGBoost models lie close to each other. Among all the models, the RF model emerges as the superior correction model for the BS, AHBS, and Heston models regarding the outperformance rate. This suggests that the RF model notably improves predictions for these parametric models.

Interestingly, when the CW model is corrected using the hybrid NN-OS architecture, it reaches the highest outperformance rate across the other ML models, whereas the other NN architectures do not exhibit similar results. This unexpected result may indicate that the option-specific features have a linear relation to the CW errors. The NN-OS also shows potential as a corrective measure for the BS model, given the outperformance rate. For the Heston and AHBS, the NN-FF model performs best across the neural network architectures.

Table 3: Out-of-sample performance of the machine-corrected cross-sectional prediction of equity options compared to the parametric models

	ENet	RF	XGB	NN	NN-OS	NN-FF
Panel A: OR						
BS	0.875	**0.981**	0.976	0.687	0.957	0.899
AHBS	0.553	**0.929**	0.926	0.514	0.514	0.673
Heston	0.778	**0.932**	0.912	0.822	0.833	0.859
CW	0.687	0.745	0.804	0.744	**0.950**	0.680
Panel B: MedLR						
BS	0.864	0.575	**0.571**	0.989	0.965	0.977
AHBS	1.000	0.822	**0.812**	1.000	0.991	0.993
Heston	0.910	0.713	**0.706**	0.980	0.958	0.970
CW	0.877	0.789	**0.726**	0.966	0.875	0.945
Panel C: Mean LR						
BS	0.876	0.594	**0.590**	0.989	0.963	0.976
AHBS	1.000	0.806	**0.799**	1.000	1.000	0.991
Heston	0.931	**0.720**	0.729	0.978	0.955	0.967
CW	1.004	0.898	**0.808**	0.973	0.878	0.999

Note. This table presents the performance metrics of the cross-sectional prediction of each machine-corrected model against their respective parametric model. Panel A reports the time-series average of the outperformance rate (the number of times the IVRMSE was lower than that of the parametric model), Panel B reports the time-series averages of the daily median loss ratios, and Panel C shows the time-series averages of the daily mean loss ratios. Each row represents a parametric model, and the columns indicate the applied machine learning correction model (Elastic Net (ENet), Random Forests (RF), XGBoost (XGB), and NN, NN-OS, and NN-FF refer to the neural network architectures without, with option-specific and with fuzzy forests direct input layer, respectively). The bold values indicate the best-performing correction model for that parametric model, while the underscored value represents the best-performing model based on all the models. The test sample ranges from June 1st, 2017, to December 18th, 2019.

Panel B and C of Table 3 present the median and mean loss ratios, respectively, providing insights into the distribution of loss ratios and their average values. Panel B shows that the XGBoost consistently performs well across all parametric models. This consistency underscores the robustness and reliability of XGBoost as a corrective model in capturing the

median values of loss ratios. Similarly, in Panel C XGBoost outperforms other ML correction models across most parametric models. However, an exception arises with the Heston model, where the RF correction model exhibits a lower MeanLR than XGBoost. This suggests that the RF model shows particular proficiency in addressing extreme values within the loss ratio distribution for the Heston model.

Lastly, to perform a cross-comparison on our models, we focus on the performance metrics with the AHBS model as benchmark instead of their parametric model. The comparative analysis is presented in Table 4. The observed pattern of best-performing model combinations aligns closely with the findings in Table 3. Notably, the combination of AHBS with RF stands out as the best-performing model, achieving an outperformance rate of 0.929.

Next, we turn our attention to Panels B and C. Consistent with our expectations, when the AHBS model is corrected using the NNs, the MedLR and MeanLR values converge to approximately 1. This suggests that the NNs have a limited impact on improving the predictions, as they do not significantly alter the distribution of loss ratios. Examining the performance across all parametric models, XGBoost consistently delivers the best MedLR. This finding indicates that XGBoost offers a superior balance between predictive accuracy and the spread of loss ratios. Furthermore, in terms of both metrics, the combination of XGBoost with the AHBS model results in the highest performance among all the examined combinations. This reinforces the effectiveness of XGBoost as a corrective model when applied in conjunction with the AHBS framework. The MeanLR is more sensitive to extreme values in the loss ratio distribution. In this regard, the RF model outperforms XGBoost only for the Heston model. This result implies that the RF model exhibits better resilience against extreme loss ratio values, leading to a more favorable average loss ratio for the Heston model compared to XGBoost.

Table 4: Out-of-sample performance of the cross-sectional prediction of equity options compared to the AHBS model

	No ML	ENet	RF	XGB	NN	NN-OS	NN-FF
Panel A: OR							
BS	0.073	0.107	**0.718**	0.715	0.077	0.084	0.083
AHBS	-	0.553	**0.929**	0.926	0.514	0.514	0.673
Heston	0.056	0.070	0.432	**0.442**	0.060	0.067	0.064
CW	0.016	0.014	0.043	**0.106**	0.017	0.027	0.013
Panel B: MedLR							
BS	1.431	1.249	0.843	**0.828**	1.414	1.379	1.401
AHBS	-	1.000	0.822	**0.812**	1.000	0.991	0.993
Heston	1.578	1.435	1.084	**1.080**	1.541	1.498	1.523
CW	2.607	2.397	2.087	**1.852**	2.501	2.223	2.534
Panel C: MeanLR							
BS	1.493	1.277	0.859	**0.853**	1.473	1.433	1.453
AHBS	-	1.000	0.806	**0.799**	1.000	1.000	0.991
Heston	1.846	1.688	**1.362**	1.376	1.805	1.759	1.780
CW	3.201	2.928	2.642	**2.452**	3.086	2.826	3.085

Note. This table presents the performance metrics of the cross-sectional prediction of each model against the AHBS model without ML correction. Panel A reports the time-series average of the outperformance rate (the number of times the IVRMSE was lower than that of the AHBS model), Panel B reports the time-series averages of the daily median loss ratios, and Panel C shows the time-series averages of the daily mean loss ratios. Each row represents a parametric model, and the columns indicate the applied machine learning correction model (Elastic Net (ENet), Random Forests (RF), XGBoost (XGB), and NN, NN-OS, and NN-FF refer to the neural network architectures without, with option-specific and with fuzzy forests direct input layer, respectively) or, in the case of no correction model (No ML). The bold values indicate the best-performing correction model for that parametric model, while the underscored value represents the best-performing model based on all the models. The test sample ranges from June 1st, 2017, to December 18th, 2019.

6.3 Feature Importance

By analyzing the temporal variation of feature importance, we can gain insights into the differing information sets of our best-performing machine learning models across the parametric models. To assess feature importance over time, we employ an iterative approach where we fix features to their mean values and measure the resulting influence on the mean squared error. The outcome of this analysis for our best ML models is presented in Figure 8. Note, the option-specific features are excluded from this evaluation due to their high feature importance, which could hinder meaningful comparisons with other feature groups[6]. However, for a comprehensive examination of feature importance, including the option-specific

[6]The grouping of features is detailed in Table A9 of Appendix A.5.

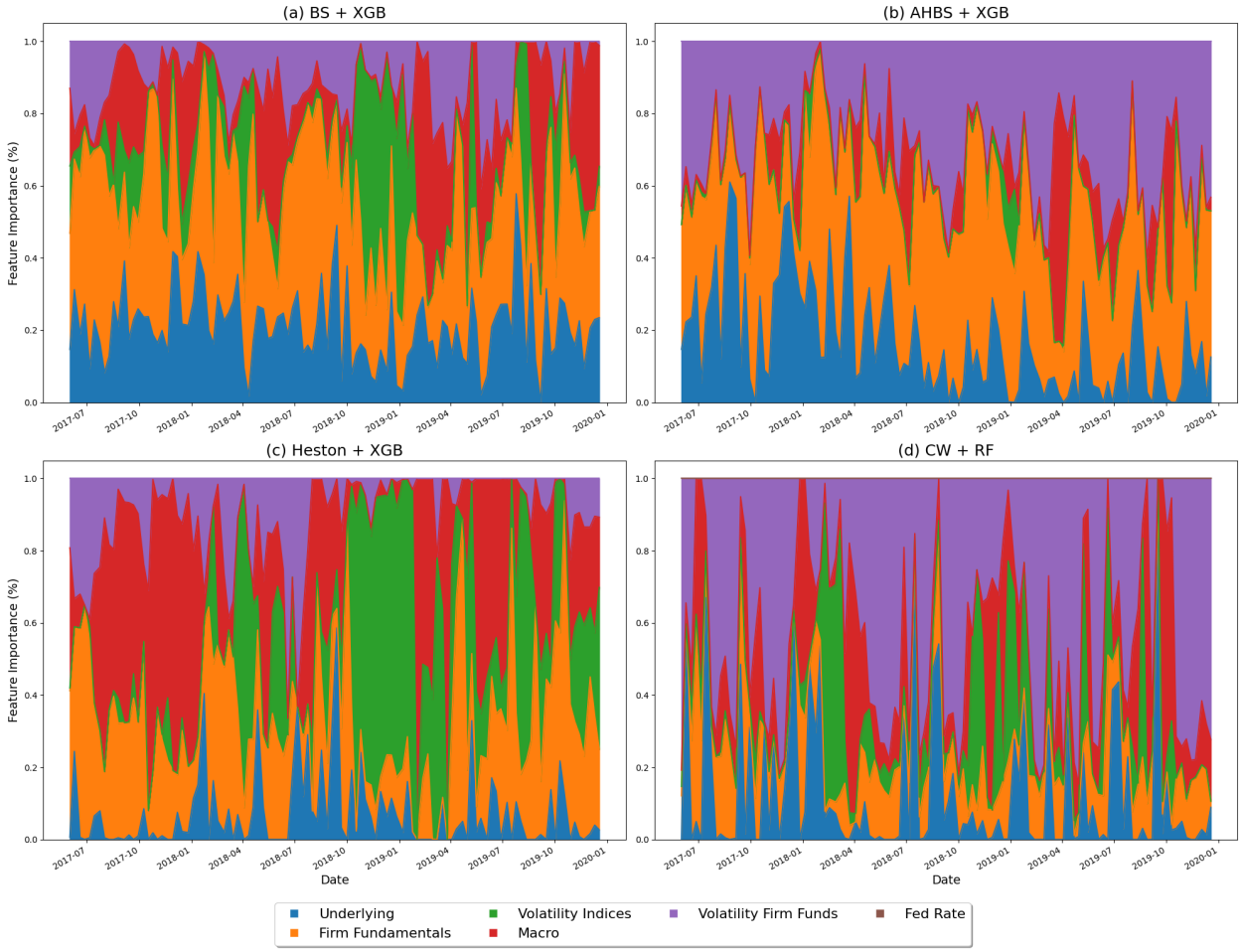
features, please refer to Appendix A.5, Figure A1.

Upon observation, we see that the feature importance across different groups within the CW + RF model exhibits less homogeneity compared to the BS + XGB model. This disparity suggests that the relative importance of individual features varies significantly between models. However, it is crucial to note that these models operate with different features. Therefore, a more meaningful exercise would involve comparing the feature importance within a specific model.

In Figure 8d, we observe considerable variation in feature importance across the different groups, particularly between the macroeconomic variables and the volatility indices. Notably, the volatility indices features demonstrate a heightened influence between October 2018 and February 2019 for models (a) and (d). This increase in importance could be attributed to the escalating trade tensions between the United States and China, which contributed to increased market volatility. Furthermore, Figures 8a and 8b highlight the consistent significance of the firm fundamentals group across the entire sample period. Additionally, Figure 8c reveals a substantial feature importance for the firm fundamental features concerning volatility.

Interestingly, graph (c) exhibits frequent spikes where macroeconomic features and volatility indices assume a more influential role, indicating a relatively varied distribution of importance across feature groups. Regarding the AHBS + XGB model, the volatility indices appear less crucial than the other feature groups. There, the importance of features is largely divided into three groups: covariates regarding the underlying asset, firm fundamentals, and covariates of firm fundamentals regarding volatility. In contrast, external information from macroeconomic variables and volatility indices are more important for the Heston model.

Figure 8: Feature importance over time for the best-performing machine-corrected model per option pricing model, excluding the option-specific features

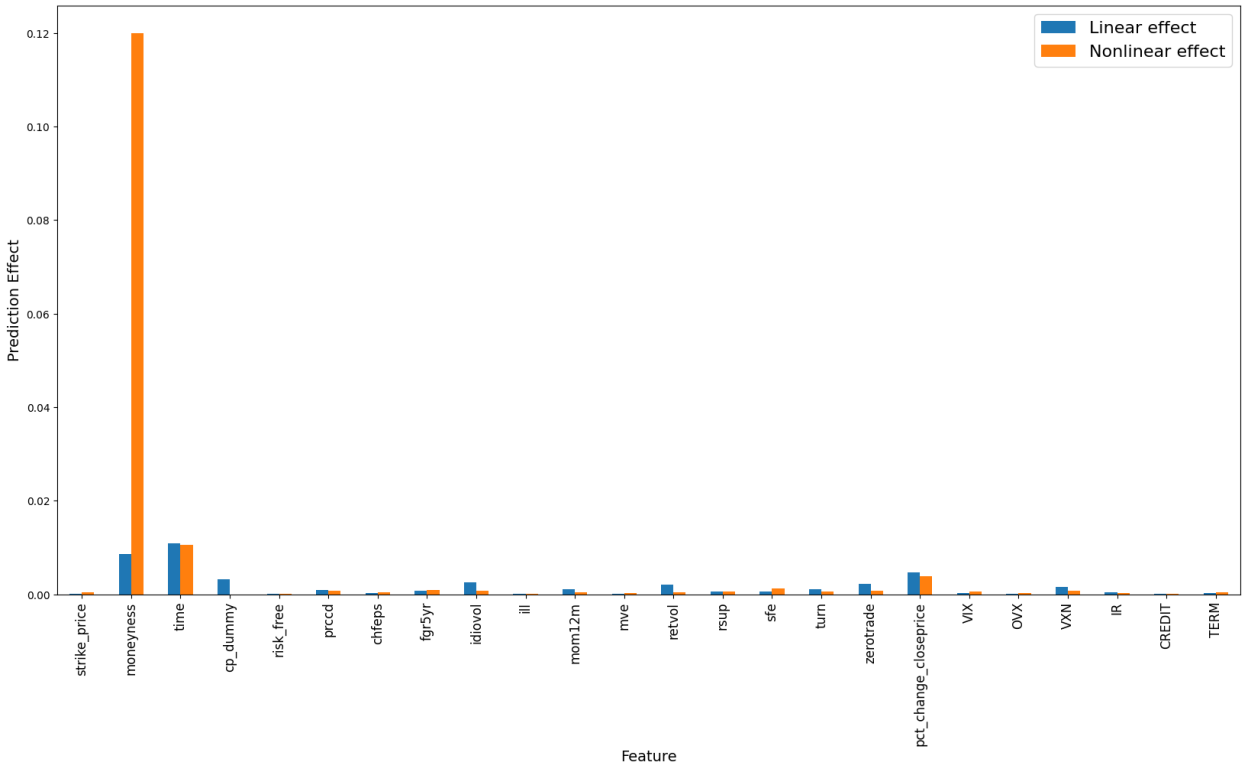


6.4 Interpretability

To improve interpretability, we computed partial predictions for each feature in our BS + XGBoost model and decomposed them into linear and nonlinear. Figure 9 presents the outcomes of the decomposition.

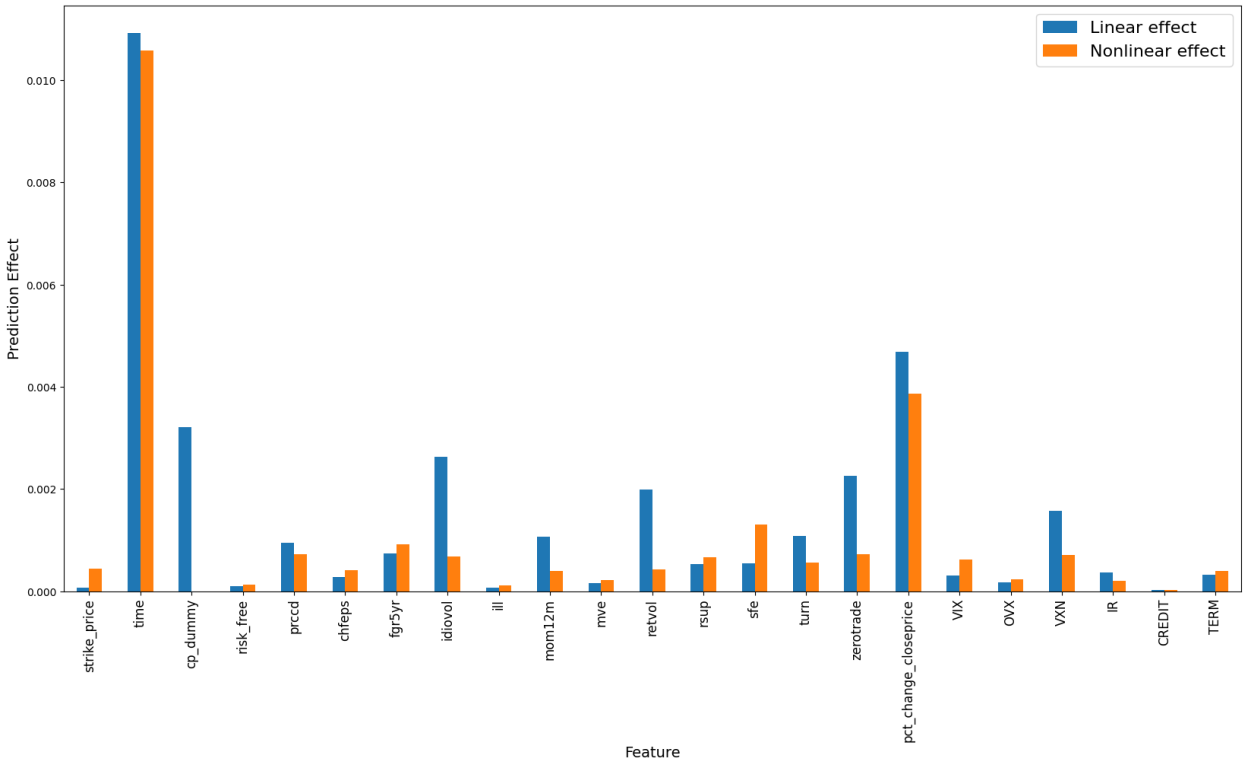
From Figure 9, we see a strong nonlinear effect of moneyness on the partial predictions, clearly carrying the largest effect. Next to this feature, the option-specific variables time-to-maturity and the dummy for call/put option are among the features with the highest predictions effects along with the percentage in price change.

Figure 9: Decompositions of the linear and nonlinear effect for the out-of-sample partial predictions from the BS + XGBoost model



Due to the large effect of moneyness, the effects of the other features become less visible. To increase visibility, we exclude moneyness from the figure. The new figure can be seen in Figure 10. The nonlinear effect is most prominent in 13 out of the 24 features. If we look at the covariates apart from the first four option-specific variables, we see that the linear effect mostly dominates the total prediction effect implying a more linear relation with the residuals of the BS model.

Figure 10: Decompositions of the linear and nonlinear effect for the out-of-sample partial predictions from the BS + XGBoost model without the moneyness feature



7 Conclusion

In this paper, we present an adaptable method to enhance the accuracy of parametric option pricing models using machine learning models inspired by the framework introduced by Almeida et al. (2022). Our main objective is to accurately predict the implied volatility surface of individual equity options. The approach involves two steps: calibrating a parametric model to align with the observed IVS and training a secondary model based on the pricing errors derived from the pricing model. The effectiveness of the nonparametric correction is evaluated using a large dataset containing options from the 50 most liquid underlying stocks in the U.S. between 2014 and 2019. This dataset is expanded with various covariate data, including firm fundamentals, macroeconomic indicators, and shape characteristics of the S&P 500 index options.

The empirical results demonstrate that the machine-corrected regression tree models perform favorably in improving option pricing models. Specifically, the XGBoost combined with the BS model results in the most accurate predictions of the IVS. The comparative analysis

of the machine-corrected models against the respective parametric models and a benchmark model (AHBS) indicates the approach's efficacy in improving option pricing. The machine-corrected models consistently outperform the parametric models regarding IVRMSE and other performance metrics, such as outperformance rate, median loss ratio, and mean loss ratio.

Furthermore, the feature importance and interpretability analysis reveal the importance of option-specific features, implying that the parametric models do not optimally extract all information from these features. Additionally, the analysis shows that firm fundamentals, macroeconomic variables, and volatility indices played significant roles in predicting the IVS. The specific importance of features varies across different parametric and machine learning models, highlighting the diversity of information captured by these models.

Additionally, the interpretability analysis provides insights into the linear and nonlinear effects of features on the predictions of the machine-corrected Black-Scholes model. The nonlinearity effects, particularly in features like moneyness, time-to-maturity, and option type, substantially influence the predictions. The firm fundamental covariates show a more linear relation to the model errors.

Although the machine-corrected models perform well, it is important to acknowledge the limitations of this study. Firstly, the CW and Heston models perform poorly as parametric models, which may be attributed to the illiquidity of individual equity options and their sensitivity to extreme time-to-maturities. The limited availability of data points for individual equities and the challenges in accurately estimating parameters for these models might contribute to their underperformance.

Secondly, the methodology employed in this study is computationally heavy, requiring significant computational resources and processing time. This computational complexity could present practical challenges and limit the scalability of the approach. Additionally, the lack of computational power restricts us to using one fixed test set.

In light of these limitations, future research could explore alternative parametric models that address the illiquidity and extreme time-to-maturity issues when predicting individual equity options. Additionally, investigating different machine learning models or ensembles of models could further enhance predictive accuracy. Furthermore, while computationally demanding, expanding or moving window analysis may provide valuable insights into the model's performance over time and predictive accuracy. Lastly, varying the train and test split in the dataset would contribute to more robust results and a more concise conclusion.

Despite these limitations, our research demonstrates that machine learning techniques can successfully improve the accuracy of parametric option pricing models to predict the IVS of equity options. The findings contribute to a better understanding of the dynamics of IVS and

provide valuable insights for market participants, traders, and investors in managing option positions and making informed investment decisions. Additionally, this paper contributes to the currently shallow body of literature on the predictability of individual equity options and can be seen as a stepping stone for further research.

References

Aït-Sahalia, Y., Li, C., & Li, C. X. (2021). Implied stochastic volatility models. *The Review of Financial Studies*, *34*(1), 394–450.

Albrecher, H., Mayer, P., Schoutens, W., & Tistaert, J. (2007). The little heston trap. *Wilmott*(1), 83–92.

Almeida, C., Fan, J., Freire, G., & Tang, F. (2022). Can a machine correct option pricing models? *Journal of Business & Economic Statistics*, 1–14.

An, B.-J., Ang, A., Bali, T. G., & Cakici, N. (2014). The joint cross section of stocks and options. *The Journal of Finance*, *69*(5), 2279–2337.

Andersen, T. G., Bollerslev, T., Diebold, F. X., & Labys, P. (2001). The distribution of realized exchange rate volatility. *Journal of the American statistical association*, *96*(453), 42–55.

Bates, D. S. (1996). Jumps and stochastic volatility: Exchange rate processes implicit in deutsche mark options. *The Review of Financial Studies*, *9*(1), 69–107.

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

Bianchi, D., Büchner, M., & Tamoni, A. (2021). Bond risk premiums with machine learning. *The Review of Financial Studies*, *34*(2), 1046–1089.

Black, F. (1976). The pricing of commodity contracts. *Journal of financial economics*, *3*(1-2), 167–179.

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

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

Buhler, C., Winkler, V., Runge-Ranzinger, S., Boyce, R., & Horstick, O. (2019). Environmental methods for dengue vector control–a systematic review and meta-analysis. *PLoS neglected tropical diseases*, *13*(7), e0007420.

Carr, P., & Madan, D. (1999). Option valuation using the fast fourier transform. *Journal of computational finance*, *2*(4), 61–73.

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

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

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Chernov, M., Gallant, A. R., Ghysels, E., & Tauchen, G. (2003). Alternative models for stock price dynamics. *Journal of Econometrics*, *116*(1-2), 225–257.

Christoffersen, P., Fournier, M., & Jacobs, K. (2018). The factor structure in equity options. *The Review of Financial Studies*, *31*(2), 595–637.

Conn, D., Ngun, T., Li, G., & Ramirez, C. M. (2019). Fuzzy forests: Extending random forest feature selection for correlated, high-dimensional data. *Journal of Statistical Software*, *91*, 1–25.

Conn, D., Ngun, T., & Ramirez, C. M. (n.d.). Fuzzy forests: Extending random forest feature selection for correlated, high-dimensional data. *Journal of Statistical Software*.

Dai, Q., & Singleton, K. J. (2000). Specification analysis of affine term structure models. *The journal of finance*, *55*(5), 1943–1978.

Das, S. P., & Padhy, S. (2017). A new hybrid parametric and machine learning model with homogeneity hint for european-style index option pricing. *Neural Computing and Applications*, *28*, 4061–4077.

Daul, S., Jaisson, T., & Nagy, A. (2022). Performance attribution of machine learning methods for stock returns prediction. *The Journal of Finance and Data Science*, *8*, 86–104.

David, A., & Veronesi, P. (2000). Option prices with uncertain fundamentals: Theory and evidence on the dynamics of implied volatilities. *Available at SSRN 199332*.

Dennis, P., & Mayhew, S. (2002). Risk-neutral skewness: Evidence from stock options. *Journal of Financial and Quantitative Analysis*, *37*(3), 471–493.

Duffie, D., Pan, J., & Singleton, K. (2000). Transform analysis and asset pricing for affine jump-diffusions. *Econometrica*, *68*(6), 1343–1376.

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

Dupire, B., et al. (1994). Pricing with a smile. *Risk*, *7*(1), 18–20.

Fang, F., & Oosterlee, C. W. (2009). Pricing early-exercise and discrete barrier options by fourier-cosine series expansions. *Numerische Mathematik*, *114*(1), 27.

Freire, G., & Kleen, O. (2023). Equity options and firm characteristics. *Available at SSRN 4342597*.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Gatheral, J. (2011). *The volatility surface: a practitioner's guide*. John Wiley & Sons.

Gatheral, J., & Jacquier, A. (2014). Arbitrage-free svi volatility surfaces. *Quantitative Finance*, *14*(1), 59–71.

Ghysels, E., & Ng, S. (1998). A semiparametric factor model of interest rates and tests of the affine term structure. *Review of Economics and Statistics*, *80*(4), 535–548.

Green, J., Hand, J. R., & Zhang, X. F. (2017). The characteristics that provide independent information about average us monthly stock returns. *The Review of Financial Studies*, *30*(12), 4389–4436.

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

Han, Y., He, A., Rapach, D., & Zhou, G. (2022). Expected stock returns and firm characteristics: E-enet, assessment, and implications. *Assessment, and Implications (August 28, 2022)*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the ieee international conference on computer vision* (pp. 1026–1034).

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, *6*(2), 327–343.

Ivaşcu, C.-F. (2021). Option pricing using machine learning. *Expert Systems with Applications*, *163*, 113799.

Jäckel, P. (2015). Let's be rational. *Wilmott*, *2015*(75), 40–53.

Kelly, B. T., Pruitt, S., & Su, Y. (2019). Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics*, *134*(3), 501–524.

Langfelder, P., & Horvath, S. (2008). Wgcna: an r package for weighted correlation network analysis. *BMC bioinformatics*, *9*(1), 1–13.

Li, Y., Turkington, D., & Yazdani, A. (2020). Beyond the black box: an intuitive approach to investment prediction with machine learning. *The Journal of Financial Data Science*, *2*(1), 61–75.

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

Loh, W.-Y. (2011). Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *1*(1), 14–23.

Luo, Q., Jia, Z., Li, H., & Wu, Y. (2022). Analysis of parametric and non-parametric option pricing models. *Heliyon*, *8*(11), e11388.

Masters, T. (1993). *Practical neural network recipes in c++*. Academic Press Professional, Inc.

Medvedev, A., & Scaillet, O. (2007). Approximation and calibration of short-term implied volatilities under jump-diffusion stochastic volatility. *The Review of Financial Studies*, *20*(2), 427–459.

Mehrdoust, F., Saber, N., & Najafi, A. R. (2017). Modeling asset price under two-factor heston model with jumps. *International Journal of Applied and Computational Mathematics*, *3*, 3783–3794.

Nesterov, Y. E. (1983). A method of solving a convex programming problem with convergence rate o\bigl(kˆ2\bigr). In *Doklady akademii nauk* (Vol. 269, pp. 543–547).

Rathgeber, A., Stadler, J., & Stöckl, S. (2021). The impact of the leverage effect on the implied volatility smile: evidence for the german option market. *Review of Derivatives Research*, *24*, 95–133.

Rubinstein, M. (1994). Implied binomial trees. *The journal of finance*, *49*(3), 771–818.

Ruf, J., & Wang, W. (2022). Hedging with linear regressions and neural networks. *Journal of Business & Economic Statistics*, *40*(4), 1442–1454.

Skiadopoulos, G., Hodges, S., & Clewlow, L. (2000). The dynamics of the s&p 500 implied volatility surface. *Review of derivatives research*, *3*, 263–282.

Sussillo, D., & Abbott, L. (2014). Random walk initialization for training very deep feed-forward networks. *arXiv preprint arXiv:1412.6558*.

Tian, M., Li, W., & Wen, F. (2021). The dynamic impact of oil price shocks on the stock market and the usd/rmb exchange rate: Evidence from implied volatility indices. *The North American Journal of Economics and Finance*, *55*, 101310.

Zheng, Y., Yang, Y., & Chen, B. (2019). Gated deep neural networks for implied volatility surfaces. *arXiv preprint arXiv:1904.12834*, *7*.

A Appendix

A.1 Options Data

The 50 tickers included in our dataset are displayed in Table A1. These specific stocks are chosen based on the stocks from the dataset provided by D. Chen et al. (2023) in their study. After applying the filters described in Section 2.1, we identify the top 50 stocks with the greatest average options liquidity between January 1, 2014, and December 31, 2019.

Table A1: Tickers of 50 most liquid stocks included in the dataset

Tickers				
AAL	C	HD	NFLX	TWRT
AAPL	CAT	IBM	NKE	UAL
ABBV	CMG	INTC	NVDA	V
ADBE	COST	ISRG	QCOM	VLO
AMD	CRM	JPM	REGN	WDC
AMZN	CVX	LRCX	SBUX	WFC
BA	DAL	LULU	SLB	WMT
BAC	DIS	MA	SWKS	WYNN
BIIB	GILD	MSFT	TGT	X
BKNG	GS	MU	TSLA	XOM

A.2 Covariate Data

Table A2 provides a full description of the variables in addition to the firm characteristics. For an overview of the firm fundamentals, please refer to the paper of D. Chen et al. (2023).

Table A2: Covariate descriptions and abbreviations of the additional predictor variables

Variable	Description
CREDIT	Moody's Seasoned Baa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity
daily_vol	Daily volume of S&P 500 index options
DFF	Federal Funds Rate
dvi	Annualized dividend
EPUI	Economic Policy Uncertainty Index
EUI	Economic Uncertainty Index
EXCH	U.S. Dollars to Euro Spot Exchange Rate
GVZ	CBOE Gold ETF Volatility Index
IR	5 Year Break-even Inflation
level	Level of IVS of SP500 index options
OVX	Crude Oil Volatility Index
pct_change_closeprice	% change in closing price of underlying asset
prccd	Close price of underlying
risk_free	Risk free rate
slope	Slope of IVS SP500 index options
SP500	Close price of SP500 index
TERM	10-Year Treasury Constant Maturity Minus 3-Month Treasury Constant Maturity
term_structure	Term structure of IVS of SP500 index options
VIX	CBOE Volatility Index
VXN	CBOE Nasdaq 100 Volatility Index

A.3 Heston Pricing Solution

Albrecher et al. (2007) provided proof of the instability of the original Heston formulation under certain conditions. To overcome this issue of discontinuity formulation, we use the alternative formulation from Gatheral (2011). The quasi-closed form solution of the Heston model for pricing European options, as defined by Gatheral (2011), becomes:

$$C(S_t, K, V_0, \tau) = P(\tau) \cdot \left[\frac{1}{2}(F - K) + \frac{1}{\pi} \int_0^\infty (F f_1 - K f_2) \mathrm{d}u \right], \tag{23}$$

with f_1 and f_2,

$$f_1 = \mathrm{Re}\left(\frac{\mathrm{e}^{-iu\ln K}\varphi(u-i)}{iuF}\right) \text{and} f_2 \quad = \mathrm{Re}\left(\frac{\mathrm{e}^{-iu\ln K}\varphi(u)}{iu}\right), \tag{24}$$

where $F = S\mathrm{e}^{\mu\tau}$ and $P(\tau) = \mathrm{e}^{-r\tau}$ as the discount factor to the expiry date of the option. The log-characteristic function $\varphi(\cdot)$ is defined as as a function of the underlying value S_τ at expiration, such that $\varphi(0) = 1$ and $\varphi(-i) = F$:

$$\varphi(u) = \mathbb{E}\left[\mathrm{e}^{-iu\ln S_\tau}\right]. \tag{25}$$

For the Heston model, this equation can be written as:

$$\varphi(u) = \mathrm{e}^{C(\tau,u)+D(\tau,u)V_0+iu\ln F}. \tag{26}$$

The new formulation of the coefficients C and D and their auxiliary functions read:

$$C(\tau,u) = \frac{\kappa\overline{v}}{\sigma_v^2}\left((\kappa-\rho\sigma_v ui - d(u))\tau - 2\ln\left(\frac{1-c(u)\mathrm{e}^{-d(u)\tau}}{1-c(u)}\right)\right), \tag{27}$$

$$D(\tau,u) = \frac{\kappa-\rho\sigma_v ui - d(u)}{\sigma_v^2}\left(\frac{1-\mathrm{e}^{-d(u)\tau}}{1-c(u)\mathrm{e}^{-d(u)\tau}}\right), \tag{28}$$

$$c(u) = \frac{\kappa-\rho\sigma_v ui - d(u)}{\kappa-\rho\sigma_v ui + d(u)}, \tag{29}$$

$$d(u) = \sqrt{(\rho\sigma_v ui-\kappa)^2 + iu\sigma_v^2 + \sigma_v^2 u^2}. \tag{30}$$

Lastly, we apply Fast Fourier Transform method from the `SciPy` library to reduce computational costs (Carr & Madan, 1999).

A.4 Hyperparameter tuning

We define a grid of hyperparameters for all models to iterate over and select the optimal model. These hyperparameters, which can take on multiple values as described below, will be evaluated against the mean squared error to determine the model with minimal error.

A.4.1 WGCNA

For both the WGCNA and the FF, we evaluate the MSE of the models at the last RF iteration. For the WGCNA framework, we only tune the number of clusters and the minimum size of each cluster. To identify appropriate hyperparameters, we initially examine a correlation heatmap displaying the cross-correlations among the predictor variables. We observe around ten distinct groups of features with strong correlations, each containing more than two variables. Therefore, we set the minimum cluster size to three and explored ten potential clusters. For further computational details, we refer to Appendix A.6.

A.4.2 FF

Table A3 provides an overview of the hyperparameters tuned to identify the optimal feature reduction model using the FF algorithm. Due to the computational heaviness of the algorithm, we only use the model implied errors of the BS model to tune the parameters. The chosen parameter values are in bold in Table A3. First, the drop fraction indicates the proportion of features that will be eliminated after each RFE-RF step, which filters out insignificant variables as part of the screening step. Starting with all predictor variables in the partition P_l, a random forest model is fitted, and the least important variables are subsequently removed. We repeat this process iteratively using the reduced set $P_l^{(1)}$ until the specified number of features determined by the keep fraction is reached. We use a wide range of possible values to tune the drop fraction. For further details on the FF algorithm and the RFE-RF procedure, we refer to Section 2.4 in the paper of Conn et al. (2019).

As mentioned above, the keep fraction represents the number of features to keep after the final FF iteration. We set this fraction at 45%, providing a broad range of potentially interesting features. From the VIM plot in Figure 6 from Section 6.1, we select the number of features that collectively account for two-thirds of the overall variable importance.

Lastly, after completing each FF algorithm, we plot the MSE as a function of the number of iterations. The MSE starts to marginally decline between 25 and 75 iterations. Therefore, we decided to consider the final number of trees equal to 25, 50, or 75 iterations. It is unlikely that the MSE would significantly improve across all other hyperparameters beyond 75 iterations, as the RF may start overfitting. Additionally, we set the minimum number of

trees for each RF at 100, not too large, as it would be too computationally expensive.

Table A3: Hyperparameter grid for WGCNA and FF

Method	Hyperparameter	Parameter Grid
WGCNA	#Clusters	{**10**}
	Min Cluster Size	{**3**}
Fuzzy Forests	Drop Fraction	{0.05, 0.15, 0.25, **0.5**}
	Keep Fraction	{**0.45**}
	Final # Trees	{25, **50**, 75}
	Min # Trees	{**100**}

A.4.3 ENet

In Table A4, we provide the tuning grid of the ENet method. We consider two hyperparameters, as mentioned in Section 3.3.1. The first hyperparameter, λ, determines the regularization in the model. The second parameter, ρ, determines the shift between the two regularization methods Lasso and ridge regression.

For λ, we focus on relatively small values as we already implement the FF algorithm to reduce the dimension of predictor variables. We consider 1000 values between the logspace starting from 10^{-100} to 10^{-1}.

To control for overfitting, we do not consider values of ρ that are too heavy on one side of the regularization methods. Hence, we take three values around 0.5.

Table A4: Hyperparameter grid for ENet

Hyperparameter	Parameter Grid	BS	AHBS	Heston	CW
λ	{min=1e-100, max=1e-01, steps=1000}	1e-12	0.1	1e-12	0.4571
ρ	{0.3, 0.5, 0.7}	0.5	0.5	0.3	0.7

A.4.4 RF

Table A5 displays the main hyperparameters for RF that require tuning. First, we adjust the number of decision trees (estimators) to control the ensemble's size and improve accuracy. Generally, higher estimator values enhance prediction accuracy. However, computational costs increase, and a point of convergence is reached where additional trees offer limited accuracy improvement. Next, we optimize the maximum depth to increase model complexity

by allowing more splits in each tree. We use a grid with a wide range of potential maximum depths. Lastly, we fine-tune the maximum number of randomly selected features used at each node split to mitigate overfitting. Our hyperparameter grid for the last parameter is narrower than Gu et al. (2020) due to the implementation of the FF feature reduction technique, which already reduces the number of predictor variables in our models.

Table A5: Hyperparameter grid for RF

Hyperparameter	Parameter Grid	BS	AHBS	Heston	CW
Max # of Trees	{100, 200, 500}	500	500	500	100
Max Depth	{6, 8, 10, 12}	12	12	10	8
Max Features	{3, 5, 10}	10	10	10	10

A.4.5 XGBoost

For XGBoost, the hyperparameter grid we use is shown in Table A6. First, the number of estimators D determines the number of trees in the ensemble. Second, we tune η, the learning rate parameter, which shrinks the weighting of new features added to our model, making the process more conservative. In their study, Friedman (2001) discovered that lower learning rates result in improved test errors, albeit requiring a larger number of estimators. Consequently, a trade-off exists between the learning rate and the number of estimators. Given the substantial size of our estimator grid, we suggest using two relatively smaller learning rate values (0.01 and 0.1). Third and last, we tune the maximum depth of the regression trees. The maximum depth and number of estimators determine the model's complexity and flexibility. It is crucial to strike a balance when selecting the depth parameter to avoid overfitting the algorithm. Therefore, in our grid, we align with the approach of Breiman (2001) and Gu et al. (2020), who use boosted regression trees to analyze US government bonds and US equities, respectively.

Table A6: Hyperparameter grid for XGBoost

Hyperparameter	Parameter Grid	BS	AHBS	Heston	CW
# estimators	{25, 50, 100, 200}	100	100	50	25
Learning Rate	{0.01, 0.1, 0.3}	0.1	0.1	0.1	0.1
Maximum Depth	{6, 12}	6	6	6	6
Subsample Size	{1, 0.75, 1}	1	1	1	0.5

A.4.6 NN

Table A7 displays the comprehensive grid used for selecting hyperparameters for our neural networks, along with the optimal combinations for each parametric model. We have a collection of optimal parameters (L1L2 Penal, Dropout, Batch Size, Learning Rate) specific to each architecture NN1 to NN5. These hyperparameters are chosen within the framework of a normal NN (i.e., without an additional input layer to the output layer). As previously mentioned, the hyperparameters for NN-OS and NN-FF consist of the optimal results obtained from all five structures. The chosen architectures for NN-OS and NN-FF are outlined in Table A8.

Table A7: Hyperparameter grid for NN

Hyperparameter	Parameter Grid	BS	AHBS	Heston	CW
L1L2 Penal	{1e-14, 1e-18, 1e-20}	1e-14	1e-20	1e-14	1e-14
Dropout	{0.4, 0.5, 0.6, 0.7}	0.6	0.6	0.4	0.4
Batch Size	{128, 256}	128	256	128	256
Learning Rate	{0.001, 0.01}	0.001	0.001	0.001	0.001
Architecture	{NN1, NN2, NN3, NN4, NN5}	NN3	NN4	NN1	NN1

Table A8: Optimal neural network architecture per adjusted structure

Structure	BS	AHBS	Heston	CW
NN-OS	NN1	NN5	NN2	NN1
NN-FF	NN1	NN1	NN1	NN1

Tables A7 and A8 reveal the selection of relatively simple neural network architectures for the BS, Heston, and CW models, primarily consisting of a single hidden layer. In contrast, the AHBS model demonstrates the lowest MSE on the validation set with more complex architectures, indicating the preference for a more intricate structure.

A.5 Feature Importance

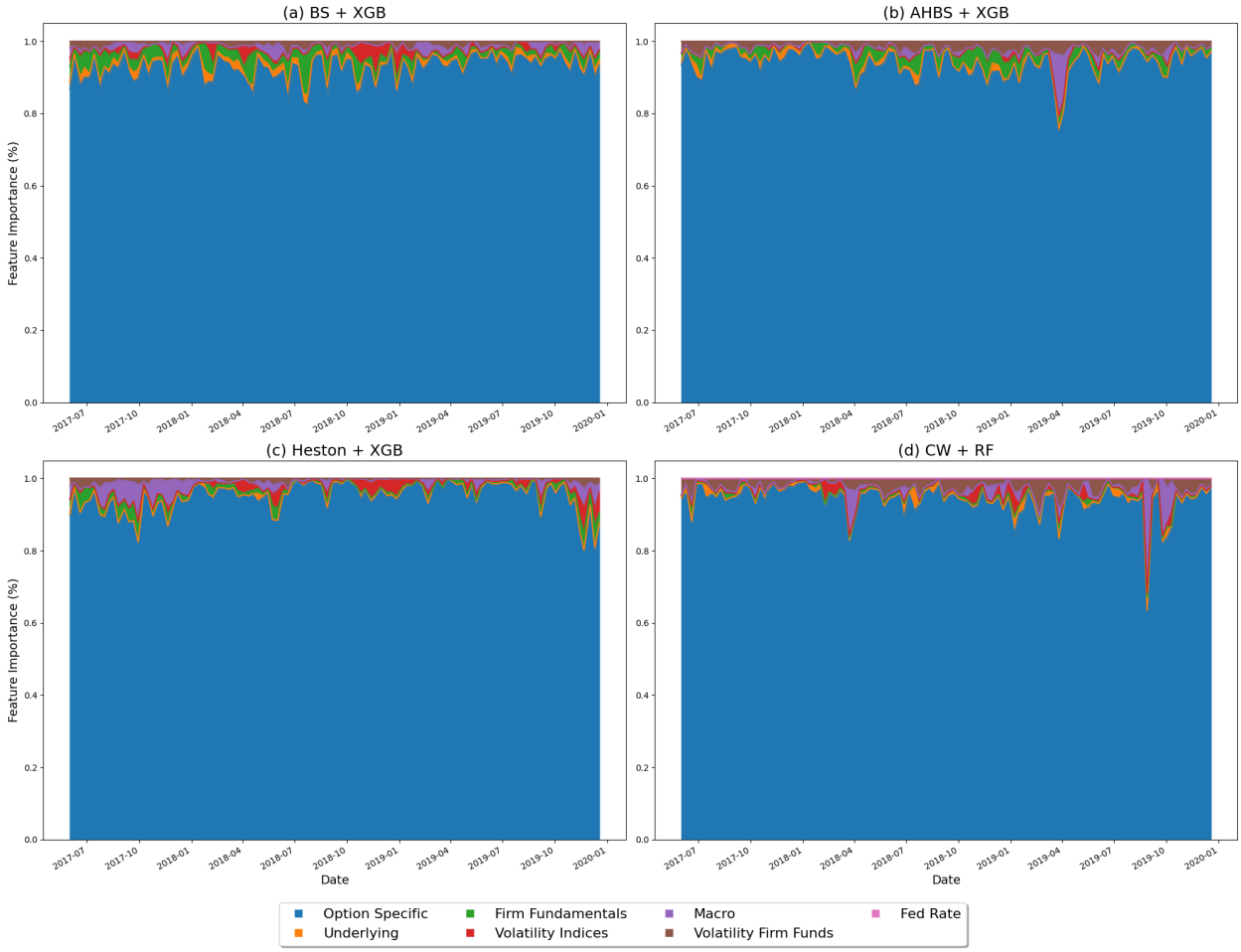
We have grouped the predictor variables based on their similar type to improve interpretations of the results from the feature importance over time. The groups are in Table A9.

Next to the results in Section 6.3, we present the feature importance over time, including the option-specific variables. Figure A1 shows that the option-specific group holds a considerably large share in feature importance. This is expected given the varying levels of implied volatility across the option cross-section.

Table A9: Groups of variables formed for the feature importance

Group	Features
Underlying	prccd
	pct_change_close
Firm Fundamentals	acc
	betasq
	chfeps
	fgr5yr
	ill
	indmom
	mom12m
	mve
	mve_ia
	roeq
	sfe
	turn
	zerotrade
Option Specific	cp_dummy
	moneyness
	strike_price
	time
Volatility Indices	OVX
	SP500
	VIX
	VXN
Fed Rate	DFF
Macro	CREDIT
	IR
	risk_free
	TERM
Volatility Firm Fundamentals	ba_spread
	idiovol
	retvol
	roavol
	std_turn

Figure A1: Feature importance over time for the best-performing machine-corrected model per option pricing model, including option-specific features



A.6 Computational details

For the implementation of the fuzzy forests algorithm, we use the R-packages `WGCNA` for the WGCNA framework (Langfelder & Horvath, 2008) and `fuzzyforests` (Conn et al., n.d.).

We use the Python package `Scikit-Learn` to process the data, split our data randomly, and standardize/scale our input variables. This package also includes the `LinearRegression()` function needed for AHBS, `ElasticNet()` for ENet, and the `RandomForestRegressor()` ensemble model for the Random Forests. For the optimization of the parametric models Heston and CW, we use the `scipy` package in Python, which includes both `differential_evolution()` and `least_squares()` functions. The Python wrapper for the `Vollib` library calculates the implied volatility in the Heston model.

To implement the other machine-learning models, we use the following Python packages:

`xgboost`, `TensorFlow 2`, and `Keras`. The NNs are trained and fitted using the high-level API called `Keras` from the `TensorFlow` library. The `Keras` wrapper provides various regularization techniques used in our paper (i.e., batch normalization, dropout, early stopping).