Erasmus University Rotterdam

Erasmus School of Economics - Econometrics and Management Science

Quantitative Finance - Master thesis

Improving Implied Volatility Predictions of Individual Equity Options: A Two-Step Procedure Integrating Machine Learning

Author:

Melih Samutoglu

Student ID:

501628

Supervisor:

Dr. G. Freire

Second assessor:

Prof. Dr. C. Zhou

April 21, 2024

Abstract

The Implied Volatility, derived from the Black-Scholes model, is a one-to-one mapping of the price of an option. It provides insights into the financial market participants' expectations of the underlying asset's behavior throughout the duration of the option. Well-known parametric models are commonly employed to predict the Implied Volatility. In recent years, machine learning models have gained significant popularity for their state-of-the-art performance in Implied Volatility prediction. This study aims to exploit the potential of machine learning, building upon the unique two-step Implied Volatility prediction procedure proposed by Almeida et al. (2022). We attempt to improve the Implied Volatility prediction of the Black-Scholes, Ad-Hoc Black-Scholes and Carr-Wu model with non-parametric models such as Random Forest, Extreme Gradient Boosting and Neural Network models. Our analysis incorporates individual American-style equity options from January, 2000 up to and including December, 2021. The findings of this research illustrate that the tree-based non-parametric models improve the Implied Volatility prediction of the parametric models in various scenarios.

Keywords: Equity Options, Implied Volatility, Machine Learning

Contents

List of Figures

List of Tables

1 Introduction

It is essential to accurately price options for financial market participants in order to effectively manage their portfolio. Hence, various parametric option pricing models, such as the well-known Black-Scholes (BS) model introduced by Black and Scholes (1973), have been developed over the years to predict the value of an option. These parametric models are commonly solved for the volatility parameter referred to as the Implied Volatility (IV). The IV is a popular measure for institutional investors to assess the relative value of an option (Carr and Wu, 2016). Furthermore, the IV is a critical component of the Implied Volatility Surface (IVS), which represents the relationship between the option's IV, maturity and strike price (moneyness). The IVS provides valuable insights into market expectations of financial market participants.

Despite the developments in option pricing theory, the parametric models produce a prediction of the IV that leaves room for improvement. In recent years, the popularity of machine learning models, also known as non-parametric models, across various research fields, including option pricing theory, has grown due to their potential to capture complex patterns and strong predictive performance. Consequently, Almeida et al. (2022) propose the innovative idea of correcting the IV predictions of a parametric model with a non-parametric model using a two-step prediction procedure. They employ a Feed-Forward Neural Network (FNN) in order to correct various parametric models and observe a significant improvement in the IV predictions.

We follow the two-step prediction procedure of Almeida et al. (2022) and extend their research by conducting a comparative study of multiple non-parametric models similar to the study of Gu et al. (2020). Specifically, the BS, Ad-Hoc Black-Scholes (AHBS) and Carr-Wu (CW) model are the relevant parametric models (Black and Scholes, 1973; Dumas et al., 1998; Carr and Wu, 2016). These are corrected by an FNN, a Random Forest (RF) or an Extreme Gradient Boosting (XGBOOST) model (Breiman, 2001; Chen and Guestrin, 2016). Furthermore, we deviate from Almeida et al. (2022) by exploring individual equity options instead of S&P 500 index options. This allows us to include firm characteristics as features in the non-parametric models. In addition, we determine the importance of the features with the use of Shapley Additive Explanation (SHAP) values

(Lundberg and Lee, 2017).

The focus of our research is to compare and analyze the ability of various non-parametric models to correct the IV prediction of the parametric models. Therefore, the main question of our research is formulated as follows: *"Can a non-parametric model correct the Implied Volatility prediction of a parametric model for individual equity options?"*. Moreover, some interesting follow-up questions are *"Which combination of the analyzed parametric and non-parametric model results in the best prediction of the Implied Volatility?"* and *"Which features are the most important for the prediction of the Implied Volatility?"*.

We obtain end-of-month data for individual American-style equity options in the period of January, 2000 up to and including December, 2021, which is accessible through OptionMetrics (OM) via Wharton Research Data Services (WRDS). Furthermore, the Center for Research in Security Prices (CRSP) provides the historical price of equities and is also available in WRDS. In addition, the firm characteristic and macroeconomic data originate from the papers by Gu et al. (2020) and Welch and Goyal (2008), respectively.

We attempt to leverage the predictive capability of non-parametric models, namely FNN, RF and XGBOOST, to correct the IV prediction of parametric models, such as BS, AHBS and CW. In the initial step, the IV is predicted with a parametric model. Subsequently, the residuals of these predictions become the target variable and are predicted using a non-parametric model. Afterwards, the corrected IV is achieved by combining the two predictions. The final step consists of evaluating the performance of the models using the Root Mean Squared Error (RMSE), Root Mean Squared Percentage Error (RMSPE) and Outperformance Rate (OR). This evaluation is conducted across three distinct empirical scenarios. In the first and second scenario, a daily and quarter year rolling window approach is employed for the prediction and evaluation step mentioned above. In the third scenario, the training time period spans the initial 19 years, while the test time period covers the remaining two years.

Our research shows that the tree-based non-parametric models consistently improve the predictions of the parametric models across all empirical scenarios. Particularly, the BS model corrected by the RF or XGBOOST model are the top-performing two-step

prediction models. In contrast, the parametric models corrected by the NN models have a subpar performance. Moreover, the correction of the NN models lead to a deterioration of the parametric models' IV prediction in some cases. While the two-step prediction procedure results in improvements in the IV predictions of the parametric models, a similar performance is achieved by predicting the IV directly with the non-parametric models. The prediction of the two-step models are mostly influenced by option characteristics and macroeconomic features based on the feature importance determined by the SHAP values. When we employ the non-parametric models with exclusively the top seven most important features, the performance of the two-step prediction models is similar to the original case with all the features included. In addition, we conduct the empirical analyses on the top five liquid firms and observe an improved or similar performance of the individual and two-step prediction models in most cases compared to the analyses involving all the firms.

The empirical findings presented in this research can have practical implications for a wide range of professionals. This includes options traders looking to refine their strategies and portfolio managers aiming for optimized investment decisions. Furthermore, financial institutions can benefit from improved risk management while analysts and researchers gain access to more accurate option pricing data to improve their research. Moreover, our paper focuses on individual equity options, which represent a significant part of the financial market's value, instead of the commonly utilized S&P 500 index options in research regarding IV prediction.

The remainder of the paper proceeds as follows. We provide a brief literature review in Section 2. Next, the relevant data is described in Section 3. In Section 4, we introduce the models and methods. Afterwards, Section 5 presents the results. Finally, we conclude and discuss our results in Section 6 and Section 7, respectively.

2 Literature

In this section, we provide the main findings of previous research that has been conducted on option pricing models. In addition, we explain crucial concepts for improved understanding of our research process and how we add to the existing literature.

Option pricing models are a common subject in financial literature and applications such as hedging, investing and trading. The BS model introduced by Black and Scholes (1973) stands out as the most notable and pioneering option pricing model. It exhibits several desirable features such as its simplicity and closed-form solution. Although, this is accompanied with the limitation of assuming constant volatility, which implies that options sharing the same underlying asset, despite differences in strike price and maturity, exhibit equivalent IVs at any given point in time. The IV is acquired by solving the option pricing model for the volatility parameter. Institutional investors effectively manage their option positions based on the IVs of the options instead of the prices as it is more comparable across the option panel (Carr and Wu, 2016). Hence, the IVS, which represents the IVs across different time-to-maturities and strike prices (moneyness), holds significant value for institutional investors.

Subsequent research explored possible extensions and improvements to the BS model. For instance, Merton (1973) derived a modified BS model to account for dividend payments on the underlying equity. Furthermore, a broadened BS framework that includes commodity options, forward contracts and future contracts is introduced by Black (1976). Comparably, Garman and Kohlhagen (1983) expanded the BS model to foreign exchange options. However, Rubinstein (1985) documents the existence of an IV SMILE, which refers to the U-shaped pattern of the IV across the strike price, in equity options after conducting robust non-parametric tests to examine the constant volatility assumption of the BS model. His first finding denotes that the IV tends to be consistently higher for (deep-)out-of-the-money call options with shorter time-to-maturity. The observation of the IV SMILE contradicts the fundamental assumption of the BS model in favor of stochastic volatility models for option pricing (Sheikh, 1991). Moreover, Shastri and Tandon (1986) utilize the framework introduced by Geske and Johnson (1984) to price American-style options on futures. They observe an IV SMILE and term-structure effects in options with S&P 500 futures or Deutsche Mark futures as the underlying asset. In addition, Heynen (1994) identifies systematic SMILE effects with a U-shaped term structure for the IV. He evaluates the IV predictions of various stochastic volatility models and finds that the predictions are inconsistent with the observed SMILE effects. Hence, his suggestion for an alternative explanation for the volatility SMILE based on market inef-

ficiencies. Thus, notable deviations persist between the IV predictions of the BS model and the actual IVs observed in the market. This has attracted significant interest in the academic world to improve the theoretical framework of Black and Scholes (1973). As a consequence, Hull and White (1987) and Heston (1993) utilize stochastic volatility as the basis for their approach, while Bates (1991) and Kou (2002) derive theoretical prices assuming jump-diffusion processes. An alternative approach models the near-term dynamics of the BS IV and defines no-arbitrage constraints on the IVS in order to derive a quadratic equation for the IV (Carr and Wu, 2016). Furthermore, Carr and Wu (2020) introduce a novel approach where the current fair value of an option's IV is linked with the current conditional moments of log changes in the underlying price. Other studies combine several existing frameworks such as Bates (1996b), who includes jumps in the framework introduced by Heston (1993), and co-jump models presented by Andersen et al. (2015), Carr and Wu (2017) and Bates (2019). These represent a limited portion of the extensive literature on option pricing models (Smith Jr, 1976; Bates, 1996a; Bakshi et al., 1997; Bates, 2003; Mitra, 2011; Orlando and Taglialatela, 2017; Bates, 2022).

All these parametric option pricing models require a specific set of assumptions such as the distribution of the price process, the interest rate process and the market price of factor risks (Bakshi et al., 1997). Therefore, alternative studies transition from the common parametric models towards non-parametric models which effectively capture non-linearity and exhibit strong prediction performance. The earliest contributions of utilizing non-parametric models for option pricing are from Malliaris and Salchenberger (1993), Hutchinson et al. (1994) and Boek et al. (1995). They demonstrate that NNs are more accurate and efficient in option pricing compared to the BS model. Likewise to Black and Scholes (1973), this initiated various option pricing studies which incorporate non-parametric models (Ghysels et al., 1997). The studies conducted by Anders et al. (1998), Garcia and Gençay (2000), Gençay and Qi (2001) and Gençay and Salih (2003) demonstrate accurate option pricing using NNs. Moreover, Mitra (2011) states that NNs offer the potential to surpass theoretical option pricing methodologies due to their ability to learn from features that are difficult to integrate into parametric approaches. In recent years, model calibration and portfolio hedging with NNs is introduced (Buehler et al., 2019; Becker et al., 2020; Cuchiero et al., 2020; Horvath et al., 2021). Ruf and

Wang (2020) and Kumar (2023) present a thorough and comprehensive overview of the extensive literature utilizing NNs in option pricing, hedging and risk management. In addition, Ivaşcu (2021) provides a similar overview and expands upon it by incorporating other notable machine learning models such as RF and XGBOOST.

Our focus lies with the research conducted by Almeida et al. (2022), who propose an innovative two-step IV prediction procedure utilizing a parametric and non-parametric model. They demonstrate that an FNN can correct the IV prediction of various parametric models. In particular, the initial prediction of the IV is done by a parametric model. Afterwards, the residuals of the initial IV prediction are predicted with an FNN by taking time-to-maturity, moneyness and, if applicable, time-varying features as an input. The last step involves correcting the IV prediction with the residual prediction. Their study serves as a robust foundation for demonstrating the feasibility of their methodology, as it incorporates various parametric models with unique fundamental frameworks. Specifically, the BS model and, its more pragmatic successor, the AHBS model are included (Black and Scholes, 1973; Dumas et al., 1998). The other two models move away from BS and are defined as stochastic volatility models. Namely, the stochastic volatility model introduced by Heston (1993) and the CW model presented by Carr and Wu (2016). Almeida et al. (2022) examine the performance of their approach on a day-to-day and h-days ahead basis. In addition, they conduct a further analysis of the correction on the Heston model based on the methodology of Andersen et al. (2015). However, limitations in the study conducted by Almeida et al. (2022) are present, which is common in research. They state that the performance of their models can be considered as a benchmark due to the choice of NN architectures, activation function and optimization algorithm. In this paper, we aim to address some of these limitations and explore possible extensions in order to build upon their robust foundation. They investigate a set of five distinct NN architectures, as originally proposed by Gu et al. (2020). These architectures vary in the number of hidden layers, which range from one to five hidden layers, and uniformly employ the sigmoid activation function. Several research papers discuss the limitations of the sigmoid activation function in NNs (Glorot and Bengio, 2010; Glorot et al., 2011; Krizhevsky et al., 2012; Sussillo and Abbott, 2014; Sharma et al., 2017; Nwankpa et al., 2018). The main drawback is the vanishing gradient problem of the sigmoid activation

function. This is the phenomenon where the gradients exponentially decrease towards zero or increase during the back-propagation procedure as they move away from the input layer. As a consequence, the NN experiences slow learning and potential convergence problems. In contrast, the Rectified Linear Unit (ReLU) activation function, which is introduced by Nair and Hinton (2010), has become a popular choice in NNs due to its advantages in overcoming the vanishing gradient problem. Furthermore, Almeida et al. (2022) omit regularization in their NN models as they only include a small amount of features. We investigate individual equity options and incorporate a total of 108 features regarding firm characteristics, option characteristics and macroeconomic data. Therefore, regularization is an important tool to prevent overfitting in our case. Popular regularization methods such as L1 and L2 regularization, dropout, max-norm and early stopping can be included to improve our NNs. In addition, despite the significant performance of NNs in various fields including option pricing, it is crucial to conduct a comparative study with alternative non-parametric models. This establishes a more robust and comprehensive foundation for future research. Therefore, we consider RF and XGBOOST as potential substitutes for the NN.

3 Data

In this section, the data of our research is introduced. We discuss the data selection, necessary transformations and present descriptive statistics to provide insights into our dataset.

We acquire end-of-month data for individual American-style equity options in the period of January, 2000 up to and including December, 2021 from OM. Each observation consists of the option's OM IV measure, strike price, expiration date, best bid and ask price, volume and delta. Afterwards, the raw data is filtered by excluding observations with null IV, bid or ask, zero volume or bid, a mid-point of bid and ask below $\frac{1}{8}$ and bid greater than ask following the approach of Freire and Kleen (2023). Furthermore, the equity prices are obtained from the CRSP. In addition, we include the firm characteristics data, which originates from Gu et al. (2020), provided on the website of Dacheng Xiu[1]. As

[1]The website of Dacheng Xiu for the relevant data is https://dachxiu.chicagobooth.edu/

per convention, we substitute the missing values in the firm characteristics with the median of the cross-section in that particular month (Kelly et al., 2019; Freyberger et al., 2020; Gu et al., 2020; Freire and Kleen, 2023). Moreover, we apply a monthly standardization to the firm characteristics by adjusting the cross-sectional ranks to fall within the interval of [-1, 1]. The macroeconomic data is found on the website of Amit Goyal (Welch and Goyal, 2008)[2]. These individual datasets are merged with the linking table provided by WRDS. Utilizing the equity price, we remove observations violating the weak no-arbitrage bounds equal to

$$\text{Call: } \max(0, S_t^j - K_{it}^j) \leq C_{it}^j \leq S_t^j, \tag{1}$$

$$\text{Put: } \max(0, K_{it}^j - S_t^j) \leq P_{it}^j \leq K_{it}^j, \tag{2}$$

where S_t^j is the price of underlying asset j on day t, K_{it}^j the strike price of option i and C_{it}^j (P_{it}^j) the price of the call (put) option calculated as the mid-point of the bid and ask. To conclude, we examine options across all maturities and moneyness range, where moneyness is defined as the ratio between the equity and strike price, spanning from 0.5 to 2.0 in our study. This results in the final sample of 13,203,378 options across 8,896 distinct firms. Each firm has 20 observations on average per day which is notably less than the thousands of daily S&P 500 index options. We do not differentiate between calls and puts as the focus lies on the IVs.

Table 1 displays the descriptive statistics of our dataset. Specifically, the dataset is split into several bins according to the moneyness, m, and time-to-maturity, τ, of the equity options. Each bin includes, from top to bottom, the time-series average of the number of observations, average IV in % and its standard deviation in parentheses, the 10% and 90% percentile of the IV in brackets and the number of firms in curly braces. We observe the well-known SMILE phenomenon across the various time-to-maturity rows. In the context of IVs, SMILE refers to the U-shaped curve when plotting the IV against the moneyness of options. In alternative terms, the IV for further In-The-Money (ITM) and Out-of-The-Money (OTM) options is higher compared to At-The-Money (ATM) options. However, the SMILE is less steep as the time-to-maturity increases. We observe a similar

[2]The website of Amit Goyal for the relevant data is https://sites.google.com/view/agoyal145

SMILE for the moneyness columns. Moreover, the number of observations and firms decreases when the time-to-maturity increases with an exception for the final row.

Table 1

The descriptive statistics of the equity options.

	$0.50 \leq m < 0.90$	$0.90 \leq m < 0.97$	$0.97 \leq m < 1.03$	$1.03 \leq m < 1.10$	$1.10 \leq m \leq 2.00$
$\tau < 0.25$	4569 62.8% (31,9%) [32.4%, 101.7%] {950}	5957 43.1% (20.5%) [23.0%, 68.3%] {1141}	8652 39.6% (18.7%) [21.3%, 62.5%] {1137}	5749 44.4% (19.7%) [25.4%, 68.5%] {1021}	5968 61.6% (29.0%) [34.2%, 96.3%] {1110}
$0.25 \leq \tau < 0.50$	2514 49.7% (23.1%) [26.3%, 78.4%] {971}	1467 38.7% (17.9%) [21.1%, 61.2%] {799}	1395 38.5% (17.4%) [21.5%, 60.2%] {703}	1191 40.6% (17.5%) [23.6%, 62.4%] {623}	2597 50.7% (20.9%) [29.9%, 76.4%] {899}
$0.50 \leq \tau < 0.75$	1400 46.9% (21.6%) [24.9%, 74.5] {592}	701 37.9% (17.4%) [20.9%, 60.1%] {425}	652 38.0% (17.1%) [21.4%, 59.4%] {373}	543 39.9% (17.1%) [23.2%, 61.6%] {324}	1401 48.5% (19.9%) [28.7%, 73.4%] {540}
$0.75 \leq \tau < 1.00$	566 33.6% (13.7%) [21.3%, 48.1%] {184}	196 29.4% (11.0%) [19.8%, 41.2%] {111}	184 29.4% (10.3%) [20.5%, 40.4%] {98}	174 31.1% (10.4%) [22.1%, 42.4%] {95}	604 37.3% (11.9%) [26.2%, 50.3%] {169}
$1.00 \leq \tau$	1320 40.5% (17.0%) [23.4%, 62.7%] {413}	405 36.9% (15.1%) [21.9%, 56.3%] {223}	398 36.7% (14.7%) [22.3%, 55.1%] {205}	362 37.8% (14.9%) [23.3%, 56.5%] {193}	1392 43.2% (16.0%) [27.2%, 63.4%] {401}

Note: This table presents the descriptive statistics of the equity options sorted into bins categorized by time-to-maturity (τ) and moneyness (m). The values represent (1) the number of observations, (2) the average Implied Volatility in % and its standard deviation in parentheses, (3) the 10% and 90% quantiles of the Implied Volatility in square brackets and (4) the number of firms in curly braces from the top to the bottom of each cell. We calculate these values as the time-series averages of the daily values from the period of January, 2000 up to and including December, 2021.

In Figure 1, the left panel illustrates the number of observations, while the right panel shows the average number of observations over the firms for the time period January, 2000 up to and including December, 2021. The plotted values are derived from the entire dataset and a subset based on a liquidity criterion. The liquid subset consists of firms with a minimum of 20 observations for a given day. Our models require a certain number of observations in order to estimate their parameters and, hence, the relevancy of the liquid subset. We observe a significant increase over time in both panels. This aligns with the general growth of the equity market over the years. The sharp decrease in some years can be explained by global events such as the dot-com bubble in the year 2000, the financial crisis in 2009 and the recent Covid-19 pandemic in 2020. In addition, the difference between the entire data set and the liquid subset plot in the left panel is

insignificant, which suggests that each non-liquid firm consists of a few observations. The right panel supports this suggestion as the average number of observations of the liquid sample ranges from two to three times that of the entire dataset. Moreover, an increasing difference between the two datasets is present in the right panel. The liquidity of the liquid subset increases relatively more than that of the non-liquid subset.

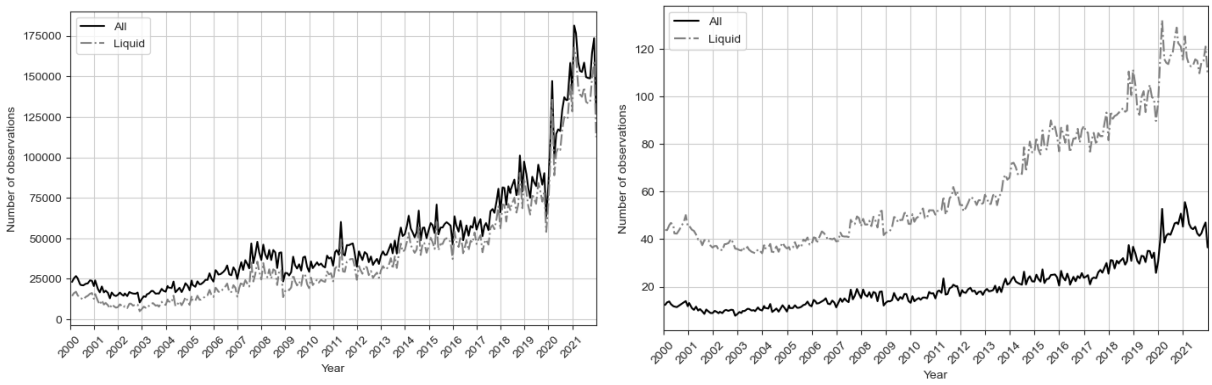

Figure 1

These figures illustrate the number of observations (left) and average number of observations (right) for all firms and liquid firms, which requires a minimum of 20 options, for each day. The sample includes the period of January, 2000 up to and including December, 2021.

Figure 2 displays the number of firms over the time period January, 2000 up to and including December, 2021. The number of firms increases over time and the difference between the datasets remains constant. Moreover, we observe that the entire dataset includes around three to four times the number of firms of the liquid subset. In addition, Figure 3 showcases the frequency of the daily average number of observations per firm. It is apparent that most firms consist of one to ten daily observations on average. Hence, the substantial difference in the number of firms between the liquid and non-liquid dataset, as seen in Figure 2, while the difference in the number of observations between these two datasets is insignificant, as illustrated in Figure 1.

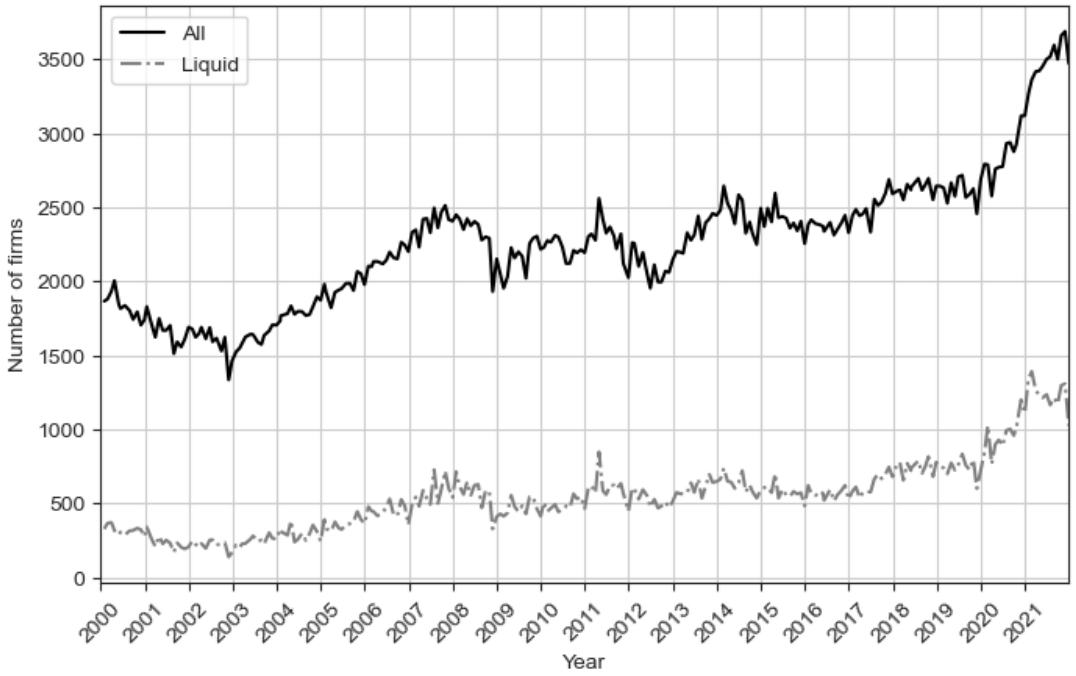

Figure 2

This figure depicts the number of firms and liquid firms, which requires a minimum of 20 options, for each day. The sample includes the period of January, 2000 up to and including December, 2021.

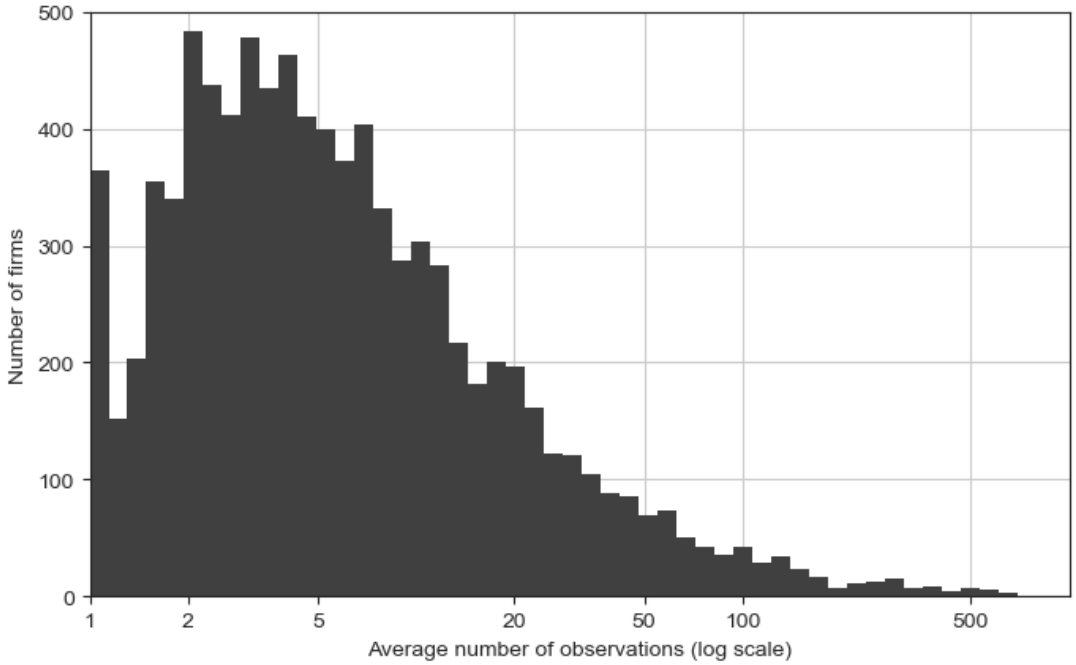

Figure 3

This figure presents the frequency of the average number of observations per firm, which is calculated as the time-series average of the daily values from the period of January, 2000 up to and including December, 2021.

4 Methodology

This section provides thorough explanations of the methods used in our research. We start by discussing the parametric and non-parametric models employed for the IV prediction. Afterwards, the empirical study is explained and the model performance measures are described. We conclude this section with the description of the method to determine the importance of the features in our models.

4.1 Parametric models

We intend to remain similar to Almeida et al. (2022) and, therefore, work with the parametric models covered in their research. Specifically, the BS, AHBS and CW model are utilized. The Heston model is computationally too expensive for our individual equity option data versus their index option data. Hence, we omit this model without losing substantial importance as the focus lies on the accurate correction accomplished by the non-parametric models, which can be employed on a diverse range of parametric models. All in all, the following sections discuss the BS, AHBS and CW model. We adopt the notation presented in Almeida et al. (2022) whenever it is feasible. In order to accommodate the cross-section, we follow Freire and Kleen (2023) and denote firm $i = 1, \ldots, N$, time period $t = 1, \ldots, T$ and option $j = 1, \ldots, J_i$. N and J_i vary over time as we work with an unbalanced panel. The subscript t, representing the time period, is omitted to maintain brevity.

4.1.1 Ad-Hoc Black-Scholes

Dumas et al. (1998) aim to address certain limitations they identify in the BS model and stochastic volatility models, such as the Heston model. Specifically, the BS model assumes constant volatility and stochastic volatility models commonly rely on the market price of the risk parameter, which is challenging to estimate. Therefore, they propose to model the IV as a quadratic function of the moneyness, m, and time-to-maturity, τ. We adjust the function for panel data with a cross-section similar to Bernales and Guidolin

(2014) and Freire and Kleen (2023). Hence, the AHBS model is denoted as

$$\sigma_{ij} = \alpha_{0,i} + \alpha_{1,i} m_{ij} + \alpha_{2,i} m_{ij}^2 + \alpha_{3,i} \tau_{ij} + \alpha_{4,i} \tau_{ij}^2 + \alpha_{5,i} m_{ij} \tau_{ij} + \varepsilon_{ij}, \tag{3}$$

where σ_{ij} is the IV of firm i and option j, m_{ij} the moneyness, τ_{ij} the time-to-maturity, $\alpha_{0,i}$ a constant, $\alpha_{k,i}$ with $k = 1, \ldots, 5$ the coefficient of the relevant feature and ε_{ij} the residual. To maintain conciseness with Almeida et al. (2022), the other models mentioned by Dumas et al. (1998) are omitted.

To estimate the parameters of the AHBS model in Equation (3), Bernales and Guidolin (2014) follow the suggestion of Hentschel (2003) and utilize Generalized Least Squares (GLS) due to the potential presence of heteroskedasticity and autocorrelation in the Ordinary Least Squares (OLS) residuals. However, they observe comparable results between the GLS and OLS method. In addition, instead of separate individual regression, Freire and Kleen (2023) utilize a pooled OLS regression. In particular, their approach involves modeling the deviation from the average firm IV as an alternative to direct IV prediction. This methodology considers the various levels of average IVs and aims for a positive bias-variance trade-off. We employ both OLS and pooled OLS in order to maintain a comprehensive analysis. The parameter estimation of Equation (3) is achieved by minimizing the MSE (Heij et al., 2004). This results in the estimated parameters $\hat{\boldsymbol{\alpha}}_i' = (\hat{\alpha}_{0,i}, \hat{\alpha}_{1,i}, \hat{\alpha}_{2,i}, \hat{\alpha}_{3,i}, \hat{\alpha}_{4,i}, \hat{\alpha}_{5,i})$. Common practice dictates that $Z + 1$ observations are necessary to estimate a regression model with Z parameters. To conclude, the AHBS IV prediction equals

$$\hat{\sigma}_{ij}^{\text{AHBS}} = \hat{\alpha}_{0,i} + \hat{\alpha}_{1,i} m_{ij} + \hat{\alpha}_{2,i} m_{ij}^2 + \hat{\alpha}_{3,i} \tau_{ij} + \hat{\alpha}_{4,i} \tau_{ij}^2 + \hat{\alpha}_{5,i} m_{ij} \tau_{ij}. \tag{4}$$

The BS model predictions also follow from Equation (3) by setting the parameters $\alpha_{1,i}$, $\alpha_{2,i}$, $\alpha_{3,i}$, $\alpha_{4,i}$ and $\alpha_{5,i}$ to zero. Hence, the prediction follows from

$$\hat{\sigma}_{ij}^{\text{BS}} = \hat{\alpha}_{0,i} \tag{5}$$

4.1.2 Carr-Wu

Another option pricing model, which aligns with the methods of institutional investors for managing options positions, is derived by Carr and Wu (2016). The framework begins with the near-term dynamics of the IVS and derives no-arbitrage constraints based on the shape of the IVS. In contrast, standard option pricing models rely on specifying the complete instantaneous variance rate dynamics. For instance, the flat and constant dynamics under the BS model and the stochastic volatility equation under the Heston model (Black and Scholes, 1973; Heston, 1993). However, the instantaneous variance rate is unobserved (Carr and Wu, 2016). Instead, we observe the IVS consisting of numerous options that span different strike prices and maturity dates. The initialization with the observed IVS and minimal specification of the current levels of the drift and diffusion processes reduces the computational complexity compared to starting with a single instantaneous variance rate and complete specification of the dynamics. Furthermore, the connection between the IVS and the instantaneous variance rate dynamics is unclear at times, which leads to institutions frequently calibrating their models in order to account for changing market conditions. Carr and Wu (2016) state that these parameters are theoretically fixed over time. Hence, they present a quadratic equation to solve the shape of the complete IVS. We follow the notation of Carr and Wu (2016) and denote the equity price under the risk-neutral dynamics as

$$\frac{dS_t}{S_t} = \sqrt{v_t} dW_t, \tag{6}$$

where S_t is the equity price at time t, v_t the instantaneous variance rate following a positive, real-valued stochastic process and W_t a standard Brownian motion under the $\mathbb{Q}$-measure. However, the risk-neutral dynamics of the instantaneous variance rate is not specified by Carr and Wu (2016). Alternatively, the framework defines the risk-neutral dynamics of the IV based on the BS model as

$$d\sigma_t(K,T) = \mu_t dt + \omega_t dZ_t, \tag{7}$$

where $\sigma_t(K,T)$ is the IV of the option with strike price K and maturity T, μ_t the drift of the IV process, ω_t the volatility of the IV process and Z_t a standard Brownian motion.

Furthermore, the standard Brownian motions W_t and Z_t have a correlation equal to $\rho_t \in [\text{-}1, 1]$ or in mathematical terms $E_t[dW_t dZ_t] = \rho_t dt$. When the drift, μ_t, and diffusion, ω_t, are established as proportional to the IV level, Carr and Wu (2016) derive

$$\frac{d\sigma_t(K,T)}{\sigma_t(K,T)} = e^{\text{-}\eta_t(T-t)}(m_t dt + w_t dZ_t), \tag{8}$$

where m_t, w_t and η_t are stochastic processes that are not conditional on K, T or $\sigma_t(K,T)$. The process w_t is bound to remain strictly positive and the inclusion of the exponential dampening term $e^{-\eta_t \tau}$ with $\tau = T - t$ accounts for the fact that IVs with long maturities tend to exhibit less movement in empirical observation (Carr and Wu, 2016). Subsequently, they denote $k = \log(\frac{K}{S_t})$ and derive a quadratic equation for the IVS as a function of k and τ under the no-arbitrage assumptions in combination with the dynamics in Equations (6)-(7) equal to

$$\begin{aligned} \frac{1}{4}e^{\text{-}2\eta_t\tau} w_t^2 \tau^2 \sigma_t^4 + (1 - 2e^{\text{-}\eta_t\tau} m_t \tau - e^{\text{-}\eta_t\tau} w_t \rho_t \sqrt{v_t}\tau)\sigma_t^2 \\ - (v_t + 2e^{\text{-}\eta_t\tau} w_t \rho_t \sqrt{v_t} k + e^{\text{-}2\eta_t\tau} w_t^2 k^2) = 0. \end{aligned} \tag{9}$$

Equation (9) indicates that the no-arbitrage constraint is dependent on the current levels of the stochastic processes $(m_t, w_t, \eta_t, v_t, \rho_t)$ rather than their exact dynamics. As a result, the IV prediction process requires extracting the current levels of the five dynamic states without estimating the underlying dynamics. Therefore, it is possible to model the IVS on day t by considering the values of the dynamic states as parameters. Particularly, we attempt to find the optimal set of $\boldsymbol{\theta}_t' = (m_t, w_t, \eta_t, v_t, \rho_t)$ for which the left-hand side of Equation (9) is approximately zero. Hence, $\hat{\boldsymbol{\theta}}_t$ is estimated with

$$\begin{aligned} \hat{\boldsymbol{\theta}}_t = \arg\min_{\boldsymbol{\theta}_t} \sum_{i=1}^{n} \Big(\frac{1}{4}e^{\text{-}2\eta_t\tau_{i,t}} w_t^2 \tau_{i,t}^2 \sigma_{i,t}^4 + (1 - 2e^{\text{-}\eta_t\tau_{i,t}} m_t \tau_{i,t} - e^{\text{-}\eta_t\tau_{i,t}} w_t \rho_t \sqrt{v_t}\tau_{i,t})\sigma_{i,t}^2 \\ - (v_t + 2e^{\text{-}\eta_t\tau_{i,t}} w_t \rho_t \sqrt{v_t} k_{i,t} + e^{\text{-}2\eta_t\tau_{i,t}} w_t^2 k_{i,t}^2) \Big)^2, \end{aligned} \tag{10}$$

where $\sigma_{i,t}$ is the observed IV of option i on day t. The optimization is done iteratively using Non-Linear Least Squares (NLS) with initialization $\boldsymbol{\theta}_0$. The final prediction of the IV is obtained by solving Equation (9) given the optimal set $\hat{\boldsymbol{\theta}}_t$, k and τ of a particular option. This process is iterated for each firm.

4.2 Non-parametric models

The goal of this paper is to use the methodology introduced by Almeida et al. (2022) which employs a two-step prediction procedure with a parametric and non-parametric model in order to predict the IVS. The IVS is defined as the mapping of the IVs, σ, with varying moneyness, m, and time-to-maturity, τ. Despite extensive research into IVS modeling, the underlying assumptions deviate from the real world and lead to misspecification. Hence, Almeida et al. (2022) introduce the Pricing Error Surface (PES) which is the difference between the observed and predicted IVS. This is denoted as

$$\hat{\varepsilon}^P(m,\tau) = \sigma(m,\tau) - \hat{\sigma}^P(m,\tau), \tag{11}$$

where $\hat{\varepsilon}^P(m,\tau)$ is the PES of parametric model P and $\sigma(m,\tau)$ ($\hat{\sigma}^P(m,\tau)$) the observed (predicted) IVS.

Almeida et al. (2022) propose the idea of decreasing the prediction errors in a two-step prediction procedure. First, a parametric model is employed to predict the IVS. Second, they leverage a non-parametric model to predict the PES and correct the predicted IVS. In our study, a parametric model is calibrated on the observed IVs $\sigma(m_{ij},\tau_{ij})$ and used to predict $\hat{\sigma}^P(m_{ij},\tau_{ij})$ per firm. In the second step, we calculate the residuals $\hat{\varepsilon}^P(m_{ij},\tau_{ij}) = \sigma(m_{ij},\tau_{ij}) - \hat{\sigma}^P(m_{ij},\tau_{ij})$ and deploy them as the dependent variable for the non-parametric model. In contrast to the parametric model, the non-parametric models are not calibrated per firm as the firm characteristics differentiate the observations. Hence, the non-parametric models are employed for all the observations in the relevant time period. The parameters of the non-parametric models are estimated by minimizing the objective function

$$\frac{1}{J}\sum_{i=1}^{N}\left[\sum_{j=1}^{J_i}\left(\hat{\varepsilon}^P(m_{ij},\tau_{ij}) - f^{NP}(\mathbf{x}_{ij}|\,\boldsymbol{\theta}^{NP})\right)^2\right], \tag{12}$$

where J is the summation of $J_1,\ldots,J_N$, $f^{NP}(\mathbf{x}_{ij}|\,\boldsymbol{\theta}^{NP})$ the PES prediction of non-parametric model NP with a vector of features $\mathbf{x}_{ij}$ and parameter set $\boldsymbol{\theta}^{NP}$. Afterwards, the IV prediction is calculated as the combination of the predictions produced by both

the parametric and non-parametric models denoted as

$$\hat{\sigma}(m_{ij}, \tau_{ij}) = \hat{\sigma}^P(m_{ij}, \tau_{ij}) + \hat{f}^{NP}(\mathbf{x}_{ij} | \, \boldsymbol{\theta}^{NP}). \tag{13}$$

The two-step prediction procedure for the IV can be generalized to a strict non-parametric prediction by slightly altering Equation (12) (Almeida et al., 2022). In particular, the IV prediction of the parametric model is included in the equation and the target variable of the non-parametric model changes from the residual to the observed IV. This results in the following minimization problem

$$\frac{1}{J} \sum_{i=1}^{N} \left[\sum_{j=1}^{J_i} \left(\sigma(m_{ij}, \tau_{ij}) - c_{ij} - f^{NP}(\mathbf{x}_{ij} | \, \boldsymbol{\theta}^{NP}) \right)^2 \right], \tag{14}$$

where c_{ij} is the IV prediction of the parametric model denoted as $\hat{\sigma}^P(m_{ij}, \tau_{ij})$.

We have established the two-step prediction procedure. The subsequent sections focus on the specifics of the machine learning models employed in the second step. We consider well-known adaptable machine learning models with a minimal demand for hyperparameter tuning considering the frequent training of the models. In addition, the machine learning models require the ability to effectively capture complex relationships between the features and the target variable. Therefore, we choose the Random Forest, Extreme Gradient boosting and Neural Network model as candidates for the non-parametric model.

4.2.1 Random Forest

Breiman et al. (1984) introduce the Classification And Regression Tree (CART) algorithm which consists of building a decision tree from the relevant features. Specifically, the data is recursively partitioned into subsets based on thresholds of the independent variables such that the observations in each region are as similar as possible. In our case, the decision tree is configured as a regression to accommodate the continuous dependent variable. Therefore, the subsets A and B in each split follow by minimizing

$$\sum_{i:\, \sigma_i \in A} (\sigma_i - \tilde{\sigma}_A)^2 + \sum_{i:\, \sigma_i \in B} (\sigma_i - \tilde{\sigma}_B)^2, \tag{15}$$

where $\tilde{\sigma}_A$ ($\tilde{\sigma}_B$) is the sample mean of the observations in subset A (B) and σ_i the observed IV of observation i. This results in a threshold for the relevant feature in each split and a tree emerges as seen in Figure 4. Unknown observations are assigned to a final node based on the splits in the tree and the dependent variable is predicted as the mean of the relevant final node.

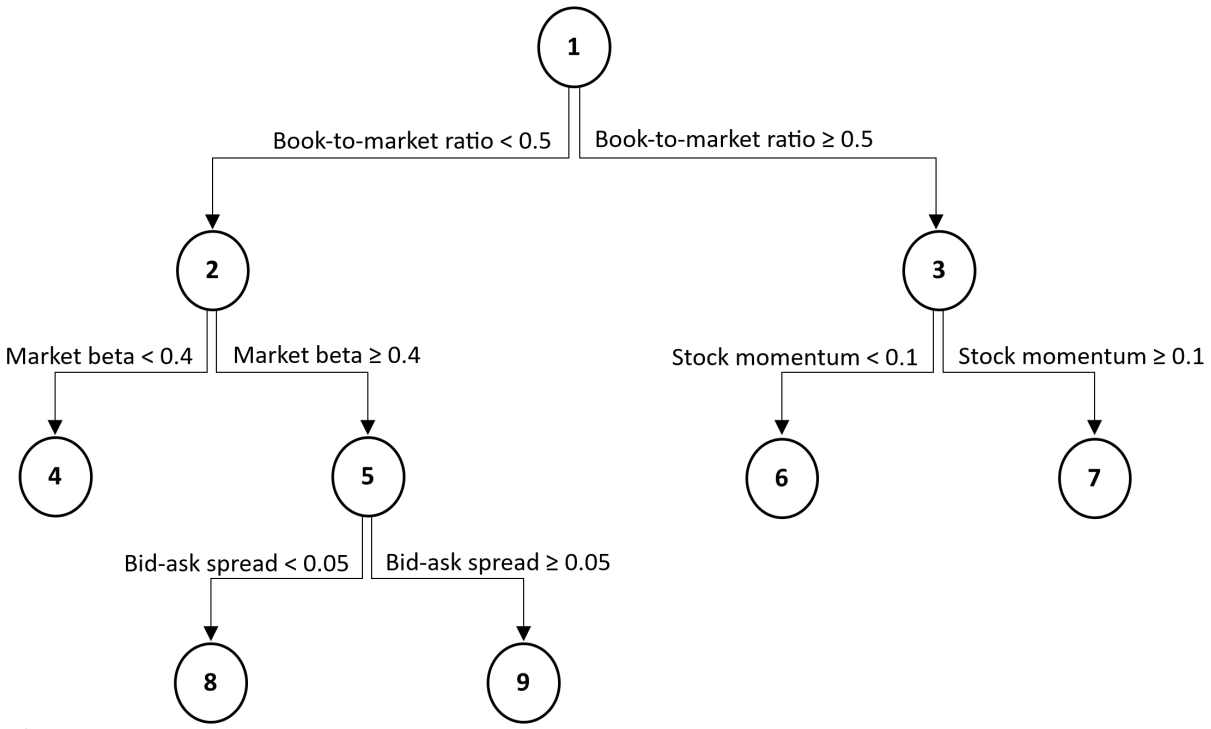

Figure 4

An example of a decision tree. The observations in nodes 1-3 and 5 are partitioned into two nodes in the subsequent layer. Unknown observations are assigned a value equivalent to the mean at the final nodes 4 and 6-9.

The resulting decision tree is considered simple and interpretable. Although, it is prone to overfit and exhibit high variance due to its sensitivity to outliers. CART can be improved by ensemble methods, such as bagging and boosting methods introduced by Breiman (1996) and Schapire (1990), respectively. These ensemble methods are employed in practice as an alternative to improve overall performance due to a net positive in the bias-variance trade-off (Dietterich, 2000; Sutton, 2005; Bramer, 2007; Ganaie et al., 2022). RF represents the bagging method, whereas XGBOOST, which will be elaborated upon in the following section, serves as an example of the boosting method. Figure 5 shows the general procedure for the bagging and boosting method. We observe that bagging includes

multiple independent models, while the models in the boosting method are dependent on each other.

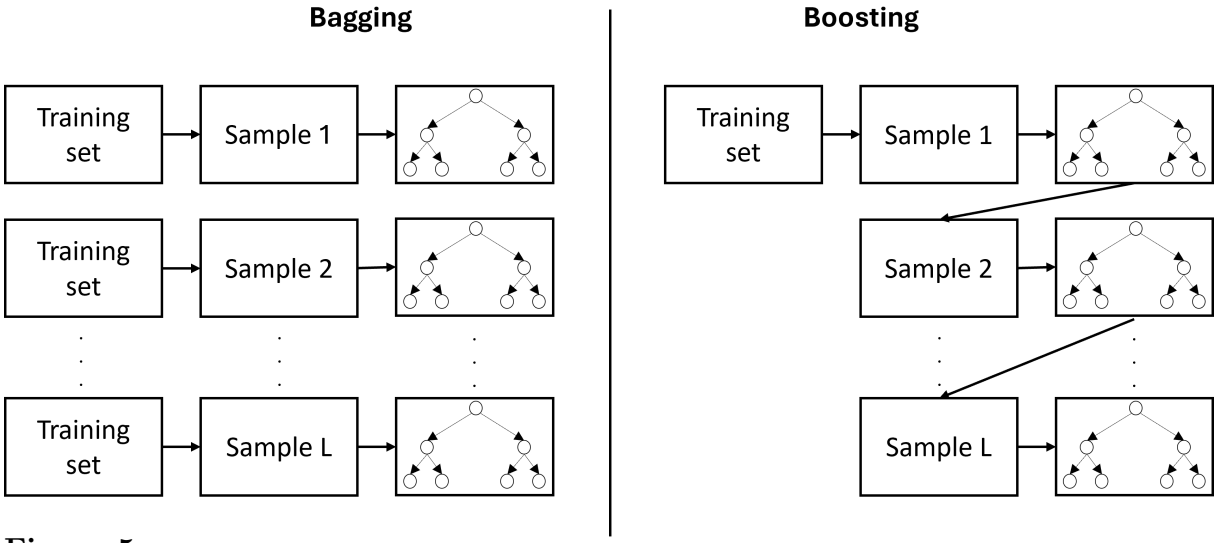

Figure 5

Ensemble methods; Bagging includes a large number of independent decision trees. Boosting consists of sequential optimization, where each new tree is improved with information from the previous tree.

We consider RF introduced by Breiman (2001) as our first non-parametric model due to its simpleness and broad applicability. This ensemble method leverages a substantial number of independent decision trees (Breiman, 1996). Each decision tree is grown with the complete dataset in the relevant period. To partition the observations, a random selection of $\sqrt{p}$ out of the total p independent variables is incorporated in each decision tree. This significantly reduces the correlation among the decision trees. The final prediction is equal to the average of all decision tree predictions. It is important to note that their are several crucial hyperparameters. Specifically, three key hyperparameters significantly influence the performance of the RF model. Namely, the number of trees which determines the size of the model and is positively correlated with the generalization error of the model. The remaining two are the maximum depth allowed for each tree and the minimum number of samples required for a node split. These affect the pruning process of the individual trees and, therefore, the proneness to overfitting. We accomplish the implementation of the RF model with the `Scikit-learn` library for Python (Pedregosa et al., 2011)[3].

[3]The documentation of the Scikit-learn Python library is found on https://scikit-

4.2.2 Extreme Gradient Boosting

The second non-parametric model in our study is a relatively newer model known for its performance named XGBOOST (Chen and Guestrin, 2016). This ensemble method utilizes a gradient boosting method where the model is iteratively optimized in order to decrease the bias of an individual decision tree (Bartlett et al., 1998). The sequential process distinguishes XGBOOST from RF. Instead of growing new independent trees through bootstrapping, XGBOOST adjusts the structure of the new tree based on the performance of the former tree. Hence, relatively simple and shallow trees, known as weak learners, are sequentially combined in order to acquire an overall strong learner with a net positive bias-variance trade-off. The gradient descent method is utilized to optimize the loss function and improve the previous tree. Moreover, XGBOOST is regularized and exploits second-order gradients for efficiency.

Following the notation of Chen and Guestrin (2016), XGBOOST utilizes K additive functions to predict the dependent variable given a dataset with n observations and m features. This is denoted as

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \; f_k \in \mathcal{F}, \tag{16}$$

where $\hat{y}_i$ is the prediction for observation i that consists of adding the K additive functions. Furthermore, $\mathcal{F} = \{f(\mathbf{x}) = w_{q(\mathbf{x})}\}$ is the space of the regression trees with structure $q : \mathbb{R}^m \to T$ with a total of T leaves and leaf weights $w \in \mathbb{R}^T$. Each f_k represents an independent tree structure characterized by the parameters q and leaf weights w. Consequently, each leaf has a continuous score w_i as we employ regression trees. Therefore, the regularized objective function, which is to be minimized, is defined as

$$\mathcal{L}(\phi) = \sum_{i} \ell(\hat{y}_i, y_i) + \sum_{k} \Omega(f_k), \tag{17}$$

where $\mathcal{L}(\phi)$ is the total loss, $\ell(\hat{y}_i, y_i)$ the differentiable convex loss function that quantifies the difference between the predicted value $\hat{y}_i$ and real value y_i, and $\Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2$ a regularization term with hyperparameters γ and λ. It serves as a penalty to regulate

learn.org/stable/index.html

the complexity of the model to decrease the risk of overfitting.

Conventional optimization methods are not viable as Equation (17) includes functions as parameters and the sequential dependency of f_k. Instead, the model follows an additive training approach. Hence, let $\hat{y}_i^t$ be the predicted value of observation i in iteration t and add f_t to the objective function as follows

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t), \tag{18}$$

where f_t is selected such that the improvement to the model performance is maximized as specified by Equation (17). This leads to a second-order approximation equal to

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n} \left(l(y_i, \hat{y}^{(t-1)}) + g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right) + \Omega(f_t), \tag{19}$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ are the first and second order derivatives of the loss function, respectively. The corresponding optimal value for a fixed structure $q(\mathbf{x})$ is then derived as

$$\mathcal{L}_q^{(t)} = \frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T. \tag{20}$$

Equation (20) is computationally too extensive and, therefore, a greedy algorithm is utilized (Chen and Guestrin, 2016). The greedy algorithm adds branches to the decision tree in each iteration. The splits are evaluated by calculating the loss reduction denoted as

$$\mathcal{L}_{\text{split}} = \frac{1}{2} \left(\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right) - \gamma, \tag{21}$$

where I_L (I_R) is the left (right) sample after the split. Similar to RF, the XGBOOST model's performance is notably affected by key hyperparameters such as the number of estimators, K, the maximum depth allowed for each tree and the minimum loss reduction, γ, required to make a node split. Furthermore, XGBOOST includes hyperparameters regarding the iterative process. Namely, the learning rate parameter, η, which is equivalent to the shrinkage of the optimal weights in each step in order to maintain a conservative process. In addition, the hyperparameters α and λ represent the L1 and L2 regularization

terms, respectively. The L1 regularization (LASSO) encourages sparse feature selection by driving some feature weights to zero (Tibshirani, 1996). On the other hand, the L2 regularization (Ridge) penalizes large weights and promotes a more balanced use of features for overall model stability (Hoerl and Kennard, 1970). By tuning these hyperparameters, XGBOOST achieves a generalizable model by finding a balance between model complexity and predictive accuracy. The last two important hyperparameters are the subsample ratio and the column subsample ratio. These control the sample and number of features utilized in the model similar to regularization, respectively. The implementation is performed with the `XGBoost` Python library created by Chen and Guestrin (2016)[4].

4.2.3 Neural Network

We step away from the tree-based models and introduce the final non-parametric model, known as the Neural Network. Theoretically, it is considered to be a universal approximator capable of handling complex and non-linear relationships in data (Cybenko, 1989; Hornik et al., 1989; Hornik, 1991; Leshno et al., 1993). Moreover, its flexibility through subsequent layers of non-linear transformations results in a complex model with state-of-the-art performances for various applications, such as asset pricing (Heaton et al., 2017; Gu et al., 2020), option pricing (Hutchinson et al., 1994; Sirignano and Spiliopoulos, 2018; Becker et al., 2019; Liu et al., 2019b,a), computer vision (Hinton et al., 2006; Krizhevsky et al., 2012) and natural language processing (Bordes et al., 2012). Hence, making it a significant component of our research. However, the flexibility is accompanied with a higher degree of parameterization and, thus, significant decrease in interpretability (Gu et al., 2020). A graphical example of a Neural Network can be seen in Figure 6.

[4]The documentation of the XGBOOST Python library is found on https://xgboost.ai/

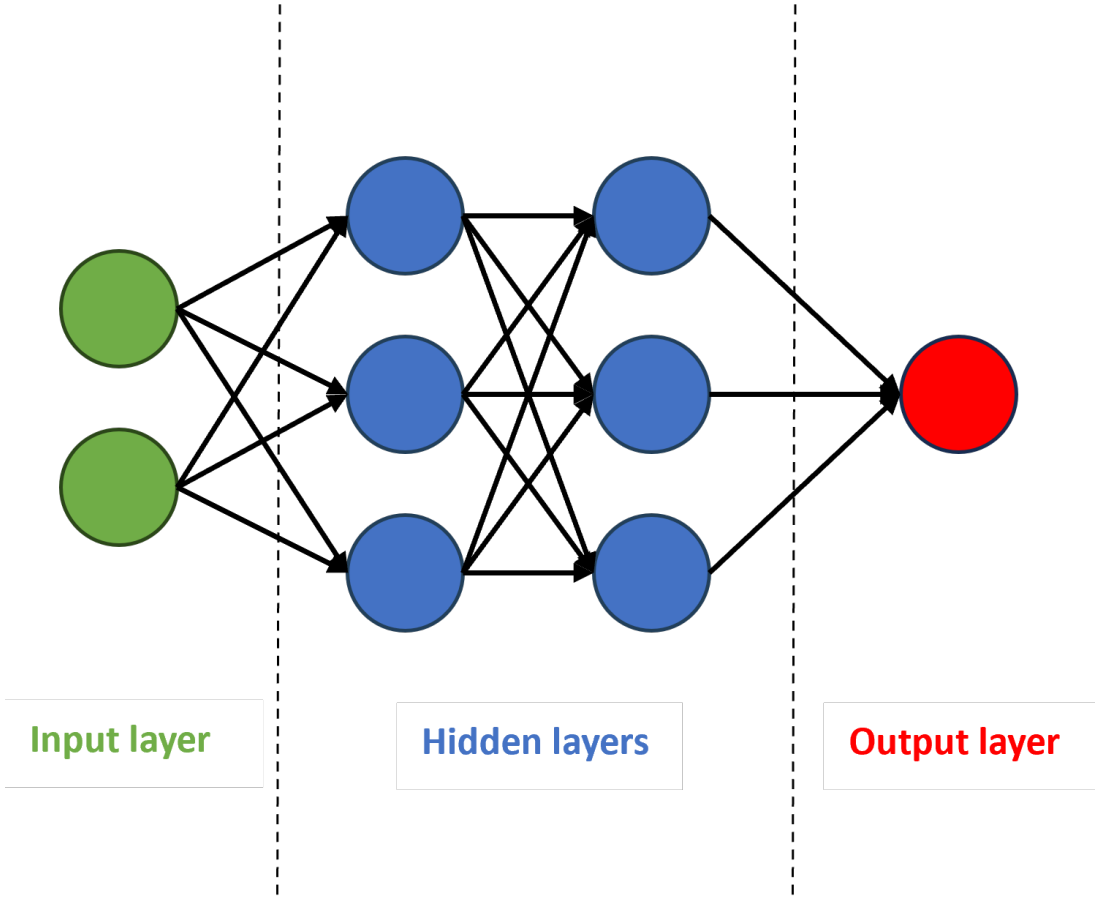

Figure 6

A graphical illustration of a Neural Network with an input layer, two hidden layers and an output layer.

In particular, we investigate the Feedforward Neural Network (FNN) with the back-propagation learning procedure popularized by Rumelhart et al. (1986). We maintain the notation introduced by Almeida et al. (2022) and denote the vector of features as $\mathbf{x}_{i,j}$, which corresponds to option i and firm j. We define the Neural Network model $f : \mathbb{R}^{|\mathbf{x}_{i,j}|} \to \mathbb{R}$, where $|\mathbf{x}_{i,j}|$ is the number of features in $\mathbf{x}_{i,j}$, and denote

$$\underset{d_l \times 1}{\mathbf{z}_l} = \mathring{h}(\underset{d_l \times d_{l-1}}{\mathbf{A}_{l-1}} \underset{d_{l-1} \times 1}{\mathbf{z}_{l-1}} + \underset{d_l \times 1}{\mathbf{b}_{l-1}}), \tag{22}$$

where $\mathbf{z}_l$ refers to the l^{th} layer in the NN with $l = 0, \ldots, L$, $\mathbf{z}_0$ is the input vector and equivalent to $\mathbf{x}_{i,j}$, d_l the amount of neurons, $\mathbf{A}_{l-1}$ the weight matrix, $\mathbf{b}_{l-1}$ the bias and $\mathring{h} : \mathbb{R}^{d_l} \to \mathbb{R}^{d_l}$ an activation function applied to each element of the vector. The final layer

consists of the calculation of the target variable which is equivalent to

$$f(\underset{1\times 1}{\mathbf{x}_{i,j}}) = \underset{1\times d_L}{\mathbf{A}_L}\ \underset{d_L\times 1}{\mathbf{z}_L} + \underset{1\times 1}{\mathbf{b}_L}, \tag{23}$$

where $f(\mathbf{x}_{i,j})$ is the final output. The FNN requires hyperparameter tuning comparable to the other non-parametric models. We start with the architectures of the FNN. Almeida et al. (2022) include the five FNN architectures presented by Gu et al. (2020), who follow the geometric pyramid rule of Masters (1993):

- Neural Network 1: 1 hidden layer with 32 neurons.
- Neural Network 2: 2 hidden layers with 32 and 16 neurons, respectively.
- Neural Network 3: 3 hidden layers with 32, 16 and 8 neurons, respectively.
- Neural Network 4: 4 hidden layers with 32, 16, 8 and 4 neurons, respectively.
- Neural Network 5: 5 hidden layers with 32, 16, 8, 4 and 2 neurons, respectively.

They combine these architectures with the sigmoid activation function defined as $\text{sigmoid}(x) = \mathring{h}(x) = \frac{1}{1+e^{-x}}$. As stated before, the sigmoid activation function suffers from the vanishing gradient problem where the gradients exponentially decrease towards zero or increase during the back-propagation procedure as they move away from the input layer. Therefore, deep NNs using the sigmoid activation function potentially encounter slow learning and convergence problems. To overcome the vanishing gradient problem, we substitute the sigmoid activation function with the popular ReLU activation function denoted as $\text{ReLU}(x) = \mathring{h}(x) = \max(0, x)$ (Nair and Hinton, 2010). Another important tool for the fine-tuning of a NN is regularization in order to prevent overfitting. Conventional methods include L1 and L2 regularization, dropout, max-norm and early stopping. L1 and L2 regularization involves including penalty terms in the loss function as follows

$$\mathcal{L}_{reg}(\mathbf{X}) = \mathcal{L}(\mathbf{X}) + \sum_{l=1}^{L} \alpha_l || \underset{d_l\times d_{l-1}}{\mathbf{A}_{l-1}} ||_{1,1} + \sum_{l=1}^{L} \lambda_l || \underset{d_l\times d_{l-1}}{\mathbf{A}_{l-1}} ||_{2,2}, \tag{24}$$

where $\mathcal{L}_{reg}(\mathbf{X})$ is the regularized loss function, $\mathcal{L}(\mathbf{X})$ the loss function, $|| \underset{d_l\times d_{l-1}}{\mathbf{A}_{l-1}} ||_{N,N}$ the N-norm of weight matrix $\underset{d_l\times d_{l-1}}{\mathbf{A}_{l-1}}$ and α_l (λ_l) the hyperparameter regarding the L1 (L2)

regularization. Similar to the idea of L1 and L2 regularization, max-norm regularization involves setting an upper bound for the 2-norm of the weight vectors. The constraint is denoted as

$$||a_{i,l}||_2 \leq c_l, \tag{25}$$

where $a_{i,l}$ is the i^{th} row of the weight matrix A_l and c_l the upper bound. Max-norm prevents individual weight vectors from becoming significantly large and influencing the learning process. Another regularization method in NNs is dropout. The idea behind dropout is to introduce noise in the NN to achieve a more robust NN. As the name suggests, a certain number of neurons in each layer, determined by the dropout rate p_{drop}, are "dropped out" which implies that the output of these neurons are omitted in the training process. The output of the remaining neurons is scaled with $\frac{1}{p_{drop}}$ to compensate for the fact that in the testing process all neurons are utilized (Hinton et al., 2012; Srivastava et al., 2014). Pairing dropout with L1, L2 and max-norm regularization optimizes its effectiveness (Srivastava et al., 2014). The final regularization method is early stopping. Despite its earlier implementation in various models and application, Morgan and Bourlard (1989) introduced early stopping in NNs. During the training process, a validation set is utilized to evaluate the performance of the NN in each epoch. An epoch refers to a single iteration of the training data through the NN. The weights are updated after each epoch to achieve a better performing NN. If the performance does not improve after a certain number of epochs, known as the patience, the training process stops. Afterwards, the final weights are acquired and the NN is calibrated with the total training data. Finnoff et al. (1993), Prechelt (1998) and Gençay and Qi (2001) state that early stopping significantly reduces overfitting in NNs.

We train the NNs with the Adam optimizer introduced by Kingma and Ba (2014). Furthermore, the ReLu activation function is initialized with the most common initialization method called the 'He Normal Initialization' (He et al., 2015; Shin and Karniadakis, 2020). We implement the NNs in the `TensorFlow` library with the `Keras` API[5]. The complete list of our non-parametric models with their respective hyperparameters can be found in Appendix B. The values of the hyperparameters are chosen based on tests run on a subset of our data and common values in similar research.

[5]The documentation of the TensorFlow library is found on https://www.tensorflow.org/

4.3 Empirical Study

In this section, we present the various scenarios of the empirical study conducted in our research. First, we evaluate the models on a daily basis similar to Almeida et al. (2022). Second, we employ a rolling window spanning three end-of-month time points as an extension of the daily analysis. Finally, we aim to include macro variables leading us to conduct a comprehensive analysis spanning the entire time frame. Freire and Kleen (2023) utilize the same dataset and, hence, we adopt some of their elements for our empirical study. The following sections provide detailed explanations of the scenarios.

4.3.1 Daily Analysis

Our research begins with a daily analysis, where the observations for each firm are randomly divided into two equally sized sets on a daily basis. The first set, assigned as the training set, is employed to calibrate our models, while the second set is appointed as the test set, utilized for model evaluation. We calibrate the parametric models per firm and, therefore, require a minimum of 10 observations per firm in both the training and test set following Freire and Kleen (2023). In contrast, the non-parametric models incorporate firm characteristics as features, which eliminates the need for individual firm calibration. The firm characteristics enable the non-parametric models to group similar observations from different firms providing them access to more information for calibration purposes compared to the individual firm calibration of the parametric models. Afterwards, the training set is partitioned once more into a smaller training set and validation set with 80% and 20% of the observations from the initial training set, respectively. The smaller training and validation set are utilized in a gridsearch to determine the optimal values for the hyperparameters of the non-parametric models. We opt out of cross-validation due to its computational demands. Note that the training, validation and test set are equal across the models to provide a fair comparison. Thereafter, the non-parametric models with the optimal hyperparameters are calibrated using the complete training set. In the end, we calculate the performance measures based on the predictions derived from the test set.

4.3.2 Quarterly Analysis

The second scenario is a quarterly analysis of the models. We calibrate, train and test the models almost identical to the first scenario. The difference lies in the time frame of the analysis. The data consists of a rolling window of three end-of-month time points instead of one. The first two end-of-month time points of the rolling window are included into the training set. Subsequently, the final end-of-month time point is randomly split into two equally sized sets. One set is added to the training set, while the other set is assigned as the test set. Hence, the test set only includes observations of the final end-of-month time point to avoid a look-ahead bias. If the test set would include observations of the first and second end-of-month time points of the rolling window, the models would be trained on data in the future compared to these time points. A quarterly analysis might offer more stable predictions by reducing the effect of possible outliers in the data. Moreover, certain investment strategies occur on a quarterly basis. Hence, an analysis with the same frequency could be more relevant.

4.3.3 Complete Dataset Analysis

In the final scenario, we focus on the predictive performance of the models on a test set outside of the training set time period. The training set consists of the observations from the year 2000 up to and including 2019 with the last two years acting as the validation set. The test set includes the years 2020 and 2021. The training, validation and test set include individual stock market crashes such as the dot-com bubble in the year 2000, the financial crisis in 2009, the cryptocurrency crash in 2018 which also affected the stock market, and the recent Covid-19 pandemic in 2020. Hence, the training and validation are representative of the test set. Gu et al. (2020) observe that some macroeconomic variables can be influential in asset pricing. The increased training and test time frame opens up the possibility to include time-varying macroeconomic variables in the non-parametric models. Moreover, the extended time frame for training allows for an even more robust model compared to the previous scenarios.

4.4 Performance Measures

A popular model performance measure utilized in similar research is the Root Mean Squared Error (RMSE) (Andreou et al., 2010; Liu et al., 2019b; Ruf and Wang, 2020; Ackerer et al., 2020; Ivaşcu, 2021). In addition, the parametric and non-parametric models in our research are calibrated with the RMSE, which is consistent with Almeida et al. (2022). The RMSE is computed as

$$\text{RMSE}_t = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(\sigma_{j,t} - \hat{\sigma}_{j,t})^2}, \tag{26}$$

where n is the number of observations, $\sigma_{j,t}$ the observed IV of option j at time t and $\hat{\sigma}_{j,t}$ the predicted IV. It is important to acknowledge that this approach does not take the difference in IV levels of the various firms in our data into account. Therefore, we also evaluate the models with a scale-invariant version of the RMSE. This is known as the Root Mean Squared Percentage Error (RMSPE) and denoted as

$$\text{RMSPE}_t = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{\sigma_{j,t} - \hat{\sigma}_{j,t}}{\sigma_{j,t}})^2}. \tag{27}$$

Furthermore, Freire and Kleen (2023) utilize a scale-invariant performance measure that is calculated relative to a benchmark model. This is the Outperformance Rate (OR) which is equal to

$$\text{OR}_t^k = \frac{1}{n}\sum_{j=1}^{n}\mathbb{1}(\text{RMSE}_{j,t}^k < \text{RMSE}_{j,t}^b), \tag{28}$$

where $\text{RMSE}_{j,t}^k$ is the RMSE of model k and $\text{RMSE}_{j,t}^b$ the RMSE of the benchmark model. In our case, the benchmark model is the simplest parametric model without a correction, which is the BS model estimated with OLS. We calculate the various performance measures for each day and take the average of the resulting time-series of performance measures.

4.5 Feature Importance

While machine learning models are known for their significant performance, they are considered as black-box models due to their limited interpretability. Hedge funds, investing firms, trading firms and other similar firms manage a substantial volume of financial resources. Hence, the understanding of a model is of great importance. Researchers have introduced methods to increase the interpretability of machine learning models. Almeida et al. (2022) utilize a permutation based approach, similar to the Permutation Feature Importance (PFI) method introduced by Breiman (2001) and further developed by Fisher et al. (2019), to evaluate the importance of a feature in the NN models. Specifically, they set one of the features to zero and observe the change in the RMSE of the model. However, the PFI method has two main drawbacks. First, when two or more features are correlated and one of the feature values is permuted, unrealistic combinations of the feature values could arise and lead to false conclusions in the importance of the feature. Second, correlated features could share their overall feature importance as they are associated with each other. There are various other methods to assess feature importance (Linardatos et al., 2020). Other popular approaches are the Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) method introduced by Ribeiro et al. (2016) and Lundberg and Lee (2017), respectively. The LIME method creates a new dataset of altered samples from the original data set with the underlying predictions. Subsequently, an interpretable model, such as a linear regression, is trained on the altered dataset. Finally, the prediction is explained by interpreting the trained interpretable model. The simplicity of LIME is accompanied with a decrease in precision compared to the SHAP method. Furthermore, the LIME method lacks a theoretical foundation in contrast to the SHAP method. Hence, we utilize the SHAP method to calculate the feature importance.

The SHAP method is based on the Shapley values introduced by Shapley (1953). These values originate from game theory and quantify the contribution of a player to the game. Specifically, the average marginal contribution of a player across all possible coalitions is calculated. In our context, the players are the features incorporated in the non-parametric models and the game is the prediction of the IV. We follow the notation

of Lundberg and Lee (2017) and specify the Shapley value of a feature as

$$\phi_j = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{j\}}(x_{S \cup \{j\}}) - f_S(x_S)], \tag{29}$$

where F is the set of features with $S \subseteq F$ and $f_{S \cup \{j\}}(x_{S \cup \{j\}})$ ($f_S(x_S)$) the machine learning model including (excluding) feature j.

Lundberg and Lee (2017) state that Shapley values have unique solutions and exhibit three desirable properties, namely the local accuracy, missingness and consistency. First, local accuracy requires that the output of the explanatory model is equivalent to the output of the original model. Second, missingness implies that a missing feature in the original input has zero importance in the model. Last, consistency ensures that the Shapley value assigned to a specific feature is greater in model A than in model B if that particular feature holds greater importance in model A compared to model B.

The calculation of the Shapley values for a model with a significant amount of features is extensive. A popular substitution is the SHAP method which combines the game theory of Shapley values and the local interpretability of model-agnostic methods, such as LIME (Lundberg and Lee, 2017). We employ TreeSHAP and DeepSHAP, which are algorithms designed by the author of Lundberg and Lee (2017), for the tree-based machine learning models and Neural Networks, respectively[6].

5 Results

The outcomes of our research are discussed in this section. We conduct a comparative analysis between the (non-)parametric models in the various scenarios of our empirical study.

5.1 Empirical Study Results

5.1.1 Daily Analysis

We start with the daily analysis, which involves calibrating the parametric and non-parametric models on the daily cross-section. Table 2 showcases the RMSE, RMSPE and

[6]python library https://github.com/shap/shap

OR of the daily analysis. The first row (column) in each panel reports the performance measures of the respective (non-)parametric model. Subsequent rows represent the performance measures of a specific parametric model corrected by a non-parametric model indicated in the corresponding column. We denote a parametric model (P) estimated with parameter estimation method (M) corrected by a non-parametric model (NP) as P-M-NP for brevity. For instance, the RMSE of 3.76% in panel A's top left corner follows from the BS model estimated with OLS and corrected by the RF model and is denoted as BS-OLS-RF. Similarly, individual performances of the parametric and non-parametric models are denoted as P-M and NP, respectively. Furthermore, the number after the NNs indicates the number of underlying hidden layers. The bold values highlight the best result per parametric model, while the bold and underlined values represent the best overall result within the relevant panel. In Panel C, we deviate from highlighting the best performance measure per parametric model as these are relative to a benchmark model. Therefore, we only emphasize the best overall result. In our study, we designate the BS model estimated with OLS, which is the simplest model in our paper, as the benchmark model. Furthermore, the hyperparameter grid search details are found in Appendix B. The values in the grid searches are based on common hyperparameter values in similar research and randomized grid searches performed on subsets of the dataset due to computational constraints (Breiman, 1996).

Table 2 illustrates the superiority of tree-based models over NNs in terms of prediction and correction. We note that the individual XGBOOST model and various parametric models corrected by the XGBOOST model are the top-performing models in each column of panel A. In contrast, the RF model surpasses the XGBOOST model in terms of correcting the BS-POLS and AHBS-POLS model in panel B. Although, the differences in performance measures of these two models are marginal, similar to panel A. The results suggest that the BS-OLS-XGBOOST is the best-performing IV prediction model across all panels with an RMSE, RMSPE, and OR of 2.59%, 5.76%, and 77.24%, respectively. The second-best predictions follow from the individual XGBOOST model with an increase equivalent to 0.05%, 0.13% and -2.12% for the RMSE, RMSPE, and OR relative to the BS-OLS-XGBOOST model, respectively. This prompts a consideration between utilizing the BS-OLS model followed by the XGBOOST model for the correction or directly

employing the XGBOOST model for IV prediction, which is a trade-off between accuracy and computational efficiency. Comparing these top two models with the individual parametric models reveals significant improvement in the performance measures.

When focusing on the performance measure of the NN models, we notice a significant increase in RMS(P)E and decrease in OR for the parametric models corrected by the NN models and the individual NN models compared to the tree-based models. A potential explanation is the high parameterization combined with the limited amount of observations in each daily cross-section.

A notable relative improvement is seen in the IV predictions of the CW-NLS model due to the corrections of the XGBOOST model. In particular, we observe a decrease of 18.76%, 58.91% and increase of 65.95% for the RMSE, RMPSE and OR, respectively. Surprisingly, the flat IVS prediction of the BS-OLS model outperforms the more complex CW-NLS model before correction due to the CW model potentially being stuck in a local optimum. After the correction with the various non-parametric models, the BS-OLS and CW-NLS model exhibit a somewhat equivalent performance.

The BS-POLS and AHBS-POLS are the worse performing models including the corrected variant. A possible reason is the difference in average firm IV of the train and test set. The prediction of the test set includes the average firm IV of the train set. A larger difference between these averages leads to larger errors in each observation and, therefore, overall worse performance measures.

All things considered, the results highlight the effectiveness of the non-parametric models, particularly the tree-based models, in capturing the complex relationship between the IV and the incorporated features. Although, the improved performance could be attributed to the inclusion of more features, in other words information, in the non-parametric models for the prediction of the IV.

Table 2

The Root Mean Squared (Percentage) Error and Outperformance Rate (in %) of the models in the daily analysis.

	Non-parametric	BS		AHBS		Carr-Wu
		OLS	POLS	OLS	POLS	NLS
Panel A: Root Mean Squared Error						
Parametric		6.07	21.47	5.19	22.07	21.67
Random Forest	3.91	3.76	15.85	4.56	16.68	3.45
XGBOOST	**2.64**	**2.59**	**15.63**	**4.04**	**16.47**	**2.91**
Neural Network 1	7.41	5.76	17.89	5.63	18.69	5.40
Neural Network 2	7.75	5.93	18.12	5.78	19.06	5.41
Neural Network 3	7.61	5.43	17.89	5.25	18.85	5.03
Neural Network 4	7.37	5.38	18.18	5.22	18.93	4.96
Neural Network 5	7.12	5.69	18.90	5.19	19.75	5.14
Panel B: Root Mean Squared Percentage Error						
Parametric		13.02	53.17	11.16	54.66	65.63
Random Forest	8.50	8.20	**40.98**	10.03	**43.13**	7.81
XGBOOST	**5.89**	**5.76**	50.00	**8.97**	43.16	**6.72**
Neural Network 1	17.58	13.10	45.55	12.65	47.65	12.45
Neural Network 2	18.45	13.72	45.97	12.96	48.46	12.52
Neural Network 3	18.01	12.05	45.44	11.41	47.60	11.29
Neural Network 4	17.30	11.92	46.24	11.33	48.02	11.21
Neural Network 5	16.74	12.38	47.85	11.21	49.62	11.29
Panel C: Outperformance Rate						
Parametric			16.38	67.29	15.97	9.28
Random Forest	76.94	75.36	20.73	70.59	19.85	74.42
XGBOOST	75.12	**77.24**	21.18	72.83	20.29	75.23
Neural Network 1	44.20	55.26	19.12	63.49	18.16	57.80
Neural Network 2	43.34	55.92	18.79	64.13	17.98	58.61
Neural Network 3	42.83	56.78	18.96	66.51	17.96	60.12
Neural Network 4	43.93	56.46	18.74	66.31	17.99	60.50
Neural Network 5	45.10	53.46	18.06	66.89	17.39	59.36

Note: This table presents the Root Mean Squared Error, Root Mean Squared Percentage Error and Outperformance Rate (in %) of the models. Each column represents the parametric model which is corrected by the non-parametric model denoted in the rows. The first row and column are the performance measures of the individual parametric and non-parametric models, respectively. Furthermore, the number after the Neural Network models indicates the number of hidden layers. The relevant performance measures are calculated as the time-series averages of the daily cross-sections from the period of January, 2000 up to and including December, 2021.

5.1.2 Quarterly Analysis

Table 3 reports the results for the quarterly analysis. The dominance of the tree-based models continues comparable to the daily analysis. The BS-OLS-RF model emerges as the best-performing IV prediction model with an RMSE and RMSPE equal to 3.75% and 8.21%, respectively. Meanwhile, the individual RF model has an OR of 76.94% and is the top-performing IV prediction model based on panel C. Once again, the differences between the performance measures of the top two IV prediction models are marginal.

Notable changes in the performances of some models are observed. While most models are robust in performance, the individual XGBOOST model demonstrates a significant

decline in terms of the performance measures compared to the daily analysis. In contrast, the individual CW-NLS model exhibits significant improvement. This improvement could be attributed to the larger training set utilized in the quarterly analysis, which includes observations from two days preceding those in the test set. The larger number of observations may lead to a better convergence in the estimation of the parameters for the more complex CW-NLS model. However, the increase in the number of observations did not improve the performance of the NN models.

Table 3

The Root Mean Squared (Percentage) Error and Outperformance Rate (in %) of the models in the quarterly analysis.

	Non-parametric	**BS**		**AHBS**		**Carr-Wu**
		OLS	**POLS**	**OLS**	**POLS**	**NLS**
Panel A: Root Mean Squared Error						
Parametric		6.06	21.40	5.16	22.00	4.40
Random Forest	**3.90**	**3.75**	**15.80**	**4.54**	**16.62**	**3.83**
XGBOOST	7.69	4.18	16.66	4.66	17.44	4.00
Neural Network 1	7.66	5.90	17.76	5.48	18.70	4.95
Neural Network 2	7.73	5.83	18.00	5.71	18.92	5.15
Neural Network 3	7.76	5.36	17.87	5.17	18.83	4.49
Neural Network 4	7.37	5.40	17.96	5.23	18.78	4.46
Neural Network 5	7.25	5.66	19.04	5.18	19.85	4.43
Panel B: Root Mean Squared Percentage Error						
Parametric		13.02	53.22	11.16	54.70	9.87
Random Forest	**8.51**	**8.21**	**41.03**	**10.02**	**43.18**	**8.54**
XGBOOST	19.29	9.08	41.90	10.21	43.94	8.92
Neural Network 1	18.39	13.53	45.48	12.19	47.82	11.42
Neural Network 2	18.44	13.52	46.08	12.95	48.42	12.13
Neural Network 3	18.27	11.85	45.33	11.26	48.04	10.07
Neural Network 4	17.36	11.87	45.64	11.38	47.95	10.00
Neural Network 5	17.21	12.52	47.78	11.26	50.22	9.94
Panel C: Outperformance Rate						
Parametric			16.38	67.39	15.97	67.40
Random Forest	**76.94**	75.34	20.73	70.70	19.85	70.38
XGBOOST	36.10	74.90	20.05	70.01	19.14	69.09
Neural Network 1	43.92	54.41	19.03	63.41	18.12	62.05
Neural Network 2	42.88	56.57	18.86	64.32	18.08	63.71
Neural Network 3	42.70	56.39	18.84	66.73	18.13	66.40
Neural Network 4	43.69	55.80	18.82	66.24	18.06	66.18
Neural Network 5	44.48	52.94	18.15	67.03	17.37	66.57

Note: This table presents the Root Mean Squared Error, Root Mean Squared Percentage Error and Outperformance Rate (in %) of the models. Each column represents the parametric model which is corrected by the non-parametric model denoted in the rows. The first row and column are the performance measures of the individual parametric and non-parametric models, respectively. Furthermore, the number after the Neural Network models indicates the number of hidden layers. The relevant performance measures are calculated as the time-series averages of the daily cross-sections from the period of March, 2000 up to and including December, 2021.

5.1.3 Complete Dataset Analysis

We observe the results of the complete dataset analysis in Table 4. This scenario deviates from the other scenarios due to inclusion of the macroeconomic features originating from

Welch and Goyal (2008). The overall performance of the models deteriorates when the out-of-sample test set is in the future. However, the two-step prediction procedure still outperforms the individual parametric models. The XGBOOST model is once again the top-performing model in terms of correction and individual prediction. Specifically, the individual XGBOOST model has a significant lower RMSE and RMSPE and higher OR compared to the second best model, which is the BS-OLS-XGBOOST model.

We explore the overall worse performance measures by calculating the performance measures without the Covid-19 crisis in the test set, which is included in Table 12 in Appendix C. However, the performance measures are better when we include this period. Another possible explanation for the deterioration of the models is the time-varying relationship between the features and the IV.

Table 4

The Root Mean Squared (Percentage) Error and Outperformance Rate (in %) of the models in the complete dataset analysis.

	Non-parametric	**BS**		**AHBS**		**Carr-Wu**
		OLS	**POLS**	**OLS**	**POLS**	**NLS**
Panel A: Root Mean Squared Error						
Parametric		16.79	28.30	16.05	29.10	24.31
Random Forest	14.01	14.96	23.65	14.73	24.80	24.94
XGBOOST	**7.57**	**10.87**	**21.00**	**11.48**	**22.94**	**23.26**
Neural Network 1	17.41	16.25	26.16	15.50	25.97	28.32
Neural Network 2	20.75	17.96	32.65	17.58	25.60	30.18
Neural Network 3	21.34	17.28	26.48	16.37	28.77	24.96
Neural Network 4	15.74	16.36	24.24	15.74	26.19	24.68
Neural Network 5	20.03	14.15	23.81	18.93	24.73	26.80
Panel B: Root Mean Squared Percentage Error						
Parametric		26.20	53.78	25.46	56.52	47.23
Random Forest	22.43	25.90	46.95	25.84	**49.90**	46.70
XGBOOST	**13.46**	**19.94**	**46.05**	**20.94**	50.48	**46.64**
Neural Network 1	29.43	27.71	51.47	25.86	51.56	54.89
Neural Network 2	34.24	30.70	66.18	30.32	53.92	60.10
Neural Network 3	38.72	28.76	53.26	27.73	57.37	48.49
Neural Network 4	26.87	28.07	49.33	25.99	52.39	47.56
Neural Network 5	32.87	23.70	49.28	33.98	51.23	51.52
Panel C: Outperformance Rate						
Parametric			28.87	53.58	28.36	37.96
Random Forest	59.46	55.38	32.64	54.65	31.36	34.80
XGBOOST	**73.09**	61.30	37.47	60.48	34.11	42.62
Neural Network 1	44.25	50.36	28.59	54.55	29.43	28.84
Neural Network 2	34.01	40.15	21.20	46.11	33.35	26.98
Neural Network 3	35.92	46.61	30.29	47.54	25.38	37.65
Neural Network 4	52.44	44.03	31.45	51.54	29.11	36.50
Neural Network 5	33.74	53.72	33.11	47.74	31.55	30.14

Note: This table presents the Root Mean Squared Error, Root Mean Squared Percentage Error and Outperformance Rate (in %) of the models. Each column represents the parametric model which is corrected by the non-parametric model denoted in the rows. The first row and column are the performance measures of the individual parametric and non-parametric models, respectively. Furthermore, the number after the Neural Network models indicates the number of hidden layers. The relevant performance measures are calculated as the time-series average of the daily cross-sections from the period of January, 2020 up to and including December, 2021.

5.2 Empirical Study Top Five Liquid Firms Results

A further analysis into the impact of the liquidity on the performance of the models is conducted. As seen in Section 3, each firm has 20 observations on average per day. While the non-parametric models do not differentiate between the different firms in their prediction, the low liquidity per firm could still have a negative impact on the performance. Therefore, we employ the models on a subset consisting of the top five liquid firms based on the number of observations in the original dataset. These are the only firms with more than 100.000 observations over the period January, 2000 up to and including December, 2021. We exclude the pooled estimation method for the BS en AHBS model due to their subpar performance in the original analyses.

5.2.1 Daily Analysis Top Five Liquid Firms

The performance measures of the daily analysis with the subset consisting of the top five liquid firms are presented in Table 5. We observe a similar pattern as the results above, where the tree-based models are superior and the NN models showcase a poor performance. Furthermore, there is an overall improvement in the performance measures. In particular, the top-performing model, which is the XGBOOST model instead of the BS-OLS-XGBOOST model as in the original daily analysis, has an RMSE, RMSPE and OR of 1.50%, 3.49% and 88.32%, respectively. This model exhibits the lowest RMSE and RMSPE among the models considered and the second-highest OR, trailing behind the BS-OLS-XGBOOST model with an OR equal to 88.60%. We notice the largest improvement in the row of the RF and XGBOOST model, and the columns of the AHBS and CW model. Taking everything into account, the increase in observations per firm seems to improve the models and, specifically, the top-performing models in the daily analysis.

Table 5

The Root Mean Squared (Percentage) Error and Outperformance Rate (in %) of the models in the daily analysis for the top five liquid firms.

	Non-parametric	**BS**	**AHBS**	**Carr-Wu**
		OLS	**OLS**	**NLS**
Panel A: Root Mean Squared Error				
Parametric		5.98	4.03	22.82
Random Forest	1.89	1.87	2.07	1.74
XGBOOST	**1.50**	**1.52**	**1.71**	**1.40**
Neural Network 1	6.74	5.91	4.47	4.87
Neural Network 2	6.87	5.99	4.73	4.81
Neural Network 3	6.50	5.50	4.28	4.28
Neural Network 4	5.97	5.30	4.00	4.28
Neural Network 5	5.95	5.55	3.99	4.65
Panel B: Root Mean Squared Percentage Error				
Parametric		15.17	9.97	75.95
Random Forest	4.33	4.30	4.85	4.27
XGBOOST	**3.49**	**3.52**	**4.01**	**3.41**
Neural Network 1	18.04	15.26	11.52	12.90
Neural Network 2	18.75	16.16	12.11	13.42
Neural Network 3	17.40	13.99	10.67	11.38
Neural Network 4	15.99	13.56	9.99	11.47
Neural Network 5	15.66	14.15	9.91	12.37
Panel C: Outperformance Rate				
Parametric			71.40	6.25
Random Forest	88.24	88.45	85.64	86.85
XGBOOST	88.32	**88.60**	87.22	88.71
Neural Network 1	50.05	56.26	65.51	63.36
Neural Network 2	49.17	56.48	67.71	64.27
Neural Network 3	51.11	58.12	70.34	66.17
Neural Network 4	51.28	56.97	70.24	65.20
Neural Network 5	53.19	53.68	71.08	62.49

Note: This table presents the Root Mean Squared Error, Root Mean Squared Percentage Error and Outperformance Rate (in %) of the models. Each column represents the parametric model which is corrected by the non-parametric model denoted in the rows. The first row and column are the performance measures of the individual parametric and non-parametric models, respectively. Furthermore, the number after the Neural Network models indicates the number of hidden layers. The relevant performance measures are calculated as the time-series averages of the daily cross-sections from the period of January, 2000 up to and including December, 2021 for the top five liquid firms.

5.2.2 Quarterly Analysis Top Five Liquid Firms

When we examine the performance measures of the models based on the top five liquid firms for the quarterly analysis in Table 6, the overall performance of the models has improved compared to the original quareterly analysis. The top-performing model remains unchanged compared to the original quarterly analysis. Nonetheless, there has been an improvement in its performance which resulted in an RMSE, RMSPE, and OR of 1.83%, 4.30%, and 88.75%, respectively. The XGBOOST model remains substandard in predicting and correcting the IV even though the liquidity of the firms has increased.

Table 6

The Root Mean Squared (Percentage) Error and Outperformance Rate (in %) of the models in the quarterly analysis for the top five liquid firms.

	Non-parametric	**BS**	**AHBS**	**Carr-Wu**
		OLS	**OLS**	**NLS**
Panel A: Root Mean Squared Error				
Parametric		5.99	4.05	3.99
Random Forest	**1.85**	**1.83**	**2.02**	**1.83**
XGBOOST	5.97	3.75	3.09	3.10
Neural Network 1	6.97	5.78	4.47	4.32
Neural Network 2	6.65	5.80	4.67	4.58
Neural Network 3	6.43	5.47	4.23	4.08
Neural Network 4	6.14	5.34	4.09	3.95
Neural Network 5	5.75	5.56	4.04	3.94
Panel B: Root Mean Squared Percentage Error				
Parametric		15.30	10.12	11.07
Random Forest	**4.33**	**4.30**	**4.82**	**4.60**
XGBOOST	16.71	9.52	7.69	8.54
Neural Network 1	18.77	15.85	11.65	12.01
Neural Network 2	18.20	15.44	12.34	12.92
Neural Network 3	17.44	13.96	10.67	11.14
Neural Network 4	16.31	13.75	10.37	10.72
Neural Network 5	15.21	14.20	10.28	10.79
Panel C: Outperformance Rate				
Parametric			71.23	67.15
Random Forest	88.61	**88.75**	85.87	85.47
XGBOOST	52.45	87.85	81.56	78.34
Neural Network 1	50.76	56.05	65.60	63.40
Neural Network 2	49.91	57.79	67.67	65.32
Neural Network 3	49.49	57.94	69.45	67.17
Neural Network 4	51.41	55.68	69.37	66.95
Neural Network 5	53.57	53.36	70.65	66.88

Note: This table presents the Root Mean Squared Error, Root Mean Squared Percentage Error and Outperformance Rate (in %) of the models. Each column represents the parametric model which is corrected by the non-parametric model denoted in the rows. The first row and column are the performance measures of the individual parametric and non-parametric models, respectively. Furthermore, the number after the Neural Network models indicates the number of hidden layers. The relevant performance measures are calculated as the time-series averages of the daily cross-sections from the period of March, 2000 up to and including December, 2021 for the top five liquid firms.

5.2.3 Complete Dataset Analysis Top Five Liquid Firms

The results of the complete dataset analysis with exclusively the top five liquid firms are shown in Table 7. The best-performing model changed from the XGBOOST to the CW-NLS-XGBOOST model. In addition, the performance of the benchmark model, which is the BS model, improved relatively more than the other models as the OR decreased significantly for all the models. Furthermore, the NN with one hidden layer deteriorated, while the other NN models improved.

Table 7

The Root Mean Squared (Percentage) Error and Outperformance Rate (in %) of the models in the complete dataset analysis for the top five liquid firms.

	Non-parametric	**BS**	**AHBS**	**Carr-Wu**
		OLS	**OLS**	**NLS**
Panel A: Root Mean Squared Error				
Parametric		13.61	11.99	14.26
Random Forest	13.27	13.40	12.61	11.48
XGBOOST	**10.27**	**10.34**	**10.55**	**9.75**
Neural Network 1	18.57	19.17	13.14	14.36
Neural Network 2	16.52	13.95	12.00	26.00
Neural Network 3	16.31	13.55	12.79	13.18
Neural Network 4	17.87	13.72	13.56	14.04
Neural Network 5	15.36	14.87	11.55	13.54
Panel B: Root Mean Squared Percentage Error				
Parametric		26.09	23.27	35.95
Random Forest	27.26	33.85	31.56	29.12
XGBOOST	**21.74**	**26.70**	**27.50**	**26.08**
Neural Network 1	33.43	41.40	26.34	34.30
Neural Network 2	33.77	29.37	25.10	59.24
Neural Network 3	37.92	26.73	25.80	31.06
Neural Network 4	33.00	26.04	25.45	31.28
Neural Network 5	28.83	27.14	24.08	31.21
Panel C: Outperformance Rate				
Parametric			58.38	38.24
Random Forest	46.34	51.00	48.39	51.04
XGBOOST	**58.49**	58.15	55.88	53.28
Neural Network 1	32.87	33.69	52.96	41.47
Neural Network 2	34.32	49.82	56.60	32.68
Neural Network 3	42.98	48.59	55.36	44.01
Neural Network 4	34.03	43.34	45.80	39.03
Neural Network 5	41.89	37.10	56.30	41.33

Note: This table presents the Root Mean Squared Error, Root Mean Squared Percentage Error and Outperformance Rate (in %) of the models. Each column represents the parametric model which is corrected by the non-parametric model denoted in the rows. The first row and column are the performance measures of the individual parametric and non-parametric models, respectively. Furthermore, the number after the Neural Network models indicates the number of hidden layers. The relevant performance measures are calculated as the time-series average of the daily cross-sections from the period of January, 2020 up to and including December, 2021 for the top five liquid firms.

5.3 Feature importance

To examine the feature importance, we calculate the SHAP values in the complete dataset analysis. Given that we employ various models, we focus on the feature importance of the best two-step prediction model. The relevant model in this case is the BS-OLS-XGBOOST model. Figure 7 shows the mean absolute SHAP value of the features. The larger the value the greater the importance of that particular feature in the predictions of the BS-OLS-XGBOOST model. Time-to-maturity, moneyness and macro equity-to-price ratio are the top three most important features based on the SHAP values. The following four features, namely the delta, bid-ask spread (baspread), stock variance (svar)

and midpoint of the bid and ask price (mid_point) are still somewhat relevant. These seven features are option characteristics and macroeconomic features. The importance of the remaining features, which are mostly firm characteristics, are individually negligible. An equivalent conclusion regarding the most important features follows from the other top-performing models as seen in Figures 13-15 in Appendix D.

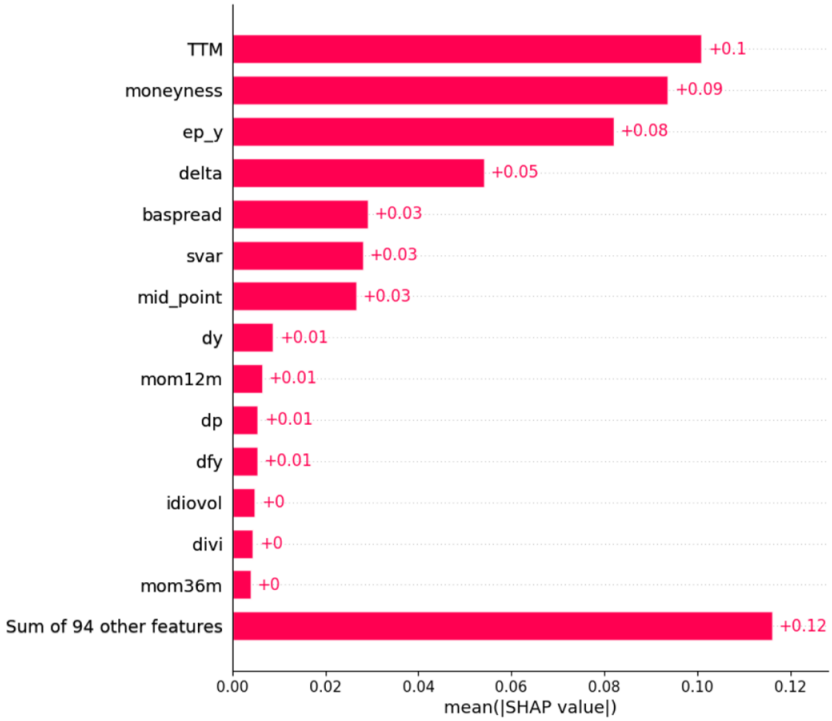

Figure 7

This figure illustrates a bar plot of the mean absolute SHAP values per feature for the best-performing two-step prediction model, which is the Black-Scholes model estimated with Ordinary Least Squares and corrected by the XGBOOST model.

We delve deeper into the top four features and plot the contribution to the prediction of each observation in Figures 8-11. For example, observations with a time-to-maturity greater than two have a SHAP value of around -0.4 to 0, while the SHAP value of observations with a time-to-maturity around zero ranges from around 0 to 1.2. Therefore, observations with a time-to-maturity of around 0 contribute more to the overall prediction. The features have a non-linear relationship with the SHAP value. Figure 8 shows that for the observations with low time-to-maturities have a strong positive contribution towards the IV prediction. Furthermore, deep ITM and OTM options have a positive contribution, while ATM options have a negative contribution as seen in Figure 9. Figure 10 showcases that most observations contribute positively to the prediction. The final

figure is a special case due to the mirrored relationship starting from a delta of 0. This is the boundary of put and call options. Put options have a negative delta, whereas call options have a positive delta. The relationship for both types of options can be seen as negative parabolic where the observations in the middle and outskirts of the parabola have the most contribution towards the correction.

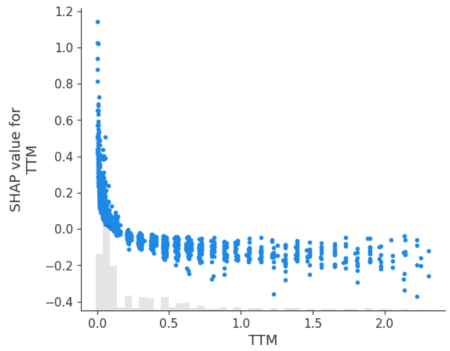

Figure 8

This figure shows a scatter plot of the time-to-maturity (τ) against SHAP values.

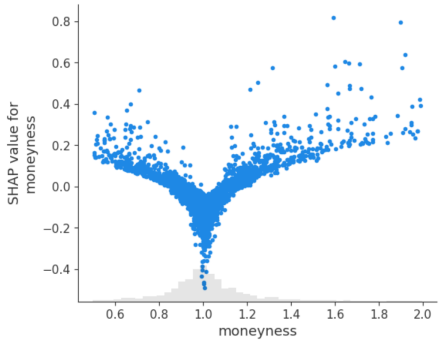

Figure 9

This figure shows a scatter plot of the moneyness (m) against SHAP values.

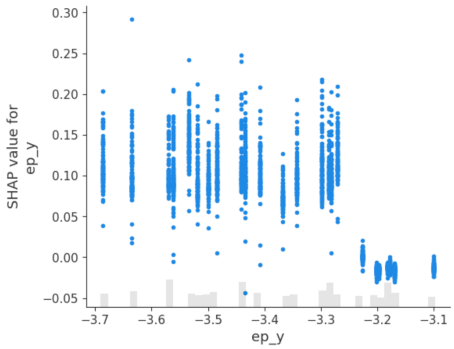

Figure 10

This figure shows a scatter plot of the macro equity-to-price (ep_y) ratio against SHAP values.

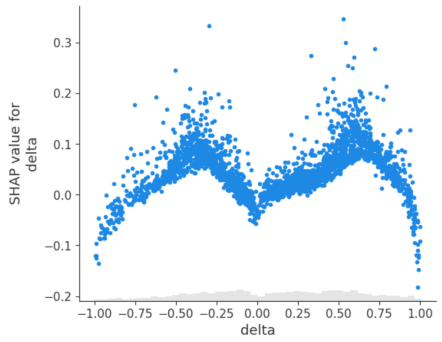

Figure 11

This figure shows a scatter plot of the delta against SHAP values.

To examine the relevancy of the feature importance, we employ the non-parametric models with a limited subset of the features based on the feature importance in the complete dataset analysis. In particular, the top seven most important features observed in Figure 7 are included. These are the time-to-maturity, moneyness, macro equity-to-price ratio, delta, bid-ask spread, stock variance and midpoint of the bid and ask price.

The performance measures are included in Table 8. The continued dominance of the

XGBOOST model is evident. Compared to Table 4, we observe the RMSE, RMSPE and OR of the individual XGBOOST model improving from 7.57%, 13.46% and 73.09% to 6.10%, 11.58% and 75.72%, respectively. In contrast, the performance of most other models deteriorated slightly or remained relatively stable. Furthermore, a substantial decrease in computational time relative to the initial complete dataset analysis is noted. This reduction can be attributed to the utilization of only six features in the non-parametric models in stark contrast to the original 108 features.

Table 8

The Root Mean Squared (Percentage) Error and Outperformance Rate (in %) of the models employed in the complete dataset analysis with exclusively the top seven most important features.

	Non-parametric	**BS**		**AHBS**		**Carr-Wu**
		OLS	**POLS**	**OLS**	**POLS**	**NLS**
Panel A: Root Mean Squared Error						
Parametric		16.79	28.30	16.05	29.10	24.31
Random Forest	12.20	14.18	21.76	14.44	23.03	23.94
XGBOOST	**6.10**	**12.74**	**19.53**	**12.85**	**21.03**	**23.50**
Neural Network 1	18.31	16.14	28.74	14.31	24.46	25.92
Neural Network 2	14.17	15.98	22.73	16.73	22.98	25.51
Neural Network 3	16.40	15.87	21.78	16.49	23.06	25.80
Neural Network 4	11.66	14.72	25.05	26.15	23.94	24.20
Neural Network 5	12.38	33.10	24.13	19.97	26.63	26.57
Panel B: Root Mean Squared Percentage Error						
Parametric		26.20	53.78	25.46	56.52	**47.23**
Random Forest	21.54	26.26	47.57	26.76	51.08	47.37
XGBOOST	**11.58**	**24.13**	**45.56**	**24.07**	**49.27**	47.37
Neural Network 1	30.17	27.25	58.73	24.09	49.75	51.79
Neural Network 2	23.87	30.09	51.19	29.38	50.91	50.89
Neural Network 3	28.07	29.41	47.96	30.11	51.31	51.12
Neural Network 4	19.88	26.29	54.87	55.88	53.96	48.59
Neural Network 5	22.48	69.91	54.31	38.47	62.08	50.36
Panel C: Outperformance Rate						
Parametric			28.87	53.58	28.36	37.96
Random Forest	58.93	51.76	38.15	52.41	37.01	37.54
XGBOOST	**75.72**	54.45	40.60	55.28	39.10	41.11
Neural Network 1	41.38	54.19	25.12	57.79	32.90	34.36
Neural Network 2	53.58	46.54	38.01	45.87	37.79	35.63
Neural Network 3	47.78	48.57	38.30	44.95	36.19	34.74
Neural Network 4	60.37	51.07	35.52	32.47	36.62	38.57
Neural Network 5	58.88	32.03	36.14	44.69	34.03	29.54

Note: This table presents the Root Mean Squared Error, Root Mean Squared Percentage Error and Outperformance Rate (in %) of the models. The non-parametric models exclusively incorporate the top seven most important features. Each column represents the parametric model which is corrected by the non-parametric model denoted in the rows. The first row and column are the performance measures of the individual parametric and non-parametric models, respectively. Furthermore, the number after the Neural Network models indicates the number of hidden layers. The relevant performance measures are calculated as the time-series average of the daily cross-sections from the period of January, 2020 up to and including December, 2021.

6 Conclusion

We conclude our research in this section. Commonly, parametric or non-parametric models are employed in order to predict the IV of options. A more recent two-step prediction procedure, which is introduced by Almeida et al. (2022), utilizes both types of models for the IV prediction of S&P 500 index options. In particular, the non-parametric model is employed in an effort to correct the IV prediction of the parametric model. We deviate from Almeida et al. (2022) by performing a comprehensive analysis of various non-parametric models as correction models in different scenarios. In addition, our dataset consists of individual equity options instead of S&P 500 index options. Hence, our primary research question is: *"Can a non-parametric model correct the Implied Volatility prediction of a parametric model for individual equity options?"*. In addition, we explore two secondary questions, namely *"Which combination of the analyzed parametric and non-parametric model results in the best prediction of the Implied Volatility?"* and *"Which features are the most important for the prediction of the Implied Volatility?"*.

This paper employs the Black-Scholes, Ad-Hoc Black-Scholes and Carr-Wu model as parametric models for the initial IV prediction (Black, 1976; Dumas et al., 1998; Carr and Wu, 2016). The non-parametric models, utilized for the correction of the initial IV prediction, are the Random Forest, Extreme Gradient Boosting and Neural Network model (Breiman, 1996; Chen and Guestrin, 2016; Rumelhart et al., 1986). In order to assess the models, three empirical scenarios are investigated. First, the parametric models are calibrated daily for each firm, while the non-parametric models do not differentiate between the firms and include firm characteristics as features. Second, we extend the calibration time period to include three days and explore the performance of the models on a quarterly basis. In the final scenario, we attempt to predict observations outside of the training time period and include macroeconomic features. Furthermore, the importance of the features included in this particular scenario are investigated with the SHAP method derived from the Shapley values (Shapley, 1953; Lundberg and Lee, 2017).

We find a consistent superiority of the tree-based non-parametric models in the correction of the parametric models over the more complex NN models in the daily and quarterly analysis. Other strong contenders are the individual XGBOOST and RF mod-

els. All things considered, the IV prediction of the parametric model improves after the correction of the XGBOOST and RF models. In case of the NN models, our choices in the architectures in combination with a subpar convergence results in a marginal improvement and, in some cases, a deterioration of the parametric models' IV prediction. Moreover, the performance of the individual NN models is also subpar. Hence, the preference for the tree-based models in IV prediction and correction. When we extend the calibration period to include multiple months, the results are similar. For the final scenario, we observe the performance of all models deteriorating due to the change in the relationship between the included features and IV. The importance of these features are based on the SHAP values (Lundberg and Lee, 2017). Option and macroeconomic characteristics are the most important features, while the importance of the individual firm characteristics are negligible for the IV prediction. When the non-parametric models are calibrated with exclusively the top seven most important features, we notice a similar performance when including all 108 features for the models corrected with the tree-based models, which are the top-performing models. In contrast, the computation time is relatively lower due to the difference in the number of features. Last, we observe an improvement in the performance of the models when we conduct the same empirical analyses for exclusively the top five liquid firms.

All things considered, the non-parametric models, specifically tree-based models, provide considerable improvements in the IV prediction of the parametric models. Moreover, the option, macroeconomic characteristics and number of observations are the main drivers of accurate IV predictions.

7 Discussion

Our research has several limitations and possible interesting areas to consider in future research. The firm characteristics data, which originates from Kelly et al. (2019), is missing 20% of the values in total. Moreover, the final two years are missing 30-40% of the values as seen in Figure 12 in Appendix A.1. These years make up the test set of our complete dataset analysis. Therefore, the results could be considerably influenced by the imputation method of the missing values. We suggest improving the dataset with other

data resources to accommodate the missing values and achieve a more complete dataset. Another limitation is the grid specification for the hyperparameter tuning of the tree-based non-parametric models. The values are partly based on a subset of the data and are fixed over time due to limited computing power and time. A possible improvement is to dynamically specify the hyperparameter grids. Similarly, we only calculate the feature importance of the complete dataset scenario due to the computational expensiveness in the other scenarios. Furthermore, we investigate a finite set of architectures for the NN models. Possible extensions are wider and deeper NN models with various activation functions and optimizers. Another extension is to create a two-step prediction model with a switching parameter. In other words, an appropriate combination of the parametric and non-parametric model will be chosen based on the underlying data in each iteration. Similarly, the length of the rolling window for the training period could be included as a parameter. The period could be adjusted based on the financial market conditions. A possible split would be between a bull and bear market. Furthermore, a common approach in similar research is to partition the options based on time-to-maturity and moneyness. Afterwards, the models are calibrated per partition. In conclusion, we suggest exploring real-world applications, such as portfolio construction and risk assessment.

References

Ackerer, D., Tagasovska, N., and Vatter, T. (2020). Deep smoothing of the implied volatility surface. *Advances in Neural Information Processing Systems*, 33:11552–11563.

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

Anders, U., Korn, O., and Schmitt, C. (1998). Improving the pricing of options: A neural network approach. *Journal of forecasting*, 17(5-6):369–388.

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

Andreou, P. C., Charalambous, C., and Martzoukos, S. H. (2010). Generalized parameter functions for option pricing. *Journal of banking & finance*, 34(3):633–646.

Bakshi, G., Cao, C., and Chen, Z. (1997). Empirical performance of alternative option pricing models. *The Journal of finance*, 52(5):2003–2049.

Bartlett, P., Freund, Y., Lee, W. S., and Schapire, R. E. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686.

Bates, D. S. (1991). The crash of 87: was it expected? the evidence from options markets. *The journal of finance*, 46(3):1009–1044.

Bates, D. S. (1996a). 20 testing option pricing models. *Handbook of statistics*, 14:567–611.

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

Bates, D. S. (2003). Empirical option pricing: A retrospection. *Journal of Econometrics*, 116(1-2):387–404.

Bates, D. S. (2019). How crashes develop: intradaily volatility and crash evolution. *The Journal of Finance*, 74(1):193–238.

Bates, D. S. (2022). Empirical option pricing models. *Annual Review of Financial Economics*, 14:369–389.

Becker, S., Cheridito, P., and Jentzen, A. (2019). Deep optimal stopping. *The Journal of Machine Learning Research*, 20(1):2712–2736.

Becker, S., Cheridito, P., and Jentzen, A. (2020). Pricing and hedging american-style options with deep learning. *Journal of Risk and Financial Management*, 13(7):158.

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

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

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

Boek, C., Lajbcygier, P., Palaniswami, M., and Flitman, A. (1995). A hybrid neural network approach to the pricing of options. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 2, pages 813–817. IEEE.

Bordes, A., Glorot, X., Weston, J., and Bengio, Y. (2012). Joint learning of words and meaning representations for open-text semantic parsing. In *Artificial intelligence and statistics*, pages 127–135. PMLR.

Bramer, M. (2007). Avoiding overfitting of decision trees. *Principles of data mining*, pages 119–134.

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

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

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). Classification and regression trees.

Buehler, H., Gonon, L., Teichmann, J., and Wood, B. (2019). Deep hedging. *Quantitative Finance*, 19(8):1271–1291.

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

Carr, P. and Wu, L. (2017). Leverage effect, volatility feedback, and self-exciting market disruptions. *Journal of Financial and Quantitative Analysis*, 52(5):2119–2156.

Carr, P. and Wu, L. (2020). Option profit and loss attribution and pricing: A new framework. *The Journal of Finance*, 75(4):2271–2316.

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

Cuchiero, C., Khosrawi, W., and Teichmann, J. (2020). A generative adversarial network approach to calibration of local stochastic volatility models. *Risks*, 8(4):101.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1*, pages 1–15. Springer.

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

Finnoff, W., Hergert, F., and Zimmermann, H. G. (1993). Improving model selection by nonconvergent methods. *Neural Networks*, 6(6):771–783.

Fisher, A., Rudin, C., and Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, 20(177):1–81.

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

Freyberger, J., Neuhierl, A., and Weber, M. (2020). Dissecting characteristics nonparametrically. *The Review of Financial Studies*, 33(5):2326–2377.

Ganaie, M. A., Hu, M., Malik, A., Tanveer, M., and Suganthan, P. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151.

Garcia, R. and Gençay, R. (2000). Pricing and hedging derivative securities with neural networks and a homogeneity hint. *Journal of Econometrics*, 94(1-2):93–115.

Garman, M. B. and Kohlhagen, S. W. (1983). Foreign currency option values. *Journal of international Money and Finance*, 2(3):231–237.

Gençay, R. and Qi, M. (2001). Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging. *IEEE transactions on neural networks*, 12(4):726–734.

Gençay, R. and Salih, A. (2003). Degree of mispricing with the black-scholes model and nonparametric cures. *Economics and Finance. Annals*, 4:73–101.

Geske, R. and Johnson, H. E. (1984). The american put option valued analytically. *The Journal of Finance*, 39(5):1511–1524.

Ghysels, E., Patilea, V., Renault, É., Torrès, O., et al. (1997). *Nonparametric methods and option pricing*. CIRANO.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings.

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

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

Heaton, J. B., Polson, N. G., and Witte, J. H. (2017). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1):3–12.

Heij, C., de Boer, P., Franses, P. H., Kloek, T., van Dijk, H. K., et al. (2004). *Econometric methods with applications in business and economics*. Oxford University Press.

Hentschel, L. (2003). Errors in implied volatility estimation. *Journal of Financial and Quantitative analysis*, 38(4):779–810.

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.

Heynen, R. (1994). An empirical investigation of observed smile patterns. *Review of Futures Markets*, 13:317–317.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

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

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Horvath, B., Muguruza, A., and Tomas, M. (2021). Deep learning volatility: a deep neural network perspective on pricing and calibration in (rough) volatility models. *Quantitative Finance*, 21(1):11–27.

Hull, J. and White, A. (1987). The pricing of options on assets with stochastic volatilities. *The journal of finance*, 42(2):281–300.

Hutchinson, J. M., Lo, A. W., and Poggio, T. (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks. *The journal of Finance*, 49(3):851–889.

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

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

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kou, S. G. (2002). A jump-diffusion model for option pricing. *Management science*, 48(8):1086–1101.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Kumar, S. (2023). A review of neural network applications in derivative pricing, hedging and risk management. *Academy of Marketing Studies Journal*, 27(3).

Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867.

Linardatos, P., Papastefanopoulos, V., and Kotsiantis, S. (2020). Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18.

Liu, S., Borovykh, A., Grzelak, L. A., and Oosterlee, C. W. (2019a). A neural network-based framework for financial model calibration. *Journal of Mathematics in Industry*, 9:1–28.

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

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Malliaris, M. and Salchenberger, L. (1993). A neural network model for estimating option prices. *Applied Intelligence*, 3:193–206.

Masters, T. (1993). *Practical neural network recipes in C++*. Morgan Kaufmann.

Merton, R. C. (1973). Theory of rational option pricing. *The Bell Journal of economics and management science*, pages 141–183.

Mitra, S. (2011). A review of volatility and option pricing. *International Journal of Financial Markets and Derivatives*, 2(3):149–179.

Morgan, N. and Bourlard, H. (1989). Generalization and parameter estimation in feedforward nets: Some experiments. *Advances in neural information processing systems*, 2.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.

Orlando, G. and Taglialatela, G. (2017). A review on implied volatility calculation. *Journal of Computational and Applied Mathematics*, 320:202–220.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Prechelt, L. (1998). Automatic early stopping using cross validation: quantifying the criteria. *Neural networks*, 11(4):761–767.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Rubinstein, M. (1985). Nonparametric tests of alternative option pricing models using all reported trades and quotes on the 30 most active cboe option classes from august 23, 1976 through august 31, 1978. *The Journal of Finance*, 40(2):455–480.

Ruf, J. and Wang, W. (2020). Neural networks for option pricing and hedging: a literature review. *Journal of Computational Finance*, 24(1):1–46.

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

Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, 5:197–227.

Shapley, L. S. (1953). *Contributions to the Theory of Games, Chapter A Value for n-person Games*. Princeton University Press.

Sharma, S., Sharma, S., and Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316.

Shastri, K. and Tandon, K. (1986). An empirical test of a valuation model for american options on futures contracts. *Journal of Financial and Quantitative Analysis*, 21(4):377–392.

Sheikh, A. M. (1991). Transaction data tests of s&p 100 call option pricing. *Journal of Financial and Quantitative Analysis*, 26(4):459–475.

Shin, Y. and Karniadakis, G. E. (2020). Trainability of relu networks and data-dependent initialization. *Journal of Machine Learning for Modeling and Computing*, 1(1).

Sirignano, J. and Spiliopoulos, K. (2018). Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364.

Smith Jr, C. W. (1976). Option pricing: A review. *Journal of Financial Economics*, 3(1-2):3–51.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Sussillo, D. and Abbott, L. (2014). Random walk initialization for training very deep feedforward networks. *arXiv preprint arXiv:1412.6558*.

Sutton, C. D. (2005). Classification and regression trees, bagging, and boosting. *Handbook of statistics*, 24:303–329.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288.

Welch, I. and Goyal, A. (2008). A comprehensive look at the empirical performance of equity premium prediction. *The Review of Financial Studies*, 21(4):1455–1508.

Appendices

A Data description

A.1 Missing values

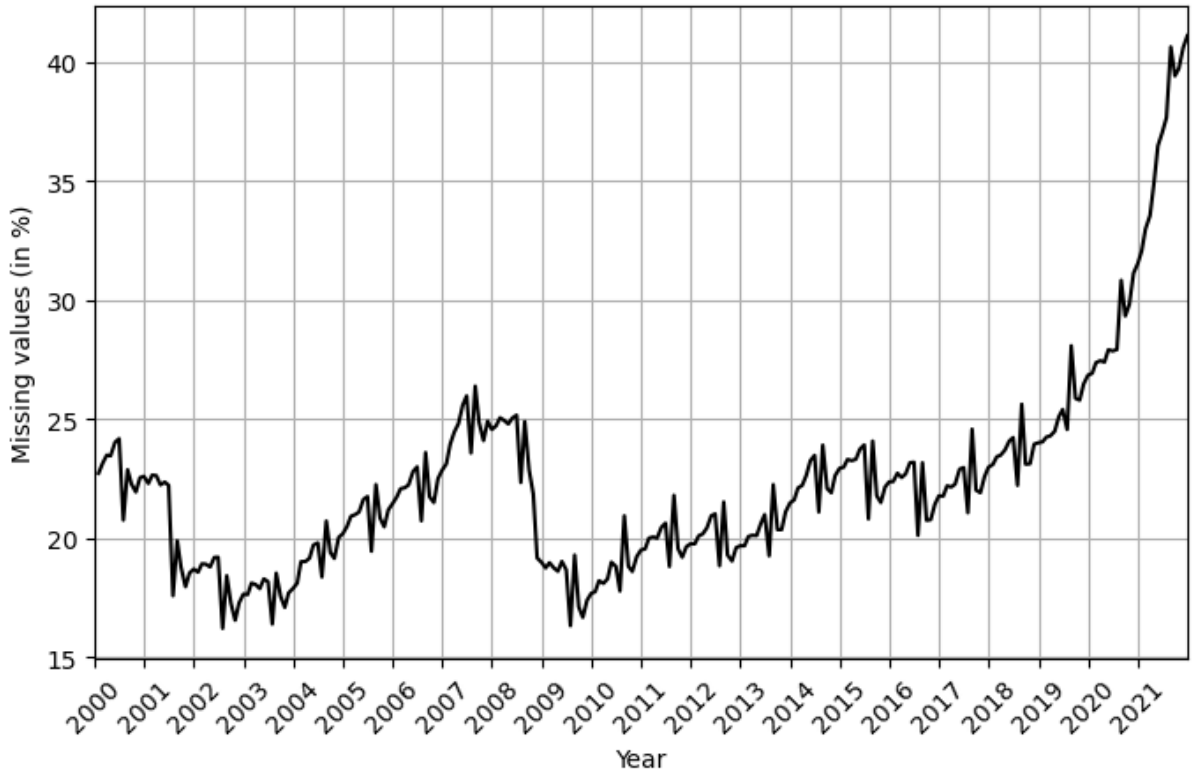

Figure 12

The percentage of missing values in the dataset for the period of January, 2000 up to and including December, 2021.

B Hyperparameter tuning

We list the hyperparameter grids utilized during the empirical analysis for the Random Forest, Extreme Gradient Boosting and Neural Network model. The unspecified hyperparameters are set to the default values of the `Scikit-learn` Python library version 1.4.1, `XGBoost` Python library version 2.0.3 and `Tensorflow` Python library version 2.15.0.

B.1 Random Forest

Table 9

Hyperparameter Grid of the Random Forest model.

Hyperparameter	General Grid
max_depth	{8, 10, 12}
max_features	{"sqrt"}
n_estimators	{100, 200, 500}

B.2 XGBOOST

Table 10

Hyperparameter Grid of the XGBOOST model.

Hyperparameter	General Grid
learning_rate	{0.01, 0.1, 0.3}
max_depth	{6, 12}
n_estimators	{25, 50, 100, 200}
reg_alpha	{0.1, 0.5, 1}
reg_lambda	{0.1, 0.5, 1}
subsample	{0.1, 0.5, 1}

B.3 Neural Network

Table 11

Hyperparameter Grid of the Neural Network models.

Hyperparameter	General Grid
initializer	{"HeNormal"}
optimizer	{"Adam"}
activation	{"ReLu"}
batch_normalisation	{"After the first layer"}
epochs	{250}
learning_rate	{0.01}
batch_size	$\{\frac{\#\text{Number of observations}}{10}\}$
patience	{50}

C Complete dataset analysis excluding the Covid-19 crisis

Table 12

The Root Mean Squared (Percentage) Error and Outperformance Rate (in %) of the models in the complete dataset analysis excluding the period January, 2020 up to and including June, 2020 (Covid-19 crisis).

	Non-parametric	**BS**		**AHBS**		**Carr-Wu**
		OLS	**POLS**	**OLS**	**POLS**	**NLS**
Panel A: Root Mean Squared Error						
Parametric		14.81	27.49	14.22	28.43	23.77
Random Forest	12.71	13.96	21.76	13.86	23.02	**24.11**
XGBOOST	**7.01**	**10.71**	**19.74**	**11.24**	**21.44**	24.74
Neural Network 1	15.99	14.93	24.24	14.07	24.48	27.36
Neural Network 2	19.35	16.69	30.70	16.17	25.12	29.70
Neural Network 3	19.85	15.94	25.55	15.02	26.57	24.49
Neural Network 4	14.39	15.36	23.39	14.17	25.01	23.98
Neural Network 5	18.66	13.53	23.16	18.32	24.06	25.58
Panel B: Root Mean Squared Percentage Error						
Parametric		24.69	55.92	24.07	59.03	48.71
Random Forest	22.13	26.26	45.97	26.32	**49.22**	**47.57**
XGBOOST	**13.42**	**20.60**	**45.42**	**21.55**	49.35	51.88
Neural Network 1	28.37	26.95	50.55	24.95	51.51	55.73
Neural Network 2	33.34	29.96	65.34	29.55	56.01	62.21
Neural Network 3	38.30	28.12	54.19	26.95	56.09	49.83
Neural Network 4	26.38	27.73	50.19	24.96	52.76	48.55
Neural Network 5	32.03	23.46	50.36	34.58	52.43	51.92
Panel C: Outperformance Rate						
Parametric			27.21	53.73	26.59	35.62
Random Forest	56.69	51.57	32.61	50.68	31.21	33.08
XGBOOST	71.19	57.77	36.96	57.22	33.81	37.43
Neural Network 1	43.73	49.36	28.57	53.38	28.69	27.96
Neural Network 2	32.69	39.11	20.68	45.12	31.27	25.16
Neural Network 3	34.95	44.94	28.85	46.24	25.67	35.23
Neural Network 4	50.68	42.10	29.92	50.78	28.07	35.05
Neural Network 5	32.03	51.21	31.53	45.33	29.64	29.71

Note: This table presents the Root Mean Squared Error, Root Mean Squared Percentage Error and Outperformance Rate (in %) of the models. Each column represents the parametric model which is corrected by the non-parametric model denoted in the rows. The first row and column are the performance measures of the individual parametric and non-parametric models, respectively. Furthermore, the number after the Neural Network models indicates the number of hidden layers. The relevant performance measures are calculated as the time-series average of the daily cross-sections from the period of July, 2020 up to and including December, 2021.

D SHAP values

D.1 SHAP values of the XGBOOST model

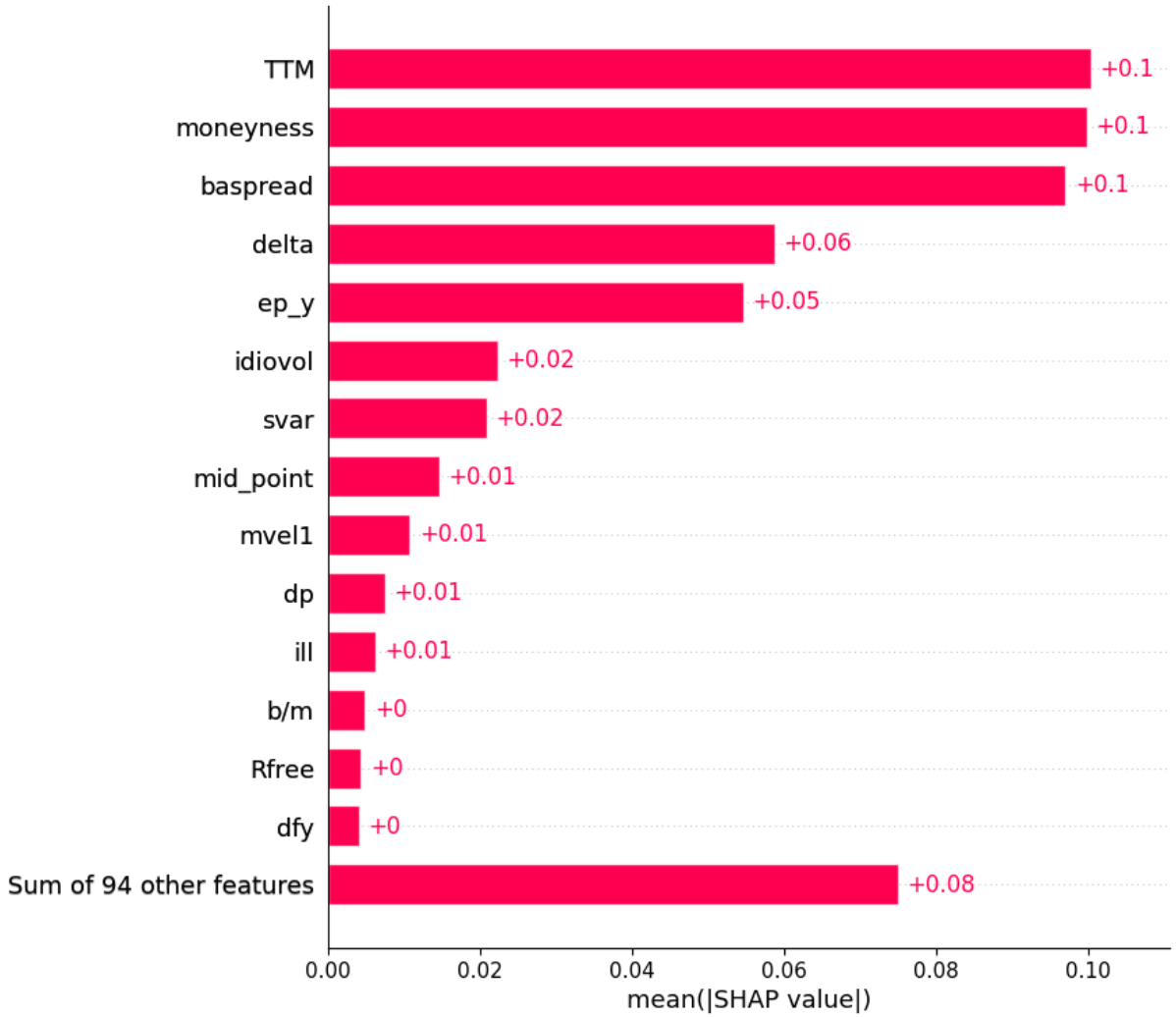

Figure 13

This figure illustrates a bar plot of the mean absolute SHAP values per feature for the XGBOOST model.

D.2 SHAP values of the CW-NLS-XGBOOST model

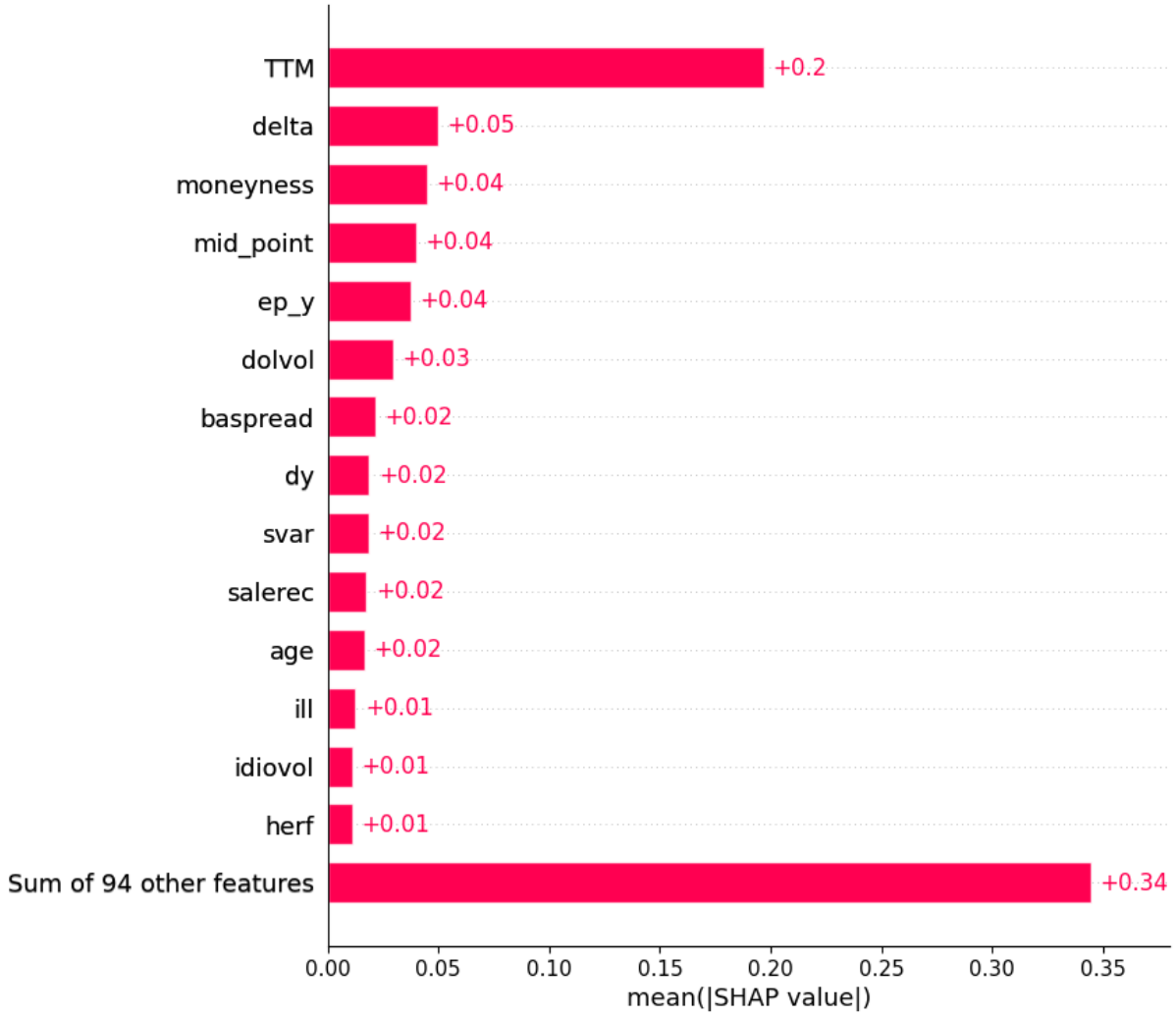

Figure 14

This figure illustrates a bar plot of the mean absolute SHAP values per feature for the Carr-Wu model estimated with Non-linear Least Squares and corrected by XGBOOST.

D.3 SHAP values of the AHBS-OLS-XGBOOST model

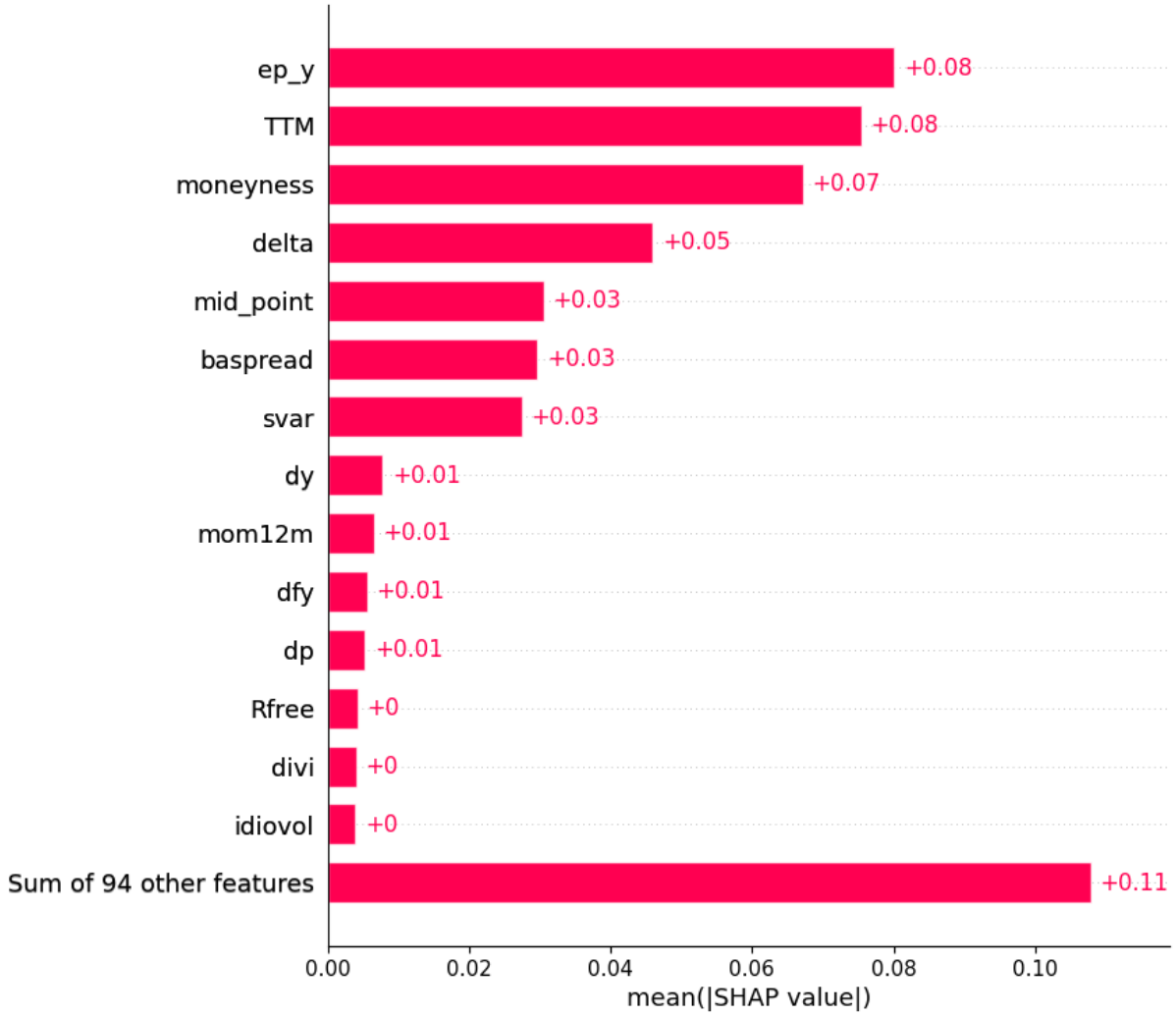

Figure 15

This figure illustrates a bar plot of the mean absolute SHAP values per feature for the Ad-Hoc Black-Scholes model estimated with Ordinary Least Squares and corrected by XGBOOST.