ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Master Thesis Quantitative Finance

Option Pricing Boosted by Machine Learning Techniques

Glenn Cato Visser (458412)

Supervisor: dr. O. Kleen

Second assessor: dr. G. Freire

Date: March 29, 2023

Abstract

I investigate the extent to which machine learning techniques can improve the performance of parametric option pricing models. Given the estimates of several models, such as the Black-Scholes and Heston models, I train random forests, support vector machines and neural networks to either correct or combine their individual forecasts. Using a dataset composed of S&P 500 options, I show that these techniques are able to outperform the parametric models significantly. Out-of-sample prediction exercises show large gains in the cross-section as well as in the option panel in fitting the implied volatility surface. Including time-varying information as features allows for the neural network to better reproduce the dynamics of the surface over time.

Contents

1 Introduction

Implied volatility is a metric that captures the market's expectation of future movements in a security's price. It is extracted from observed option prices using the Black and Scholes (1973) formula. As implied volatilities of different options are easier to compare with each other across time and moneyness than option prices, institutional investors often manage their option positions using the implied volatility surface (Carr & Wu, 2016). This is a three-dimensional plot of the implied volatilities of a security's options across different times to maturity and strike prices.

Furthermore, as shown by Andersen, Fusari and Todorov (2015), option pricing models are often estimated by representing the pricing errors in terms of the implied volatility and by subsequently minimizing these errors. This shows the importance of accurately estimating and predicting the implied volatility surface for researchers as well as traders or investors that are active in the option market.

Over the years, many parametric option pricing models have been developed to capture the observed implied volatility dynamics. Almeida et al. (2022) show that training a feed-forward neural network on parametric model-implied pricing errors will most likely enhance performance in terms of accuracy. The results of this nonparametric correction - which amounts to a semiparametric methodology - show the potential of machine learning techniques in modeling the dynamics of the implied volatility surface. This potential is explained by the fact that machine learning techniques are able to handle a large number of potential predictor variables (high-dimensional data) and approximate nonlinear specifications of functional form or relations.

This invites me to investigate the extent to which machine learning techniques can benefit parametric models' fitting of the option panel. In this paper, I look at the performance of the random forest algorithm of Breiman (2001) and the support vector machine when nonparametrically correcting the parametric models; two methods that are suited for classification as well as regression purposes.

In addition, I research the ability of neural networks to combine the individual parametric forecasts, following along the lines of Donaldson and Kamstra (1996). They show that neural

network combinations generally outperform linear combinations of individual forecasts.

I use a dataset comprising 1,196,437 observations of S&P 500 options over the period between January 4, 2016, and June 28, 2019, to test the corrections and combinations of the Black and Scholes (1973) model, the ad-hoc correction of Black-Scholes (Dumas et al., 1998) which describes the volatility as a quadratic function of an option's moneyness and time to maturity, the Carr and Wu (2016) model that specifies the dynamics of the implied volatility surface parametrically, and the Heston (1993) structural stochastic volatility model. In evaluating their performance, I consider out-of-sample cross-sectional results and results in the option panel.

In general, the random forest and neural network corrections are able to outperform the parametric models significantly when prediction exercises in the cross-section are conducted. Similarly, the performance of the neural network combinations of the individual forecasts exceeds that of the parametric models in the cross-sectional prediction exercises. In the option panel, the neural network combinations outperform the individual forecasts to a large extent, as well as their simple and weighted averages. Including time-varying covariates as features further allows neural networks to fit the option panel even more accurately and capture its dynamics. I find that a measure of economic policy uncertainty and the VIX index as a measure of market risk are the two most important features in providing neural networks with the information needed about the state of the economy. This is a sensible result, as L. Liu and Zhang (2015) show that economic policy uncertainty is strongly positively correlated with markets' future volatility. Furthermore, the VIX index is constructed from a portfolio of out-of-the-money S&P 500 options, and captures the expected volatility of the index under the risk-neutral distribution.

The remainder of this paper is structured as follows. In Section 2, I discuss the four parametric option pricing models. Section 3 sets out the procedure leading to a nonparametric correction of these models. The option data I use for my research is described in Section 4. Sections 5, 6 and 7 compare the performances of the different machine learning techniques correcting and combining the parametric models within the scope of the exercises in the cross-section. Section 8 discusses the exercise in the option panel. Lastly, Section 9 concludes this paper.

1.1 Related Literature

This thesis is related to the vast literature on option pricing. One of the earliest works on the subject is Black and Scholes (1973), which derives a closed-form solution for the price of a European call option under the assumption of constant volatility. Due to the limitations of this model, several extensions and generalizations have been proposed to capture the dynamics of the implied volatility surface. One approach is to use local volatility as a deterministic function of the underlying asset price and time. Dupire (1994) and Rubinstein (1994) estimate local volatility using a binomial lattice. Another approach is to correct Black-Scholes by smoothing implied volatilities, which is shown by Dumas et al. (1998) to outperform local volatility models. Structural continuous-time option pricing models that incorporate additional sources of risk, such as stochastic volatility and jumps, comprise a different category of models. Heston (1993), Bates (2000), Duffie et al. (2000), and Andersen et al. (2015) are some prominent examples in this category. Several notable works use approximation theory to express implied volatilities as a function of parameters of stochastic volatility models. Medvedev and Scaillet (2007), Gatheral and Jacquier (2014), and Carr and Wu (2016) are instances of this important type of model.

Nonparametric methods have gained increasing attention in the field of financial econometrics, particularly in option pricing. An example is the nonparametric approach that was introduced by Aït-Sahalia and Duarte (2003), who used a nonparametric kernel-based estimator for the state-price density implicit in the market prices of traded options. Several works provide results on combining parametric and nonparametric models in financial settings. J. Fan and Mancini (2009) propose a novel way of option pricing based on nonparametric methods guided by a parametric model. They fit a nonparametric model on the errors of a parametric model. Along these lines, Almeida et al. (2022) train a feedforward neural network on the pricing errors of parametric option pricing models, and achieve boosted performance. They furthermore show that incorporating time-varying covariates as features in a neural network, allows for the network to better learn the shape of the misspecification as a function of different states of the economy. Kumar and Thenmozhi (2014) compare the performance of three different hybrid models in forecasting stock index returns. They examine combinations of linear ARIMA models with support vector machines, random forests and

neural networks, and find that the first combination is the best forecasting model to achieve high forecast accuracy. No research has been conducted comparing these nonparametric methods when correcting parametric predictions of the implied volatility surface. Therefore, I look at the performance of these support vector machine and random forest corrections, and compare them with the neural network correction proposed by Almeida et al. (2022).

Several other academic papers utilize neural networks to price options and manage risk, as they are effective at modeling complex relationships. Hutchinson et al. (1994), Garcia and Gençay (2000), and Amilon (2003) use neural networks to estimate the option pricing function nonparametrically. Others, such as Dugas et al. (2009) and Ackerer et al. (2020), expand on this by incorporating no-arbitrage constraints. S. Liu et al. (2019) apply neural networks to decrease computational time by numerically solving option pricing models. Harrald and Kamstra (1997) describe an experiment using a neural network combining forecasts of stock price volatility, outperforming simple linear models and a nonparametric kernel method. Following a similar approach in the option pricing setting, I use neural networks to combine individual implied volatility forecasts of parametric models, before and after providing the networks with information about the options and the state of the economy.

2 Parametric Option Pricing Models

2.1 Black-Scholes Model

The model developed by Black and Scholes (1973) expresses the dynamics of a market containing derivative instruments. The model allows for computation of the theoretical value of a European option based on certain assumptions. One of those assumptions is the underlying asset's price, denoted by S_t, following a geometric Brownian motion with constant drift μ and volatility σ:

$$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t, \tag{1}$$

where W_t represents a standard Wiener process. A partial differential equation can be derived and solved from this process for a European call option. Given the annualized risk-free interest rate r, the equation presents the following solution for the price of such a call with

strike price K and time to maturity $\tau = (T - t)$:

$$C(S_t, K, \tau, \sigma) = \Phi(d_1)S_t - \Phi(d_2)Ke^{-r\tau}, \tag{2}$$
$$d_1 = \frac{1}{\sigma\sqrt{\tau}}\left[ln\left(\frac{S_t}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)\tau\right],$$
$$d_2 = d_1 - \sigma\sqrt{\tau},$$

where $\Phi(x)$ denotes the standard normal cumulative distribution function evaluated at x. Equation (2) is known as the Black-Scholes formula. By means of the put-call parity, the corresponding price of a put with the same strike price K is given by:

$$P(S_t, K, \tau, \sigma) = \Phi(-d_2)Ke^{-r\tau} - \Phi(-d_1)S_t. \tag{3}$$

Given observed market prices $\overline{C}$ and $\overline{P}$, the implied volatility of the options is computed by solving the inverse of equations (2) and (3) for σ, respectively. Alternatively stated, the implied volatility is the value of σ for which the theoretical option price is equal to its market-observed counterpart, i.e., $C = \overline{C}$ or $P = \overline{P}$.

The model predicts that the implied volatility is constant over time, as well as over maturity and strike price, resulting in a flat implied volatility surface. This is in contradiction with observed implied volatility values, as the cross-section of for example S&P 500 index options shows a "smile", as shown by Rubinstein (1994). Furthermore, the market observed implied volatilities change over time, thus deforming the shape of the implied volatility surface (Cont & Da Fonseca, 2002). The inconsistency between the Black-Scholes option pricing model (BS) and the observed implied volatility values, has led to a vast amount of literature concerning models that are able to capture the dynamics of the implied volatility surface better. In the rest of this section, I discuss three examples.

2.2 Ad-hoc Black-Scholes Model

As the Black-Scholes implied volatilities of S&P 500 options tend to have a parabolic shape - this phenomenon is also referred to as the volatility smile - Dumas et al. (1998) decide to model the volatility as a quadratic function of the time to maturity and moneyness of an option. Given a cross-section of $i = 1, 2, ..., n$ options on day t, the moneyness of option i

is defined as $m_{i,t} = S_t/K_i$. Dumas et al. (1998) consider several functional forms, the most general of which is:

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

The model is called ad-hoc Black-Scholes (AHBS), as it violates the constant implied volatility assumption and is thus inconsistent with the Black-Scholes model. The ad-hoc model is further analyzed and augmented by Christoffersen and Jacobs (2004), who named it the Practitioner Black-Scholes model. Implementation of the model is done in three stages. First, on a particular day t with a cross-section of $i = 1, 2, ..., n$ options, the Black-Scholes implied volatility $\sigma_{i,t}$ for every option is backed out of equation (2) or (3) as described in Section 2.1. Next, in order to estimate the model, the volatilities are regressed on $m_{i,t}$ and $\tau_{i,t}$ as described in equation (4) via ordinary least squares. This is equivalent to minimizing the implied volatility root mean squared error (IVRMSE):

$$\hat{\boldsymbol{\theta}}_{AHBS} = \arg\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n}\left[\sigma_{i,t} - (\theta_{0,t} + \theta_{1,t}m_{i,t} + \theta_{2,t}m_{i,t}^2 + \theta_{3,t}\tau_{i,t} + \theta_{4,t}\tau_{i,t}^2 + \theta_{5,t}m_{i,t}\tau_{i,t})\right]^2, \tag{5}$$

where $\hat{\boldsymbol{\theta}}_{AHBS}$ is the vector containing the parameters that need to be estimated. Third, the implied volatilities predicted by the AHBS model are calculated by plugging in $\hat{\boldsymbol{\theta}}_{AHBS}$. Note here that for equation (4), restricting $\theta_1, \theta_2, ..., \theta_5 = 0$ results in a constant volatility model. This amounts to the standard Black-Scholes model, which assumes that volatility is constant over time, maturity and strike price. That implies that the Black-Scholes predicted implied volatility for any option in the cross-section is given by the average implied volatility observed on day t.

2.3 Heston Model

An alternative to the assumption of constant volatility is stochastic volatility. This gives rise to the category of stochastic volatility models, which assume that the underlying asset's volatility follows a random process. The Heston (1993) model is amongst the first of these models. They allow for a quasi-closed form solution for the price of a European call and put option. The model assumes a correlation between the volatility process and the asset price,

and the risk-neutral stock price process is given by:

$$\frac{dS_t}{S_t} = (r - \frac{1}{2}V_t)dt + \sqrt{V_t}dW_{S,t}, \tag{6}$$
$$dV_t = \kappa(\overline{v} - V_t)dt + \sigma_v\sqrt{V_t}dW_{V,t}.$$

Here, V_t is the instantaneous variance, κ denotes the rate at which V_t reverts to the long-term average $\overline{v}$, σ_v is the volatility of the volatility process, and $W_{S,t}, W_{V,t}$ are two correlated Wiener processes with correlation ρ. Under the Heston (1993) framework, option prices are computed using the following expressions:

$$C_{Heston} = S_t e^{-q\tau}P_1 - Ke^{-r\tau}P_2, \tag{7}$$

$$P_{Heston} = C_{Heston} + Ke^{-r\tau} - S_t e^{-q\tau}, \tag{8}$$

$$P_j = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty \text{Re}\left[\frac{e^{-i\phi\ln(K)}f_j(\phi)}{i\phi}\right]d\phi, \qquad j \in \{1,2\}, \tag{9}$$

where q is the continuous dividend yield, C_{Heston} and P_{Heston} are the model's call and put price, respectively, and i is a unit imaginary number. Furthermore, P_1, P_2 are the probabilities of $S_t > K$ under the asset price measure and risk-neutral measure, respectively, and $f_j(\phi)$ is the characteristic function for P_1, P_2, with ϕ as the characteristic function variable. The continuous integral for the inverse Fourier transform is evaluated using numerical integration.[1]

As the performance of the Heston model will be evaluated in terms of its implied volatility surface predictions, it makes sense to estimate the set of parameters and the instantaneous variance $\boldsymbol{\xi}_t = (V_t, \overline{v}, \kappa, \sigma_v, \rho)$ by minimizing the loss function with respect to the observed implied volatilities $\sigma_{i,t}$. Model-implied option prices are converted to implied volatilities, and the loss function is minimized using nonlinear least squares. Denote the fitted values of the Heston model as $\sigma_H(\boldsymbol{\xi}_t, S_t, K_{i,t}, \tau_{i,t}, r_t)$. For a set of options $i = 1, 2, .., n$ on day t, estimates $\hat{\boldsymbol{\xi}}_t$ are then the result of minimizing:

$$\sum_{i=1}^{n}\left[\sigma_{i,t} - \sigma_H(\boldsymbol{\xi}_t, S_t, K_{i,t}, \tau_{i,t}, r_t)\right]^2. \tag{10}$$

[1]For the pricing of options under the Heston model, MATLAB function "optByHestionNI" is used. For further documentation, see https://nl.mathworks.com/help/fininst/optbyhestonni.html.

2.4 Carr and Wu Model

Carr and Wu (2016) develop a new option pricing framework. They recognize that both practitioners and academics are used to employing the Black-Scholes implied volatility surface to manage their option positions instead of through option prices. Therefore, they thought it ideal to, instead of modeling all the dynamics of the (unobservable) instantaneous variance, model the near-term dynamics of the implied volatility surface across a range of different strike prices and times to maturity. Subsequently, they derive no-arbitrage restrictions on its shape. The model, with its assumptions on the implied volatility dynamics, allows for the obtainment of the whole implied volatility surface by solving a simple quadratic equation.

Consider an option with strike price K and time to maturity τ. The model developed by Carr and Wu (2016) (CW) states that under the risk-neutral measure, the dynamics of the underlying asset price S_t and implied volatility $\sigma_t(K,\tau)$ are given by:

$$dS_t/S_t = \sqrt{v_t}dW_t, \tag{11}$$

$$d\sigma_t(K,\tau)/\sigma_t(K,\tau) = e^{-\eta_t\tau}(m_t dt + w_t dZ_t), \tag{12}$$

where v_t denotes the time-t instantaneous variance of the asset log-returns. For the implied volatility, m_t denotes its average drift, w_t its volatility, and the term $e^{-\eta_t\tau}$ ensures that the implied volatility is decreasingly volatile as τ increases. m_t, w_t and η_t are stochastic processes independent of the implied volatilities, and independent of K and τ. Wiener processes W_t and Z_t are correlated through stochastic process ρ_t. Carr and Wu (2016) propose that in order to prevent dynamic arbitrage, it is required that as a function of relative strike $k = ln(K/S_t)$ and τ, the implied variance surface $\sigma_t^2(k,\tau)$ must satisfy the following quadratic equation:

$$\begin{aligned} &1/4e^{-2\eta_t\tau}w_t^2\tau^2\sigma_t^4 + \left(1 - 2e^{-\eta_t\tau}m_t\tau - e^{-\eta_t\tau}w_t\rho_t\sqrt{v_t}\tau\right)\sigma_t^2 \\ &- \left(v_t + 2e^{-\eta_t\tau}w_t\rho_t\sqrt{v_t}k + e^{-2\eta_t\tau}w_t^2k^2\right) = 0. \end{aligned} \tag{13}$$

Equation (13) shows that the constraint does not depend on the dynamics between the processes $(v_t, m_t, w_t, \eta_t, \rho_t)$; these are left unspecified. Therefore, the implied volatility surface on a given day t can be fitted by treating the levels of these processes as parameters.

Consequently, the model-implied volatility is given by $\sigma_{CW}^2(\boldsymbol{\theta}_t, k, \tau)$, with $\boldsymbol{\theta}_t = (v_t, m_t, w_t, \eta_t, \rho_t)$. For a cross-section of n options on day t with Black-Scholes implied volatilities $\sigma_{i,t}$, relative

strike $k_{i,t}$ and times to maturity $\tau_{i,t}$, $\boldsymbol{\theta}_t$ is estimated by minimizing the next equation through nonlinear least squares:[2]

$$\hat{\boldsymbol{\theta}}_t = \arg\min_{\boldsymbol{\theta}_t} \sum_{i=1}^{n} \left[1/4 e^{-2\eta_t \tau_{i,t}} w_t^2 \tau_{i,t}^2 \sigma_{i,t}^4 + \left(1 - 2e^{-\eta_t \tau_{i,t}} m_t \tau_{i,t} - e^{-\eta_t \tau_{i,t}} w_t \rho_t \sqrt{v_t} \tau_{i,t}\right) \sigma_{i,t}^2 - \left(v_t + 2e^{-\eta_t \tau_{i,t}} w_t \rho_t \sqrt{v_t} k_{i,t} + e^{-2\eta_t \tau_{i,t}} w_t^2 k_{i,t}^2\right)\right]^2. \tag{14}$$

Using the estimated parameters $\hat{\boldsymbol{\theta}}_t$ as input, $\sigma_{CW}^2(\hat{\boldsymbol{\theta}}_t, k, \tau)$ is given by the value that satisfies equation (13).

3 Nonparametric Correction of Model-Implied Estimate

A fair amount of research has been conducted showing the theoretical advantages of a nonparametric approach that is guided by a parametric pilot estimate, for example by Glad (1998), Y. Fan and Ullah (1999), and J. Fan et al. (2009). This inspired Almeida et al. (2022) to develop a procedure implementing this type of semi-parametric approach as a means to predict implied volatility surfaces. In adopting their proposed two-step method, I consider a nonparametric correction of the fitted parametric option pricing models.

First, given a cross-sectional set of $i = 1, 2, ..., n$ options on a particular day t, I fit the implied volatility surface $\sigma(m_{i,t}, \tau_{i,t})$ using one of the four parametric models mentioned earlier, obtaining $\hat{\sigma}_p(m_{i,t}, \tau_{i,t})$. Then, the model-implied pricing errors are given by:

$$\hat{\epsilon}_p(m_{i,t}, \tau_{i,t}) = \sigma(m_{i,t}, \tau_{i,t}) - \hat{\sigma}_p(m_{i,t}, \tau_{i,t}). \tag{15}$$

Using a feedforward neural network, the pricing error surface $\epsilon_p(m, \tau)$ is approximated by the function $\hat{f}(m, \tau)$ obtained through the following minimization:

$$\min_f \frac{1}{n} \sum_{i=1}^{n} \left[\hat{\epsilon}_p(m_{i,t}, \tau_{i,t}) - f(m_{i,t}, \tau_{i,t})\right]^2. \tag{16}$$

Subsequently, the corrected implied volatility surface provided by the method proposed by Almeida et al. (2022), is computed as: $\hat{\sigma}_p(m_{i,t}, \tau_{i,t}) + \hat{f}(m_{i,t}, \tau_{i,t})$.

[2]The minimization in equation (14) is performed with the use of MATLAB function "fmincon". For further documentation, see https://nl.mathworks.com/help/optim/ug/fmincon.html.

This two-step procedure is a generalization of a direct nonparametric fit of the implied volatility surface. This is explained by the fact that the Black-Scholes model predicts constant implied volatility, which means that this model does not provide any information about the curvature of the implied volatility surface. Correcting it is thus equivalent to a direct nonparametric fitting. This provides a motivation for this method, as any model p that does provide information about the shape of the implied volatility surface, should have a pricing error surface $\epsilon_p(m, \tau)$ that is relatively flat compared to that of the Black-Scholes model. As a consequence, such a model should be easier to estimate nonparametrically.

3.1 Feedforward Neural Networks and Their Implementation

Hornik et al. (1989) have shown that any Borel measurable function can be approximated with arbitrary precision using a feedforward neural network with even a single hidden layer, arbitrary Sigmoid activation functions and a linear output layer. This is called the "universal approximation" property. In other words, (multilayer) feedforward neural networks can be seen as a class of universal approximators.

Building on this fact, Rolnick and Tegmark (2017) show that natural classes of multivariate polynomials can be approximated more efficiently using deeper neural networks. Moreover, feedforward neural networks have been researched extensively within finance because of their ability to model nonlinear relations nonparametrically, for example by Malliaris and Salchenberger (1993) and Garcia and Gençay (2000). Gu et al. (2020) show that, pertaining to prediction within the field of finance, feedforward neural networks have a good performance relative to other machine learning techniques. All the aforementioned literature suggests that feedforward neural networks are a good option for performing the nonparametric corrections described above. In the rest of this section, I describe the workings of such a network and I discuss its implementation.

The idea of a feedforward neural network is inspired by the workings of the brain and neural system. They perform a mapping of the inputs into outputs with signals flowing in one direction only, and comprise several layers. When denoting the vector containing the explanatory variables moneyness and time to maturity for option i on day t as $\mathbf{x}_{i,t} = (m_{i,t}, \tau_{i,t})' \in \mathbb{R}^2$, the neural network model $f : \mathbb{R}^2 \to \mathbb{R}$ is given by the following iterative

definition:

$$\begin{aligned} \underset{d_l \times 1}{\mathbf{z}_l} &= h(\underset{d_l \times d_{l-1}}{\mathbf{A}_{l-1}} \underset{d_{l-1} \times 1}{\mathbf{z}_{l-1}} + \underset{d_l \times 1}{\mathbf{b}_{l-1}}), \quad \text{for } l = 1, \ldots, L, \\ \underset{1 \times 1}{f(\mathbf{x}_{i,t})} &= \underset{1 \times d_L}{\mathbf{A}_L} \underset{d_L \times 1}{\mathbf{z}_L} + \underset{1 \times 1}{\mathbf{b}_L} . \end{aligned} \tag{17}$$

The first and last layers are called the input and output layer, respectively. In between these is an arbitrary number of hidden layers L, with each hidden layer containing a certain number of neurons d_l in vector $\mathbf{z}_l$. For layer $l = 1$, the proceeding layer $l-1$ is the input layer 0, with $\mathbf{z}_0 = \mathbf{x}_{i,t}$ and $d_0 = 2$. In each layer, a nonlinear activation function $h : \mathbb{R}^{d_l} \to \mathbb{R}^{d_l}$ is applied to the linear combination $\mathbf{A}_{l-1}\mathbf{z}_{l-1} + \mathbf{b}_{l-1}$ of all neurons from the previous layer, where $\mathbf{A}_{l-1}$ is the transformation matrix and $\mathbf{b}_{l-1}$ the intercept vector. The parameters $\{\mathbf{A}_l, \mathbf{b}_l\}_{l=0}^{L}$ of this linear combination are estimated when training the network on the data.

Several decisions have to be made regarding the implementation of the feedforward neural network - such as which activation function is to be used, the number of hidden layers called the network depth, as well as the number of neurons in each layer. Following Almeida et al. (2022) and Gu et al. (2020), I use a set of five different network structures with the network depth ranging from one to five. The amount of neurons is decreasing in every layer, following the geometric pyramid rule, as proposed by Masters (1993). This allows for the assessment of the tradeoff of network depth. Concretely, the neural networks one to five (NN1-NN5) are composed of $L = 1, 2, 3, 4, 5$ hidden layers, respectively. When present in a particular network, the hidden layers contain $d_i = 32, 16, 8, 4, 2$ neurons for layer $i = 1, 2, 3, 4, 5$, respectively.

With respect to the choice of nonlinear activation function, the sigmoid function is selected following Almeida et al. (2022). This is a commonly used option and is given by:

$$h(x) = \frac{1}{1 + e^{-x}}. \tag{18}$$

The neural network parameters are estimated by performing the minimization in equation (16). This is done using the Levenberg-Marquardt algorithm, which makes use of both the method of gradient descent and the Gauss-Newton method. In my choice of algorithm, I deviate from Almeida et al. (2022), as the Levenberg-Marquardt algorithm exhibited superior performance over the scaled conjugate gradient algorithm of Møller (1993). The results produced by the use of the latter are omitted for the sake of brevity and to avoid redundancy.

The results from this procedure should be seen as a "benchmark" for this two-step correction, as it is likely possible to iterate over the different choices for the options described above and obtain superior results in terms of performance.

4 S&P 500 Option Data

Following in the footsteps of Almeida et al. (2022), I consider data on European-style S&P 500 index options that were traded at the Chicago Board Options Exchange (CBOE) between January 4, 2016, and June 28, 2019. Almeida et al. (2022) consider this data specifically, on account of the fact that this type of option generally has the highest trade volume and healthy liquidity. The data is obtained from OptionMetrics and is preprocessed in the following manner. Every observation that violates the standard non-arbitrage conditions is deleted, as well as every observation that either has a price lower than \$0.125, or that has zero volume. Additionally, I compute the option closing price for each remaining observation as the average of the best closing bid price and closing ask price. I determine the time to maturity for every observation - with consideration of the question whether an option is settled at the market open or market close - and I compute the dividend yield through the put-call parity for every day and available maturity, using the pair of put and call options closest to at-the-money. I opt for this approach, as the dividend yield data provided by OptionMetrics is of poor quality. When no such at-the-money pair is available, I drop the corresponding observations from the dataset. Data on the underlying asset, the S&P 500 index, is obtained from Bloomberg, and the 3-month Treasury bill rate serves as a proxy for the risk-free rate. The latter is obtained from the Federal Reserve Economic Data (FRED) database of the Federal Reserve Bank of St. Louis. This allows for the extraction of the observed implied volatilities corresponding to each observation by means of inverting the Black-Scholes formula.

I consider options whose moneyness, as defined in Section 2.2, is contained in the interval $[0.80, 1.60]$, and that have a time to maturity between 20 and 240 calendar days. As in-the-money options, compared to at-the-money or out-of-the-money options, are very infrequently traded, their prices are unreliable (Aït-Sahalia & Lo, 1998). For that reason, it is common practice to exclude them from the dataset. More motivation for this custom is grounded on

the fact that in-the-money options can, in theory, be ignored without any loss of information because of the put-call-parity.

As the observed implied volatility surface is not flat across moneyness and maturity, as discussed in Section 2.1, the observations are ordered in several different categories. Regarding time to maturity, the options are either labeled short-term (20 to 60 days until expiration) or long-term (61 to 240 days until expiration). The options have also been ordered with respect to moneyness, with each observation falling into one of five categories defined by a certain moneyness-interval: deep out-of-the-money calls (DOTMC) with moneyness $m_{i,t} \in [0.80, 0.90)$, out-of-the-money calls (OTMC) with $m_{i,t} \in [0.90, 0.97)$, at-the-money options (ATM) with $m_{i,t} \in [0.97, 1.03)$, out-of-the-money puts (OTMP) with $m_{i,t} \in [1.03, 1.10)$, and deep out-of-the-money puts (DOTMP) with $m_{i,t} \in [1.10, 1.60]$. The data preparation leads to a final sample containing 1,196,437 observations. The option chain on the average trading day consists of 1363 options.

This is a slightly smaller dataset than the one used by Almeida et al. (2022), due to the difference in data preparation. An example is my decision to drop observations where the dividend yield could not be extracted, whereas the aforementioned paper uses OptionMetrics dividend yield data. This causes the results of the performed exercises to differ. However, the interpretation of the results remains the same, as the extra observations in the dataset of Almeida et al. (2022) are scattered across the whole spectrum of moneyness and time to maturity.

Table 4.1 presents summary statistics of the option data used for my research. The number of options in each category is listed, as well as their average implied volatility and standard deviation. Options that qualify for the short-term category make up 67.59% of the dataset, while categories DOTMC and OTMC contain the fewest observations. The statistics provide evidence for the misspecification of the Black-Scholes model. As expected, the volatility smile is present in the data, represented by the implied volatilities decreasing over moneyness from DOTMC to OTMC, after which they increase again. This characteristic of the data is most apparent for the short-term options. Variances of the implied volatilities are smallest for DOTMC, and largest for DOTMP, regardless of their category concerning time to maturity.

Table 4.1: Summary statistics of S&P 500 implied volatility data.

	Number		**Mean IV**		**Std. dev.**	
Time to maturity	Short	Long	Short	Long	Short	Long
Moneyness						
$[0.80, 0.90)$	8,617	13,643	16.23	12.45	2.92	2.40
$[0.90, 0.97)$	127,659	65,828	11.06	10.86	3.18	2.70
$[0.97, 1.03)$	234,293	78,094	11.62	13.10	3.93	3.29
$[1.03, 1.10)$	209,073	73,511	17.51	17.15	3.80	3.05
$[1.10, 1.60]$	229,070	156,649	29.43	25.98	8.28	5.46
Total	808,712	387,725	18.15	18.67	9.26	7.58

Note. Table 4.1 presents statistics summarizing the implied volatility data of the S&P 500 index options. The sample is taken over the period from January 4, 2016, to June 28, 2019. The options have been grouped by moneyness:= $(S_t/K_{i,t})$ and time to maturity (short-term and long-term). The columns show, respectively, the number of options, the average implied volatility over the options in %, and the standard deviation of the implied volatilities in %.

5 Prediction in the Option Cross-Section Using Individual Models

5.1 Implementation

In this section, I compare the performance of the individual models mentioned in Section 2, regarding the prediction of implied volatility in the option cross-section. The models are estimated on a daily basis, which, in theory, is inconsistent with structural models like Black and Scholes (1973) and Heston (1993), who assume that the model parameters are constant over time. In practice, however, these models are implemented in a daily manner.

In order to reproduce the results of Almeida et al. (2022) in broad lines, the cross-sectional prediction exercise is performed in the same way. This means that the exercise is split into

two parts. For the first part, for each day t, the option data is split into training data and test data, based on the strike prices. All options with a strike price divisible by 10 are used as training data, while options for which this condition does not hold are branded test data. Splitting the data this way, ensures that both the training set and test set contain options over the whole range of moneyness. Approximately 60% of my dataset is classified as training data by this procedure. The exercise simulates the scenario in which an investor observes certain option prices on day t, and, using these observations, tries to correctly predict the prices of unobserved options on the same day t. It is a same-day interpolation pricing exercise, where the models are tested on their ability to capture the implied volatility surface.

The second exercise aims to analyze the models' ability to learn from observed option data on day t, and then predict option prices on day $t+h$, with $h \in \{1, 5, 21\}$ indicating the number of business days predicted into the future. For each day t, the models are trained on all available option data and tested on all the observations of day $t+h$. The pitfall here is that models might overfit when trying to use all information available in the training set.

As the goal is to analyze the out-of-sample performance of the models in predicting the implied volatility, the error metric is ideally expressed in terms of implied volatility as well. Hence the roots of the mean squared errors of the implied volatility are used to compare the prediction errors, denoted by "IVRMSE".

Each parametric model described in Section 2 is first tested individually. Then, nonparametric corrections are performed as described in Section 3. The performances are compared between models, both before and after the different corrections are performed by the array of neural networks. These corrections are based on the options' moneyness and time to maturity data. The comparisons provide an indication of neural networks' ability to correct the parametric models. In addition, I look at the tradeoff of network depth.

5.2 Empirical Results

Table 5.1 shows that for every exercise, the Heston model has the best performance amongst the parametric models. Moreover, any of the five neural network structures correcting any of the four parametric models, increases the performance in terms of the IVRMSE for all four prediction exercises. As expected, the decrease in IVRMSE is most prominent for the

corrected Black-Scholes predictions. Generally, the larger the existing prediction error, the more significant the improvement. As a result, the IVRMSEs of the corrected predictions are of the same magnitude.

Table 5.1: Results of the cross-sectional prediction exercises.

	$\sim$	**NN1**	**NN2**	**NN3**	**NN4**	**NN5**	$\sim$	**NN1**	**NN2**	**NN3**	**NN4**	**NN5**
	Panel A: Same-day						Panel B: 1 day ahead					
BS	7.49	0.31	0.26	0.25	0.30	0.76	8.13	0.96	0.89	0.91	0.88	0.91
AHBS	1.32	0.24	0.20	**0.18**	0.19	0.20	1.74	0.85	0.81	0.79	0.79	0.83
Heston	0.88	0.27	0.21	0.19	0.19	0.20	1.23	0.80	0.76	0.76	**0.74**	0.75
CW	1.55	0.24	0.20	0.19	0.19	0.20	1.86	0.84	0.81	0.80	0.81	0.82
	Panel C: 5 days ahead						Panel D: 21 days ahead					
BS	8.23	1.68	1.63	1.61	1.63	1.63	8.44	2.64	2.48	2.47	2.48	2.46
AHBS	2.23	1.58	1.53	1.52	1.53	1.55	2.90	2.46	2.35	2.32	2.32	2.36
Heston	1.76	1.51	1.47	1.47	1.47	**1.46**	2.45	2.35	2.29	2.28	**2.26**	2.26
CW	2.31	1.55	1.50	1.49	1.51	1.51	2.96	2.42	2.34	2.30	2.30	2.32

Note. Table 5.1 shows the IVRMSE in % of the described models for the cross-sectional prediction exercises. The different panels present the same-day and 1, 5 and 21-day ahead exercises, and the columns show the neural network structure used for the correction of the different parametric models as indicated by the rows - "$\sim$" corresponding to no correction. The bold numbers indicate which model performed best in that exercise. The sample ranges from January 4, 2016, to June 28, 2019.

The improvement of the neural network correction over the parametric models decreases with the forecast horizon. For the ad-hoc Black-Scholes model for example, the improvement of the predictions by NN3 is 633%, 120%, 47% and 25% for Panel A, B, C and D, respectively. This is explained by the fact that the implied volatility surface changes as time passes - meaning that the neural networks are less capable of correctly predicting it based on current data.

Furthermore, the table shows that correcting a parametric model that already incorporates the features of the implied volatility surface to some extent, prevails over directly fitting the surface using neural networks. This is borne out by the fact that any Black-Scholes cor-

rection is outperformed by the corrections of each of the other three models. This result is robust for each neural network and each forecast horizon. Taken in conjunction with the given that correcting the constant Black-Scholes predictions is equivalent to directly fitting the neural network to the implied volatility surface, this proves the statement.

Gu et al. (2020) find that when forecasting stock prices, the peak ability of neural networks is achieved at a network depth of three hidden layers. When more layers are added, the performance usually deteriorates. This paper replicates this result in the option pricing setting. Table 5.1 shows that for almost every parametric model and panel, the best performing method is amongst the corrections applied by the neural networks with three or four hidden layers. The differences in IVRMSEs between these two options are minimal in the majority of the cases. In general, neural networks with only one or two hidden layers cannot properly capture the nonlinear dynamics of the implied volatility surface, whereas neural networks with five hidden layers tend to overfit. Both these instances are naturally detrimental to the performance.

6 Other Nonparametric Corrections

In general, little research has been conducted as to what the most appropriate nonparametric or semi-parametric method is for fitting and predicting the implied volatility surface. Almeida et al. (2022) opt to focus solely on feedforward neural networks for nonparametric corrections. As is confirmed in Section 5, these corrections outperform the parametric models as well as a neural network fitted directly to the implied volatility surface. This begs the question whether other machine learning techniques can perform equally well, or even better, in fitting and predicting the implied volatility surface - directly or by correcting parametric option pricing models. Although existing literature bears out that a parametrically guided nonparametric approach has an advantage over a direct nonparametric model, so far no research has been conducted comparing these techniques in the setting of the implied volatility surface. In this section, I test and analyze the performance of two other nonparametric methods in predicting the option cross-section.

In order to compare these performances to those of the parametric models and the neural

network corrections, I make use of the model confidence set (MCS) introduced by Hansen et al. (2011). They define the MCS as a subset of models $\widehat{\mathcal{M}}^*$, that has been composed in such a manner that it contains the best model(s) from the original set $\mathcal{M}^0$ comprising all models, with a given level of confidence. It is a concept similar to the confidence interval of a parameter, except constructed for models.

The MCS procedure repeatedly uses an equivalence test followed by an elimination rule. Initially setting $\mathcal{M} = \mathcal{M}^0$, it compares all models in this set by means of the equivalence test using a user-identified criterion. If the equivalence test is rejected, there is a significant difference between the quality of at least two models in $\mathcal{M}$, and the elimination rule is used to remove the poorest-performing model from it. This is repeated until the equivalence test is accepted, meaning that there is no significant difference in the specified criterion between the remaining models - i.e., they are "equally good". These models now constitute the model confidence set $\widehat{\mathcal{M}}^*$.

For the purpose of this paper, the criterion to compare the different methods predicting the implied volatility surface is their time series of IVRMSEs calculated on a daily basis.[3] Hansen et al. (2005) propose several test statistics for the equivalence test. I opt for the most commonly used alternative, defined by $T_R = max_{i,j\in\mathcal{M}}|t_{i,j}|$, where $t_{i,j}$ is the t-statistic for the difference in losses between model i and j. The distribution of T_R is non-standard and estimated using bootstrap methods. When $T_R = 0$ is rejected at an α-percent significance level, the implemented elimination rule ensures that the poor-performing model causing the largest loss difference is removed from the remaining models. The more informative the dataset is, the smaller the number of models in $\widehat{\mathcal{M}}^*$ will be, as the MCS procedure will be better able to distinguish between the different models.

I set $\alpha = 0.05$, which means that the best model(s) will be in $\widehat{\mathcal{M}}^*$ with a probability of $1 - \alpha = 95\%$, denoted by $\widehat{\mathcal{M}}^*_{95\%}$. The MCS procedure also provides p-values for all models in $\mathcal{M}^0$. The p-value $\hat{p}_i$ corresponding to model $i \in \mathcal{M}^0$ represents the threshold at which

[3] I determine the model confidence sets and compute the corresponding p-values with the use of the Oxford MFE Toolbox provided by Kevin Sheppard. Following Hansen et al. (2011), I set the block length equal to 2. The number of bootstrap replications is equal to 1,000,000. For further documentation, see https://www.kevinsheppard.com/code/matlab/mfe-toolbox/.

$i \in \widehat{\mathcal{M}}^*_{1-\alpha}$ if and only if $\hat{p}_i \geq \alpha$. The smaller the p-value of a certain model, the less likely it is to be (one of) the best model(s) in $\mathcal{M}^0$.

6.1 Random Forest

A well-known method within the machine learning literature is the decision tree. A decision tree is a nonparametric supervised learning algorithm suited for both classification and regression problems. Starting from the root node, it repeatedly splits the training data into smaller subsets at each decision node, based on a splitting criterion for one or more of the input variables. This results in a tree with leaf nodes that indicate a class or numerical output. This way, a decision tree is grown that is able to provide a classification or numerical prediction for the test data.

The popular random forest (RF) algorithm proposed by Breiman (2001) is a procedure combining predictions of several decision trees in order to get more accurate results. By using multiple trees, the random forest is less prone to overfit the data. In the case of a classification problem, the output of the random forest is the class that is selected by the largest number of trees. When used in a regression context, the random forest returns the average of the individual decision trees. The latter is the case with the correction of the parametric models.

In order to get an indication of the ability of a random forest to correct the parametric implied volatility predictions, I implement the method with its default parameters in the cross-sectional prediction exercise.[4] The algorithm randomly samples a fraction of the training data, on which it trains a regression tree. The default value of this fraction is equal to one, meaning that each tree draws a bootstrap sample from the input data that is the same size as the input dataset. The sampling is done with replacement, which means that each tree is grown using the same number but not the same set of observations. Just like the feed-forward neural network, the random forest (and thus each decision tree) uses the available cross-sectional moneyness and time to maturity data of the options in the training sample

[4]The random forests are implemented with the use of MATLAB function "Treebagger". All parameters are set equal to their MATLAB default values. For further documentation, see https://nl.mathworks.com/help/stats/treebagger.html.

as predictors, and the corresponding errors of the parametric model as the response variable. The random forest algorithm grows 500 trees, where each split within a tree is based on a single predictor variable. The forest can now provide a predicted error for each option in the test set by averaging the 500 predictions of the individual trees that result when these are presented with the moneyness and time to maturity values of said options.

6.2 Support Vector Machine

Another established supervised learning algorithm is the support vector machine (SVM). Like the random forest, it is suited for classification as well as regression purposes. In a regression setting, the method is called support vector regression. It tries to fit a hyperplane that is as flat as possible to higher dimensional training data, in such a manner that the training points lie within a distance of $\epsilon > 0$ of the plane. This ϵ is called the margin. The data points closest to the hyperplane on either side are called support vectors.

As I want to implement the support vector machine to obtain an indication of its performance when correcting the parametric models, I use its default parameters.[5] This means that I opt for a linear kernel resulting in a straight hyperplane, and set $\epsilon = iqr(Y)/13.49$, where $iqr(Y)$ is the interquartile range of the response variable, i.e., the difference between its 75^{th} and 25^{th} percentiles.[6] In my three-dimensional case, the predictor variables are once again the moneyness and time to maturity of each option in the training set, and the errors of the parametric models constitute the response variable.

6.3 Empirical Results

I repeat the two cross-sectional prediction exercises of Section 5. This time, however, the corrections of the four parametric models are obtained using the random forest and support vector machine algorithms. Table 6.1 shows the results of these two exercises, comparing the new corrections with the parametric models as well as their neural network corrections

[5]The support vector machines are fitted using MATLAB function "fitrsvm" and its default parameter values. For further documentation, see https://nl.mathworks.com/help/stats/fitrsvm.html.

[6]This value for ϵ is an estimate of one-tenth of the response variable's standard deviation.

using three and four hidden layers. The bold underlined numbers denoted with an asterisk indicate which models are included in $\widehat{\mathcal{M}}^*_{95\%}$ for that exercise.

It is evident from Table 6.1 that, like with the neural network corrections, for every exercise, the random forest correcting any of the four parametric models improves the performance in terms of the IVRMSE. As expected, the decrease in IVRMSE is again the largest for the corrected Black-Scholes predictions. Also similar to the results of the neural network corrections is the fact that the improvement of the random forest corrections over the parametric models decreases with the forecast horizon. For the ad-hoc Black-Scholes model, the improvement is 180%, 77%, 37% and 21% for the same-day, and 1-, 5- and 21-day ahead exercises, respectively.

The results of the support vector machine corrections, however, are less exciting. Even though the corrections of the Black-Scholes model result in significant improvements in terms of the IVRMSEs, they are nowhere near those of the neural networks or the random forests. Furthermore, when looking at the corrections for the other three models, the support vector machine struggles to make an impact, often decreasing the IVRMSE only slightly and, in some cases, even increasing it.

Comparing the same-day results of the two new nonparametric corrections to those of the four parametric models and different deployed neural networks, shows that the latter dominate in terms of performance. All MCS p-values of the parametric models, random forest and support vector machine corrections are equal to zero. This means that none of them will be included in $\widehat{\mathcal{M}}^*_{1-\alpha}$, regardless of the significance level. At an $\alpha = 5\%$ level, only the ad-hoc Black-Scholes and Carr and Wu corrections by the neural network comprising three hidden layers are included.

Table 6.1: MCS for parametric models and random forest, support vector machine and neural network corrections predicting in the option cross-section.

	Same day		**1 day ahead**		**5 days ahead**		**21 days ahead**	
	IVRMSE	p_{MCS}	IVRMSE	p_{MCS}	IVRMSE	p_{MCS}	IVRMSE	p_{MCS}
BS	7.487	0	8.126	0	8.226	0	8.439	0
AHBS	1.318	0	1.742	0	2.234	0	2.899	0
Heston	0.876	0	1.228	0	1.764	0	2.446	0
CW	1.554	0	1.863	0	2.310	0	2.956	0
BS-RF	0.901	0	1.400	0	1.924	0	2.619	0
AHBS-RF	0.470	0	0.984	0	1.631	0	2.389	0
Heston-RF	0.265	0	0.780	0	1.472	**0.4631***	2.258	**1***
CW-RF	0.402	0	0.921	0	1.573	0	2.35	0.0002
BS-SVM	2.870	0	2.714	0	3.070	0	3.614	0
AHBS-SVM	1.308	0	1.775	0	2.266	0	2.928	0
Heston-SVM	0.889	0	1.228	0	1.768	0	2.451	0
CW-SVM	1.390	0	1.827	0	2.293	0	2.959	0
BS-NN3	0.249	0	0.913	0	1.606	0	2.468	0
AHBS-NN3	0.181	**1***	0.794	0	1.517	0.0002	2.322	0.0053
Heston-NN3	0.191	0.0077	0.759	0.0246	1.468	**0.5637***	2.278	0.0350
CW-NN3	0.186	**0.0768***	0.800	0	1.494	0.0182	2.304	0.0350
BS-NN4	0.296	0	0.880	0	1.629	2e-06	2.482	0
AHBS-NN4	0.189	0.0454	0.795	0	1.531	2e-06	2.322	0.0016
Heston-NN4	0.195	0.0005	0.745	**1***	1.466	**1***	2.258	**0.9992***
CW-NN4	0.189	0.0454	0.806	0	1.505	0.0027	2.301	0.0332

Note. Table 6.1 shows the IVRMSE in % of each model for the cross-sectional prediction exercises, as well as the MCS p-values. The columns indicate the same-day and 1, 5 and 21-day ahead exercises, with the rows referring to the parametric option pricing model and its correction, if applicable. The bold numbers indicate which models are included in $\widehat{\mathcal{M}}^*_{95\%}$. The sample ranges from January 4, 2016, to June 28, 2019.

The longer the forecast horizon becomes, the closer the performances of the random forests come to those of the neural networks. When predicting one day ahead, the only model included in the MCS is the neural network Heston correction using four hidden layers (Heston-NN4). Five days ahead, both neural network Heston corrections are included, as well as the random forest correction of the Heston model. The latter decreases the IVRMSE almost to the level of the Heston-NN3 model. Moreover, for the exercise predicting 21 days ahead, the Heston-RF correction is even the best-performing amongst all models in $\mathcal{M}^0$, with a minimal advantage in terms of IVRMSE over Heston-NN4, which is included in $\widehat{\mathcal{M}}^*_{95\%}$ as well.

These results indicate that as the implied volatility surface changes over time, the random forest correction models better identify the predictive signal in the data for longer horizons. The neural network corrections, on the other hand, suffer more from overfitting the longer the forecast horizon becomes.

7 Combined Forecasts

Almeida et al. (2022) state that incorporating neural networks as a correction of parametric models can significantly improve performance when predicting the implied volatility surface. This is shown again in previous sections. It is interesting to research what other ways neural networks are able to perform this task. Clemen (1989) reviews the literature regarding the combination of forecasts, and concludes that combining multiple individual forecasts can lead to significant improvements in forecast accuracy. This finding is robust over a wide range of forecasting targets. In this section, I analyze the performance of several methods of combining forecasts in predicting the implied volatility surface.

7.1 Simple and Weighted Average

Makridakis et al. (1982) show that a simple average of forecasts from six different models performs very well for multiple forecast targets, and generally better than each of the individual models. It even outperforms a weighted average of the forecasts based on the sample covariance matrix of fitting errors. This robustness invites this simple average to be used as

a benchmark for combining forecasts. I create this benchmark by taking, for each option in the dataset, the simple average (SA) of the four predictions of the parametric models.

As Section 5 has shown that the performance of the Black-Scholes model in terms of the IVRMSE is generally significantly worse than that of the other three models, it is to be expected that this will harm the performance of the simple average as well. Stock and Watson (1998) look at a method of forecast pooling that accounts for the difference in the performance of the individual models. They construct the weights in such a manner that they are inversely proportional to the out-of-sample mean squared error of the respective forecast, thus realizing a higher weight for a model that does better. In this paper's setting, incorporating a similar method would deal with at least part of the problem of the poor performance of the Black-Scholes model. Therefore, for the cross-sectional exercise, I include and analyze a weighted average model (WA) where the weights of the four models are inversely proportional to their in-sample IVRMSEs.

7.2 Neural Network Combinations

Neural networks are able to combine forecasts in an efficient manner, often improving over the individual forecasts. Donaldson and Kamstra (1996) combine forecasts of several countries' stock market volatility using neural networks. They find that these combined forecasts generally dominate traditional linear combinations of forecasts. They attribute this success to the ability of neural networks to account for the often complex and hidden nonlinear relations between the predictor variables (in their case the individual forecasts) and the target variable. I investigate the extent to which neural network combinations can outperform all previously mentioned models, by including four different methods relying on a neural network consisting of again three hidden layers. The first neural network (4M) makes predictions based on the predictor variables that are composed of the four parametric models' forecasts. It is trained on the individual predictions for the options within the training set, with the respective implied volatilities comprising the target variable. Including the cross-sectional variables - moneyness and time to maturity - allows the neural network to differentiate between options based on these values. For example, the predictions of the Heston model might, on average, be better for options with a longer time to maturity. Three more neural network methods are

implemented, similar to 4M, but including moneyness as an explanatory variable besides the individual forecasts (4M-Moneyness), including time to maturity (4M-TtM), and including both (4M-Moneyness-TtM). Section 7.3 compares these combined forecasts with those of the simple and weighted averages, the individual parametric forecasts, and again the neural network corrections with a network depth of three and four hidden layers.

7.3 Empirical Results

The results in Table 7.1 help answer the question of whether combining forecasts can benefit the prediction of the implied volatility surface. For every cross-sectional prediction exercise, the simple average is only able to outperform Black-Scholes amongst the parametric models. By accounting for the sizeable in-sample prediction error of the Black-Scholes model caused by its constant volatility prediction - thus lowering its weight while assigning the largest weight to the parametric prediction with the lowest in-sample IVRMSE - the weighted average method is able to substantially lower the IVRMSE further. It outperforms three of the four parametric models for every prediction horizon, with only the Heston model having a lower IVRMSE. Neither of the two averaging methods comes close to the performance of the neural network corrections or combinations.

The neural networks combining the individual predictions perform very well in the same-day exercise, both with and without including moneyness and time to maturity as explanatory variables. 4M-TtM performs best, with 4M-Moneyness-TtM coming in a close second. These two models are the only ones included in the model confidence set $\widehat{\mathcal{M}}^{*}_{95\%}$, with the MCS p-values of all other models equal to zero. This means that the best model(s) are amongst these two, with very high certainty. The results show that including the moneyness and time to maturity of the options as explanatory variables besides the individual forecasts, helps the neural network predict their implied volatility more accurately.

Table 7.1: MCS for parametric models, simple and weighted averages, and neural network combinations and corrections predicting in the option cross-section.

	Same day		**1 day ahead**		**5 days ahead**		**21 days ahead**	
	IVRMSE	p_{MCS}	IVRMSE	p_{MCS}	IVRMSE	p_{MCS}	IVRMSE	p_{MCS}
BS	7.487	0	8.126	0	8.226	0	8.439	0
AHBS	1.318	0	1.742	0	2.234	0	2.899	0
Heston	0.876	0	1.228	0	1.764	0	2.446	0
CW	1.554	0	1.863	0	2.310	0	2.956	0
SA	2.225	0	2.498	0	2.831	0	3.362	0
WA	0.936	0	1.317	0	1.838	0	2.521	0
4M	0.293	0	0.813	2e-06	1.586	1.4e-05	2.423	1e-05
4M-Moneyness	0.207	0	0.766	0.0303	1.539	0.0041	2.345	0.0368
4M-TtM	0.144	**1***	0.805	0.0015	1.564	1.4e-05	2.425	1e-05
4M-Moneyness-TtM	0.145	**0.7016***	0.793	0.0303	1.519	0.0009	2.353	0.0031
BS-NN3	0.249	0	0.913	2e-06	1.606	1e-06	2.468	0
AHBS-NN3	0.181	0	0.794	9.9e-05	1.517	0.0005	2.322	0.0070
Heston-NN3	0.191	0	0.759	0.0303	1.468	**0.5637***	2.278	0.0368
CW-NN3	0.186	0	0.800	2e-06	1.494	0.0101	2.304	0.0368
BS-NN4	0.296	0	0.880	1e-06	1.629	8e-06	2.482	0
AHBS-NN4	0.189	0	0.795	2e-06	1.531	1.4e-05	2.322	0.0031
Heston-NN4	0.195	0	0.745	**1***	1.466	**1***	2.258	**1***
CW-NN4	0.189	0	0.806	2e-06	1.505	0.0041	2.301	0.0368

Note. Table 7.1 shows the IVRMSE in % of each model - as indicated by the rows - for the cross-sectional prediction exercises, as well as the MCS p-values. The columns indicate the same-day and 1, 5 and 21-day ahead exercises. The bold numbers indicate which models are included in $\widehat{\mathcal{M}}^{*}_{95\%}$. The sample ranges from January 4, 2016, to June 28, 2019.

This result is replicated in the exercise where the models make predictions about the future. Including one or both of the cross-sectional variables always leads to a decrease of the IVRMSE relative to 4M, except for when predicting 21 days ahead, where 4M-TtM has a

slightly higher IVRMSE than 4M. For none of the three forecast horizons, the neural network combinations are included in the MCS. They consistently outperform the four parametric models and the two methods of averaging them, attested by their MCS p-values. However, they are dominated by the neural network corrections of the Heston model - Heston-NN3 and Heston-NN4. The latter is the best-performing model for all three horizons, being the sole model in the MCS when predicting one day and 21 days ahead. For the predictions made five days into the future, Heston-NN3 is included in $\widehat{\mathcal{M}}^*_{95\%}$ as well.

The neural network combinations perform better than the neural network corrections when predicting within the same day, whereas this is the other way around for predicting the implied volatility of future days. This indicates that the neural network is better able to extract information about the changing implied volatility surface as time passes, when correcting a parametric model. The optimal weights combining the individual forecasts change to a larger extent than the errors of the parametric models, as the implied volatility surface changes over time.

8 Prediction in the Option Panel

In the exercises discussed thus far, the four parametric models are estimated on a daily basis. As mentioned before, this is in theory inconsistent with structural models like Black and Scholes (1973) and Heston (1993), who assume that the model parameters are constant over time. Therefore, I conduct a prediction exercise in the option panel (the sequence of implied volatility surfaces), following a similar approach to Andersen et al. (2015). Almeida et al. (2022) show that a neural network is able to correct the misspecification of the Heston model significantly in the option panel. In this section, I investigate the extent to which a neural network can achieve the same by combining individual forecasts.

Almeida et al. (2022) furthermore present evidence that it is beneficial to include time-varying covariates such as the VIX index as explanatory variables in the neural network correcting the Heston model. As machine learning techniques are able to accommodate high-dimensional data, it is interesting to consider using multiple covariates in order to combine the different forecasts. This provides an indication of whether certain models or techniques

perform better in different states of the world. For example, some models could perform better in times of high volatility. I use a feature importance measure to determine the explanatory variables' predictive value.

8.1 Implementation

In the option panel exercise, all implemented models are estimated and/or trained on the whole training sample, which comprises the first two years of my data set, i.e., the option data from January 4, 2016, until December 29, 2017. In order to reduce the computational complexity, for each day t in the training sample, every alternate maturity is removed. This results in a smaller sample size, while keeping the same range of maturities.[7] The estimated parameters are kept fixed over time. The time-varying parameters of the ad-hoc Black-Scholes and Carr and Wu models make that they are designed to fit an implied volatility surface on a given day rather than an option panel. These models are, however, included in the prediction exercise in the option panel to serve as a reference for the performance of the other models. Their parameters are assumed constant over time and estimated using the whole training panel dataset. The out-of-sample predictions in the option panel are then obtained analogously to Section 2, keeping the fixed parameters estimated in-sample and using the available moneyness, time to maturity, and relative strike data of the options in the test set.

Equation (6) in Section 2.3 shows that the Heston model contains the parameters V_t, κ, $\overline{v}$, σ_v and ρ. V_t is a state variable representing the spot variance for day t with $t = 1, 2, ..., T$, while the rest are parameters that are fixed over time. Following Almeida et al. (2022), I set V_t equal to the daily observed option-implied measure of spot volatility of Todorov (2019) that is available on the website owned by Viktor Todorov and Torben Andersen.[8] The other parameters are estimated as described in Section 2.3, except this time over the whole training panel.

Fitting the option panel by means of the neural network correction of the Heston model

[7] The performances of the models are compared with those of Almeida et al. (2022), confirming that this does not significantly change the results.

[8] https://tailindex.com/index.html.

is analogous to the method in Section 3. The approximation of the model-implied pricing errors now becomes:

$$\min_f \frac{1}{T}\sum_{t=1}^{T}\frac{1}{n_t}\sum_{j=1}^{n_t}\left[\hat{\epsilon}(t,m_{j,t},\tau_{j,t})-f(m_{j,t},\tau_{j,t})\right]^2, \tag{19}$$

with $j = 1, 2, ..., n_t$ moneyness and times to maturity for each day t, and the corrected predictions given by $\hat{\sigma}(t, m_{j,t}, \tau_{j,t}) + \hat{f}(m_{j,t}, \tau_{j,t})$.

When using the neural network to combine the predictions, the neural network is directly fitted on the option panel by the minimization:

$$\min_f \frac{1}{T}\sum_{t=1}^{T}\frac{1}{n_t}\sum_{j=1}^{n_t}\left[\sigma(t,m_{j,t},\tau_{j,t})-f(\mathbf{x}_t)\right]^2, \tag{20}$$

with $\mathbf{x}_t$ representing (a combination of) the individual forecasts, moneyness, times to maturity and time-varying covariates, and where $\{\sigma(t, m_{j,t}, \tau_{j,t}), j = 1, 2, ..., n_t\}_{t=1}^{T}$ is the observed in-sample option panel.

Given the previous results, all neural networks used in this prediction exercise contain three hidden layers. Following Almeida et al. (2022), I consider a set of seven covariates that are observed on a daily basis. As a proxy for market risk and sentiment, I use the VIX index provided by the CBOE. Bollerslev et al. (2015) develop a method to capture market jump risk. This method forms the basis for the Left Tail Volatility (LTV) and Left Tail Probability (LTP) measures of Viktor Todorov and Torben Andersen, provided on their website. The former estimates the expected volatility in returns that is caused by large negative price jumps, and the latter the probability that the S&P 500 index will drop by at least 10% in the next week. Like Almeida et al. (2022), I include macroeconomic measures and measures of uncertainty as well. `https://www.policyuncertainty.com/` provides me with a measure of economic policy uncertainty (EPU) proposed by Baker et al. (2016), and the Federal Reserve Bank of Philadelphia offers the business conditions index of Aruoba et al. (2009) (ADS). Furthermore, I include the first differences of the term spread (TMS), defined as the difference between the 10-year and 3-month Treasury rates, and the first differences of the credit spread (CRS), defined by the difference between Moody's Seasoned Baa Corporate Bond yield and the 10-year Treasury yield.

The models are evaluated on their ability to fit the option panel in two ways. The first evaluation gives an indication of the pricing errors, by means of again the average IVRMSE pooled over all days in the test sample, as well as the median IVRMSE. This second metric accounts for the possibility of more significant errors causing very large daily IVRMSEs, which drive high values of the average IVRMSE. The second way of evaluating the models focuses on their ability to reproduce the dynamics of the implied volatility surface over time. Andersen et al. (2015) state that the characteristics level, term structure, skew and skew term structure summarize these dynamics. The level characteristic is a measure of the general level of volatility and is calculated as the average implied volatility of short-term at-the-money options.

The term structure characteristic (TS) indicates the slope of the term structure curve, which shows the implied volatilities of options with the same strike price but different maturities. A positive slope indicates that market participants expect the underlying asset to become more volatile over time, and a negative slope indicates an expected decrease in volatility. TS is calculated as the difference between the average implied volatility of long- and short-dated at-the-money options.

The implied volatility skew refers to the fact that options with the same underlying asset and expiration date still have different implied volatilities, depending on the strike price. Usually, for index options, the implied volatilities of out-of-the-money puts exceed the implied volatilities of out-of-the-money calls, a sign that investors perceive the risk to the downside rather than the upside. I determine the skew characteristic by calculating the difference between the average implied volatility of short-dated out-of-the-money put and call options.

The skew term structure (skew TS) is a measure of the skew over time. Generally, the skew of short-term options is steeper than that of those with a long-term expiration. An explanation is that for short-term options, a trader knows whether an option is a downside option or not. In contrast, for long-term options this is harder to determine as it is unknown at what price the underlying asset will be trading in the future. Skew TS is determined as the difference between long- and short-dated skew, where long-dated skew is computed in the same manner as its short-term counterpart.

In order to evaluate the ability of the implemented models to capture these salient characteristics, I compute the root mean squared errors (RMSEs) over the whole sample between the aforementioned characteristics that are implied by the data, and those that are implied by the models. When calculating the RMSE, every error is squared before the averaging, so a relatively high weight is assigned to larger errors. Therefore I calculate a second metric for every model, the mean absolute error (MAE), which is more robust to bigger mistakes in predictions and generally produces more interpretable values.

8.1.1 Feature Importance

I determine the impact that the different explanatory variables have by being included in the neural networks combining the individual forecasts. To this end, I use a feature importance measure congruent with Gu et al. (2020). The importance of a specific feature is determined by including all time-varying covariates and individual forecasts as explanatory variables in the neural network, and computing the IVRMSE of the consequent predicted implied volatilities when the neural network is fitted to the option panel. The feature importance of the variable is then defined as the increase in IVRMSE that follows from setting all values of the feature equal to zero and, *ceteris paribus*, fitting the implied volatility panel again. The lower the value of a feature's importance is, the more negligible the feature itself is. Naturally, a variable with a high predictive value has a higher feature importance.

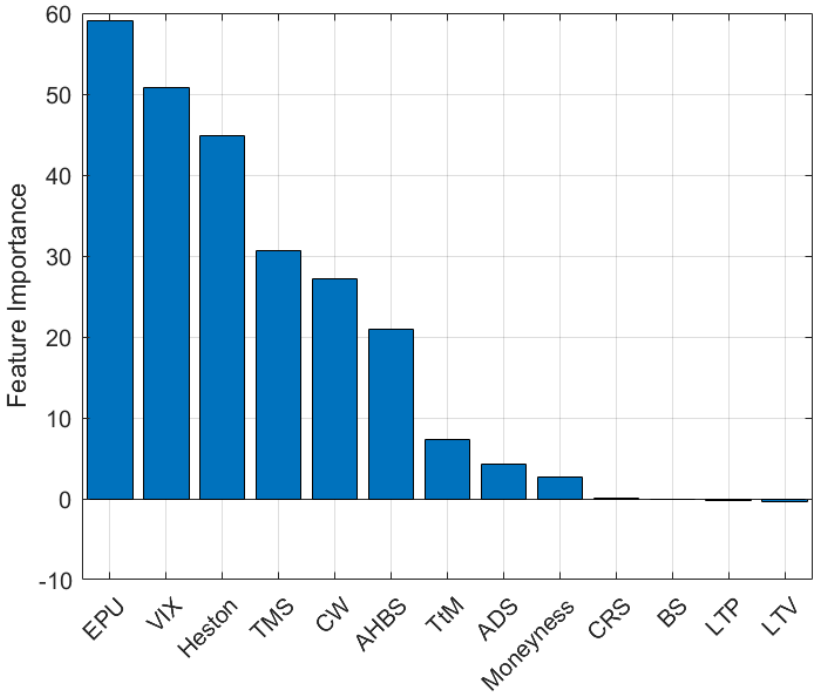

Figure 8.1: **Prediction excercise in the option panel - Feature importance.** This figure reports the importance of each feature in the option panel prediction exercise when combined by the neural network. The importance of a particular feature is defined as the increase in the out-of-sample IVRMSE (in %) that follows from setting each value of that feature equal to zero while keeping all other feature values the same. The in-sample period ranges from January 4, 2016, to December 29, 2017, and the out-of-sample period from January 2, 2018, to June 28, 2019.

Figure 8.1 shows the feature importance per variable. The economic policy uncertainty (EPU) and market risk (VIX) measures prove to be even more important than the forecasts of the best-performing parametric model in the cross-sectional exercise, the Heston model. This shows the significance of time-varying covariates in fitting the option panel. The Heston model's implied volatility predictions have the highest feature importance amongst the parametric models. All parametric models, excluding the Black-Scholes model, have a higher feature importance than the options' times to maturity and moneyness. A reason for this might be that these three parametric models' predictions already account for the moneyness and times to maturity, leaving little need for two separate measures. The feature importances of EPU, TMS and ADS indicate that measures of uncertainty and macroeconomic conditions can help with identifying the time-varying shape of the implied volatility surface.

Concerning the EPU measure, its significance is in line with L. Liu and Zhang (2015). They find that including EPU as a predictive variable significantly improves the ability of existing models to forecast stock market volatility. The high feature importance of the VIX measure is according to expectation, as it is widely regarded as the benchmark for market volatility.

It comes naturally that the feature importance of the Black-Scholes model's predictions is zero, as they are constant. The other negligible features are CRS, LTP and LTV. The last two even have a negative feature importance, meaning that the IVRMSE is lower when these features are set equal to zero. This indicates that separate measures of market jump risk do not help with predicting the shape of the option panel, given all information enclosed in the other features.

The feature importances in Figure 8.1 lead me to consider several different neural networks, besides the neural network correcting the Heston model. They are differentiated by the explanatory variables that are included when fitting the option panel. The first network only includes the individual forecasts of the parametric models (4M), the second includes the cross-sectional variables too (4M-M-TtM), and the third incorporates, besides the individual forecasts, all other features as well (4M-All). When iteratively adding the most important feature to a network already containing the individual forecasts, the lowest average IVRMSE is achieved by the network that has EPU and VIX included as explanatory variables (4M-EPU-VIX).[9]

To assess the value added by the two most important time-varying covariates, they are included in the last neural network containing the individual forecasts and cross-sectional features as well (4M-M-TtM-EPU-VIX).

8.2 Empirical Results

Table 8.1 shows the results of the prediction exercise in the option panel. The Heston model is the best amongst the parametric models in capturing the option panel's dynamics, as it has the lowest RMSE and MAE for every characteristic. In terms of IVRMSE, the simple average and weighted average perform better than the BS and CW models. Of these two averaging

[9]The results of this iterative exercise are omitted for the sake of brevity, as only the best performing network (4M-EPU-VIX) provides interesting comparisons and conclusions.

methods, only the weighted average outperforms AHBS in terms of the average IVRMSE. Both SA and WA fail to improve upon the individual Heston model's results concerning the mean and median IVRMSE. The weighted average does, however, improve the parametric models' ability to capture the salient characteristics of the implied volatility surface over time. With the exception of the skew, WA has a lower RMSE and MAE for every characteristic than the BS, AHBS and CW models, and can compete with the Heston model.

Table 8.1: Prediction in the option panel.

	IVRMSE		**Level**		**TS**		**Skew**		**Skew TS**	
	Mean	Median	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
BS	8.99	8.81	6.37	5.77	1.59	1.41	7.71	7.59	1.09	0.86
AHBS	3.53	2.43	3.84	3.01	1.24	1.05	1.35	1.08	1.40	1.19
Heston	2.24	2.04	2.71	2.11	0.94	0.75	1.22	0.90	0.80	0.62
CW	5.36	5.14	3.90	3.20	2.83	2.53	2.55	2.24	1.05	0.82
SA	3.74	3.48	3.43	2.89	0.88	0.62	2.66	2.38	0.72	0.57
WA	2.80	2.46	2.89	2.30	0.79	0.55	1.65	1.31	0.80	0.64
Heston-NN3	1.60	1.39	2.10	1.61	0.66	0.47	**1.19**	0.89	0.69	0.51
4M	1.82	1.65	1.25	0.87	1.34	1.05	1.35	1.02	1.02	0.81
4M-M-TtM	2.19	1.56	1.86	1.16	0.97	0.70	1.35	0.98	0.96	0.70
4M-M-TtM-EPU-VIX	1.77	**1.15**	1.81	0.66	**0.59**	**0.38**	1.25	**0.77**	**0.61**	**0.42**
4M-EPU-VIX	**1.41**	1.19	**0.84**	**0.52**	0.67	0.44	1.24	0.82	0.95	0.54
4M-All	4.23	3.02	3.57	1.49	1.31	0.68	3.08	1.53	1.95	1.03

Note. The first two columns of Table 8.1 show the average and median out-of-sample IVRMSE in % of the described models for the prediction exercise. The others depict the RMSE and MAE over the whole sample for the level, term structure, skew and skew term structure of the implied volatility surface (definitions provided in this section). The bold numbers indicate which model performed best. The in-sample period ranges from January 4, 2016, to December 29, 2017, and the out-of-sample period from January 2, 2018, to June 28, 2019.

The neural network correction of the Heston model dominates all parametric models as well as the simple and weighted averages in terms of both IVRMSE metrics, and in the ability to capture the surface dynamics. This confirms the conclusion of Almeida et al. (2022) that

the nonparametric corrections of models outperform the original models - generally to a large extent.

The neural network that directly fits the option panel by combining the individual forecasts (4M) does not perform as well as Heston-NN3 in terms of IVRMSE. It is significantly better at capturing the level characteristic than Heston-NN3, resulting in a lower RMSE and MAE, but it cannot replicate this for the other three characteristics. 4M does provides significant improvements in terms of the out-of-sample pricing performance compared to the parametric models. Including moneyness and times to maturity of the options as features (4M-M-TtM) does not lower the average IVRMSE, but does so for its median. This indicates larger errors on certain days, but arguably a better pricing performance in general. Moreover, it does help reduce the RMSE and MAE for the term structure, skew and skew term structure.

Adding the EPU and VIX features (4M-M-TtM-EPU-VIX) has a large positive effect on the pricing ability, with the average and median IVRMSE lower than those of 4M and 4M-M-TtM. It does the same for the ability to reproduce the characteristics, with every RMSE and MAE lower than their counterparts provided by the two aforementioned neural network combination models. This shows the added benefit of including time-varying covariates as explanatory variables in neural networks regarding pricing in the option panel, and reproducing the implied volatility surface's characteristics.

The best-performing model in terms of the average IVRMSE is 4M-EPU-VIX, whose median IVRMSE is slightly higher than that of 4M-M-TtM-EPU-VIX, indicating that 4M-EPU-VIX has fewer days with large errors, i.e., the distribution of its IVRMSEs is less right-skewed. The latter model is able to capture the dynamics very well, too, with a lower RMSE and MAE for the level characteristic than all other models. These metrics for the other characteristics are within 0.1 of those of the respective best-performing model, with the exception of the RMSE of skew TS.

Including all features in the neural network (4M-All) does not provide a significant advantage over the other neural network combination models. It cannot compete with any of them in terms of IVRMSE, while it has the highest RMSE and MAE for the level, skew and skew TS amongst this set of models. This can be explained by the bias-variance tradeoff: the

neural network overfits the data as a result of too many features being included. Nakkiran et al. (2021) find that the phenomenon of double descent is often present in deep learning exercises. They show that while increasing the model size, the performance of the model first gets worse before it improves again past a certain size. This tipping point, called the interpolation threshold, is generally reached when the number of parameters in the model roughly matches the number of sample observations. This is an indication that adding more and more features to 4M-All, might eventually increase its performance again.

Figure 8.2 compares, over time, the characteristics of the implied volatility surface that are implied by the data with those implied by model 4M-M-TtM. The level observed from the data represents the general level of volatility, thus always being positive. It has a few spikes, which can be seen to work through into the other characteristics. The term structure has a few negative outliers but is mostly positive, whereas the skew is always positive and has higher spikes corresponding to the spikes in level. The skew term structure is primarily negative. Figure 8.2 shows that the model 4M-M-TtM is able to recreate the surface dynamics quite well overall, but usually fails to capture the spikes.

To further investigate the benefit of including time-varying covariates in the fitting of the option panel, Figure 8.3 shows the same comparison, this time between the characteristics implied by the data and those by 4M-M-TtM-EPU-VIX. The time-varying covariates allow the neural network to learn the shape of the implied volatility surface as a function of the state of the economy, whereas without them, it can only learn the average shape over time. It can be seen that the in-sample fit significantly improves for all four characteristics. The same goes for the out-of-sample period, with the model better capturing the spikes. However, it sometimes does predict a spike that is not matched by the observed data, or at least not to the extent the model forecasted. This is the reason that, for level and skew, the improvement of 4M-M-TtM-EPU-VIX over 4M-M-TtM is much more significant in terms of MAE than in RMSE.

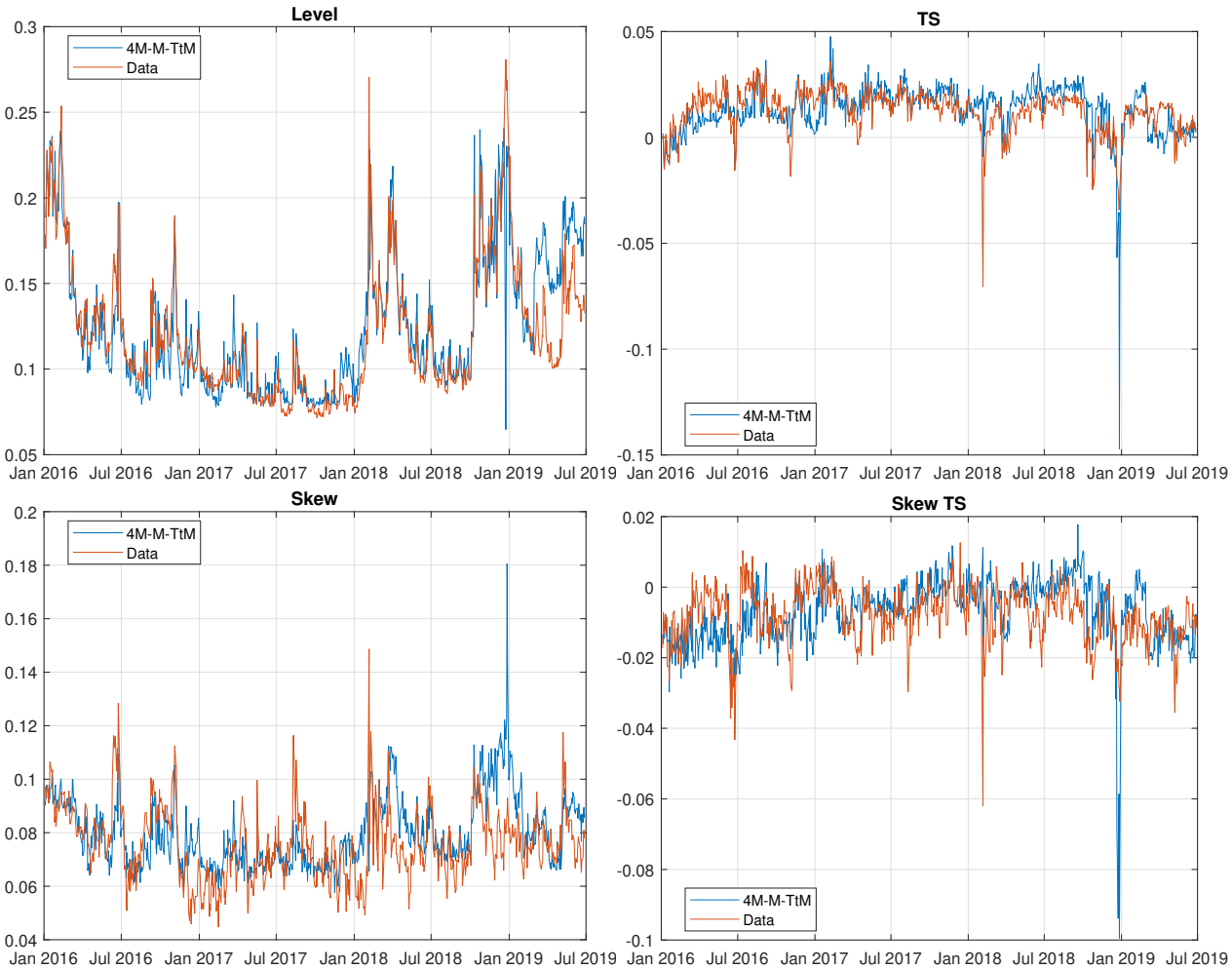

Figure 8.2: **Implied volatility surface characteristics - Models, Moneyness and Time to Maturity.** This figure plots the data-implied time series of the implied volatility surface characteristics level, term structure, skew and skew term structure (red) against their counterparts implied by model 4M-M-TtM (blue). The in-sample period ranges from January 4, 2016, to December 29, 2017, and the out-of-sample period from January 2, 2018, to June 28, 2019.

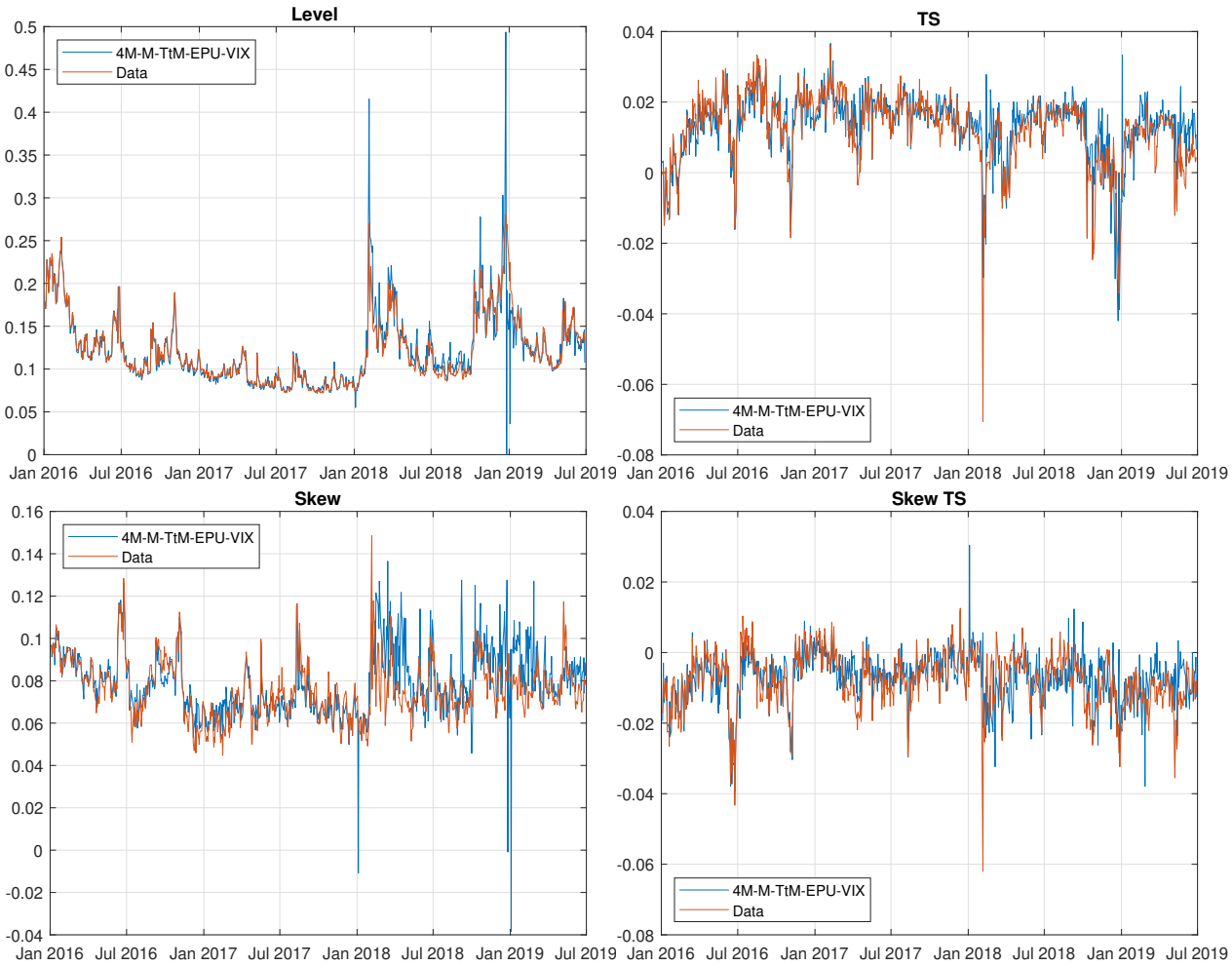

Figure 8.3: **Implied volatility surface characteristics - Models, Moneyness, Time to Maturity, EPU and VIX.** This figure plots the data-implied time series of the implied volatility surface characteristics level, term structure, skew and skew term structure (red) against their counterparts implied by model 4M-M-TtM-EPU-VIX (blue). The in-sample period ranges from January 4, 2016, to December 29, 2017, and the out-of-sample period from January 2, 2018, to June 28, 2019.

Summing up the findings provided in this section, the combination of different forecasts using neural networks is in the option panel able to outperform the individual forecasts made by structural parametric option pricing models like Black and Scholes (1973) and Heston (1993), their simple and weighted averages, and the neural network with three hidden layers correcting the Heston model, proposed by Almeida et al. (2022). Using conditioning information by including time-varying covariates as explanatory variables can further improve the fit of the option panel and the ability to reproduce its salient characteristics. It allows

the neural network to capture the time-varying shape of the implied volatility surface as a function of states of the economy, most notably the measures of volatility (VIX) and economic policy uncertainty (EPU).

9 Conclusion

In this paper, I research the extent to which machine learning techniques are able to correct or combine parametric models in order to fit the implied volatility surface. By means of a cross-sectional prediction exercise, I compare the performance of random forest and support vector machine corrections with that of the parametric models Black and Scholes (1973), ad-hoc Black-Scholes (Dumas et al., 1998), Carr and Wu (2016), and Heston (1993), as well as their neural network corrections proposed by Almeida et al. (2022). They are evaluated based on their out-of-sample performance for different forecast horizons.

I find that the support vector machine corrections do not significantly outperform the parametric models, whereas the random forest corrections do. The latter perform better the longer the forecast horizon is. Especially the random forest correction of the Heston model can compete with the neural network corrections of the parametric models.

The parametric models' simple average and inverse prediction error-weighted average are able to outperform the individual forecasts as well, but they are dominated by the individual forecasts' neural network combinations. I observe this result irrespective of whether moneyness and times to maturity are included as features in the network or not. When predicting within the same day, the neural network combinations have an even better performance than the neural network corrections of the Heston model.

With a prediction exercise in the option panel, I show that using neural networks to combine individual forecasts significantly improves the fit of the implied volatility surface over time. Including time-varying covariates as features, helps a neural network learn the shape of the surface as a function of the state of the economy. This allows for an even better fit and it enables the neural network to capture the salient characteristics of the option panel more accurately.

The results in this paper provide several implications for option pricing in practice. Pro-

fessionals who price options using parametric models to fit the implied volatility surface, can improve their estimates by using machine learning techniques to correct or combine parametric forecasts. Incorporating information about the state of the economy can help these techniques in doing so. Furthermore, combining individual forecasts using neural networks - with or without the use of time-varying covariates - is a method that can potentially deliver significant improvements in other areas as well; a result that invites further research.

References

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

Aït-Sahalia, Y., & Duarte, J. (2003). Nonparametric option pricing under shape restrictions. *Journal of Econometrics*, *116*(1-2), 9–47.

Aït-Sahalia, Y., & Lo, A. W. (1998). Nonparametric estimation of state-price densities implicit in financial asset prices. *The Journal of Finance*, *53*(2), 499–547.

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

Amilon, H. (2003). A neural network versus black–scholes: a comparison of pricing and hedging performances. *Journal of Forecasting*, *22*(4), 317–335.

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

Aruoba, S. B., Diebold, F. X., & Scotti, C. (2009). Real-time measurement of business conditions. *Journal of Business & Economic Statistics*, *27*(4), 417–427.

Baker, S. R., Bloom, N., & Davis, S. J. (2016). Measuring economic policy uncertainty. *The Quarterly Journal of Economics*, *131*(4), 1593–1636.

Bates, D. S. (2000). Post-'87 crash fears in the s&p 500 futures option market. *Journal of Econometrics*, *94*(1-2), 181–238.

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

Bollerslev, T., Todorov, V., & Xu, L. (2015). Tail risk premia and return predictability. *Journal of Financial Economics*, *118*(1), 113–134.

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

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

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

Clemen, R. T. (1989). Combining forecasts: A review and annotated bibliography. *Interna-*

tional Journal of Forecasting, *5*(4), 559–583.

Cont, R., & Da Fonseca, J. (2002). Dynamics of implied volatility surfaces. *Quantitative Finance*, *2*(1), 45.

Donaldson, R. G., & Kamstra, M. (1996). Forecast combining with neural networks. *Journal of Forecasting*, *15*(1), 49–61.

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

Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., & Garcia, R. (2009). Incorporating functional knowledge in neural networks. *Journal of Machine Learning Research*, *10*(6).

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

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

Fan, J., & Mancini, L. (2009). Option pricing with model-guided nonparametric methods. *Journal of the American Statistical Association*, *104*(488), 1351–1372.

Fan, J., Wu, Y., & Feng, Y. (2009). Local quasi-likelihood with a parametric guide. *Annals of Statistics*, *37*(6B), 4153.

Fan, Y., & Ullah, A. (1999). Asymptotic normality of a combined regression estimator. *Journal of Multivariate Analysis*, *71*(2), 191–240.

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

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

Glad, I. K. (1998). Parametrically guided non-parametric regression. *Scandinavian Journal of Statistics*, *25*(4), 649–668.

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

Hansen, P. R., Lunde, A., & Nason, J. M. (2005). Model confidence sets for forecasting models.

Hansen, P. R., Lunde, A., & Nason, J. M. (2011). The model confidence set. *Econometrica*, *79*(2), 453–497.

Harrald, P. G., & Kamstra, M. (1997). Evolving artificial neural networks to combine financial forecasts. *IEEE Transactions on Evolutionary Computation*, *1*(1), 40–52.

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.

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

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

Kumar, M., & Thenmozhi, M. (2014). Forecasting stock index returns using arima-svm, arima-ann, and arima-random forest hybrid models. *International Journal of Banking, Accounting and Finance*, *5*(3), 284–308.

Liu, L., & Zhang, T. (2015). Economic policy uncertainty and stock market volatility. *Finance Research Letters*, *15*, 99–105.

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

Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., ... Winkler, R. (1982). The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting*, *1*(2), 111–153.

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

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

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

Møller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, *6*(4), 525–533.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., & Sutskever, I. (2021). Deep dou-

ble descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, *2021*(12), 124003.

Rolnick, D., & Tegmark, M. (2017). The power of deeper networks for expressing natural functions. *arXiv preprint arXiv:1705.05502*.

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

Stock, J. H., & Watson, M. W. (1998). *A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series.* National Bureau of Economic Research Cambridge, Mass., USA.

Todorov, V. (2019). Nonparametric spot volatility from options. *The Annals of Applied Probability*, *29*(6), 3590–3636.

10 Appendix

10.1 Overview of Abbreviations

- BS - Black-Scholes
- AHBS - ad-hoc Black-Scholes
- CW - Carr and Wu
- IVRMSE - implied volatility root mean squared error
- RMSE - mean squared error
- MAE - mean absolute error
- DOTMC - deep out-of-the-money calls
- OTMC - out-of-the-money calls
- ATM - at-the-money options
- OTMP - out-of-the-money puts
- DOTMP - deep out-of-the-money puts
- TtM - time to maturity
- NN - neural network
- RF - random forest
- SVM - support vector machine
- MCS - model confidence set
- SA - simple average
- WA - weighted average
- VIX - volatility index
- EPU - economic policy uncertainty measure
- ADS - the business conditions index of Aruoba et al. (2009)
- TMS - the first differences of the term spread
- CRS - the first differences of the credit spread
- LTV - Left Tail Volatility
- LTP - Left Tail Probability
- TS - term structure

10.2 MATLAB Codes

The MATLAB codes I wrote to conduct my research are included in a zip-file. Each file is briefly explained below. "$*$" indicates that any of the abbreviations for the parametric models can be filled in instead, i.e., $* \in \{\text{“}BS\text{”}, \text{“}AHBS\text{”}, \text{“}Heston\text{”}, \text{“}CW\text{”}\}$.

- *Heston_estimation.m*: Estimate parameters Heston for cross-section exercises.
- *Heston_cross_section*1.*m*: Get Heston predictions for same day exercise using parameters.
- *Heston_cross_section*2.*m*: Get Heston predictions for days ahead exercise using parameters.
- *CW_estimation.m*: Estimate parameters CW for cross-section exercises.
- *CW_cross_section*1.*m*: Get CW predictions for same day exercise using parameters.
- *CW_cross_section*2.*m*: Get CW predictions for days ahead exercise using parameters.
- $*$*_NN.m*: Get neural network corrections of $*$ predictions for same day exercise.
- $*$*_NN_prediction.m*: Get neural network corrections of $*$ predictions for days ahead exercise.
- $*$*_RF_Same_Day.m*: Get random forest corrections of $*$ predictions for same day exercise.
- $*$*_RF_Prediction_Ahead.m*: Get random forest corrections of $*$ predictions for days ahead exercise.
- $*$*_SVM_Same_Day.m*: Get support vector machine corrections of $*$ predictions for same day exercise.
- $*$*_SVM_Prediction_Ahead.m*: Get support vector machine corrections of $*$ predictions for days ahead exercise.
- *SA_Same_Day.m*: Get simple average of parametric model predictions for same day exercise.
- *SA_Prediction_Ahead.m*: Get simple average of parametric model predictions for days ahead exercise.
- *inverse_IVRMSE_weighted.m*: Get weighted average of parametric model predictions for same day exercise.
- *inverse_IVRMSE_weighted_Prediction_Ahead.m*: Get weighted average of parametric model predictions for days ahead exercise.
- *bar_chart.m*: Create bar chart of feature importances.
- *Combined_NN_Same_Day.m*: Get neural network combinations of parametric model predictions (possibly with moneyness and/or ttm included) for same day exercise.
- *Combined_NN_Prediction_Ahead.m*: Get neural network combinations of parametric model predictions (possibly with moneyness and/or ttm included) for days ahead exercise.
- *BS.m*: Get BS predictions for option panel exercise.

- *Heston.m*: Estimate Heston model for option panel exercise.
- *get_predictions_Heston.m*: Get predictions Heston model for option panel exercise.
- *CW.m*: Estimate CW model for option panel exercise.
- *get_predictions_CW.m*: Get predictions CW model for option panel exercise.
- *SA_Option_Panel.m*: Get simple average of parametric model predictions for option panel exercise.
- *WA_Option_Panel.m*: Get weighted average of parametric model predictions for option panel exercise.
- *NN_Correction_Option_Panel.m*: Get 3-layered neural network corrections of Heston model predictions for option panel exercise.
- *Combined_NN_Option_Panel_all_vars.m*: Determine feature importance for neural network combinations by setting one feature at a time equal to zero, and perform option panel exercise.
- *Combined_NN_Option_Panel.m*: Get neural network combinations for option panel exercise with specific features.
- *all_vars_one_by_one.m*: Add variables one by one based on feature importance, and perform option panel exercise.
- *performance_time_series.m*: Get mean and median IVRMSE of model for option panel exercise, given predictions.
- *get_in_sample_predictions.m*: Get in-sample predictions of combining neural networks, for characteristics evaluation in option panel.
- *get_surface_characteristics.m*: Get the surface characteristics, given predictions for option panel.
- *MCS_RF_SVM.m*: Determine the model confidence set for Table 6.1 of the same day exercise.
- *MCS_RF_SVM_**d.m*: Determine the model confidence set for Table 6.1 of the $**$-day ahead exercise, with $** \in \{1, 5, 21\}$.
- *MCS_NN_combination.m*: Determine the model confidence set for Table 7.1 of the same day exercise.
- *MCS_NN_Combination_**d.m*: Determine the model confidence set for Table 7.1 of the $**$-day ahead exercise, with $** \in \{1, 5, 21\}$.