



# Pricing American Put Options

A Comparison of Parametric and Non-parametric  
Approaches

Master Thesis  
Spring 2022

## Supervisor

Anders Rahbek

Keystrokes: 125,000

## Contribution

*Edmund Christensen Clink hsr387*

1, 2, 3.1, 4.1, 5.1, 5.3, 6.2, 7.2, 8.1, 9

*Jeppe Taagegaard Martinussen nkl401*

1, 2, 3.2, 4.2, 5.2, 6.1, 7.1, 7.3, 8.2, 9

## ABSTRACT

This thesis implements and compares two approaches in order to estimate American put option prices, namely the parametric Least-squares Monte Carlo (LSM) approach [Longstaff & Schwartz (2001)] and a data-driven neural network (NN) approach. For the parametric LSM approach, the underlying asset is simulated with the Heston-CIR model [Heston (1993)] of stochastic volatility, in which the instantaneous interest rate follows a CIR-process [Cox, Ingersoll & Ross (1985)]. Our findings suggest that NN methods achieve greater precision in estimating American put option prices both in-sample and out of sample. More complex NN models that account for the level of volatility yield more accurate out of sample estimates during times of high volatility. However, there is a clear trade-off between added complexity and computational effort. The LSM approach accurately estimates American put option prices in-sample and close to the in-sample period, and the approach requires much less data.

# Contents

	Page
<b>1 Introduction</b>	<b>3</b>
<b>2 The Option Pricing Problem</b>	<b>6</b>
<b>3 Models of Stochastic Volatility</b>	<b>7</b>
3.1 The Cox-Ingersoll-Ross Model . . . . .	7
3.2 The Heston-CIR Model . . . . .	7
<b>4 Simulation and Discretisation</b>	<b>8</b>
4.1 Milstein Discretisation . . . . .	9
4.2 Applying the Milstein Discretisation Method . . . . .	11
4.2.1 The Heston Model . . . . .	11
4.2.2 The Cox-Ingersoll-Ross Model . . . . .	12
<b>5 Least-squares Monte Carlo</b>	<b>12</b>
5.1 Least-squares Monte Carlo Algorithm . . . . .	13
5.2 Improving Accuracy: Control Variates . . . . .	15
5.3 Model Calibration . . . . .	16
5.3.1 Calibrating the CIR Model . . . . .	16
5.3.2 Calibrating the Heston Model . . . . .	18
<b>6 Neural Networks for Option Pricing</b>	<b>19</b>
6.1 Activation Functions . . . . .	20
6.2 Multilayer Perceptron . . . . .	21
6.2.1 The Backpropagation Algorithm . . . . .	23
<b>7 Research Setup and In-sample Results</b>	<b>25</b>
7.1 In-sample Data and Data Pre-processing . . . . .	25
7.2 Least-squares Monte Carlo . . . . .	31
7.2.1 Calibration Results & In-sample Performance . . . . .	31
7.3 Neural Network . . . . .	37
7.3.1 Network Architecture . . . . .	38
7.3.2 Model Training & In-sample Performance . . . . .	39
<b>8 Out of Sample Results</b>	<b>45</b>
8.1 6 Months Out of Sample . . . . .	46
8.1.1 January 8th 2018 . . . . .	46
8.1.2 February 1st 2018 . . . . .	48
8.1.3 June 29th 2018 . . . . .	50
8.2 Period of High Volatility . . . . .	53

8.2.1 March 1st - March 7th 2022 . . . . .	53
<b>9 Concluding Discussion and Potential Applications</b>	<b>56</b>

# 1 Introduction

Pricing American style options is an important and challenging aspect within finance. Many options that are traded are American options and can be found in all major financial markets including the equity, foreign exchange or commodity markets. Even among private investors trades with American options have increased significantly as of late. At the time of writing a record number of put options are being issued in order for investors to take on short positions. Options of American style have the feature that they can be exercised at any time up until maturity. This results in the pricing problem being of a rather complex nature when compared to European options, for which closed form solutions are typically available, since European options can be exercised *only* at maturity.

In order to estimate the price of American put options at time  $t$  on the S&P 500, denoted  $\hat{V}_t$ , we consider two approaches: A parametric simulation based method, Least-Squares Monte Carlo (LSM) first proposed by Longstaff and Schwartz (2001) and a data-driven machine learning method, Neural Networks (NN).

Paramount for the LSM procedure is model-based simulation of the underlying asset, in our case the S&P 500. Several stochastic processes exist in the literature for this purpose. We propose the Heston-CIR model as a means of describing the underlying asset at time  $t$ ,  $S_t$ , in order to price the American put option, since it introduces stochastic differential equations (SDE) for the volatility,  $v_t$ , and the instantaneous interest rate,  $r_t$ . We define  $X_t = (S_t, v_t, r_t)$  as the entire underlying process. Given the continuous nature of the SDEs we are not able to simulate the model without taking certain measures. This entails transforming the SDEs into discrete processes. This procedure is called discretisation. There are several methods of discretisation and we specifically apply the Milstein discretisation scheme [Milstein (1975)] since it has proved rather effective, yet simple. Once we have applied the discretisation scheme to the SDEs that characterise the Heston-CIR model, we are left with discrete processes. Following the discretisation procedure we are able to simulate the model for the underlying asset. Specifically, we simulate  $n = (1, 2, 3, \dots, 35000)$  paths of the S&P 500.

Due to the option being of American nature, the option holder must evaluate whether or not to exercise the option at any point in time. Therefore, it is relevant what the payoff of immediate exercise is as well as the value of not exercising the option referred to as the continuation value denoted  $C_{t,n}$  at time  $t$  and for path  $n$ . The value of immediate exercise at any time  $t$  for path  $n$  is denoted  $h_{t,n}(X_{t,n})$  and is given by the difference between the strike price,  $K$ , and the spot price of underlying asset,  $S_t$ . The value of immediate exercise is observable at all times, but the continuation value is associated with uncertainty as it depends on future cash-flows and therefore needs to be estimated. Longstaff & Schwartz (2001) present a simple yet powerful method in order to estimate the value of American put options,  $\hat{V}_t^{LSM}$ , at selected times,  $t$ , maturities,  $T$ , and strike prices,  $K$ . The main idea behind the method is to estimate the above mentioned continuation value at each discrete time point using only Ordinary Least Squares (OLS). Once the continuation value has been estimated accordingly, the decision between immediate exercise and continuation

can be made at each discrete time point. At each time  $t$  for each path  $n$  we evaluate whether the value of immediate exercise exceeds the value of continuation. When this is the case the option is exercised and the estimated value of the option is the discounted payoff of immediate exercise at time  $t$ .

The above method of pricing American put options can be improved with one of several techniques within the field of variance reduction. We consider the technique called control variates and propose using the price of the European put option as a control variate. The price of the European put option is an ideal candidate, since it is assumed to correlate with the American counterpart. Moreover, a closed form solution is available such that we know the value of the European option precisely. This allows us to estimate the European option price with the LSM approach and compare it to the closed form solution leaving us with an estimation error for the European put option price. This estimation error can be used to correct the estimates of the American put option price. In addition, variance is reduced significantly. Adjusting  $\hat{V}_t^{LSM}$  by the estimation error of the European put option we have the proposed estimate of the American put option price,  $\hat{V}_t^{CV}$ .

The precision of the model is highly dependent of the input parameters we use. Therefore, we require a method, which adjusts the parameter values of the Heston-CIR model accordingly, such that we are able to generate prices that are close to the prices we observe in the market. This procedure is commonly referred as the calibration procedure. The main idea is to minimise the differences between observed market prices,  $V_t^{Market}$ , and model predicted prices,  $\hat{V}_t^{CV}$ , by adjusting the model's parameters. Specifically, we minimise the Mean Squared Error (MSE) between model and market prices.

As an alternative to the parametric LSM procedure we consider a model-free strictly data-driven approach, namely a NN such as the one proposed by Hutchinson et al. (1994). Their results indicate that NNs are able to explain the option price and their approach proved to be useful in pricing American style options. The model architecture we propose is a feed-forward NN which uses the backpropagation algorithm [Rumelhart et al. (1986)] in order to determine the optimal weights of the network. Specifically, we consider two model specifications. First, a simple model which serves as a baseline. This model's inputs,  $X_t$  at time  $t$ , are the same as for the LSM procedure. In addition, we propose an extended model with a more complex specification and an extended input vector which also includes a range of volatility measures following the lead of Culkin & Das (2017). Akin to the model calibration procedure for the LSM approach the two NN models are trained such that the estimated option prices,  $\hat{V}_t^{NN}$ , replicate real market data precisely. The training procedure consists of weight adjustments for each of the two NN models.

In order to perform the calibration and training procedure of the LSM and NN approaches respectively, we consider data for American put options on S&P 500 for the entire year of 2017 as our in-sample period. Once calibrated and trained, we estimate American put option prices at a range of dates. The results suggest that all models are able to realistically estimate option prices for a range of strike prices and maturities in-sample, although both NN models consistently outperform

the LSM approach. In addition to the in-sample performance, the models are evaluated out of sample. We find that both NN models perform similarly during the first 6 months out of sample. The extended model is slightly more accurate, but both NN models outperform LSM. Immediately out of sample LSM is still able to estimate American put option prices realistically, however as we move further away from the in-sample period, the approach only produces realistic option prices for in the money (ITM) options. Additionally, we test the approaches out of sample for March 2022, where uncertainty and high volatility dominate the financial markets. LSM is still only able to price ITM options, but this pricing performance is consistent out of sample. For this out of sample period we find that the extended NN model, which accounts for volatility, outperforms the baseline NN model for the entire range of options.

## 2 The Option Pricing Problem

An option is a financial derivative that gives the holder the right to buy or sell a certain underlying asset at a certain date for a certain price. The holder of the option is not obligated to trade the underlying asset, and the option can simply be left to expire unlike forward or future contracts. We distinguish between two types of options, namely put and call. A call option gives the option holder the right to *buy* the underlying asset at a fixed "strike price", usually denoted  $K$ , whereas a put option gives the option holder the right to *sell* the underlying asset at the fixed strike price. Moreover, options can be classified into different styles. More commonly considered styles are of European and American style. A European option gives the option holder the right to exercise only at maturity, denoted  $T$ , whereas an American option can be exercised at any time up to maturity. Given the difference in exercise policy, European and American options with identical maturity and strike price can be priced differently. Pricing American options is more complex, since the option can be exercised at any time up to maturity. Analysing European options is a simpler matter, since a closed form solution is typically available.

This paper will specifically focus on pricing American put options where the underlying asset is the S&P 500 index. A put option is considered in the money (ITM) when  $S < K$ , where  $S$  denotes the price of the underlying asset. In contrast, a put option is out of the money (OTM) when  $S > K$ . In case of the option being OTM, the optimal exercise policy is not to exercise the option resulting in a payoff of 0. Hence, the payoff of the option at the time of maturity,  $T$ , is given by:

$$\max[K - S_T, 0]$$

Due to the option being of American nature, the option holder must evaluate whether or not to exercise the option at any point in time. Therefore it is relevant what the payoff of immediate exercise is as well as the value of not exercising the option which is referred to as the continuation value. At each time,  $t$ , the option holder must compare the payoff of immediate exercise to the value of continuation. Since the payoff of immediate exercise is known for all times, we require a method in order to estimate the value of continuation in order to determine when the option is optimally exercised, referred as the optimal stopping rule.

Longstaff & Schwartz (2001) proposed a simulation based method to determine this optimal stopping rule called Least-Squares Monte Carlo (LSM). By simulating paths for the underlying asset, which are determined by a parametric model, we can estimate the conditional expectation of the continuation value using the cross-sectional information in the simulated paths by use of least squares.

An alternative approach more prominent in newer literature is the use of Neural Networks (NN) in order to solve the option pricing problem. Hutchinson et al. (1994) proposed the use of NN for estimating prices of European options, which is generally considered a reference point for the use of NN in order to estimate option prices, and the later literature applies the approach on both

European and American style options. Unlike the parametric simulation based approach, NNs are data-driven and we make no assumptions on the process of the underlying asset.

### 3 Models of Stochastic Volatility

#### 3.1 The Cox-Ingersoll-Ross Model

Classical models in the option pricing literature make tight assumptions for the short rate. A prominent example is the Black-Scholes valuation formula [Black & Scholes (1973)], where they assume "ideal conditions" in the market for the stock and for the option. One of these assumptions is that the short-term interest rate is known and is constant through time. Such assumptions are quite strict and it can be discussed to what extent this is reflected by reality. Interest rates can be characterised by a degree of stochasticity, and particularly move in a random fashion in the short term. Interest rates do not trend to zero or infinity in the long term such that there must always be mean reversion. The yields of the benchmark rates vary with the time to maturity, which implies different (instantaneous) forward rates. A model to describe the short rate is the Cox-Ingersoll-Ross (CIR) model [Cox, Ingersoll & Ross (1985)]. The short interest rate,  $r_t$ , is assumed to follow the SDE specified below:

$$dr_t = \kappa_r(\theta_r - r_t)dt + \sigma_r\sqrt{r_t}dW_{r,t} \quad (3.1)$$

The short rate described in (3.1) is a mean reverting process towards its long-term average,  $\theta_r$ , where  $\kappa_r$  determines the speed of mean reversion,  $\sigma_r$  is the volatility of the short rate and  $W_{r,t}$  is a standard Brownian motion.

#### 3.2 The Heston-CIR Model

As introduced above, the Black-Scholes model is a classical option pricing model, which is widely used and recognised. For long it has been regarded as one of the best models for determining the price of options. This method of asset pricing is a geometric Brownian motion that only depends on the asset price and time. Despite being well-recognised, it relies on simplified and often considered unrealistic assumptions. In particular the model assumes constant volatility, which has been observed to not be constant over time. In order to price assets within a more realistic framework, the volatility can be assumed to follow a stochastic process. Moreover, volatility has been observed to have other characteristics, namely from Hilpisch (2015) volatility clustering, the leverage effect, and mean reversion. Volatility clustering entails that a period with high volatility is followed by periods with high volatility as well (and vice versa).

Such a model of stochastic volatility, which deals with the above mentioned concerns, is the Heston model [Heston (1993)]. The model specified in Heston (1993) assumes a constant interest rate. Expanding the model with the stochastic process specified in (3.1) yields the Heston-CIR model. The model is described by the bi-variate stochastic process for stock price,  $S_t$ , and its time-varying variance,  $v_t$ :

$$dS_t = r_t S_t dt + \sqrt{v_t} S_t dW_{S,t} \quad (3.2)$$

$$dv_t = \kappa(\theta - v_t)dt + \sigma\sqrt{v_t}dW_{v,t},$$

where  $r_t$  is the short rate and is assumed to follow the CIR-process specified in (3.1),  $\theta$  is the long-term average of variance,  $\kappa$  controls the speed of mean reversion and degree of volatility clustering,  $\sigma$  is the volatility of the variance, and  $W_{S,t}, W_{v,t}$  are standard Brownian motions. We note, that the process for the variance in the Heston model follows a CIR-process, which is identical in construction to the process for the short rate in the CIR-model. The process of the variance is simplified to be a deterministic process such as in the Black-Scholes model in the case of  $\kappa$  and  $\sigma$  being equal to zero. In the Black-Scholes model this would result in the volatility being equal to the square root of the initial variance in the Heston model, denoted  $v_0$ .

Correlations are given by  $E[W_{S,t}W_{v,t}] = \rho$ , and we assume  $\kappa, \theta, \sigma > 0$ . Moreover,  $W_r$  is assumed independent of the two standard Brownian motions,  $W_{S,t}, W_{v,t}$ . The leverage effect is captured in the Heston model by assuming  $\rho < 0$ , which ensures negative correlation between the stock price,  $S_t$ , and volatility captured by the variance,  $v_t$ . The leverage effect refers to the well-established relationship between stock returns and volatility, namely that volatility increases when the stock price falls. Moreover, it is assumed that the process for the interest rate specified in (3.1) is uncorrelated with the process for both the stock,  $S_t$ , and variance,  $v_t$ , i.e.  $dW_{S,t}dW_{r,t} = dW_{v,t}dW_{r,t} = 0$ .

It can be shown that if  $\kappa, \theta$  and  $\sigma$  satisfy the condition

$$2\kappa\theta > \sigma^2 \wedge v_0 > 0$$

the variance,  $v_t$ , is always positive. This condition is known as the Feller condition [Feller (1951)]. However, the Feller condition is hardly satisfied in the market, and in many cases its violation is required in order to replicate market data realistically. This particularly holds true for options with maturity longer than a few weeks. Instead of requiring the Feller condition, the variance process is floored to 0 to avoid negative variance. This is called a full truncation scheme and is applied in the discretisation process.

## 4 Simulation and Discretisation

The Heston-CIR model is a model consisting of three continuous stochastic differential equations (SDE). When using Monte Carlo simulation in order to price options, we are interested in generating underlying values of the underlying asset or the interest rate over time. The dynamics of the underlying asset and the short rate are assumed to be driven by continuous stochastic processes. Simulation, however, is done at discrete time steps. Therefore, we must transform the three continuous SDEs into discrete processes, a procedure commonly referred to as discretisation.

Since discretisation has to be performed for all three continuous processes defined in (3.1) and

(3.2), we consider a general SDE for  $X_t$ .

$$dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dW_t, \quad (4.1)$$

where  $W_t$  is a Brownian motion. We wish to simulate  $X_t$  over the time interval  $[0, T]$  in  $I$  equally spaced sub-intervals with the size of each sub-interval of  $dt$ . By integrating  $X_t$  in (3.1) from  $t$  to  $t + dt$  we get:

$$X_{t+dt} = X_t + \int_t^{t+dt} \mu(X_u, u)du + \int_t^{t+dt} \sigma(X_u, u)dW_u \quad (4.2)$$

Since the value of  $X_t$  is known at time  $t$  we wish to obtain the value,  $X_{t+dt}$ . By definition, the SDEs in (3.1) and (3.2) are continuous, and approximating the processes with a discretisation scheme results in a finite number of evaluations for the processes. Errors from this finite approximation will occur, but this error can be reduced by increasing the number of time steps. However, this is associated with increased computational costs. Therefore, it is relevant to discuss which approximation converges to the continuous process most efficiently. The simplest approximation is the Euler method. Kinghi (2019) compares the Euler method to the Milstein method [Milstein (1975)], which increases the accuracy of the Euler method by adding a second-order “correction” term, which is derived from the stochastic Taylor series expansion. The author finds that the Milstein method performs better in terms of convergence. Therefore, this thesis will approximate the stochastic continuous processes using the Milstein method.

## 4.1 Milstein Discretisation

When considering the two approximation methods introduced above, Gatheral (2006) argues ”In practice, with typical parameters, the frequency with which the process goes negative is substantially reduced relative to the Euler case. As it is no more computationally expensive to implement the Milstein discretisation than it is to implement the Euler discretisation, the Milstein discretisation is always to be preferred”. However, to describe the Milstein approximation method, we first need to define of the Euler method.

The Euler method results from approximation of the integrals with the left-point rule in (4.2), which for the first integral yields [Glasserman (2003)]:

$$\int_t^{t+dt} \mu(X_u, u)du \approx \mu(X_t, t) \int_t^{t+dt} du = \mu(X_t, t)dt.$$

In similar fashion, the second integral is approximated:

$$\begin{aligned} \int_t^{t+dt} \sigma(X_u, u)dW_u &\approx \sigma(X_t, t) \int_t^{t+dt} dW_u \\ &= \sigma(X_t, t)(W_{t+dt} - W_t) = \sigma(X_t, t)\sqrt{dt}Z, \end{aligned} \quad (4.3)$$

where  $Z$  is a standard normal variable such that  $W_{t+dt} - W_t$  and  $\sqrt{dt}Z$  are identical in distribution. This leaves us with the Euler discretisation of (4.2):

$$X_{t+dt} = X_t + \mu(X_t, t)dt + \sigma(X_t, t)\sqrt{dt}Z. \quad (4.4)$$

This result is crucial and will be applied in the derivation of the Milstein scheme. The scheme works for SDEs where the coefficients  $\mu(X_t)$  and  $\sigma(X_t)$  depend only on  $X$  and not directly on time,  $t$ . Therefore, we can write the SDE for  $X_t$  described in (4.1) as:

$$dX_t = \mu(X_t)dt + \sigma(X_t)dW_t = \mu_t dt + \sigma_t dW_t.$$

This can be rewritten in integral form as

$$X_{t+dt} = X_t + \int_t^{t+dt} \mu_s ds + \int_t^{t+dt} \sigma_s dW_s. \quad (4.5)$$

The accuracy of the discretisation is increased by considering expansions of the coefficients  $\mu_t = \mu(X_t)$  and  $\sigma_t = \sigma(X_t)$ . We can apply Itô's lemma to the coefficients' functions giving

$$\begin{aligned} \mu_s &= \mu_t + \int_t^s (\mu'_u \mu_u + \frac{1}{2} \mu''_u \sigma_u^2) du + \int_t^s (\mu'_u \sigma_u) dW_u, \\ \sigma_s &= \sigma_t + \int_t^s (\sigma'_u \mu_u + \frac{1}{2} \sigma''_u \sigma_u^2) du + \int_t^s (\sigma'_u \sigma_u) dW_u, \end{aligned}$$

for  $s \geq t$ . Substituting for  $\mu_s$  and  $\sigma_s$  in (4.4) yields

$$\begin{aligned} X_{t+dt} &= X_t + \int_t^{t+dt} \left( \mu_t + \int_t^s (\mu'_u \mu_u + \frac{1}{2} \mu''_u \sigma_u^2) du + \int_t^s (\mu'_u \sigma_u) dW_u \right) ds \\ &\quad + \int_t^{t+dt} \left( \sigma_t + \int_t^s (\sigma'_u \mu_u + \frac{1}{2} \sigma''_u \sigma_u^2) du + \int_t^s (\sigma'_u \sigma_u) dW_u \right) dW_s. \end{aligned}$$

Defining the order  $dW_u dW_s = \mathcal{O}(dt)$  as order one, we ignore terms higher than order one,  $dsdu = \mathcal{O}((dt)^2)$  and  $dsdW_u = \mathcal{O}((dt)^{\frac{3}{2}})$  as they are negligibly small. This leaves us with

$$X_{t+dt} = X_t + \mu_t \int_t^{t+dt} ds + \sigma_t + \int_t^{t+dt} dW_s + \int_t^{t+dt} \int_t^s (\sigma'_u \sigma_u) dW_u dW_s. \quad (4.6)$$

Applying Euler discretisation specified earlier to the last term:

$$\begin{aligned} \int_t^{t+dt} dW_s + \int_t^{t+dt} \int_t^s (\sigma'_u \sigma_u) dW_u dW_s &\approx \sigma'_t \sigma_t \int_t^{t+dt} \int_t^s dW_u dW_s \\ &= \sigma'_t \sigma_s \left( \int_t^{t+dt} W_s dW_s - W_t W_{t+dt} + W_t^2 \right). \end{aligned} \quad (4.7)$$

A further application of Itô's lemma gives

$$\int_t^{t+dt} W_s dW_s = \frac{1}{2} W_{t+dt}^2 - \frac{1}{2} W_t^2 - \frac{1}{2} dt \quad (4.8)$$

Substituting (4.8) back into (4.7) yields:

$$\int_t^{t+dt} dW_s + \int_t^{t+dt} \int_t^s (\sigma'_u \sigma_u) dW_u dW_s \approx \frac{1}{2} \sigma'_u \sigma_u [(\Delta W_t)^2 - dt],$$

where  $\Delta W_t = W_{t+dt} - W_t$ , which is equal in distribution to  $\sqrt{dt} Z$ . Combining (4.8) and (4.6) produces the general form of Milstein discretisation:

$$X_{t+dt} = X_t + \mu_t dt + \sigma_t \sqrt{dt} Z + \frac{1}{2} \sigma'_t \sigma_t dt (Z^2 - 1). \quad (4.9)$$

The latter shows that the Milstein approximation method adds a second order correction term to the Euler method in (4.4).

## 4.2 Applying the Milstein Discretisation Method

The overall goal is to approximate the stochastic continuous processes for the underlying asset and the short rate. Therefore, the next step is to apply the general Milstein approximation method in (4.9) to the SDEs in (3.1) and (3.2).

### 4.2.1 The Heston Model

The Heston model for stochastic volatility consists of the two continuous processes for the stock price,  $S_t$ , and volatility,  $v_t$ . The coefficients for the stock price process are  $\mu(S_t) = rS_t$  and  $\sigma(S_t) = \sqrt{v_t} S_t$ . Applying the coefficients and (4.9) yields the process for the stock price under Milstein discretisation:

$$S_{t+dt} = S_t + rS_t dt + \sqrt{v_t dt} S_t Z_s + \frac{1}{2} S_t v_t dt (Z_s^2 - 1),$$

where  $dt = t_i - t_{i-1}$  for  $i = 1, 2, 3, \dots, I$ .

From the bi-variate stochastic process in (3.2) the coefficients of the variance process are  $\mu(v_t) = \kappa(\theta - v_t)$  and  $\sigma(v_t) = \sigma\sqrt{v_t}$ . Once more by directly applying the general Milstein discretisation scheme from (4.9) for  $v_t$  we get:

$$v_{t+dt} = v_t + \kappa(\theta - v_t) dt + \sigma \sqrt{v_t dt} Z_v + \frac{1}{4} \sigma^2 dt (Z_v^2 - 1), \quad (4.10)$$

where  $dt = t_i - t_{i-1}$  for  $i = 1, 2, 3, \dots, I$ . In order to strictly avoid negative values for variance,  $v_t$ , we apply a full truncation scheme to, (4.10) which entails

$$v_t = \max[v_t, 0].$$

#### 4.2.2 The Cox-Ingersoll-Ross Model

Recall, that the process for the variance in the Heston model follows a CIR-process, which is identical in construction to the process for the short rate in the CIR-model. As such, Milstein discretisation of the short rate is *also* identical in construction. Applying the Milstein discretisation method to the SDE in (3.1) yields:

$$r_{t+dt} = r_t + \kappa_r(\theta_r - r_t)dt + \sigma_r \sqrt{r_t dt} Z_r + \frac{1}{4}\sigma_r^2 dt(Z_r^2 - 1), \quad (4.11)$$

where  $dt = t_i - t_{i-1}$  for  $i = 1, 2, 3, \dots, I$ . Once more we apply a full truncation scheme to (4.11) to ensure non-negative short interest rates such that:

$$r_t = \max[r_t, 0].$$

## 5 Least-squares Monte Carlo

The option holder faces a choice at any point in time until maturity,  $T$ , of whether to immediately exercise the option or to not exercise the option. At any time  $t$ , the value of immediate exercise is known to the option holder. However, the value of not exercising is unknown, since it depends on future cash flows. We require a method that allows us to compare the value of immediate exercise to the unknown value of continuation, which will allow us to decide when to optimally exercise the option.

The objective of the LSM method boils down to approximating the above-mentioned optimal stopping rule such that value of the American option is maximised for each path. This optimal stopping rule is approximated at each discrete exercise point,  $t$ . At the expiration date of the option,  $T$ , the option holder decides to exercise the option in case of it being in the money. On the other hand, the option is simply allowed to expire when OTM. At the prior exercise point,  $T - 1$ , the option holder must decide whether to exercise immediately or to continue the life of the option and revisit the exercise decision at the next exercise time. Once the immediate exercise value is greater than or equal to the value of continuation, the option holder exercises the option. For each path this will maximise the value of the option.

The optimal stopping policy is contingent on the cash flow from immediate exercise, which is known to the option holder, and the unknown value of continuation. Therefore, we need to approximate the continuation value, which is given by the conditional expectation of the remaining discounted cash flows with respect to the risk-neutral pricing measure,  $Q$ , which is a probability measure that accounts for risk-aversion among investors. The main idea of Longstaff and Schwartz (2001) is that this conditional expectation can be estimated from the cross-sectional information in the simulation using least squares. By estimating the continuation value at each discrete point in time at which the option could be exercised, the optimal exercise strategy for each path can be fully specified. When the optimal exercise strategy for the option is known, the option can be valued accurately.

## 5.1 Least-squares Monte Carlo Algorithm

As mentioned, whether or not an American option is exercised before maturity depends on the value of continuation. To simulate the value of the American put option, we simulate  $N$  paths for the underlying process,  $X_t$ , described in Section 3 with  $I + 1$  equally spaced time steps. For each simulated path,  $n$ , we define the underlying process for the Heston-CIR model as  $X_{t,n} = (S_{t,n}, v_{t,n}, r_{t,n})$  for  $t \in [0, 1, 2, \dots, T]$ ,  $n \in [1, 2, \dots, N]$ . For each path  $n$  at maturity  $T$  the optimal exercise strategy is to exercise the option when it is in the money. In this case the payoff will be  $K - S_T$ . Therefore, the payoff function of immediate exercise at time  $T$ , denoted  $h_{T,n}$ , is

$$h_{T,n}(X_{T,n}) = \max[K - S_{T,n}, 0]. \quad (5.1)$$

From maturity, time  $T$ , we iterate backwards to  $t = 0$  and at each time  $t$  for each path  $n$ , the payoff of immediate exercise is denoted

$$h_{t,n}(X_{t,n}) = K - S_{t,n}. \quad (5.2)$$

Instead of immediate exercise of the option, the option holder has the choice of not exercising i.e. letting the option continue to run. This continuation value can be described by the conditional expected value of holding the option under the risk-neutral measure at time  $t_i$  for path  $n$ :

$$C_{t,n}(x) = E^Q \left[ \exp(-dt \cdot r) V_{t_{i+1},n}(X_{t+1,n}) | x = X_{t,n} \right] \quad (5.3)$$

At each potential exercise point,  $t_i$ , the holder exercises the option when the payoff of immediate exercise is larger than the value of continuation. This behaviour of choice between immediate exercise and continuation can be captured by the following payoffs:

$$V_{t,n}(X_{t,n}) = \begin{cases} h_{t,n}(X_{t,n}) & \text{if } h_{t,n}(X_{t,n}) \geq C_{t,n}(x) \\ \exp(-dt \cdot r) V_{t_{i+1},n}(X_{t+1,n}) & \text{if } h_{t,n}(X_{t,n}) < C_{t,n}(x) \end{cases} \quad (5.4)$$

for  $i = 0, 1, 2, \dots, I - 1$ , and where  $dt = t_{i+1} - t_i$ . We note, that when the continuation value exceeds the payoff of immediate exercise,  $h_{t,n}(X_{t,n}) < C_{t,n}(x)$ , the payoff the option holder receives is the *actual* discounted value of continuation. The actual continuation values are known since asset prices are simulated for every point in time. However, when deciding whether or not to immediately exercise, the choice is dependent on the conditional expectation. If the choice was based on the actual discounted value of continuation this would translate into perfect foresight, which would result in a better than optimal exercise strategy [Hilpisch (2015)].

At each potential exercise point the value of immediate exercise is known to the option holder, as it is simply the difference between strike price,  $K$ , and stock price,  $S_t$ , which are both observable. Therefore, in order to determine the optimal stopping rule, the approach boils down to estimating the continuation value at each potential exercise point.

We regress the  $N$  realised cash flows from continuation on a set of basis functions of the values

of the relevant state variables, in this case  $X_t = (S_t, v_t, r_t)$ . At each time  $t$  we assume that the unknown functional form of continuation can be specified as a linear combination of  $M$  basis functions. Given  $M = 10$  basis functions,  $B_m(X_{t,n})$  (Appendix A), for the regression, the estimate for the continuation value is

$$\hat{C}_{t,n} = \sum_{m=1}^M \alpha_{m,t}^* \cdot B_m(X_{t,n}), \quad (5.5)$$

where the optimal parameters,  $\alpha_{m,t}^*$ , in equation (5.5) are determined by the solution to the minimisation problem:

$$\min_{\alpha_{1,t}, \dots, \alpha_{M,t}} \frac{1}{N} \sum_{n=1}^N \left( (\exp(-dt \cdot r) V_{t_{i+1},n}(X_{t+1,n})) - \sum_{m=1}^M \alpha_{m,t} \cdot B_m(X_{t,n}) \right)^2. \quad (5.6)$$

Using the cross-sectional observations of the state variable generated by a Monte Carlo simulation, the parameter vector  $\alpha_{m,t}$  can be computed using the standard OLS formula. This allows us to determine the optimal stopping time for each path  $n$ .

The American option is then valued by starting at time zero, moving forward along each path until the first stopping time occurs, discounting the resulting cash flow from exercise back to time zero, and then taking the average over all paths.

We note, that when obtaining the regression coefficients, only ITM paths of the option are considered, as this increases efficiency of the algorithm and decreases computational time [Longstaff & Schwartz (2001)], since the decision whether or not to exercise the option is only relevant, when the option is ITM.

For each asset, at each date, we simulate 35,000 paths ( $N = 35,000$ ) discretized into  $I = 100$  time steps a year. The valuation of American options by LSM can be summarised by the following procedure (see Hilpisch (2011)):

---

**Algorithm 1** The LSM approach

---

- 1: **Input:**  $S_0, V_0, r_0, T, I, K, \kappa, \theta, \sigma, \kappa_r, \theta_r, \sigma_r$
- 2: Simulate  $N$  paths of the underlying asset  $(X_{t,n})$   $t = 0, 1, \dots, T$  and  $n = 1, 2, \dots, N$ .
- 3: At  $t = T$  set  $V_{T,n} \leftarrow \max(h_T(X_{T,n}), 0)$
- 4: **for**  $t = T - 1, T - 2, \dots, 1$  **do**
- 5:     Discount:  $V_{t+1,n}(X_{t+1,n}) \leftarrow \exp(-dt \cdot r) V_{t+1,n}(X_{t+1,n})$
- 6:     Use paths where  $h_t(X_{t,n}) > 0$  to compute coefficients from (5.6) and return  $\hat{C}_{t,n}$
- 7:     Compute  $V_{t,n} = \begin{cases} h_{t,n}(X_{t,n}) & \text{if } h_{t,n}(X_{t,n}) \geq \hat{C}_{t,n} \\ \exp(-dt \cdot r) V_{t+1,n}(X_{t+1,n}) & \text{if } h_{t,n}(X_{t,n}) < \hat{C}_{t,n} \end{cases}$
- 8:     continue until  $t = 0$ .
- 9: **end for**
- 10: **return**  $\hat{V}^{LSM} = \frac{1}{N} \sum_{n=1}^N V_{0,n}$

---

## 5.2 Improving Accuracy: Control Variates

The method of control variates is one of several techniques for improving the efficiency of Monte Carlo simulation. It is considered one of the most effective techniques within the field. The main idea of control variates is to exploit the estimation error of a known quantity which is correlated with the estimate of the variable of interest, in our case the American put option.

For options of American type a commonly used control variate is the European option counterpart. As such, we will use the European put option as the control variate in our Monte Carlo simulation with the LSM method. The value of the European put option is obtained using the closed form solution[Christomo (2014)]. The European counterpart can be considered a good choice as one would assume the two option types to be widely correlated. Moreover, the correlation is considered to be decreasing in the moneyness of the option. If the option is in the money to a larger extent, the importance of early possibility of exercise becomes more apparent. If an American option is exercised earlier, the correlation between the European option, which is always exercised at maturity (when in the money), becomes smaller. On the other hand an American option which is closer closer to being OTM is exercised at a later date closer to maturity and will to a larger degree mimic the European counterpart leading to higher correlation. E.g. in our case with the put option, we would expect a lower strike price to lead to fewer options in the money and hence a higher correlation and more improved accuracy. The true value of the European option is known, since there exists a closed form solution for European options in the Heston framework. Hence, the control variate error is known exactly for all paths.

When we have simulated  $N$  paths of the underlying asset to price the American put option, we also have  $N$  simulated present values of the European option, which are given by  $P_n^{LSM} = E_0^Q \exp(-rT) \max[K - s_T, 0]$ . Using the true model value of the European option,  $P^M$ , the control

variate estimator for an American put option with the European counterpart as control variate is:

$$\hat{V}^{CV} = \frac{1}{N} \sum_{n=1}^N (V_n^{LSM} - \hat{b}(P_n^{LSM} - P^M)). \quad (5.7)$$

As such, the control variate estimator adjusts the estimate from the LSM procedure by using the estimation error of pricing the European option with the LSM method. This estimation error is observable, since a closed form solution, and hence the true price of the European put option, exist. The adjustment of the estimate is given by the factor  $b$ . The choice of coefficient  $b$  is crucial when discussing the effectiveness of reducing the variance of the estimate of the American put option. The goal of the procedure is to reduce the variance of the estimates. Therefore, the optimal  $b$  minimises the variance of (5.7) and is estimated by  $\hat{b}$  which is given by

$$\hat{b} = \frac{\sum_{n=1}^N (P_n^{LSM} - \bar{P}^{LSM})(V_n^{LSM} - \bar{V}^{LSM})}{\sum_{n=1}^N (P_n^{LSM} - \bar{P}^{LSM})^2}, \quad (5.8)$$

where  $\bar{V}^{LSM}$  is the sample mean of the American put option price for all simulations and  $\bar{P}^{LSM}$  is the sample mean of the European put option price for all simulations. For the derivation of  $\hat{b}$  we refer to Appendix B.

### 5.3 Model Calibration

In a historical setting theoretical valuation is considered as the process where a range of input parameters of a formula of a pricing model determines the value of a financial derivative such as an American put option. The key assumption behind such reasoning, is that anything in the market that determines the value of such a financial derivative can be observed. A prominent example of this is the Black-Scholes (1973) formula which as input considers six variables, namely initial price level of the underlying asset, the underlying asset's volatility, the strike price of the option at hand, maturity, the short rate and dividends paid by the underlying asset. With numerical input for variables, the formula will return a value for the option.

When working with models of stochastic volatility such as the Heston-CIR model in our case, the input parameters cannot be directly observed in the market. The model is highly sensitive to input parameters, and therefore the process of correctly setting the parameter values is extremely important. The goal of the model is to price options that realistically reflect market prices. Therefore, the parameters must be such that the model output matches real world data. This procedure is formally known as the calibration procedure.

#### 5.3.1 Calibrating the CIR Model

In order to calibrate the parameters of the Heston-CIR model, such that real market prices can be replicated well, we must first turn our attention to the underlying process for the instantaneous short rate, namely the CIR-process. The goal is for this process to simulate the instantaneous short

rate according to real market data. Recall the SDE for the instantaneous short rate described in (3.1):

$$dr_t = \kappa_r(\theta_r - r_t)dt + \sigma_r\sqrt{r_t}dW_{r,t},$$

When modelling the instantaneous short rate realistically, the parameter values,  $(\kappa_r, \theta_r, \sigma_r, r_0)$ , should be calibrated according to market data. Once calibrated, these parameter values can be fed into the CIR-process, which the underlying asset depends on.

The task of the calibration procedure is for all times considered,  $t$ , and for the parameter set  $\Theta = (\kappa_r, \theta_r, \sigma_r, r_0)$ , to minimise the differences simultaneously:

$$\Delta f(0, t) = f(0, t) - f^{CIR}(0, t; \Theta), \quad (5.9)$$

where  $f(0, t)$  is the current market implied forward rate at time  $t$  and  $f^{CIR}(0, t; \Theta)$  is the current CIR-model implied forward rate at time  $t$  given the parameter set  $\Theta$ . However, the forward rates are not quoted directly in the market such that data is not readily available. Instead, the instantaneous forward rate at time  $t$  at time  $T$  can be defined by:

$$f(t, T) = -\frac{\partial B_t(T)}{\partial T}, \quad (5.10)$$

where  $B_t(T)$  denotes the price of a zero-coupon bond at time  $t$  with maturity  $T$ , where  $T > t$ , and paying one unit of currency at that date.  $f(0, t)$  is a special case and it holds for the short rate that  $r_t = f(t, t)$ . The relationship between zero-coupon bond yields for different maturities and forward rates follows [Baxter & Rennie (1996)]:

$$f(0, T) = Y(0, T) + \frac{\partial Y(0, T)}{\partial T} \cdot T, \quad (5.11)$$

where  $Y(0, T)$  is the yield today of a bond with maturity  $T$ . The (continuous) yield for a zero-coupon bond solves the equation:

$$\begin{aligned} B_T(T) &= B_0(T)\exp(Y(0, T) \cdot T) \Leftrightarrow \\ Y(0, T) &= \frac{\log B_T(T) - \log B_0(T)}{T}. \end{aligned}$$

Normalising the final value of the bond to 1, we get:

$$Y(0, T) = -\frac{\log B_0(T)}{T}. \quad (5.12)$$

As such, relying on zero-coupon bond data allows as to estimate the partial derivative in (5.12), such that the calibration process for the CIR-model can be carried out.

Unfortunately, zero-coupon bond rates are quoted for selected discrete maturities. To be able to calculate the partial derivative in (5.12) the data must instead be continuous. Therefore, we interpolate between the single data-points of several discrete maturities. The interpolation process,

however, should be based on continuous annualised rates (continuous yield of a unit zero-coupon bond) rather than rates with different maturities (Hilpisch (2011)). Consider for instance a 31 day zero-coupon bond rate of 1.5%. The corresponding factor is:

$$f_s^{31d} = 1 + \frac{31}{365} \cdot 0.015,$$

with the equivalent continuous annualised rate equal to:

$$f_c^{31d} = \frac{365}{31} \cdot \log(L_s^{3m}).$$

This ensure that the following holds:

$$1 + 31/365 \cdot f_c^{31d}$$

This annualisation is done for all discrete maturities. Now we interpolate to get continuous data in order to calculate the partial derivative and hence forward rates for arbitrary times  $T$ . Specifically, we use cubic splines regression [Brandimarte (2006)].

In order to calibrate the CIR-model to the forward rates, we consider the Mean Squared Error (MSE) of the market implied and model implied forward rate curve. At discrete points in time this is minimised. Given an initial short rate,  $r_0$ , and equidistant spacing in  $[0, T]$  by  $dt$  with  $M = \frac{T}{dt}$ , we have the following minimisation problem:

$$\min_{\Theta} \frac{1}{M} \sum_{m=0}^M (f(0, mdt) - f^{CIR}(0, mdt; \Theta))^2, \quad (5.13)$$

where  $f(0, mdt)$  is the current market implied forward rate for the discrete time step  $mdt$  and  $f^{CIR}(0, t; \Theta)$  is the current CIR-model implied forward rate for the discrete time step  $mdt$  given the parameter set  $\Theta$ .

### 5.3.2 Calibrating the Heston Model

Calibrating the Heston-CIR model is the process of finding parameter values that yield model price predictions that are as close to market observed prices of the American put option. Since the preceding step is the calibration of the CIR-model for the instantaneous short rate, the parameters found in this procedure are fed into to the Heston-CIR model. In short, when calibrating the Heston-CIR model we wish to minimise the difference between the real put option prices and the price predictions by the model by finding the set of parameters,  $\Psi = (v_0, \kappa, \theta, \sigma, \rho)$ , which minimises this difference. This difference can be characterised by a range of error functions. Christoffersen & Jacobs (2004) argue that the objective itself, e.g. valuation or hedging, plays a role when choosing which error function is minimised. In accordance with their findings, we consider the MSE, which

they find performs best across different error functions.

$$\min_{\Psi} \sum_{j=1}^J (V_j^{Market} - V_j^{CV})^2, \quad (5.14)$$

where  $J$  is the number of options used for calibration,  $V_j^{Market}$  is the market price of the American put option,  $V_j^{CV}$  is the Heston-CIR model price with control variates of American put option  $j$  and  $\Psi$  is the vector of model parameters with  $\Psi = (v_0, \kappa, \theta, \sigma, \rho)$ .

## 6 Neural Networks for Option Pricing

One of the main advantages of considering a NN approach is that it is not necessary to assume any underlying process for the data unlike classical model-based simulation methods. A NN approach is non-parametric and entirely data-driven.

In line with Albargouthi (2021), we first define NNs generally. The shape of these NN models is referred to as the architecture of the network. The architecture is determined by a combination of different layers that consist of numerous artificial neurons. In the network, each node of the model performs an operation. We consider the simplest model that is composed of a single layer of a single neuron.

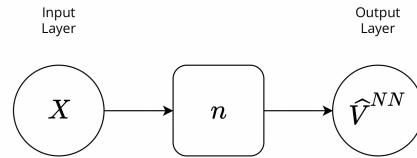


Figure 6.1: A Very Simple Neural Network

The input node passes the input,  $X$ , to node  $n$ , which is the neuron. An operation on the input is performed and the output, in our case the price of the American put option,  $\hat{V}^{NN}$ , is computed to the output node. This is the very simple idea behind a NN, but in reality the operation within the neuron is rather complex. We do not only consider a single input node, but rather a range of  $K$  inputs,  $X_K$ , that determine the price of an American put option. Specifically, the input vector,  $X_K$ , for each option consists of the strike price of the option, the maturity of the option, the closing price of the S&P 500, the interest rate and measures of volatility.

The neuron consists of three consecutive operations: a summation of weighted inputs, the addition of bias,  $b$ , to this summation and an activation function,  $\phi$ , which allows for the calculation of the output. This is illustrated with the simple figure of a neuron below.

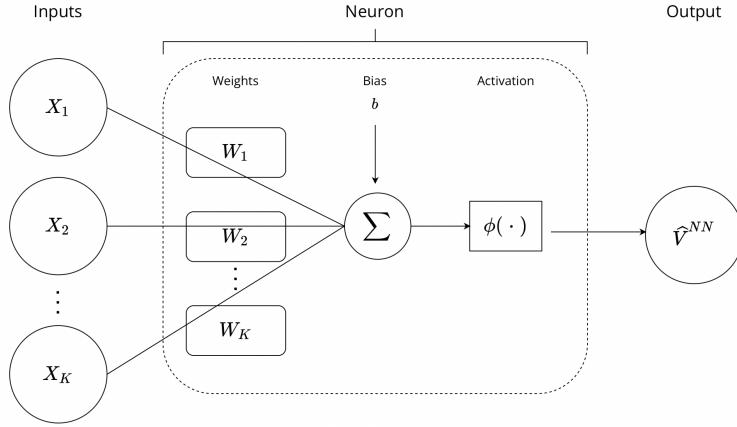


Figure 6.2: A Neuron

The activation function transforms the weighted input after the addition of bias to output of the neuron. The activation function plays an essential role as it is used to add non-linearity into the NN and directly controls the output of each neuron.

## 6.1 Activation Functions

In case of an activation function not being used in a NN the network is linear. More specifically, in order for the NN model to be able to approximate a non-linear function, the activation function must also be non-linear. Linear functions are simple and easily solvable, but they are limited when it comes to solving complex relationships between the input vector, \$X\_K\$ and the output, \$\hat{V}^{NN}\$. The network we consider in order to price American put options will persist of the following activation functions as suggested by Culkin & Das (2017).

The ReLU activation function is considered the most popular activation function and was introduced by Hahnloser et al. (2000). The function returns the input value directly in case the input value is strictly positive or zero in case the input value is less than or equal to zero.

$$ReLU(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \quad (6.1)$$

Given the simplicity of the ReLU activation function it is computationally efficient. However, this simplicity does come with a drawback. In case of input values less than or equal to zero, the output will also be zero and the NN cannot continue its computation and "dies", commonly referred to as the "dead ReLU problem".

The Leaky ReLU activation function is similar to the ReLU activation function. The difference

is that for input values less than or equal to zero the slope,  $\alpha$ , is between 0 and 1 [Liu et al. (2019)],

$$\text{LeakyReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0, \end{cases} \quad (6.2)$$

where  $0 < \alpha < 1$ . This specification solves the dead ReLU problem but comes at a computational cost.

The Exponential Linear Unit (ELU) activation function is similar to the Leaky ReLU as negative inputs are transmissible through the neuron, however negative or zero values take on a different transformation

$$\text{ELU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(\exp(x) - 1) & \text{if } x \leq 0 \end{cases} \quad (6.3)$$

The final relevant activation function is the exponential activation function. The exponential function does not allow for negative output values.

$$\text{EXP}(x) = \exp(x) \quad (6.4)$$

This characteristic of non-negativity is particularly relevant when pricing options, since option prices cannot be negative. Therefore, we apply the exponential activation function when computing the output layer in order to guarantee non-negative American option prices,  $\hat{V}^{NN}$ .

## 6.2 Multilayer Perceptron

Now we have introduced the main ingredients in order to form the NN such that we are able to estimate American put option prices.

It is not realistic that the simple example with a single layer consisting of a single neuron is able to price an American put option,  $i$  for  $i = (1, 2, 3, \dots, I)$ . Therefore, we expand the network to consist of several layers and neurons. When designing the NN, it is important to consider in which way nodes are interconnected. The way in which nodes are connected determines how computations proceed and constitutes an important early design decision for a NN developer. This form of network architecture with several layers is called multilayer perceptron. The goal of our NN model is to price the American put option,  $i$ , for a range of American put options  $i = (1, 2, 3, \dots, I)$ . The input layer consists of the set of  $K$  inputs,  $X_{k,it}$ , where

$$X_{k,it} = (X_{1,it}, X_{2,it}, \dots, X_{K,it}), \quad (6.5)$$

which is specific to option  $i$  at time  $t$ .

Such a multilayer perceptron is illustrated below in Figure 6.3, which is a model which represents a nonlinear relationship between the input vector,  $X_{k,it}$ , and the output vector  $\hat{V}_{it}$  for input variable  $k$  and option  $i$  at time  $t$ :

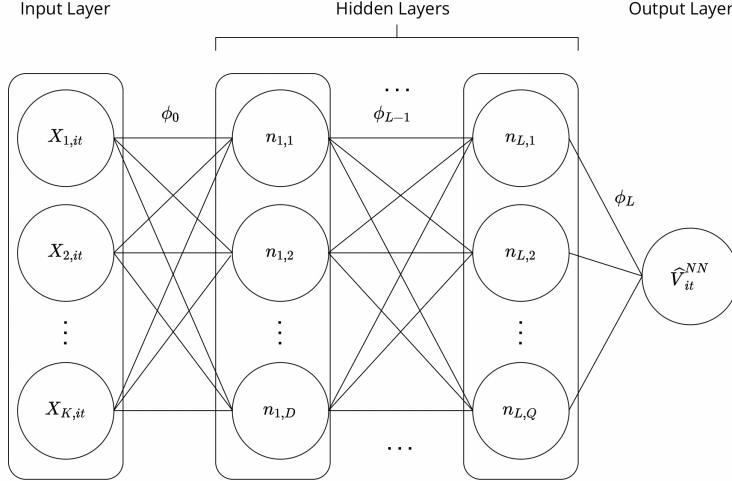


Figure 6.3: A Multilayer Perceptron With  $L$  Hidden Layers

The layers between the input and output layers are referred as the hidden layers. The first hidden layer,  $n_{1,d}$  for  $d = (1, 2, 3, \dots, D)$  consists of  $D$  neurons. The  $L$ th hidden layer,  $n_{L,q}$  for  $q = (1, 2, 3, \dots, Q)$  consists of  $Q$  neurons. By connecting the neurons of adjacent layers, output signals of the previous layer enter the next layer as the input signal. Between each hidden layer the output of each layer is transferred to the next layer as input with an activation function of the kind introduced above. Starting with the first connection between the input layer,  $X_{k,it}$ , and the first hidden layer,  $n_{1,d}$ , each hidden node is obtained as

$$n_{1,d} = \phi_0 \left( \sum_{k=1}^{N_0} w_{d,k} X_{k,it} + b_d \right), \quad (6.6)$$

where  $N_0$  is the number of input variables each hidden node is fed,  $w_{d,k}$  is the weight of the input layer  $k$  with respect to the hidden node  $d$ ,  $b_d$  is the bias and  $\phi_0$  is the activation function.

The  $(L - 1)$ th hidden layer,  $n_{L-1,s}$  for  $s = (1, 2, 3, \dots, S)$  consists of  $S$  neurons. For the final connection between hidden layers, which is between the  $L$ th hidden layer and the  $(L - 1)$ th hidden layer, each hidden node is obtained as

$$n_{L,q} = \phi_{L-1} \left( \sum_{s=1}^{N_{L-1}} w_{q,s} n_{L-1,s} + b_q \right), \quad (6.7)$$

where  $N_{L-1}$  is the number of inputs from the previous hidden layer,  $L - 1$ , each hidden node is fed,  $w_{q,s}$  is the weight of the previous hidden layer  $L - 1$  with respect to the hidden node  $q$ ,  $b_q$  is the bias and  $\phi_{L-1}$  is the activation function.

It has been shown that the multilayer perceptron can be trained to approximate any continuous function for certain activation functions. This is known as the universal approximation theorem

[Hornik et al. (1989)]. In fact, any non-polynomial activation function works [Leshno et al. (1993)].

In a multilayer perceptron, the output of each node is fed forward to be the input of the nodes in the next layer of the network after being scaled by the connecting weight. Therefore, there is a clear direction in which the information is being processed. Given the forward direction, the multilayer perceptron is considered a feed-forward NN, which is one of the simplest forms of an artificial NN. The output layer calculates a weighted sum of all the outputs from the neurons of the final hidden layer, layer  $L$ , in order to estimate the price of the American put option,  $\hat{V}_{it}^{NN}$  and is given by

$$\hat{V}_{it}^{NN} = \phi_L \left( \sum_{q=1}^{N_L} v_q n_{L,q} + b \right), \quad (6.8)$$

where  $N_L$  is the number of inputs from the previous hidden layer,  $v_q$  is the weight of the node  $n_{L,q}$ ,  $b$  is the bias and  $\phi_L$  is the activation function.

### 6.2.1 The Backpropagation Algorithm

We are interested in a NN which is able to model the relationship between the input vector,  $X_{k,it}$ , and the American put option price accurately. In order for the model to accurately price the American put option, we introduce a procedure called model training. By this procedure, the network learns which weights and biases minimise the measurement errors between the NN estimates for the American put option price and observed put option prices.

For simplicity, we consider an example of a multilayer perceptron with only two weights. In this simple case, the network error can be plotted as an error surface for given weights in a three-dimensional space as presented below.

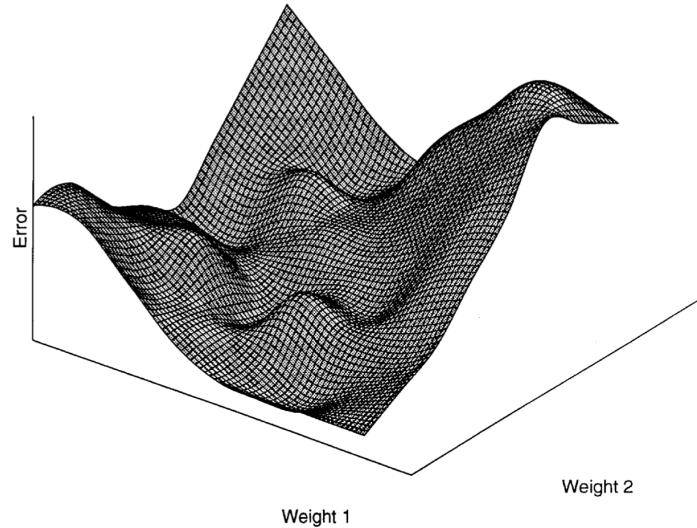


Figure 6.4: An Error Surface for a Simple Multilayer Perceptron Containing Two Weights [Gardner & Dorling (1997), Figure 4]

Since we in practice have a large number of weights it is not possible to plot an error surface like the one above. By training, we wish to select the combination of weights and biases that produce the smallest error which is the global minimum of the error surface. The error is given by the MSE

$$C = \frac{1}{N} \sum_{i=1}^N (V_{it}^{Market} - \hat{V}_{it}^{NN}(\Gamma))^2,$$

where  $N$  is the number of observations for option  $i$  at time  $t$  for the set of weight and biases,  $\Gamma$ . A possible method is to use backpropagation [Rumelhart et al. (1986)], which uses a procedure known as gradient descent. The idea is to randomly choose a set of weights at step  $z = 0$  which corresponds to selecting a random point on the error surface. The local gradient at that specific point is calculated and the weights and biases,  $\Gamma$ , are updated in the direction of the steepest local gradient. Repeating this process iteratively, the weights and biases will converge to the global minimum of the error surface. For each observation at step  $z$  weights,  $w$ , and biases,  $b$ , are updated by

$$\Delta w(z) = -\alpha \nabla w(z)(z-1), \quad \Delta b(z) = -\alpha \nabla b(z)(z-1), \quad (6.9)$$

where  $\alpha$  is the learning rate and the local gradient,  $\nabla w$  or  $\nabla b$ , can be obtained by applying the chain rule for a single weight,  $w$

$$\nabla w = \frac{\partial C(\hat{V}^{NN}(\Gamma))}{\partial w} = \frac{\partial C(\hat{V}^{NN}(\Gamma))}{\partial Y} \frac{\partial Y}{\partial X} \frac{\partial X}{\partial w}, \quad (6.10)$$

where  $X$  is the input of each NN neuron and  $Y$  is the output of each NN neuron.

The backpropagation algorithm for a single weight,  $w$ , can be summarised in the following steps.

---

**Algorithm 2** The backpropagation algorithm

---

- 1: initialise network weights,  $w(z = 0)$
  - 2: present first input vector to the network
  - 3: propagate the input vector through the network to obtain an output
  - 4: calculate an error signal by comparing actual output to the desired (target) output,  $C(z)$
  - 5: propagate error signal back through the network
  - 6: adjust weights to minimise overall error according to (6.9)
  - 7: repeat steps 2—7 with next input vector, until overall error is satisfactorily small
- 

The backpropagation algorithm contains an adjustable parameter of significance, the learning rate,  $\alpha$ . The learning rate determines the step size of the weight adjustment during the iterative gradient descent process and can assist in the algorithm not getting stuck in a local minimum. In case the learning rate is too small, training will either take too long to converge or get stuck in a local minimum. On the other hand, if it is too large there is a possibility that the training process will skip the global minimum in the error surface, since the error will change too erratically as the

weight changes are large.

## 7 Research Setup and In-sample Results

The goal for each approach is to estimate American put option prices that are close to the observed market prices. Parametric approaches such as LSM are highly dependent on input parameters and different sets of parameters can produce widely different estimates of option prices. We consider option price data for three dates, namely January 3rd 2017, June 29th 2017 and December 29th 2017. For each date we select a range of put options on the S&P 500 with different strike prices and maturities. The parameters in the Heston-CIR model are calibrated to this data according to the procedure described in Section 5.3. Following the calibration procedure we evaluate the in-sample pricing performance of the LSM approach at each date.

Analogously, in order for the NN approach to accurately estimate American put option prices we train the NN model according to Section 6.2.1. Since the training procedure requires much more data in order for the model to achieve an accurate fit, we consider in-sample data for all of 2017 rather than a select few options at discrete dates. We are unable to observe the optimal weights and biases unlike for LMS procedure, where we directly observe the optimal parameters. As such, the NN approach constitutes a "black box". In addition, we must consider which NN model specifications yield the best pricing results. As such, we carry out a range of model selection steps in order to determine which activation functions to use, how many layers the network should consist of and which inputs should be considered.

We implement both approaches in Python Version 3.7.4. For the NN approach we use the TensorFlow and Keras modules. The relevant code documentation can be found in Appendix E.

### 7.1 In-sample Data and Data Pre-processing

The two methods are dependent on a solid data foundation in order to be able to price American put options realistically. Therefore, we present the in-sample data which is used in order to calibrate and train the two methods, respectively. However, before doing so the data must be prepared so as to remove non-representative data-points that lead to pricing inaccuracies.

The option data considered are put options of American style for S&P 500. Options on the index have the benefit of being among the most popular and widely used by investors, speculators, and hedgers. As such, this option market is one of the most liquid and therefore plenty of data-points each day are available of options with high trading volume. Since the S&P 500 reflects the 500 largest publicly traded U.S. stocks it captures the market situation in the U.S. and arguably worldwide due to the broad nature of the index. Moreover, it is not as susceptible to industry-specific shocks as compared to options traded on single company stocks. However, only options of European type are traded directly on the S&P 500. Instead, when trading American options on the S&P 500,

the underlying asset is an Exchange Traded Fund (ETF) which tracks the S&P 500 index, denoted SPY. The objective of tracking the S&P 500 is achieved by holding a portfolio of the common stocks included in the fund, with the weight of each of stock in the fund corresponding to the weight of each stock in the index itself. Specifically, we consider data for the year 2017 since this represents a period with no notable shocks to the market. Before the data cleaning process the original data set consists of 611,836 options across the year i.e. 252 trading dates. Data for the price of the American put option is acquired from the OptionMetrics database and consists of the average of the daily highest bid and daily lowest ask for selected maturities and strike prices.

In addition to the option prices themselves, data for the underlying asset and the instantaneous interest rate is necessary in order to apply the Heston-CIR model. As such, we need closing quotes for the SPY index as well as data for a benchmark rate. Data for closing quotes of the SPY are gathered from Yahoo! Finance. For the instantaneous interest rate we consider data for the zero-coupon bond yield at select maturities as reported by IvyDB. This metric has the advantage of being quoted for several maturities ranging from 1 week to 10 years. Closing quotes for SPY and interest rate data also figure as input in the NN architecture i.e.  $X_{k,it}$  in (6.5) discussed later.

To avoid certain biases in the data set various exclusion criteria are applied. Options priced lower than 10 basis points i.e.  $V_i^{Market} \leq 0.1$  are excluded from the data set. Anders et al. (1998) argue that not doing so leads to relatively high deviations between theoretical and market prices when option prices are very low. Options tend to be traded at whole numbers or numbers rounded at one or two decimal points. OTM options traded at the lowest quoted price have varying degrees of moneyness e.g. an option with moneyness -1.1% and an option with moneyness -0.25% are both quoted at a price of 0.05. This leads to pricing inaccuracies. Therefore, we also choose to exclude deep OTM options, where the moneyness of the put option is greater than 2.5.

In line with the literature a liquidity filter is applied. Gaspar et al. (2020) exclude options with less than 20 trades per trading day. We apply the same restriction to trading volume per option so as to ensure there is no liquidity related bias. Finally, we exclude all options with zero maturity, as some options trading on the last execution day are included during the data gathering process. From the original data set of 611,836 options across the year we end up with a data set consisting of 145,842 options.

Figure 7.1 presents the daily closing prices of the underlying asset, the SPY. The index starts in \$225.24 and trends upwards throughout the period of interest with the highest recorded closing value of \$268.2 at the end of the period. There are no notable breaks in the upward trend of the time-series.



Figure 7.1: SPY Daily Closing Price 1/1/2017 - 31/12-2017

Based on the date in Figure 7.1 the daily returns for the SPY are computed and presented in Figure 7.2 below. The average daily return is 0.06%, and we note there are no absolute returns greater than 1.77%. Moreover, from the figure it is evident that larger absolute returns of more than 1% are few.

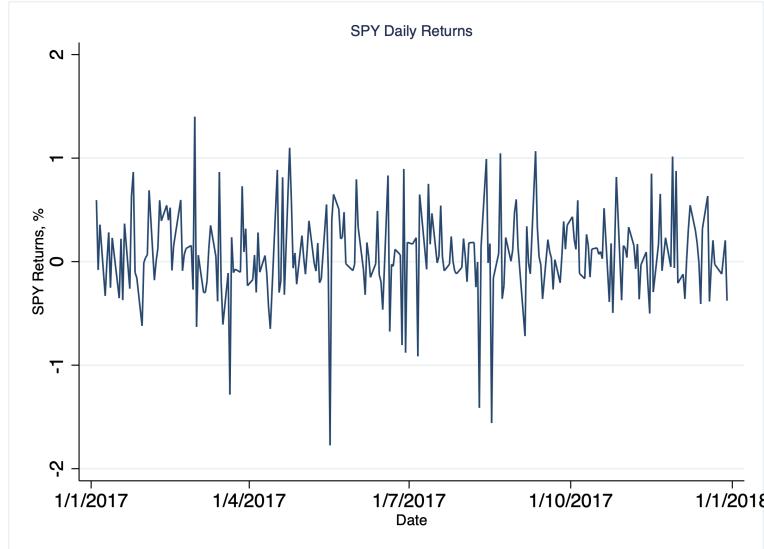


Figure 7.2: SPY Daily Returns, %, 1/1/2017 - 31/12-2017

Another relevant factor for option pricing is volatility of the underlying asset. Typically, periods of high volatility are correlated with low/negative returns. This is captured in the Heston-CIR model as the leverage effect introduced earlier. Following the lead of Culkin & Das (2017) we consider the

historical volatility of the SPY. The historical volatility is computed as:

$$\sigma_{HV} = \sqrt{\frac{1}{T-1} \sum_{t_1}^T (r_t - \bar{r})^2}, \quad (7.1)$$

where  $\sigma_{HV}$  is the historical volatility estimate,  $T$  is the number of periods over which the volatility is measured and  $\bar{r}$  is the average of the stock returns,  $r_t$ , over the time-period in question.

In addition, we consider the volatility index (VIX), commonly referred to as the index of fear, as an alternative measure for volatility. VIX is specifically based on the S&P 500 and can therefore be considered a relevant measure in our case. The two volatility measures are illustrated below with  $\sigma_{HV}$  and VIX on the left and right axis respectively:

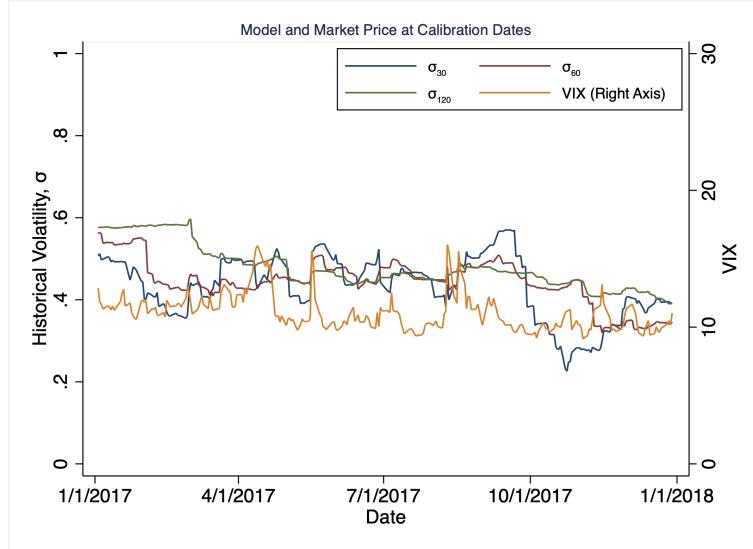


Figure 7.3: Historical Volatility,  $\sigma_{HV}$  (left axis), and VIX (right axis) 1/1/2017 - 31/12-2017

Figure 7.4 presents the zero-coupon bond yield at three selected maturities, namely 3 months, 1 year and 2 years. In general, the rates are stable throughout the period of interest. However, the short rate trends upwards throughout. In the final quarter of 2017 all rates trend upwards and increase by about 50 basis points.



Figure 7.4: Zero-Coupon Bond Yields at Selected Maturities

Below put prices for selected maturities and strike prices are presented. Specifically, the figure depicts the distribution of put option prices against the price of the underlying asset, SPY, for the strike prices \$240, \$250, \$260 and \$270. We clearly observe the ramp function shape which is typical for option prices. This shape shows a negative relationship between the value of the put option and the price of the SPY. This is just as expected, since a larger price of the underlying asset requires a larger outlay for the put option holder when exercising the option. Moreover, we note that long maturity options in general record higher prices. The lowest strike price recorded in the data is \$95 and the highest is \$410.

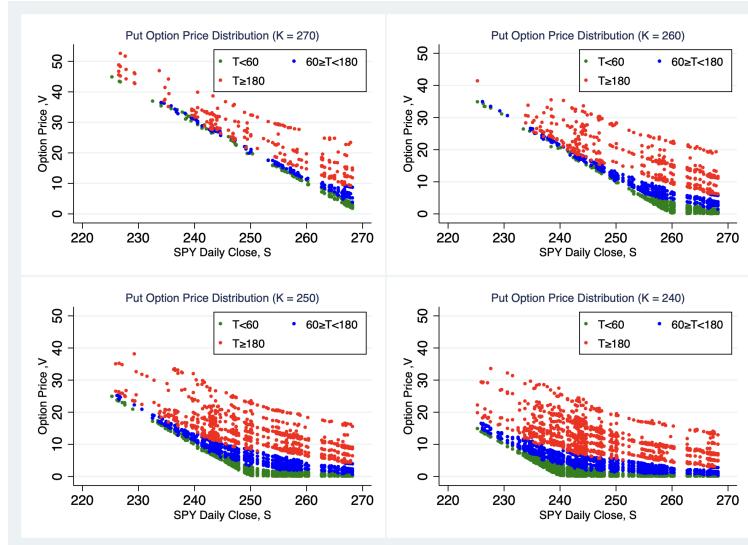


Figure 7.5: Put Option Price Distribution at Selected Maturities

As introduced, the degree of moneyness,  $m$ , for each option is calculated by normalising the underlying asset price by the strike price, such that  $m_i = S_i/K_i$  for option  $i$ . The vast majority of the options in the data set have moneyness in the range between 0.8 and 1.3, specifically 93.8% of the data falls in this range of moneyness. Following Gradojevic et al. (2009), who examine Modular Neural Networks for option pricing, we consider options to be ITM when  $m < 0.97$ , NTM when  $0.97 \leq m < 1.05$ , and OTM when  $m \geq 1.05$ . In similar fashion, another division in the data is considered based on time to maturity,  $T$ . The options are split into three maturities such that options are of short maturity when  $T < 60$  and of long maturity when  $T \geq 180$ . This separates short and long maturity options by a large subset of medium maturity options.

Figure 7.6 presents the distribution of the put options' moneyness in the range of moneyness between 0.8 and 1.3 for short, medium and long term maturities. The division between ITM, NTM and OTM options is graphically illustrated by the vertical lines at  $m = 0.97$  and  $m = 1.05$ , respectively.

Two clear patterns emerge: Firstly, the value of the option is increasing in time to expiration as was clear in Figure 7.5. Secondly, the option price is increasing in the option's degree of moneyness. This is just as expected, as options with higher moneyness have a higher probability of being exercised at a higher payoff,  $K - S$ .

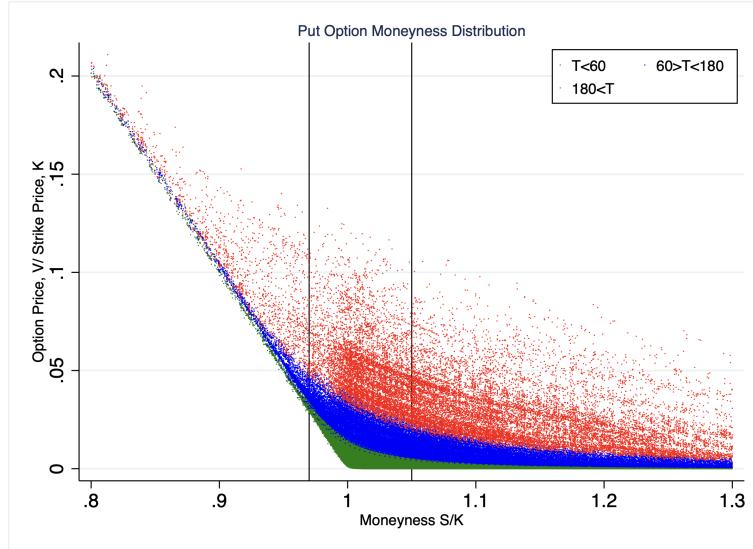


Figure 7.6: Put Option Moneyness Distribution at Selected Maturities

Summary statistics for the above defined data set division are presented in Table 7.1 below. The largest subset of data is for options NTM, which accounts for just over 50% of the data. Moreover, a vast majority of the NTM options are short maturity options. OTM options also account for a large share of the data set, and the subset is fairly well balanced in terms of maturity. Overall, option prices are increasing in moneyness and maturity as expected from the graphical analysis above.

Days to expiration, $T$	$T < 60$	$60 \leq T < 180$	$T \geq 180$	All Options
Moneyness, $m (S/K)$	Average Option Price			
ITM ( $m < 0.97$ )	29.229	26.465	42.690	32.144
NTM ( $0.97 \leq m < 1.05$ )	1.817	5.091	11.848	3.400
OTM ( $m \geq 1.05$ )	0.323	1.260	4.648	1.925
Moneyness, $m (S/K)$	# Observations			
ITM ( $m < 0.97$ )	2,407	2,359	1,936	6,702
NTM ( $0.97 \leq m < 1.05$ )	53,391	13,390	7,329	74,110
OTM ( $m \geq 1.05$ )	20,894	25,594	18,542	65,030

Table 7.1: SPY Put Options Summary Statistics for Varying Degrees of Moneyness and Selected Maturities

## 7.2 Least-squares Monte Carlo

Now we focus on the LSM approach for estimating American put option prices introduced in Section 5. First, we consider the performance of the model for the in-sample period, 2017. The first step consists of the model calibration procedure introduced in Section 5.3.

### 7.2.1 Calibration Results & In-sample Performance

In order to price options with the Heston-CIR model, we must first be able to simulate the instantaneous rate,  $r_t$ , according to (4.11). As such, we calibrate the CIR-model to the zero-coupon bond rates introduced in the previous section. Recall, that this calibration procedure consists of the minimisation problem in (5.13). The calibration procedure is performed at 3 selected discrete points in time, namely January 3rd 2017, June 29th 2017 and December 29th 2017. The calibration results for the three dates are presented in Table 7.2 below.

Date of calibration	$\kappa_r$	$\theta_r$	$\sigma_r$	$r_0$	MSE
03/01/2017	0.420	0.063	0.001	0.007	7.924e-07
30/06/2017	0.216	0.064	0.001	0.012	5.011e-08
29/12/2017	0.444	0.060	0.231	0.015	5.575e-07

Table 7.2: Calibration Results for the CIR-Model at Selected Dates

We note, that the CIR-model is able to simulate the instantaneous rate precisely at all three dates for the parameters obtained by calibration, with a MSE no larger than 7.924e-07.

The long-term average,  $\theta_r$ , is consistent for all three dates. The same is observed for the volatility of the short rate. However, the parameter is significantly larger for the final date considered. The speed of mean-reversion is also fairly consistent, although it is lower for 30/06/2017. Below, the model and market implied forward for 30/06/2017 are plotted against each other. This graphically illustrates

the precision of the forward rates predicted by the CIR-model using the parameters obtained by calibration at this date.

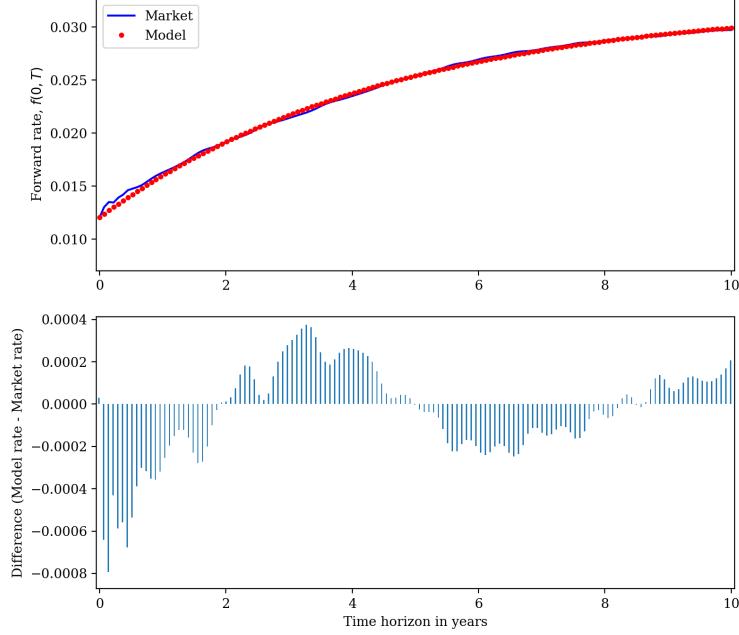


Figure 7.7: Model and Market Implied Forward Rates, 30/06/2017

Once we have the calibrated parameters for the CIR-model, they are fed into the Heston-CIR model such that the instantaneous rate for each date can be simulated precisely and the calibration process for the Heston-CIR model is performed.

Recall, that the calibration procedure of the Heston-CIR model corresponds to solving the minimisation problem in (5.14). Therefore, the goal of the calibration procedure is to minimise the MSE between observed market prices for American put options and the put option prices estimated by the Heston-CIR model for the same set of options on a given day. Calibrating the model to observed market prices takes place by changing the model parameter vector,  $\Psi = (v_0, \kappa, \theta, \sigma, \rho)$ . For consistency, the calibration procedure is performed for the same three dates as for the CIR-model. In practice, the calibration procedure consists of two steps. In the first step a brute force optimiser is considered. For this step a starting interval and a step-size for the five parameters we wish to calibrate the model to must be specified. Specifically, we consider the same specifications as Hilpisch (2011), who calibrates a Heston model to EURO STOXX 50 options. For every combination of parameters within the five parameter intervals, the corresponding MSE is determined in order to scan for potential regions where the minimum could be. The process of determining option prices and the corresponding MSE for each parameter combination is computationally expensive, and therefore the trade-off between computation time and the size of the interval and steps should be considered. The second step consists of local optimisation. Specifically, we consider the built-in python optimiser `fmin`, which can be used to minimise non-convex, non-linear functions. The resulting parameters

from the first step are used as starting values for the optimiser.

For the optimisation problem of the objective function in (5.14), it is worth noting there is a possibility of the function having more than one global minimum. In addition, even in the case of the function only having one global minimum, it is unclear whether or not the unique minimum can be reached, and the possibility exists that the found minimum is not the global minimum but rather a local minimum. For a comprehensive discussion of these issues as well optimisation techniques we refer to Milan Mrázek and Jan Pospíšil [Mrázek & Pospíšil (2017)].

The above introduced calibration procedure is performed for the three dates in question. For each date we consider a range of strike prices so as to calibrate varying degrees of moneyness as well as three maturities varying between about 2 weeks and just shy of 6 months. The option characteristics at each date are presented in the table below:

Date of calibration	Maturities, $T$	Strikes, $K$	# Options	Mean Option Price, USD (\$)	$S_0$
03/01/2017	(17,73,164)	[205,250]	77	4.36	225.24
30/06/2017	(14,77,168)	[205,250]	70	3.80	241.80
29/12/2017	(14,77,168)	[240,275]	81	3.41	267.03

Table 7.3: Option Characteristics at Calibration Dates

In order to evaluate the in-sample performance, we consider the two pricing errors mean absolute error (MAE) and the relative mean absolute error (RMAE), which are defined as:

$$MAE = \sum_i^I |V_i^{Market} - V_i^{MODEL}| \quad (7.2)$$

$$RMAE = \sum_i^I \frac{|V_i^{Market} - V_i^{MODEL}|}{V_i^{Market}}, \quad (7.3)$$

where  $I$  is the number of options priced,  $V_i^{Market}$  is the observed put option price for option  $i$  and  $V_i^{MODEL}$  is the model predicted price by either approach for option  $i$ . The advantage of the MAE is that it can be interpreted as the average USD (\$) deviation from the true option price. The RMAE can in turn be interpreted as the percentual deviation from the true option value. The calibration results i.e. the parameter vector,  $\Psi = (v_0, \kappa, \theta, \sigma, \rho)$ , which minimises the MSE at the selected dates for the selected options are presented in Table 7.4 along with the two pricing errors.

Date of calibration	$v_0$	$\kappa$	$\theta$	$\sigma$	$\rho$	# Options	MAE	RMAE
03/01/2017	0.002	11.264	0.035	0.835	-0.914	77	0.147	0.124
30/06/2017	0.002	10.926	0.025	0.742	-0.996	70	0.203	0.147
29/12/2017	0.002	7.721	0.023	0.586	-0.999	81	0.249	0.180

Table 7.4: Calibration Results for the Heston-CIR Model at Selected dates

From Table 7.4 we firstly note, that the MAE is lower than 0.249 for all three calibration dates, with the highest MAE recorded on 29/12/2017 with a relative pricing error of 18%. In general, the option prices that result from the Heston-CIR model with parameters obtained by the calibration procedure replicate market prices well. This is particularly the case on the first calibration date, 03/01/2017, where the lowest pricing errors are recorded. For this date market prices are replicated well both in absolute terms and relative to the mean option price in the calibration sample, which is the highest at this date.

Overall, the parameters obtained by calibration for the three dates are consistent and similar in magnitude. This is particularly the case for the three parameters  $v_0$ ,  $\theta$  and  $\rho$ . We note that  $\rho < 0$  for all dates in line with expectation, meaning that the stock price and variance are negatively correlated such that the model captures the leverage effect discussed in Section 3.2. In addition, the consistency in the parameter  $\theta$  indicates the long-term average of the variance of the SPY is stable throughout the time period.

On the other hand, the volatility of the variance,  $\sigma$ , is slightly more volatile than the three aforementioned parameters, although it is in a fairly narrow range between just below 0.6 and just over 0.8. The parameter which exhibits the largest fluctuations in magnitude is  $\kappa$ , which captures the speed of mean reversion towards the long-term average of the variance,  $\theta$ . This is the case for the final date of calibration, 29/12/2017, where  $\kappa$  is noticeably smaller than for the two other dates. Since the initial variance,  $v_0$ , is below the long-term average,  $\theta$ , a smaller  $\kappa$  will lead to a higher value of the underlying asset through the leverage effect and therefore lower put option prices.

For the three dates we plot the observed market prices and the option prices predicted by the model using the parameters obtained by calibration at each selected date at selected strike prices. This is done for all three maturities.

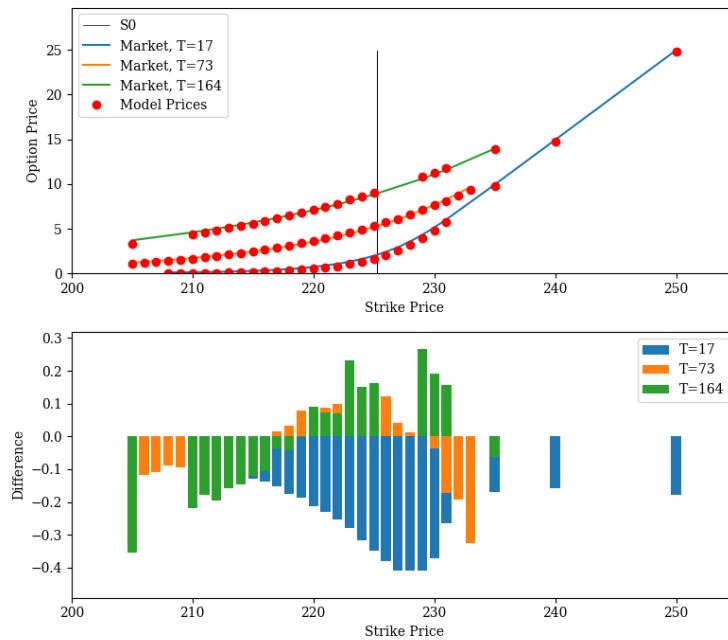


Figure 7.8: Market and Model Prices for SPY Options, 03/01/2017

For the first date in question, we graphically see that the market observed prices are replicated extremely well across all maturities and strike prices. There is a general tendency for the model to underestimate option prices as indicated by the differences between the model and market prices. The largest deviations are for the short maturity options with the largest deviation of about 40 cents. Moreover, OTM and NTM options record the largest absolute deviations between market prices and model prices. Since lower moneyness options have lower market prices, the pricing errors are relative larger for options of this specification.

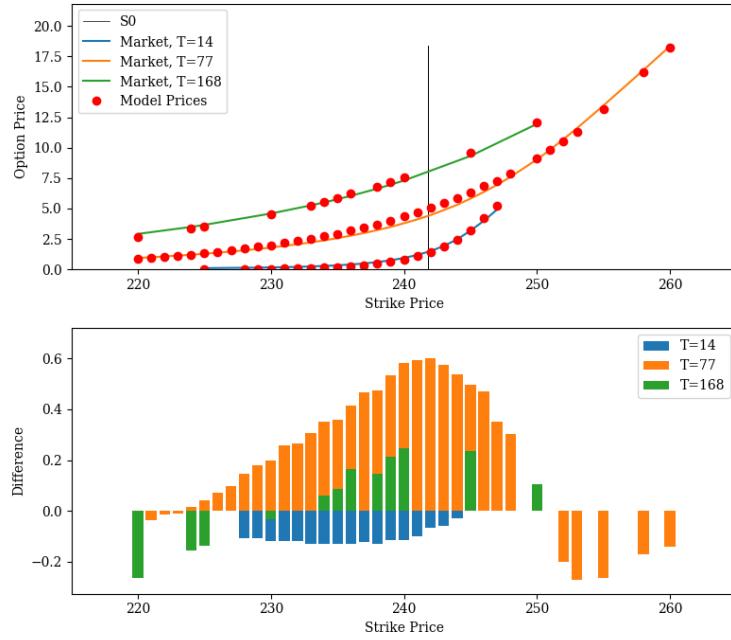


Figure 7.9: Market and Model Prices for SPY Options, 30/06/2017

For the second date in question we observe larger deviations between observed market prices and model prices with the largest deviation of about 60 cents. In particular, the pricing errors are large for the medium maturity options,  $T = 77$ . However, the market prices are replicated accurately for the short and long maturity options. Overall, the model at this date tends to overestimate the option prices.

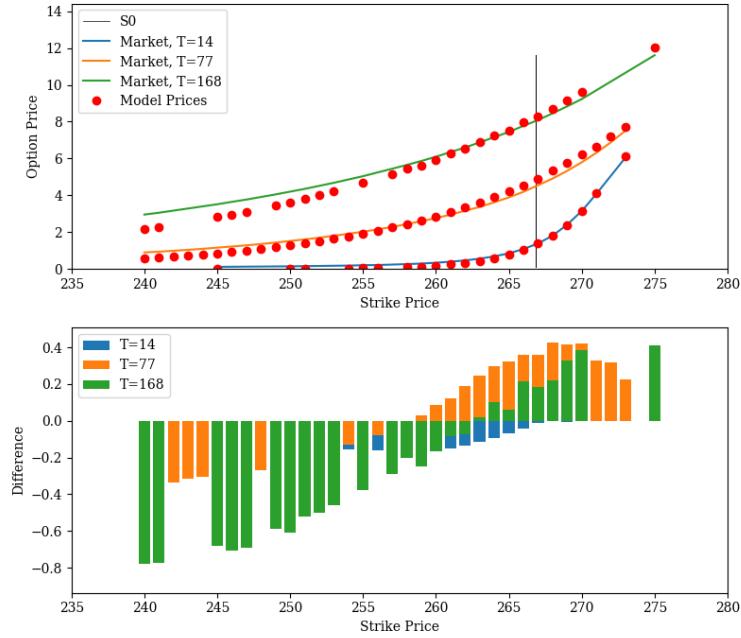


Figure 7.10: Market and Model Prices for SPY Options, 29/12/2017

For the final calibration date we observe the largest absolute deviation across the three dates. The largest absolute deviation recorded at this date is 75 cents. Larger pricing errors are apparent for deeper OTM long maturity options, and therefore the pricing errors are percentually larger for OTM options. The model tends to underestimate OTM options and overestimate NTM and ITM options across all maturities.

In Figure 7.11 the observed market prices are plotted against the model prices for all three dates.

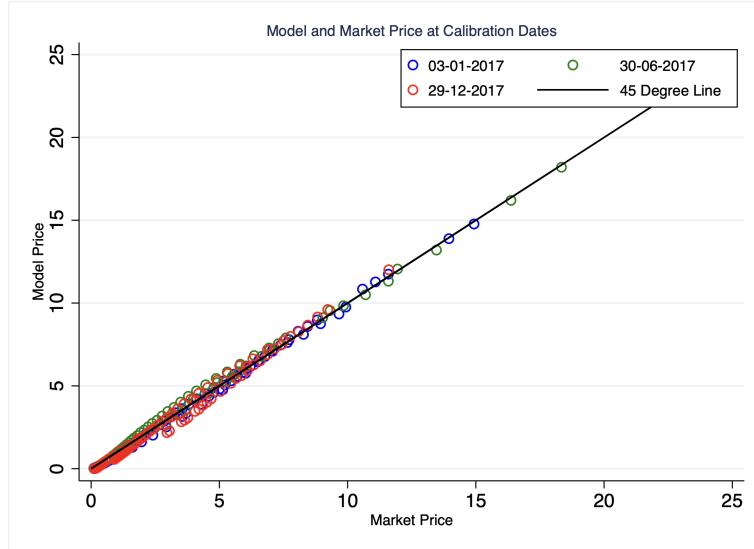


Figure 7.11: Market and Model Prices for SPY Options for all three dates

The model is clearly able to produce put option prices that are close to real observed market data. In particular options with high observed market prices perform well, corresponding to long maturity, high strike price options in our calibration sample. The largest absolute deviations are for options priced at \$4 - \$6 corresponding to options that are NTM or OTM which confirms the tendency of OTM options recording larger pricing errors. The question now is, how well does the Heston-CIR model fare for out of the sample with the parameters we have just obtained by the calibration procedure.

### 7.3 Neural Network

In order to estimate American put option prices we will consider two NN models: one simple NN model with a single hidden layer, which takes the same inputs as the LSM procedure and a more advanced model with several hidden layers and a larger set of inputs,  $X_{k,it}$ .

The first step in the NN procedure is akin to the calibration step for the LSM method, which is referred to as training the model. This is done by the backpropagation procedure introduced

in Section 6.2.1. This step is crucial, as it adjusts the model weights and biases in order for the model to produce the most realistic option prices. Model training of the NN is performed over the entire in-sample data set rather than at discrete time points which was the case for calibration of the Heston-CIR model.

### 7.3.1 Network Architecture

#### Neural Network Model 1

For our first NN model, we opt for a model as simple as possible with only one hidden layer and a minimal number of inputs. Specifically, the inputs we consider are the same as for the LSM approach, namely the underlying asset price, the strike price, time to maturity and the interest rate. This leaves us with the simple NN Model 1 to price the American put option:

$$\hat{V}_{it}^{NN_1} = f(S_t, K_{it}, \frac{T_i - t_i}{252}, r_t), \quad (7.4)$$

where  $\hat{V}_{it}^{NN_1}$  is the price of option  $i$  at time  $t$ ,  $S_t$  is the SPY closing price at time  $t$ ,  $K_{it}$  is the strike price of option  $i$  at time  $t$ ,  $\frac{T_i - t_i}{252}$  is the annualised time until maturity for option  $i$  and  $r_t$  is the risk-free interest rate at time  $t$ .

Data scaling or normalisation is a process of making model data in a standard format so that the training is improved such that it is more accurate and faster. As a general guideline, variables should be small values in the range of zero to one or standardised with a zero mean and a standard deviation of one. Hutchinson et al. (1994) scale their data by the strike price of option  $i$  at time  $t$ ,  $K_{it}$ . Specifically, they scale the output such that the price of option  $i$  at time  $t$ ,  $\hat{V}_{it}^{NN_1}$ , is divided by the strike price of option  $i$  at time  $t$ ,  $K_{it}$ . Furthermore, inputs are scaled such that the spot price of the underlying asset at time  $t$ ,  $S_t$  and the strike price of option  $i$  at time  $t$ ,  $K_{it}$ , are both divided by the strike price for option  $i$  at time  $t$ ,  $K_{it}$ . Following their approach the model in equation (7.4) becomes:

$$\frac{\hat{V}_{it}^{NN_1}}{K_{it}} = f\left(\frac{S_t}{K_{it}}, 1, \frac{T_i - t_i}{252}, r_t\right). \quad (7.5)$$

Regarding the structure of the network, we test for the single hidden layer to consist of a range of nodes, specifically 60, 80, 100 or 120 nodes. Moreover, we consider three of the four activation functions introduced in Section 6.1, namely ELU, ReLU and Leaky ReLU. The output layer consists of a single neuron which outputs the option price for option  $i$  at time  $t$ ,  $\hat{V}_{it}^{NN_1}$ . Since the price of an option cannot be negative, the activation function of the output layer is set to the exponential activation function in order to guarantee strictly positive option prices.

#### Neural Network Model 2

As an alternative to the simple NN Model 1, we also consider a more complex NN model with more hidden layers and inputs. The architecture of this model is based on Culkin & Das (2017), who price a range of simulated options using deep learning.

Although by the universal approximation theorem it can be shown that a single hidden layer network can approximate any non-linear continuous function, Deeper Neural Networks, that consist of several hidden layers, are empirically preferred when the learning task is performed for a more difficult objective. This may be able to help the network learn a difficult pattern more quickly [Balázs (2001)]. Specifically, the model consists of four hidden layers of 120 nodes each. The data is passed forward through the network using the activation functions introduced in Section 6.1. The final output layer is a single output node which uses the exponential activation function to once more guarantee strictly positive output prices,  $\hat{V}_{it}^{NN_2}$ .

In addition to the increased number of hidden layers in the model, we also increase the number of inputs to be passed through the model. In particular, volatility and returns are negatively correlated, introduced in this paper as the leverage effect. Therefore, it is relevant in the NN approach to account for volatility. Culkin & Das (2017) do so by including historical volatility of the underlying asset measured over different time periods in the input vector,  $X_{k,it}$ . In line with their approach, we introduce three historical volatility measures measured over 30 days, 60 days and 120 days at time  $t$  given by equation (7.1) denoted by  $\sigma_{t,30}$ ,  $\sigma_{t,60}$  and  $\sigma_{t,120}$ , respectively. In addition, we opt to include the VIX index at time  $t$  as a volatility measure. Since VIX in our sample is in the range between 10 and 20, we scale VIX by dividing the index-level by 100. Therefore, NN Model 2 is given by:

$$\frac{\hat{V}_{it}^{NN_2}}{K_{it}} = f\left(\frac{S_t}{K_{it}}, 1, \frac{T_i - t_i}{252}, r_t, \sigma_t\right), \quad (7.6)$$

where  $\sigma_t = (\sigma_{t,30}, \sigma_{t,60}, \sigma_{t,120} \frac{VIX_t}{100})$ .

### 7.3.2 Model Training & In-sample Performance

In order for the NN models to be able to produce realistic option prices, we need to train the models within the in-sample data set to reduce pricing errors by adjusting the model weights. The model training procedure is performed on a subset of the in-sample data, referred to as the training data set. This training data set consists of 75% of the entire in-sample data, whereas the remaining 25% is considered test data.

To evaluate the performance of the model on the two samples we should consider an error function between the observed put option prices and the put option prices predicted by the models. In our case, we use the MSE between observed and predicted option prices defined as:

$$MSE = \sum_i^I (V_i^{Market} - \hat{V}_i^{NN})^2,$$

where  $I$  is the number of options in either the training or test data set,  $V_i^{Market}$  is the observed put option price for option  $i$  and  $\hat{V}_i^{NN}$  is the NN model predicted price for option  $i$ . As such, we have two MSEs: one for the training data and one for the test data referred to as the training loss and test loss, respectively. During training process, we compare the models' performance on the training

and test data subsamples in order to evaluate how well the model can generalise the data structure. For a model to be well-fitted, the MSE should be similar for the training and test subsamples.

The optimal weights of each model are determined through the backpropagation algorithm described in Section 6.2.1. In order to initialise model training we set initial weights from the normal distribution with mean 0 and a standard deviation of 0.5. In line with Culkin & Das (2017) we select the Root Mean Squared Propagation optimiser (RMSProp) introduced by Geoffrey Hinton [Hinton, G. (2012)], which is an extension of gradient descent. Setting the initial learning rate to  $\alpha = 0.001$ , the RMSProp optimiser uses an adaptive learning rate instead of treating the learning rate as a constant, such that the learning rate changes over time. We select a batch size of 128 for the simple NN Model 1 and 64 for the extended NN Model 2, such that the models loop over 128 and 64 samples for each model respectively until all data has been passed through. We set the number of epochs to 400 for NN Model 1 and 1200 for NN Model 2. Therefore, the entire training data set is passed through our baseline model 400 times and our extended model 1200 times before the training procedure is completed. Given the more simple nature of the first model, convergence is expected to be faster and therefore a lower number of epochs is expected to be sufficient.

As introduced, the simple NN Model 1 is specified for a range of nodes and activation functions. Before settling on a final model specification, we consider each combination of nodes and activation functions. The model with each combination is trained and the MSE for the training sample is calculated. The distribution of the MSEs are reported in Figure 7.12 below for each combination.

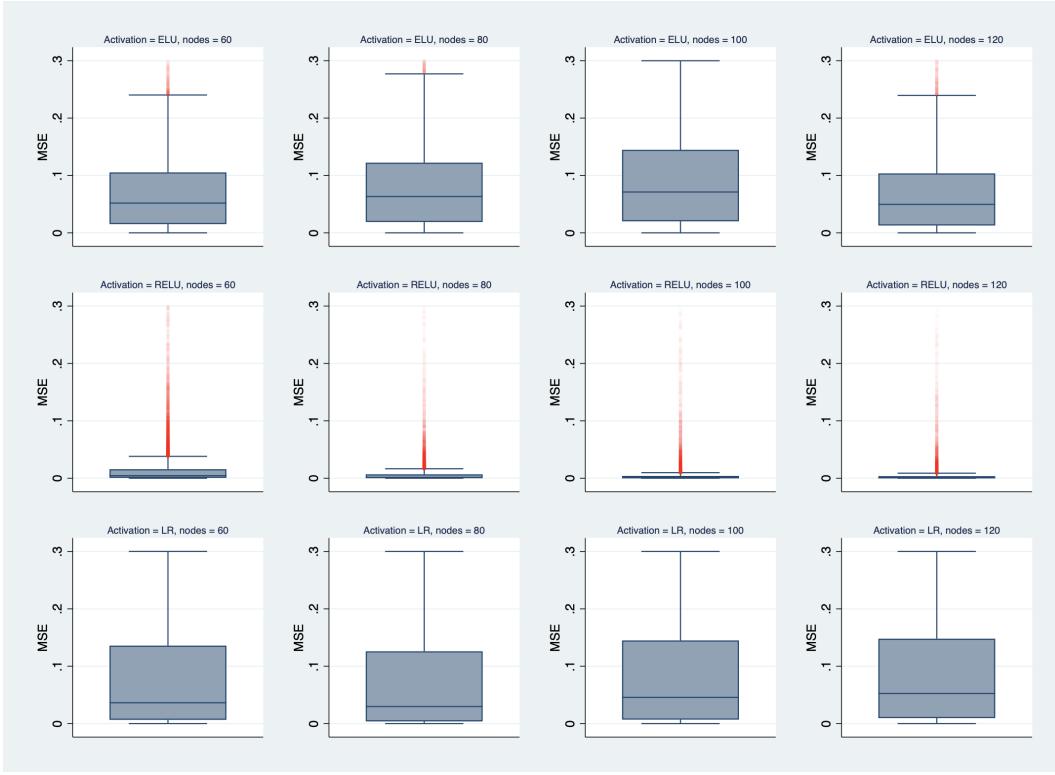


Figure 7.12: NN Model 1: Number of nodes (One hidden layer)

From the figure above we clearly see that the ReLU activation function performs the best in terms of MSE and the spread of the MSE for the entire range of nodes. In addition, 120 nodes overall appears to perform the best, in particular for the ELU and ReLU activation functions. Therefore, the hidden layer of the baseline NN Model 1 will be comprised of the ReLU activation function and 120 nodes.

In order to determine which volatility measure(s) to include in NN Model 2, we test a range of models with different volatility measures. Overall, we test three models with the architecture given by (7.6) by either including only historical volatility or VIX, or by including both volatility measures at the same time. For the model that only includes VIX as the volatility measure, the MSE is consistently lower for the training sample than for the test sample. This indicates, that the model weights are adjusted such that the model specifically fits the patterns in the training sample rather than a generalised structure. This phenomenon is referred to as overfitting. However, when including VIX as well as historical volatility, we observe a better model fit such that the model performs similarly for both the training and test set subsamples. Therefore, our two selected NN models have the following characteristics:

	NN Model 1	NN Model 2
Output Variable	Put Price (Scaled)	Put Price (Scaled)
Input Variables	SPY Close (Scaled) Strike Price (Scaled) Annualised Maturity Interest Rate	SPY Close (Scaled) Strike Price (Scaled) Annualised Maturity Interest Rate
		$\sigma_{30}$ $\sigma_{60}$ $\sigma_{120}$ $\frac{\text{VIX}}{100}$
Learning Rate	$\alpha = 0.001$ RMSProp Adjustment	$\alpha = 0.001$ RMSProp Adjustment
Hidden Layers	1	4
Hidden Nodes	120	120
Activation Function(s)	ReLU	Leaky ReLU ELU ReLU ELU
Epochs	400	1200
Batch Size	128	64

Table 7.5: NN Models' Specifications

Below are the learning curves for NN Model 1 for the training and test data set. This allows us to analyse the (scaled) pricing error for each epoch, reported as the MSE. For the first epochs the pricing error decreases extremely quickly both for the training and test data set. However, there are no noticeable improvements in the network's last epochs, indicating rather fast convergence for the model. We note, that the test pricing error is much more volatile than the training pricing error. This may indicate that the model is slightly overfitted to the training data, such that the model cannot generalise to new data. However, we see that the model's error is low for both in-sample data sets.

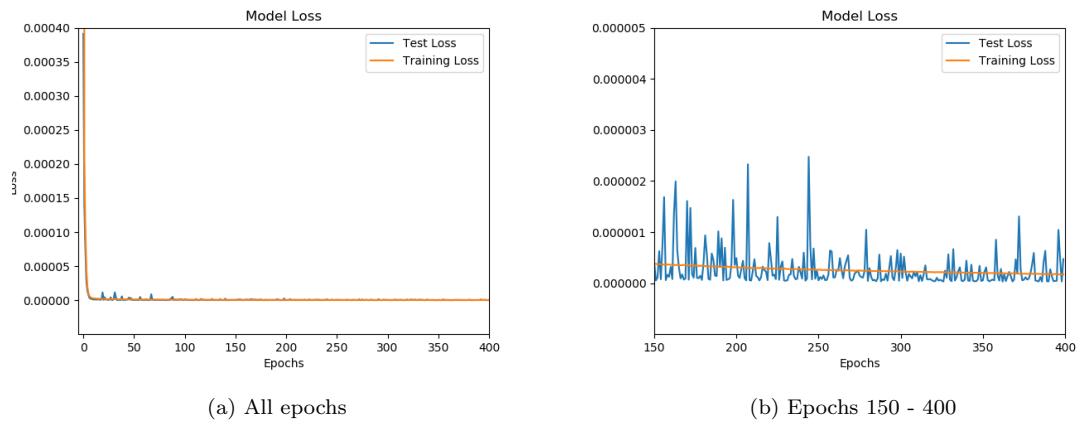


Figure 7.13: NN Model 1 Learning Curves

A similar pattern emerges in Figure 7.14 for the extended model. Once more, we observe vast improvements in pricing accuracy for both data sets in the model's first epochs. However, we do observe improvements after epoch 400, but not in the very last epochs as illustrated by subpanel (b). Moreover, subpanel (b) indicates a better fitted model, since the pricing errors for the test data converge towards the pricing errors for the training data in the last epochs. In addition, the test errors are observed to be much less volatile than for the simple NN Model 1.

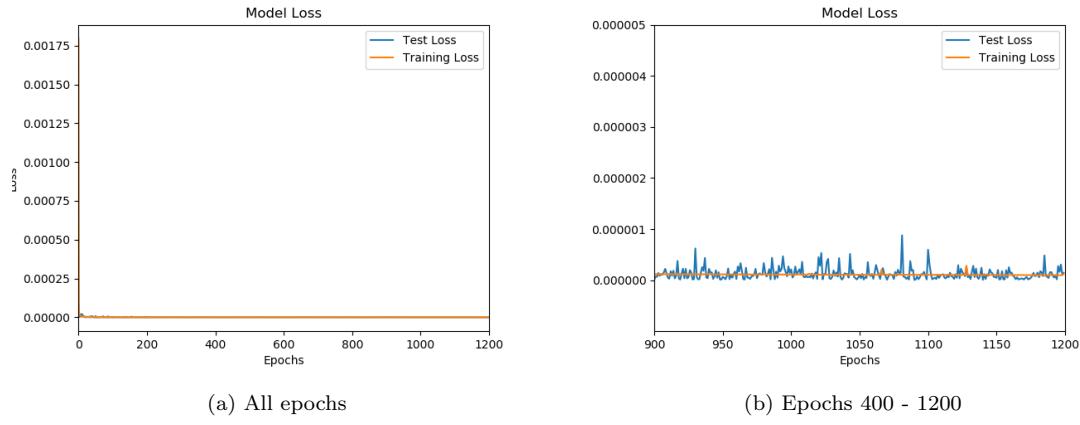


Figure 7.14: NN Model 2 Learning Curves

Now that both models have been trained such that the weights have been adjusted in order to replicate the pricing dynamics for American put options, we are able to examine exactly how well they are able to replicate the in-sample data. In order to evaluate the in-sample performance, we again consider the MAE and RMAE given by (7.2) and (7.3). The pricing errors for the two NN models on the training and test data sets are reported below:

Model	Training MAE	Test MAE	Training RMAE	Test RMAE
NN Model 1	0.0459	0.0486	0.0735	0.0769
NN Model 2	0.0659	0.0659	0.0557	0.0567

Table 7.6: In-sample NN Model Pricing Errors

We see that both NN models are able to replicate the in-sample data accurately. Given the MAE, NN Model 1 appears to perform better than NN Model 2 in-sample with an absolute pricing error of just under 5 cents for both the test and training data, whereas the absolute pricing errors for NN Model 2 around just under 7 cents. However, NN Model 2 performs better in-sample in terms of relative pricing errors of just over 5% compared to just over 7%. Finally, we note that the pricing errors for NN Model 1 are consistently lower for the training data, which may indicate a slight overfit, although these differences are not large in magnitude.

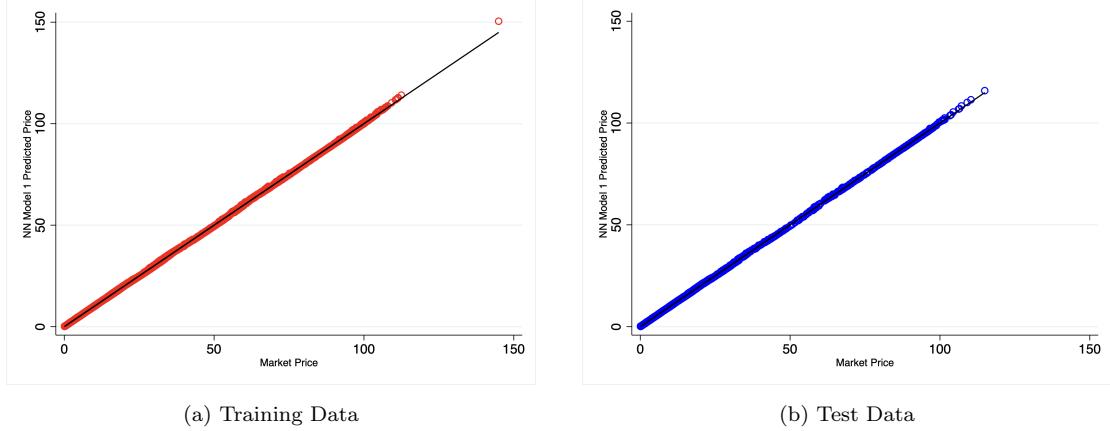


Figure 7.15: NN Model 1 In-sample predictions - Training and Test

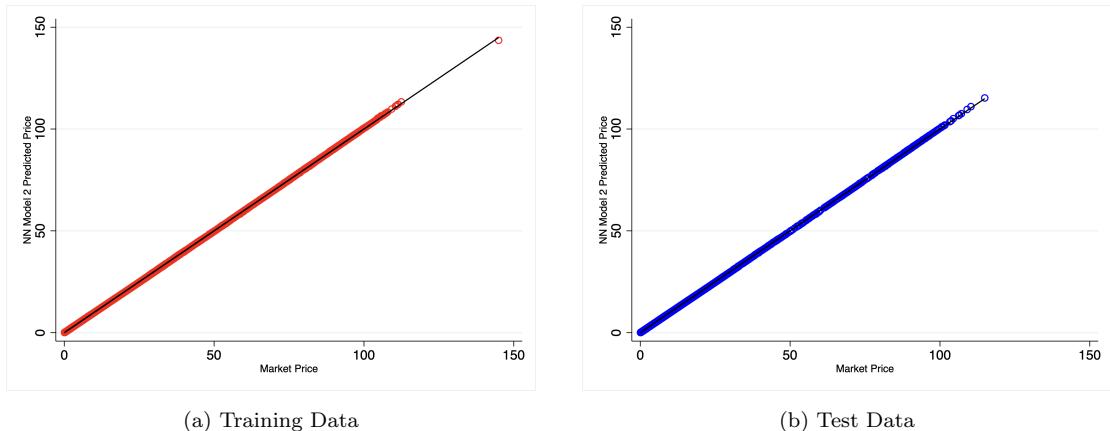


Figure 7.16: NN Model 2 In-sample predictions - Training and Test

Figure 7.15 and 7.16 graphically confirm that both NN models are able to capture the pricing dynamics for the American put option well. This is the case for all degrees of moneyness, as both options with lower and higher market prices are estimated accurately.

Recall, that the starting point for the extended NN Model 2 is the architecture from Culkin & Das (2017). Between each layer they apply a dropout rate of 0.25. Doing so, 25% of the data is discarded between each layer. This is a common practice in NNs in order to alleviate overfitting of the model to the training data. However, in NN Model 2 we observe no indications of overfitting. Below, we present the in-sample option price predictions for the training and test data when we also apply a dropout rate of 0.25 in line with Culkin & Das (2017).

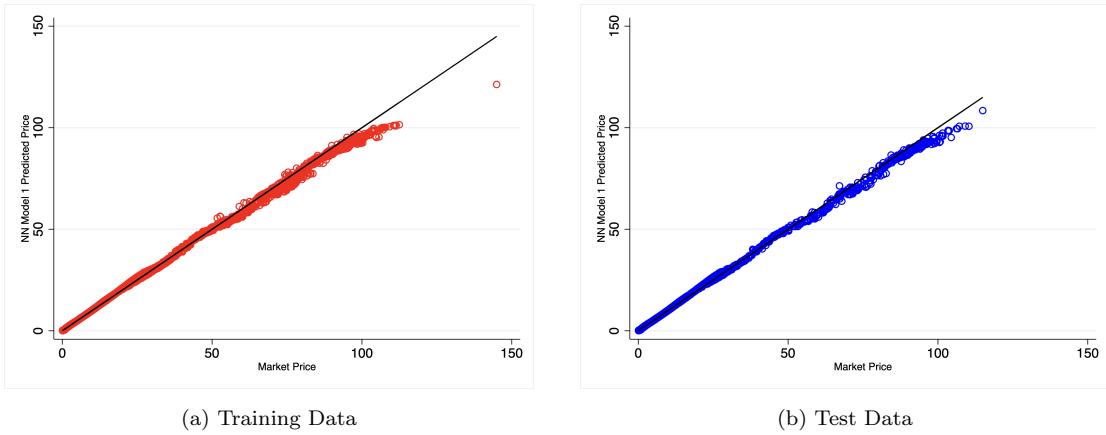


Figure 7.17: NN Model 2 In-sample predictions - Training and Test

Using their approach on our sample there are issues pricing deep ITM options. As we note from the in-sample data description, the dataset is heavily skewed towards NTM and OTM options. Using a dropout rate, the model cannot generalise to the data due to a lack of data-points for ITM options. We simply cannot afford to throw away valuable information for ITM options. Therefore, we opt to not apply a dropout rate and the architecture remains as described in Table 7.5.

## 8 Out of Sample Results

Now that the parameters in the Heston-CIR model have been calibrated and the two NN models have been trained to the in-sample data, we wish to examine the performance of each approach for new out of sample data in order to evaluate how well each approach can generalise to new data. First, we consider three dates within 6 months after the in-sample date. Specifically, we estimate the price of American put options 1 week, 1 month and 6 months after the final in-sample date. We do so using either parameters obtained by calibration on the 29th of December 2017 for the LSM approach or the NN models with optimal weights determined by training on the in-sample data.

In the first 6 months of 2018 no larger structural changes are observed in the financial markets,

such that out of sample estimates for this time-period serve as a test of the approaches' general validity. Furthermore, it is of interest how well the models perform when large shifts are observed in the financial markets. Therefore, we consider a fourth out of-sample period, namely March 1st 2022 - March 7th 2022. Large structural changes are observed throughout the first quarter of 2022. We observe increasing interest rates, and further expectation of increasing interest rates, as the FED promises future interest-rate hikes, are priced in. A high level of year-on-year inflation is recorded both as a consequence of the coronavirus pandemic, supply-chain issues as well as the ongoing conflict in Ukraine. Fear of a period with lower real economic growth or even a recession looms large. In addition, the financial markets are observed to be extremely volatile and a record number of put options are being issued, which indicates an increased interest in the market of taking on short positions.

For the LSM approach, we observe the zero-coupon bond rates at different maturities, re-calibrate the CIR-process in (5.13) and simulate the instantaneous short rate. The calibrated parameters at each out of sample date can be found in Appendix C. After observing strike prices, maturities and the SPY closing price, we are able to estimate American put option prices at out of sample dates with the approach. Analogously, we observe the relevant input variables for the two NN models reported in Table 7.5, feed them through the network and estimate the American put option prices at each out of sample date. We apply the same exclusion criteria to the data as described in Section 7.1. For both methods, we compute the pricing errors to compare results between models as well as to their in-sample performance. Specifically, we consider MAE and RMAE defined in (7.2) and (7.3).

For summary statistics on the out of sample options considered, we refer to Appendix D.

## 8.1 6 Months Out of Sample

### 8.1.1 January 8th 2018

The first date of interest is January 8th 2018, which is a week after the calibration date for the LSM approach and the final in-sample date for the NN training procedure. On this date we estimate 534 American put option prices with the three models and the two pricing errors are presented in Table 8.1 below.

	MAE	RMAE
LSM	0.385	0.247
NN Model 1	0.091	0.138
NN Model 2	0.066	0.062

Table 8.1: Out of sample pricing errors, 08/01/2018

Both NN models outperform the LSM approach in terms of MAE and RMAE. Overall, the extended NN Model 2 produces the most accurate option prices with an average pricing error of

\$0.066 corresponding to a relative pricing error of 6.2%. Compared to the test and training pricing errors, the model performs equally as well one week out of sample compared to the in-sample estimates. Based on the out of sample results for January 8th, the baseline NN model 1 is the second-best performing model with an average pricing error of \$0.091 or 13.8%, which is significantly larger than the in-sample pricing errors recorded. The LSM is clearly the worst performing model and records an average pricing errors of \$0.385 corresponding to almost 25%. However, the three models perform relatively differently for varying degrees of moneyness. Table 8.2 reports the pricing errors for options OTM, NTM and ITM.

	OTM		NTM		ITM	
	MAE	RMAE	MAE	RMAE	MAE	RMAE
LSM	0.566	0.403	0.238	0.127	0.276	0.015
NN Model 1	0.085	0.173	0.085	0.115	0.332	0.015
NN Model 2	0.054	0.067	0.070	0.060	0.209	0.011

Table 8.2: Out of sample pricing errors for different moneyness, 08/01/2018

For all models, ITM options are priced the most accurately in terms of relative pricing errors, whereas OTM options are priced the most inaccurately. Once more, NN Model 2 performs the best for all moneyness categories, but the pricing errors for LSM and the simple NN Model 1 are very similar for NTM and ITM options. The results for different moneyness highlight a potential drawback of the LSM approach, which clearly exhibits the largest OTM pricing errors.

	$T < 60$		$60 \leq T < 180$		$T \geq 180$	
	MAE	RMAE	MAE	RMAE	MAE	RMAE
LSM	0.122	0.228	0.279	0.219	1.190	0.342
NN Model 1	0.225	0.225	0.084	0.069	0.167	0.047
NN Model 2	0.054	0.089	0.067	0.046	0.096	0.023

Table 8.3: Out of sample pricing errors for different maturities, 08/01/2018

Finally, we consider the pricing errors for options of different maturities. The picture is the same; NN Model 2 performs the best for all maturities. The LSM approach shows no indication of performing better for shorter or longer maturities at this date. However, for NN Model 1 the pricing errors are decreasing in time to maturity, with the short maturity exhibiting the largest pricing errors of 22.5%. Figure 8.1 plots the observed market prices against the model predicted prices.

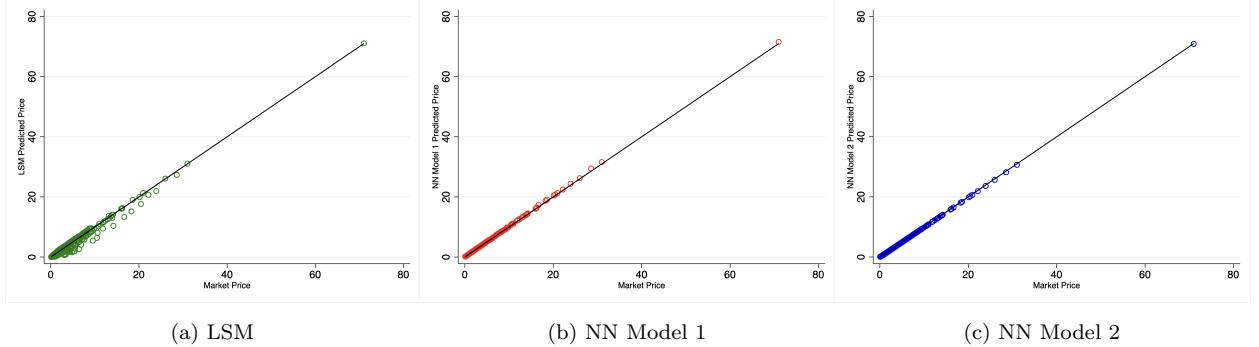


Figure 8.1: Model Estimated Option Prices - 08/01/2018

Overall, Figure 8.1 graphically confirms that the LSM approach performs worse than both NN models. However, the pricing errors are clearly larger for low moneyness options, whereas high moneyness options are priced extremely accurately. In addition, the figure indicates that the LSM approach generally suffers from a downward bias when estimate American put option prices at this date.

### 8.1.2 February 1st 2018

The next date of interest is February 1st 2018 corresponding to exactly one month after the final in-sample date. For this date we estimate 750 American put option price. Once more we report the two pricing errors for the three models in Table 8.4 below.

	MAE	RMAE
LSM	0.588	0.348
NN Model 1	0.104	0.094
NN Model 2	0.079	0.046

Table 8.4: Out of sample pricing errors, 01/02/2018

A similar pattern emerges. NN Model 2 performs the best of all three models. The RMAE is in fact improved over the previous out of sample date and has a relative pricing error of less than 5%. Moreover, NN Model 1 also performs better than the previous out of sample date based on the RMAE, which is now lower than 10%. Overall, LSM appears to perform significantly worse than for the previous date and the average pricing error is almost 60 cents corresponding to 34.8%. Presenting the pricing errors in Table 8.5 below, we can observe the pricing discrepancies for varying degrees of moneyness. In Table 8.5 below the pricing errors for the three models are presented for varying degrees moneyness.

	OTM		NTM		ITM	
	MAE	RMAE	MAE	RMAE	MAE	RMAE
LSM	0.566	0.550	0.619	0.262	0.529	0.038
NN Model 1	0.076	0.186	0.092	0.041	0.431	0.012
NN Model 2	0.053	0.067	0.084	0.036	0.212	0.008

Table 8.5: Out of sample pricing errors for different moneyness, 01/02/2018

As for the previous date, all models perform relatively best for ITM options based on the RMAE. For NTM and ITM, both NN models show improvements compared to the previous out of sample date. The LSM approach is still able to accurately estimate ITM option prices and only exhibits a small increase in both pricing errors with a pricing error in percentage of 3.8%. Unlike for the previous out of sample date, the pricing errors for NTM options are much larger and are observed at 26.2%. The estimates for OTM option prices are still inaccurate. For the two NN models we observe large differences in the pricing errors for OTM options unlike for NTM and ITM options. For NN Model 1 we report a relative pricing error of more than 18% whereas for NN Model 2 we observe an error of 6.7%.

	$T < 60$		$60 \leq T < 180$		$T \geq 180$	
	MAE	RMAE	MAE	RMAE	MAE	RMAE
LSM	0.507	0.406	0.390	0.220	1.458	0.322
NN Model 1	0.075	0.120	0.134	0.054	0.194	0.038
NN Model 2	0.069	0.056	0.088	0.032	0.117	0.021

Table 8.6: Out of sample pricing errors for different maturities, 01/02/2018

For the LSM approach we observe no systematic difference in pricing errors for the varying degrees of maturities. We note, that the approach performs the worst for  $T < 60$ . Both NN models perform the worst for the shorter maturity category,  $T < 60$ , just like the LSM approach. NN Model 2 outperforms the simple NN Model 1 for all maturities, and in particular is able to price short term options relatively better.

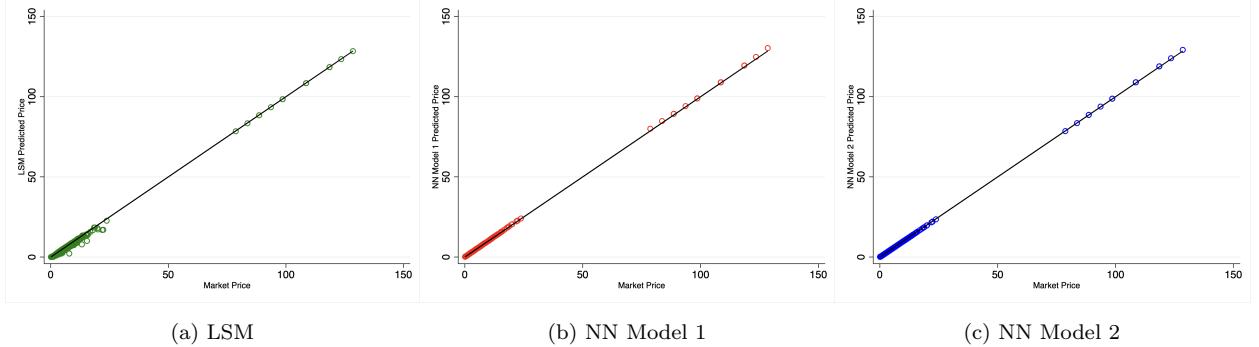


Figure 8.2: Model Estimated Option Prices - 01/02/2018

Figure 8.2 captures the overall pricing performance of the three models. We clearly see, that the LSM approach accurately estimates ITM option prices observed in the north-eastern corner of subpanel (a). For NTM and OTM options with lower market prices, the approach consistently appears to underestimate the American put option prices, which was also the case for the previous out of sample date. The figure confirms that both NN models outperform the LSM approach.

### 8.1.3 June 29th 2018

Next we consider the three models' performance exactly six months after the final in-sample date i.e. 29th June 2018 where we estimate 726 option prices. Below the two pricing errors are reported for each model.

	MAE	RMAE
LSM	0.880	0.468
NN Model 1	0.088	0.077
NN Model 2	0.092	0.063

Table 8.7: Out of sample pricing errors, 29/06/2018

Overall, the two NN models perform significantly better than the LSM approach. Based on the pricing errors for all options for this date it appears LSM is unable to estimate American put option prices with parameters obtained by calibration 6 months earlier. Based on Table 8.7 it is ambiguous which of the two NN models performs the best, since NN Model 1 reports the lowest MAE but the highest RMAE between the two NN models. Both models perform well with relative pricing errors of 7.7% and 6.3% for NN Model 1 and 2 respectively. We further analyse the performance of each model for different degrees of moneyness in Table 8.8 below.

	OTM		NTM		ITM	
	MAE	RMAE	MAE	RMAE	MAE	RMAE
LSM	0.933	0.716	0.861	0.354	0.746	0.049
NN Model 1	0.067	0.139	0.084	0.044	0.235	0.014
NN Model 2	0.072	0.094	0.097	0.049	0.166	0.012

Table 8.8: Out of sample pricing errors for different moneyness, 29/06/2018

Just as for the two other out of sample dates the LSM approach shows huge improvements when only considering ITM options. Once more, LSM is able to estimate ITM option prices relatively accurately with a percentual pricing error of less than 5% corresponding to just under 75 cents. The NTM and OTM performance is, however, quite inaccurate. In particular the OTM pricing errors are large. Both NN models overall estimate option prices accurately at all moneyness categories. Once more ITM option prices are estimated the most accurately with percentual pricing errors less than 1.5% for both models. NTM option prices are estimated less accurately, but the NN models still perform well with pricing errors less than 5%. The largest pricing errors are observed for OTM options. For these select options NN Model 2 outperforms NN Model 1 with the average percentual pricing error observed at less than 10% against 13.9% for NN Model 1.

	$T < 60$		$60 \leq T < 180$		$T \geq 180$	
	MAE	RMAE	MAE	RMAE	MAE	RMAE
LSM	0.562	0.493	1.102	0.393	2.726	0.470
NN Model 1	0.070	0.098	0.098	0.031	0.201	0.031
NN Model 2	0.085	0.076	0.102	0.040	0.121	0.025

Table 8.9: Out of sample pricing errors for different maturities, 29/06/2018

As for the previous two dates, there are no systematic pricing differences for the LSM approach for varying maturities, and no particular maturity generally reports low pricing errors. For medium ( $60 \leq T < 180$ ) and long ( $T \geq 180$ ) maturity options both NN models perform better than for short maturity options. Based on the two pricing errors reported in Table 8.9, it is ambiguous which of the two NN models performs the best.

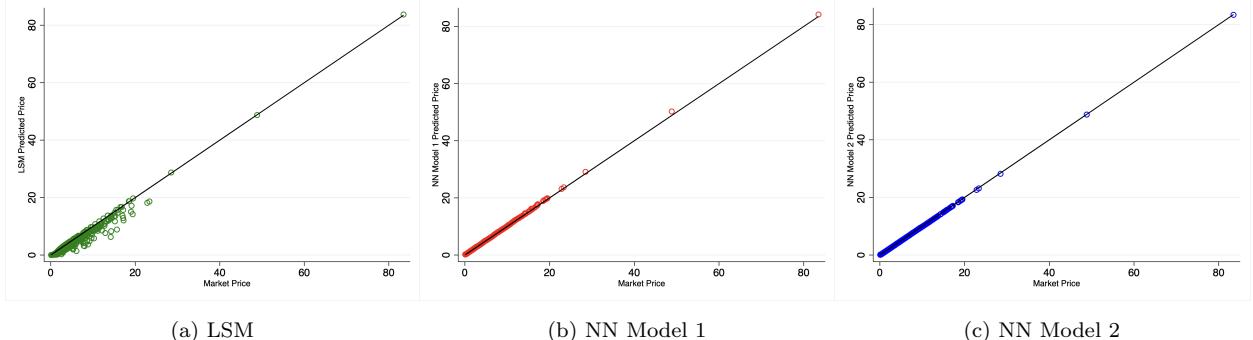


Figure 8.3: Model Estimated Option Prices - 29/06/2018

Plotting the model predicted prices against the observed market prices, the tendency of the LSM approach underestimating NTM and OTM option prices remains and becomes even more evident for this date. On the other hand, we clearly observe that ITM options with higher market prices are still estimated well. Based on subpanel (b) and (c) both NN models accurately estimate the range of option prices for the date.

Overall, based on the estimation results within the first 6 months out of sample a clear pattern emerges. As we move further away from the LSM calibration date, the LSM approach performs significantly worse. The model calibration parameters result in accurate option prices within a week from calibration, however for the 1 month and 6 month results the model can no longer generally estimate American put option prices accurately. This highlights the parameter sensitivity of this parametric modelling approach. However, the model's performance for ITM options remains accurate even 6 months after the calibration date. As such, the LSM approach remains a valid option pricing tool for ITM option even 6 months after the date calibration, especially seen in the light of the much fewer data-points necessary in order for the calibration procedure compared to the NN model training.

Both NN models perform well at all out of sample dates even as we move 6 month ahead of the last in-sample date. When considering the absolute pricing error i.e. values in USD (\$), naturally the most expensive options which are ITM tend to present larger deviations than NTM or OTM options. However, just like for the LSM approach ITM options present the lowest relative pricing errors for both NN models. As we move towards options of low moneyness the relative pricing errors increase for both NN models. For OTM options NN Model 2 is consistently more accurate than NN Model 1. As such, it appears that accounting for volatility may be of greater relevance for risk-loving investors, who prefer OTM options. Moreover, both NN models perform the best for medium and long maturity options in relative terms but not necessarily in absolute terms. The same intuition applies; options with longer maturities are generally more expensive in USD (\$) terms. Although the extended NN Model 2 overall outperforms the simple NN Model 1, there are instances of the NN Model 1 producing more accurate put option price predictions. This may indicate, that the additional layers and in particular the broader selection of input data, namely volatility measures,

are relatively constant and therefore do not contribute noticeably to the option pricing problem in this period.

## 8.2 Period of High Volatility

The final out of sample American put option estimates are for the first week of March 2022. This time period is chosen, since important dynamics which determine asset and option pricing exhibit large structural changes. The financial markets as a whole exhibit a large degree of uncertainty. This translates into extremely volatile markets, even for indices such as the S&P 500. Empirically, high volatility is associated decreasing returns. A record number of put options are issued leading up to and during this period such that there are significantly more short positions in the market. This may indicate a general expectation of short term losses.

### 8.2.1 March 1st - March 7th 2022

Now we analyse how well the three models perform for the first 5 trading days of March between March 1st and March 7th. After applying our exclusion criteria described in Section 7.1 to the datasample, we are left with 7,784 American put options. This corresponds to 1,556.8 put options on average each day, which is a sizeable increase from the selected days in the first 6 months of 2018. Moreover, the sample is highly skewed toward short maturity and OTM options. The complete summary statistics for the out of sample date can be found in Appendix D, Table D.4. The pricing errors for the three models for all 5 days are reported in Table 8.10 below.

	MAE	RMAE
LSM	5.048	0.793
NN Model 1	0.413	0.547
NN Model 2	0.180	0.105

Table 8.10: Out of sample pricing errors, 01/03/2022 - 07/03/2022

NN Model 2 performs the best as indicated by both pricing errors. On average, the pricing error in percentage is just over 10%. This is an increase when compared to all three previous out of sample dates considered, but we still consider the results reasonably accurate. Unlike for the previously considered dates, the simple NN Model 1 performs significantly worse than the extended NN Model 2 and we observe an average pricing error in percentage of almost 55%. LSM performs the worst of the three models based on the entire data sample. In particular, we note that the MAE for this model exceeds that of NN Model 1 ten-fold. However, the differences in the RMAE are not as large. We consider the models' performance for varying degrees of moneyness in Table 8.11 below.

	OTM		NTM		ITM	
	MAE	RMAE	MAE	RMAE	MAE	RMAE
LSM	3.390	0.955	7.821	0.703	4.611	0.137
NN Model 1	0.434	0.921	0.313	0.086	0.662	0.018
NN Model 2	0.188	0.166	0.148	0.032	0.254	0.006

Table 8.11: Out of sample pricing errors for different moneyness, 01/03/2022 - 07/03/2022

LSM performs the worst of all models for all moneyness categories. In particular, it is unable to estimate options prices accurately for OTM and NTM options. Results do, however, improve dramatically when only considering ITM options. The percentual pricing error of 13.7% is much higher compared to the NN models. Moreover, LSM model's performance is much worse compared to the 3 previous out of sample date, where the largest RMAE recorded was 0.049.

For all moneyness categories we observe lower absolute and percentual pricing errors for NN Model 2 compared to NN Model 1. Both NN Model 1 and 2 exhibit the best pricing performance for increasing moneyness, with the relatively most accurate prices being estimated for ITM options. Both models are extremely accurate and we observe pricing errors in percentage of 1.8% and 0.6% for NN Model 1 and 2, respectively. For NTM and particularly OTM options, NN Model 2 outperforms NN Model 1. This is not surprising, since NN Model 2 is able to account for the increasing volatility in the financial markets. With the expectation of decreasing returns the pricing dynamics of put options change, and the payoff from a put option increases. If fears of significant downturns on the financial markets are not apparent, OTM options will rarely lead to a positive payoff, since the S&P 500 index typically trends upwards. However, for this period the fear is prevalent, many investors may be expecting negative returns and we observe high volatility. Therefore, a model which does not account for volatility, will particularly underestimate prices of OTM put options. Finally, we note that even when accounting for volatility, NN Model 2 does not perform as well compared to the three previous dates.

	$T < 60$		$60 \leq T < 180$		$T \geq 180$	
	MAE	RMAE	MAE	RMAE	MAE	RMAE
LSM	3.761	0.813	8.037	0.748	12.174	0.677
NN Model 1	0.431	0.690	0.266	0.100	0.559	0.030
NN Model 2	0.187	0.124	0.135	0.048	0.217	0.021

Table 8.12: Out of sample pricing errors for different maturities, 01/03/2022 - 07/03/2022

Just as for the three previous dates LSM is unable to generally price American put options for all three maturity categories. NN Model 1 and Model 2 perform similarly for long maturity,  $T \geq 180$ , options with observed pricing errors in percentage of 3% and 2.1% respectively. As the maturity

length shortens the discrepancies between the two NN models become more apparent. The simple NN Model 1 is unable to price short maturity options accurately with an observes RMAE of 0.690. On the other hand, NN Model 2 performs much better, although short maturity options are priced with the highest observed error.

The underlying structural changes observed during this period can be considered as short term shocks rather than permanent of nature. Therefore, by including measures which may be able to capture the uncertainties caused by the short term shocks, the precision when pricing short maturity options improves. Buying American style options with long maturity requires a premium, such that if you are taking a short position based on the short-term shocks buying an option with a shorter maturity is the obvious choice. This explains why NN Model 2 is relatively better than NN Model 1 for short and medium maturity options.

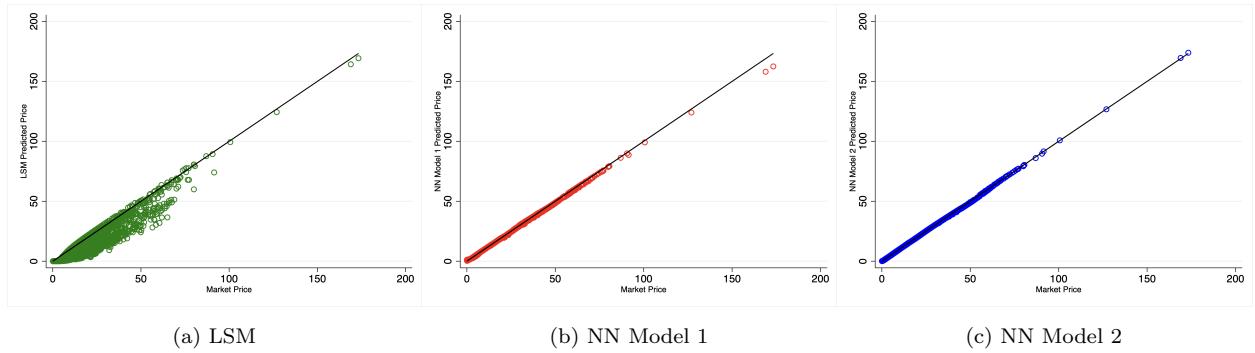


Figure 8.4: Model Estimated Option Prices - 01/03/2022 - 07/03/2022

Based on Figure 8.4 it is evident that the LSM approach generally struggles to estimate the American put option prices for the period March 1st - March 7th with parameters obtained by calibration on December 29th 2017. However, we can graphically observe that ITM options are priced with reasonable accuracy and very deep ITM options are priced extremely well. Generally, the estimated option prices appear to be underestimated. Intuitively this makes sense, since the parameters are not adjusted to capture the increased volatility which in turn will lead to lower spot prices for the underlying asset through the leverage effect and higher put option prices in the market.

It is difficult to compare NN Model 1 and 2 graphically for low priced options. As mentioned earlier, the sample for this period is primarily comprised of OTM options with lower market prices. Graphically, these are all clumped together in the south-western corner of the graphs. However, we can observe a general underestimation of the American put option prices for the period for NN Model 1. Moreover, we are graphically able to see greater precision for NN Model 2 for deep ITM options, as there are simply fewer data-points.

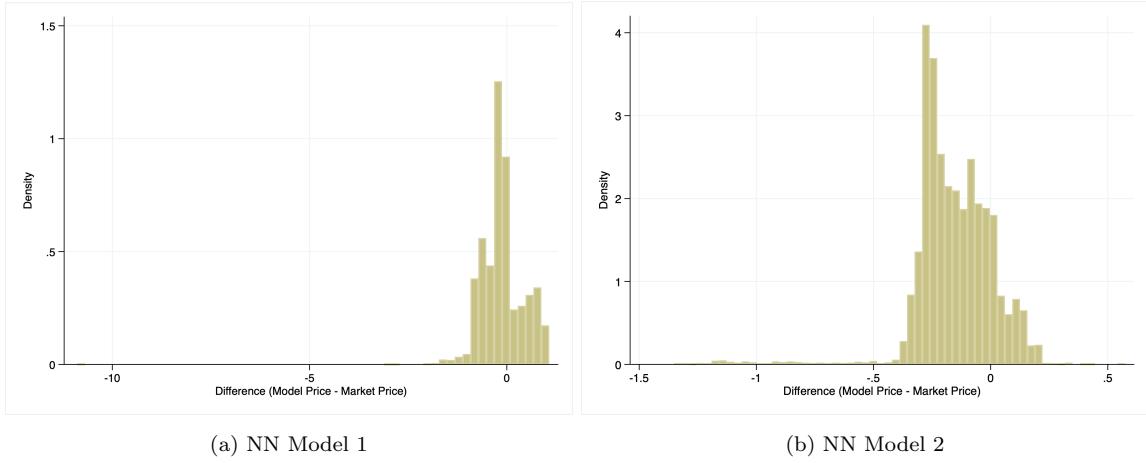


Figure 8.5: Actual Pricing Errors - 01/03/2022 - 07/03/2022

Figure 8.5 above confirms, that NN Model 1 to a greater extent underestimates American put option prices for this out of sample date, note the different axis-ranges for panel (a) and panel (b). We clearly observe that the difference between the model price and market price is more frequently negative for NN Model 1, and larger in absolute magnitude. Therefore, in cases where NN Model 1 underestimates the price it also does so more extremely.

## 9 Concluding Discussion and Potential Applications

Analysing the results in and out of sample, we find that each method presents its strengths and weaknesses. In-sample, we find that all methods are able to estimate prices of American put options well. The data-driven NN models perform significantly better overall, which is no surprise given the large amounts of data that we feed through the models in order to fit to observed market prices. For our sample, both NN models are able to fit the data-structure extremely precisely, and even the simple NN Model 1 with a single hidden layer achieves an accurate fit to the data. Therefore, our in-sample results indicate that complexity is not necessarily a strength if the goal is simply to achieve a good in-sample model fit. This is further highlighted when we consider the computational cost for each NN model, since training the extended model takes significantly more time. However, a strong model fit is difficult to achieve if the data foundation is not large enough, as highlighted when including a dropout rate. This inhibits the extended model from achieving a good fit to the data for ITM options, since we simply do not have enough data-points available for ITM options when including a dropout rate.

In contrast, this is a great strength of the LSM approach. In-sample, the model is observed to be a good option pricing tool, although not as accurate as the NN approach. Even when the data-points are few, the model is able to realistically estimate American put option prices. Using parameters obtained by calibration a week ago, the model still performs well, particularly for NTM and ITM options. As we move away from the calibration date, the pricing errors are aggravated. Structural

changes appear to inhibit the model from performing well. However, for all out of sample dates ITM options are priced accurately. Those who prefer less risk may therefore still use the LSM approach. Similar findings are prevalent in the literature, when comparing parametric and non-parametric approaches. Malliaris & Salchenberger (1993) show that a parametric approach performs better in case of ITM options whereas the NN approach is dominant in the out of sample prediction. The empirical study of Jang and Lee (2019), who consider S&P 100 American put option prices, also points out the better performance of NNs when compared with the classical financial option models.

As we noted initially, the Heston-CIR model is extremely sensitive to the choice of model parameters. Therefore, it is no surprise that parameters obtained up to 6 months ago are unable to result in realistic option prices. This motivates the degree to which the model can be used in pricing options, when only model parameters from an earlier date are available. Several underlying conditions should be considered. The parameters obtained by calibration determine the process for the variance,  $v_t$ . If we wish to consider parameters obtained by calibration at an earlier date, we must therefore consider structural changes that affect volatility of returns. Moreover, fundamentals within the economy or changes within behaviour and expectation are certainly relevant, such as fears of inflation and/or higher interest rates which could be a cause of uncertainty and more volatility. When taking such things into consideration investors may be able to adjust the parameters and price options with more precision.

American options on the SPY have a large supply and are the most frequently traded American options, in that market prices at any given time for any given strike price and maturity are available. This is a reason why the paper chose SPY as the underlying asset in the first place. However, we may consider a case where we are dealing with options on an underlying asset which are scarcely traded. In this case, investors can use the LSM approach to calibrate parameters to the (relatively) few market prices with different maturities and strike prices. Once calibrated, the approach can be used to price the unique option on the scarcely traded underlying asset. Furthermore, we can consider a case in which investors are the holders of an option where the underlying asset is non-standard, whereby a market price is not readily available. In this regard parametric option pricing models, such as the Heston-CIR model, can play a key role. We argue, that the LSM approach can be used effectively in order to price the option at hand precisely if one wishes to either sell or simply value the option. In such a case the model parameters can be calibrated to option prices on a comparable underlying asset and subsequently used as a pricing tool for the non-standard asset. Since the parameter values play such an important role in the model's pricing accuracy, this comparable asset should be chosen wisely. This is an area of research which may be explored further. Which key factors identify the comparable asset from which model calibration should be carried out? Some examples that come to mind are certain indices, comparable companies within the same industry or even a selection of several comparable companies.

The results from the two NN models considered indicate that if enough data is readily available, the NN approach outperforms the parametric approach for all maturities and moneyness. As we

move away from the final in-sample date, both the simple and extended NN models estimate option prices accurately. However, as we move further away from the in-sample period, the added complexity of NN Model 2 becomes more relevant. Both NN models systematically underestimate prices of American put options in times of high volatility, but accounting for volatility in the extended NN Model 2 yields better out of sample results. Accounting for volatility results in more robust option price estimates. However, the extended model still underestimates the put option prices, indicating that the volatility measurements, when trained to our in-sample data cannot completely capture the structural changes. There is clearly a trade-off between the complexity of NN models allowing for better out of sample predictions and the effort associated with added complexity.

Given the "black box" nature of the procedure, we are unable to determine how the performance of the NN models changes, except by comparing a range of models, of which there are unlimited specifications. This is obviously a time-consuming process. Since the approach is not driven by economic theory and strictly relies on data, the only thing that matters is the model fit. The question is how optimal model fit is achieved, to which there is no one-size fits all solution.

Our results suggest that the two trained NN models estimate American put option prices differently for varying degrees of moneyness. Therefore, the models may be able to achieve a better fit, if the pricing problem is solved for each moneyness category. The data is partitioned on the basis of the moneyness of the options before the training procedure. This approach is known as Modular Neural Networks. Gradojevic et al. (2009) apply a Modular Neural Network, where the modules are based on time to maturity and moneyness of the options, and find that modularity improves the generalisation properties of standard feed-forward NN models for option pricing such as ours. In a similar way, Yao et al. (2000) use a backpropagation NN to forecast option prices of Nikkei 225 index future. For periods with high volatility present in financial markets, they find that a NN option pricing model outperforms the traditional Black-Scholes model. However, in their case the Black-Scholes model is still adequate for pricing NTM options. They also suggest partitioning the data according to moneyness in order to improve the performance of NNs.

Nonetheless, in terms of NN applicability to the option pricing problem, the results are promising. Even the simple NN Model 1 overall outperformed the parametric LSM procedure. As for the the extended NN Model 2, it far outperforms the LSM procedure and the simple NN Model when estimating option prices far out of sample. If NN models continue to produce accurate option price estimates, investors who have the resources could start using the approach to price an option and complete trades instantaneously.

## Appendix A. Selected $M = 10$ Basis Functions

$$B_{10}(x) = S_t * v_t * r_t$$

$$B_9(x) = S_t * r_t$$

$$B_8(x) = S_t * v_t$$

$$B_7(x) = v_t * r_t$$

$$B_6(x) = S_t^2$$

$$B_5(x) = v_t^2$$

$$B_4(x) = r_t^2$$

$$B_3(x) = S_t$$

$$B_2(x) = v_t$$

$$B_1(x) = r_t$$

$$B_0(x) = 1$$

## Appendix B. Derivation of Control Variate Adjustment Parameter, $\hat{b}$

Consider a general setting based on Glassermann (2003). Let  $Y_1, Y_2, \dots, Y_N$  be outputs from  $N$  replications of a simulation and we wish to estimate  $E[Y_n]$ . Further suppose, that we have another random variable,  $X_n$ , which is calculated along with  $Y_n$  for each replication. Suppose that the pairs  $(X_n, Y_n)$  are *i.i.d.* and that  $Y$  and  $X$  are correlated. Furthermore, for each  $X_n$  the expectation  $E[X]$  is known. Now, for any fixed  $b$ , we can calculate:

$$Y_n(b) = Y_n - b(X_n - E[X]), \quad (\text{I})$$

from the  $n$ th replication. We compute the sample mean, and the control variate estimator is given by:

$$\bar{Y}(b) = \bar{Y} - b(\bar{X} - E[X]) = \frac{1}{N} \sum_{n=1}^N (Y_n - b(X_n - E[X])), \quad (\text{II})$$

where  $\bar{Y}$  is the usual estimator given by the sample mean  $\bar{Y} = (Y_1 + \dots + Y_N)/N$ . Hence, in (II) the observed error,  $\bar{X} - E[X]$  serves as a control when estimating  $E[Y]$ . The control variate estimator is an unbiased and consistent estimator of  $E[Y]$ . The choice of coefficient  $b$  is crucial when discussing the effectiveness of reducing the variance of the estimate of  $E[Y]$ . Hence, we select a  $b$  that minimises the variance of (II), because  $Var(\bar{Y}(b)) = \frac{1}{N} Var(Y_n(b))$ , which is given by:

$$\begin{aligned} Var[Y_n(b)] &= Var[Y_n - b(X_n - E[X])] \\ &= \sigma_Y^2 - 2b\sigma_X\sigma_Y\phi_{XY} + b^2\sigma_X^2 = \sigma^2(b), \end{aligned} \quad (\text{III})$$

where  $\sigma_X^2 = Var[X]$ ,  $\sigma_Y^2 = Var[Y]$ , and  $\phi_{XY}$  is the correlation coefficient between  $X$  and  $Y$ . As such, the optimal coefficient,  $b^*$ , minimises (III) and is given by:

$$b^* = \frac{\sigma_Y}{\sigma_X}\phi_{XY} = \frac{Cov[X, Y]}{Var(X)} \quad (\text{IV})$$

In practice  $E[Y]$  is unknown, but there can still be benefits to the control variate estimator by estimating  $b^*$ . Hence, we estimate  $b^*$  by replacing the population parameters with the sample counterparts in (IV) such that:

$$\hat{b} = \frac{\sum_{n=1}^N (X_n - \bar{X})(Y_n - \bar{Y})}{\sum_{n=1}^N (X_n - \bar{X})^2} \quad (\text{V})$$

Dividing the numerator and denominator by  $N$  and applying the strong law of large numbers shows that  $\hat{b} \rightarrow b^*$  with probability 1, which suggests replacing  $b^*$  with  $\hat{b}$  in (II)

## Appendix C. Out of Sample CIR Calibrated Parameters

Date of calibration	$\kappa_r$	$\theta_r$	$\sigma_r$	$r_0$	MSE
06/01/2018	0.545	0.060	0.256	0.015	7.470e-07
01/02/2018	0.764	0.063	0.005	0.015	6.467e-07
29/06/2018	1.533	0.062	0.002	0.020	6.394e-07
01/03/2022	2.008	0.039	0.005	0.001	4.366e-06
02/03/2022	1.935	0.039	0.391	0.001	4.874e-06
03/03/2022	2.010	0.039	0.398	0.002	4.316e-06
04/03/2022	2.270	0.037	0.409	0.002	1.559e-06
07/03/2022	2.320	0.037	0.316	0.002	3.777e-07

Table C.1: Out of Sample Calibration Results for the CIR-Model at Selected dates

## Appendix D. Out of Sample Data Description

Days to expiration, $T$	$T < 60$	$60 \leq T < 180$	$T \geq 180$	All Options
Moneyness, $m (S/K)$	Average Option Price			
ITM ( $m < 0.97$ )	19.595	27.2425	22.0475	22.70269
NTM ( $0.97 \leq m < 1.05$ )	1.291845	4.612986	12.41396	3.080018
OTM ( $m \geq 1.05$ )	0.2300862	1.308155	4.29039	2.010273
Moneyness, $m (S/K)$	# Observations			
ITM ( $m < 0.97$ )	5	4	4	13
NTM ( $0.97 \leq m < 1.05$ )	187	72	24	283
OTM ( $m \geq 1.05$ )	58	103	77	238

Table D.1: SPY Put Options Summary Statistics for Varying Degrees of Moneyness and Selected Maturities, January 8th 2018

Days to expiration, $T$	$T < 60$	$60 \leq T < 180$	$T \geq 180$	All Options
Moneyness, $m (S/K)$	Average Option Price			
ITM ( $m < 0.97$ )	12.84125	69.49694	87.79083	49.57875
NTM ( $0.97 \leq m < 1.05$ )	3.115498	6.567836	13.83531	4.451174
OTM ( $m \geq 1.05$ )	0.4552083	1.982364	4.2502	1.732839
Moneyness, $m (S/K)$	# Observations			
ITM ( $m < 0.97$ )	16	18	6	40
NTM ( $0.97 \leq m < 1.05$ )	331	67	32	430
OTM ( $m \geq 1.05$ )	120	110	50	280

Table D.2: SPY Put Options Summary Statistics for Varying Degrees of Moneyness and Selected Maturities, February 1st 2018

Days to expiration, $T$	$T < 60$	$60 \leq T < 180$	$T \geq 180$	All Options
Moneyness, $m (S/K)$	Average Option Price			
ITM ( $m < 0.97$ )	15.50058	14.07844	22.95357	16.10092
NTM ( $0.97 \leq m < 1.05$ )	2.75847	7.468	13.37706	3.896278
OTM ( $m \geq 1.05$ )	0.5314855	2.079121	5.229146	1.766444
<hr/>				
Moneyness, $m (S/K)$	# Observations			
ITM ( $m < 0.97$ )	26	16	7	49
NTM ( $0.97 \leq m < 1.05$ )	330	60	17	407
OTM ( $m \geq 1.05$ )	138	91	41	270

Table D.3: SPY Put Options Summary Statistics for Varying Degrees of Moneyness and Selected Maturities, June 29th 2018

Days to expiration, $T$	$T < 60$	$60 \leq T < 180$	$T \geq 180$	All Options
Moneyness, $m (S/K)$	Average Option Price			
ITM ( $m < 0.97$ )	30.2964	39.7728	60.25097	34.5961
NTM ( $0.97 \leq m < 1.05$ )	9.970615	21.98269	36.74279	12.75348
OTM ( $m \geq 1.05$ )	2.093155	7.249845	13.59531	3.981234
<hr/>				
Moneyness, $m (S/K)$	# Observations			
ITM ( $m < 0.97$ )	531	109	67	707
NTM ( $0.97 \leq m < 1.05$ )	2,235	362	120	2,717
OTM ( $m \geq 1.05$ )	3,198	809	353	4,360

Table D.4: SPY Put Options Summary Statistics for Varying Degrees of Moneyness and Selected Maturities, March 1st 2022 - March 7th 2022

## Appendix E. Code

The content of the repository is as follows:

- CIR\_zcb\_valuation.py  
Calculates the implied forward rates given zero-coupon bond data, which is fed into the CIR calibration process
- CIR\_calibration.py  
Calibrates the CIR process to the implied forward rates given (5.13)
- Heston\_pricing.py  
Defines the LSM approach for the Heston-CIR process given the CIR calibrated parameters from (5.13)
- Heston\_Calibration.py  
Calibrates the Heston-CIR model parameters according to (5.14) given the CIR calibrated parameters from (5.13)
- Build\_Model.py  
Builds the two NN models used for American option pricing specified in Table 7.5
- NN\_Option\_Pricing.py  
Uses the two local NN models defined in Build\_Model.py. The training procedure and model evaluation is carried out.



GitHub Repository  
[https://github.com/EdmundClink/speciale\\_options](https://github.com/EdmundClink/speciale_options)

## References

- [Albarghouthi (2021)] Albarghouthi, A. (2021). Introduction to Neural Network Verification, University of Wisconsin–Madison, arXiv preprint arXiv:2109.10317
- [Anders et al. (1998)] Anders, U., Korn, O. & Schmitt, C. (1998) “Improving the pricing of options: a neural network approach,” Journal of Forcasting, vol. 17, pp. 369–388,
- [Balázs (2001)] Balázs C. C. (2001). ”Approximation with artificial neural networks.” Faculty of Sciences, Etvs Lornd University, Hungary 24: 48.
- [Baxter & Rennie (1996)] Baxter, M. and Rennie, A. (1996). Financial Calculus—An Introduction to Derivative Pricing, Cambridge University Press, Cambridge.
- [Brandimarte (2006)] Brandimarte, P. (2006). Numerical Methods in Finance and Economics with MATLAB. Statistics in Practice. 2nd Edition, Wiley-Interscience, Hoboken.
- [Black & Scholes (1973)] Black, F. & Scholes, M. (1973). ‘The pricing of options and corporate liabilities’, Journal of Political Economy 81(3), 637–654.
- [Christoffersen & Jacobs (2004)] Christoffersen, P. & K. Jacobs (2004). “Which garch model for option valuation?” Management science 50(9): pp. 1204–1221.
- [Christomo (2014)] Crisostomo, R. (2014). “An Analysis of the Heston Stochastic Volatility Model: Implementation and Calibration using Matlab”, CNMV Working Paper 58: 1-46.
- [Cox, Ingersoll & Ross (1985)] Cox, J.C., Ingersoll, J.E. and Ross, S.A. (1985). ”A Theory of the Term Structure of Interest Rates”. Econometrica. 53 (2): 385–407.
- [Culkin & Das (2017)] Culkin, R. & Das, S. R. (2017). “Machine learning in finance: The case of deep learning for option pricing.” Journal of Investment Management 15(4): pp. 92–100.
- [Feller (1951)] Feller, W. (1951). ”Two singular diffusion problems”, Ann. Math. 54(1), 173–182.
- [Milstein (1975)] Milstein, G. N. (1975). ”Approximate Integration of Stochastic Differential Equations”. Theory of Probability & Its Applications. 19 (3): 557–000.
- [Kinghi (2019)] Kingi, H. (2019). Numerical SDE Simulation - Euler vs Milstein Methods.
- [Gardner & Dorling (1997)] Gardner, W. and Dorling, S. R. (2000). Artificial Neural Networks (The Multilayer Perceptron)- A Review of Application in the Atmospheric Sciences, Atmospheric

Environment Vol. 32, No. 14/15, pp. 2627—2636, 1998

[Gaspar et al. (2020)] Gaspar R. M., Lopes S. D. and Sequeira B. (2020). Neural Network Pricing of American Put Options, Risks 8, 73, 2020

[Gatheral (2006)] Gatheral J. (2006). The volatility surface: A practitioner's guide, John Wiley & Sons, Hoboken, New Jersey.

[Geigle (1999)] Geigle, D. S. (1999). An artificial neural network approach to the valuation of options and forecasting of volatility. Nova Southeastern University.

[Glasserman (2003)] Glasserman, P. (2003). Monte Carlo Methods in Financial Engineering, Springer.

[Glorot et al. (2011)] Glorot, X., Bordes, A. & Bengio, Y. (2011). “Deep sparse rectifier neural networks.” In “Proceedings of the fourteenth international conference on artificial intelligence and statistics,” pp. 315–323.

[Gradojevic et al. (2009)] Gradojevic, N., Gençay, R. & Kukolj, D. (2009). “Option pricing with modular neural networks.” IEEE transactions on neural networks 20(4): pp. 626–637.

[Hahnloser et al. (2000)] Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., Seung H. S. (2000). “Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit”. In: Nature 405.6789 (2000), pp. 947–951.

[Heston (1993)] Heston S.L. (1993), A closed-form solution for options with stochastic volatility with applications to bond and currency options. Rev. Financ. Stud. 6(2), 327–343.

[Hilpisch (2011)] Hilpisch, Y. (2011). Fast Monte Carlo Valuation of American Options under Stochastic Volatility and Interest Rates.

[Hilpisch (2015)] Hilpisch, Y. (2015). Derivatives Analytics with Python: Data Analysis, Models, Simulation, Calibration and Hedging, John Wiley & Sons, Hoboken, New Jersey.

[Hinton (2012)] Hinton, G. (2012). Neural Networks for Machine Learning: Lecture 6, <https://www.coursera.org/learn/neural-networks/home/welcome>

[Hornik et al. (1989)] Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Networks 2: 359–66.

[Hutchinson et al. (1994)] Hutchinson, J. M., Lo, A. W., and Poggio, T. (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks. The Journal of

Finance 49: 851–89.

[Kingi (2019)] Kingi, H. (2019). Numerical SDE Simulation - Euler vs Milstein Methods.

[Jang & Lee (2019)] Jang, H. and Lee, J. (2019). Generative Bayesian neural network model for risk-neutral pricing of American index options. Quantitative Finance 19: 587–603.

[Leshno et al. (1993)] Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. (1993). Multilayer feed-forward networks with a nonpolynomial activation function can approximate any function. Neural Networks 6: 861–67.

[Liu et al. (2019)] Liu, S., Oosterlee, C. and Bohte, S. (2019). Pricing options and computing implied volatilities using neural networks. Risks 7: 16.

[Longstaff & Schwartz (2001)] Longstaff, F. A. & Schwartz, E. S. (2001), ‘Valuing american options by simulation. a simple least-squares approach’, The review of financial studies 14(1), 113–147.

[Malliaris & Salchenberger (1993)] Malliaris, M. & L. Salchenberger (1993). “A neural network model for estimating option prices.” Applied Intelligence 3(3): pp. 193–206.

[Milstein (1975) ] Milstein, G. N. (1975). ”Approximate Integration of Stochastic Differential Equations”. Theory of Probability & Its Applications. 19 (3): 557–000.

[Mrázek & Pospíšil (2017)] Mrázek, M., & Pospíšil, J. (2017). Calibration and simulation of Heston model, Open Mathematics, 15(1), 679-704.

[Rumelhart et al. (1986)] Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning representations by back-propagating errors. Nature 323(6088), 533–536.

[Yao et al. (2000)] Yao, J. , Yili, L., and Chew L. T. (2000). Option price forecasting using neural networks. Omega 28: 455–66.