Expert Systems With Applications 251 (2024) 123979

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Pricing options with a new hybrid neural network model

Yossi Shvimer [a,*], Song-Ping Zhu [b,*]

[a] School of Finance and Management, SOAS University of London, UK
[b] School of Mathematics & Applied Statistics, University of Wollongong, Australia

ARTICLE INFO

Keywords:
Option pricing
Neural networks
Put-call parity
Computational finance

ABSTRACT

A novel hybrid option pricing model using a deep learning neural network has been developed. The hybrid model keeps the traditional option pricing model with the same input parameters while simultaneously adjusting the model with neural network methods to improve accuracy when applied to real market data, especially in OTM options. The new hybrid model demonstrates superior accuracy compared to both traditional parametric and non-parametric option pricing models for both Call and Put options across all moneyness levels. The empirical results of the hybrid model provide an explanation for the deviation from the Put-Call parity observed in real market data.

1. Introduction

1.1. Background

Option pricing is a methodical process that entails calculating the theoretical fair price under certain conditions. The price of an option is affected by basic financial factors, including the spot price of an underlying asset, the option's strike price, the volatility of the underlying assets, and the interest rate. Option prices can be calculated using either parametric or nonparametric models. The parametric approach typically determines the option price within a well-defined framework, such as the well-known Black and Scholes (1973) model. Since the seminal work of Black and Scholes (hereinafter BS model), practitioners have used the model extensively all over the world. Despite its success in quantitively revealing some fundamental insights of financial derivatives, the model fails to capture empirical phenomena like the leptokurtic distribution of returns (not a normal one) or the existence of a volatility smile that suggests that volatility is not constant as the model assumes.

Numerous models have been proposed to relax the BS model restrictions and improve the empirical results. Some of the most popular approaches are (i) statistical series expansion, such as the models suggested by Corrado and Su (1996) or Jarrow and Rudd (1982), (ii) local volatility models, such as the Schroder model (1989), (iii) stochastic volatility models such as Hull and White (1987), Heston (1993) and (iv) models with jump diffusion process like Merton (1976) and Bates (1996). In addition, more advanced models, such as the Levy process presented by Carr and Wu (2004) and the variance gamma process presented by Madan et al. (1998), have been studied. Liang et al. (2009) showed that numerical methods like Monte-Carlo, finite differential method, and binomial tree methods are used for pricing an option when it is difficult to solve a model with regular form owing to a large number of variables or when the BS model is difficult to apply when a closed-form does not exist. However, although these models have proved to be more efficient in terms of valuation, they are much more computationally costly, requiring many implicit parameters to be calibrated. Also, they do not present a closed-form solution of the valuation equation, meaning that numerous optimization procedures are needed. In addition, they are still restricted by some economic and statistical assumptions, like no-arbitrage or market completeness assumptions. Finally, parametric models fail to adjust to the fast and frequent changes in market participant approaches toward options trading, which reduces the attractiveness to market practitioners.

Therefore, to efficiently price financial derivatives in a fast and accurate way, another line of research based on data-driven models has been developed. If options are considered a functional mapping between contracted terms (inputs) and the premium (output), then the machine learning algorithms become very handy.

1.2. Machine learning algorithms in option pricing

In recent years, machine learning techniques such as neural networks (NN) and support vector regression (SVR) have been used in pricing the options in the nonparametric model. Empirical results show that machine learning techniques can outperform traditional parametric models

* Corresponding authors.
E-mail addresses: yossi.shvimer@gmail.com (Y. Shvimer), spz@uow.edu.au (S.-P. Zhu).

https://doi.org/10.1016/j.eswa.2024.123979
Received 10 July 2023; Received in revised form 25 February 2024; Accepted 11 April 2024
Available online 16 April 2024

(Jang et al., 2021) because of their inherent adaptive learning mechanism. The nonparametric models proposed by different authors (i.e., Gradojevic and Kukolj, 2022; Almeida et al., 2022) use the input variables directly from option market data or some form of derived values such as the spot price of the underlying asset, the strike price of the option contract, the volatility of the underlying asset, the time to maturity of the option contract, the interest rate, and the option moneyness. There is a substantial literature in this domain, starting with the seminal study of Hutchinson et al. (1994) who argued that a data-driven approach is adaptive and responds to structural changes in the data in ways that parametric models cannot. Also, since the models do not rely on restrictive parametric assumptions, they are robust to the specifications errors that limit classical models. In addition, the nonparametric approach is flexible enough to be used in the valuation of a wide variety of derivatives. Garcia and Gençay (2000), Yao et al. (2000), Amilon (2003), and Liu et al. (2019) confirm that an artificial NN is a very performant vehicle to approximate the option pricing function and is more accurate and computationally more efficient than the BS model.

Other papers focus also on different algorithms. Grace (2000) suggests that genetic algorithms are also fitted to evaluate options. Park, Kim, and Lee (2014) realize a comprehensive study comparing the performance of state-of-art parametric no-arbitrage models such as Merton and Heston and non-parametric machine learning models such as the SVR model and Gaussian Process (GP). They found that SVR, GP, and NN have comparable results, all of them outperforming parametrical approaches. Of course, these advantages come with some costs, respectively a large quantity of historical data required for the training set which may be available for less traded derivatives. Also, if the dynamics of the asset are very well modeled, then the parametric models will be better. However, these conditions occur rarely enough so a non-parametric approach can add great practical value to option pricing.

Liang et al. (2009) proposed a cascade architecture on the pre-processed results calculated from the traditional parametric option pricing models (Binomial trees, finite difference, and Monte Carlo) to forecast the option price using NN and Support SVR. They demonstrated the effectiveness of this forecasting model in extensive experiments using data from 122 options collected from the Hong Kong option market in 2006–2007.

Garcia and Gençay (2000) trained their NN model by categorizing and clustering the options data based on the moneyness and time-to-maturity of the option contract, an approach that they call the homogeneity hint (HH) technique. The hint guides the learning process based on the prior supplementary and add-on information about the properties of the unknown function to be learned by the nonparametric machine learning model. The experimental results of Garcia and Gençay (2000) showed that using HH improves option pricing accuracy whereby the out-of-sample mean square prediction error is always lower with the HH information than that achieved by NN without HH information. Improving the predictive performance of forecasting models remains a key concern in the financial domain because even a minor improvement in forecasting accuracy can positively impact financial investments.

Park et al., (2014) investigated the performance of parametric and non-parametric methods concerning the in-sample pricing and out-of-sample prediction performances of the index on the KOSPI 200 Index options in 2001–2010. They found that non-parametric methods significantly outperform parametric methods on both in-sample pricing and out-of-sample pricing. Ivascu (2021) continues the work of Park et al. (2014) and uses three additional models: Decision tree algorithms, XGBoost, and LightGBM. Ivascu (2021) evaluated the model on Call options on crude oil futures between January 2017 and November 2018 and found that the machine-learning algorithms outperformed by a significant margin the classical parametric models (BS and an extension of it, Corrado and Su, 1996).

Das and Padhy (2017) proposed an improved forecasting hybrid model for price options based on the multi-stage cascade architecture of Liang et al. (2009) and the HH concept of Garcia and Gençay (2000). In their hybrid model, Das and Padhy (2017) divided the data into three moneyness levels (in-the-money, at-the-money, and out-the-money) and two maturity levels (less than 30 days and more than 30 days). The model takes as an input the implied volatility for a given day t and calculates the option value for the day t using the BS model Monte-Carlo method and Finite difference, and the option price, and the target is calculating one-day-ahead option price. The model was trained with Support Vector Regression (SVR) and an Extreme learning machine (ELM). Das and Padhy (2017) demonstrated the effectiveness of this forecasting model in extensive experiments using Call option data from the BANK NIFTY index, which includes twelve banks' stocks in 2013–2014. The experimental results of Das and Padhy (2017) showed that using their improved model provides better predictive performance than parametric models (BS model, standard MC, and standard SVR model). These existing hybrid ML models predetermine the moneyness level and maturity before running the algorithm. Therefore, the accuracy of the model depends on the classification made by Das and Padhy (2017), which can be changed from one underlying asset to another.

1.3. Main contribution of this paper

While the nonparametric approach had better empirical results than parametric models, machine learning (ML) algorithms are still considered a “black box”. For example, in neural networks, one cannot calculate the weights to understand the interconnection between neurons or layers, even though some feature importance analysis techniques help to overcome this problem. Also, as Malliaris and Salchenberger (1993) argue, there is no formal theory for determining option network topology. Therefore, decisions like the appropriate number of layers and middle nodes must be determined using experimentation. Furthermore, while the parametric approach had some initial assumptions (continuous of the underlying asset price, risk-neutral, etc.), ML methods do not assume any of those pre-assumptions. Thus, some models might impose great problems with rational player assumption. Therefore, a new method that will keep the parametric approach up to some level, while supporting the moneyness level is needed.

In addition, most of the previous models focused on pricing Call options without providing explicit option prices for Put options. For example, Cao et al. (2021) synthesize virtual calls and obtain put prices via the put-call parity. Put-Call parity is a fundamental concept in options trading. It is based on the idea that the price of a European call option and a European put option with the same strike price and expiration date should be related in a specific way. The importance of Put-Call parity lies in its ability to help ensure that options prices remain consistent without assuming any probability distribution of the future price of the underlying asset. However, Cremers and Weinbaum (2010) showed empirical evidence for significant deviations from put-call parity in options.

This paper took a different approach: keeping the BS model with the current assumption and simplicity, but in the areas where the BS model has some pricing inaccuracy, calibrate, and modify the model using the same parameters with NN. The NN optimization will set the values to add the features to adjust the model. The hybrid model proposed here is a continuous nonlinear combination of the BS model and a nonparametric NN model, which simultaneously determines using NN when to use the BS model price or the nonparametric approach. This paper brings the following contribution to the literature. Firstly, we demonstrate the improvement in the accuracy of out-of-sample forecasting using proposed new hybrid option pricing models as opposed to benchmark models, especially when parametric models have deficiencies. We also expand our model analysis for both Call option and Put option pricing and discuss the Put-Call parity in the hybrid model. Our model price Call and Put separately and then suggest an explanation for the deviation from the Put-Call parity. Finally, by using feature importance we show the sensitivity analysis and the contribution of the parametric and non-parametric parts to the option price.

The remainder of the paper is structured as follows. Section 2 discusses the methodology of the hybrid option pricing model used in this study for both Call and Put options. Section 3 describes the data, in-sample analysis, and input feature importance for the hybrid model. Section 4 compares the performance of the hybrid model with parametric and non-parametric models. Section 5 discusses the Put-Call parity deviation and compares the hybrid model to the real market parity. Finally, Section 6 provides some concluding remarks.

2. The hybrid option pricing model

The BS model calculates the theoretical price of a European option. The model assumes that the financial markets are efficient and that the price movements of the underlying asset follow a geometric Brownian motion. While the model assumes constant volatility for all strikes, in practice, out-of-the-money (OTM) options are often overpriced and exhibit a phenomenon known as the "volatility smile" or "volatility skew". This deviation from the model's constant volatility assumption reflects the market's recognition that volatility tends to vary across strike prices and maturities. We propose a new hybrid model with two sub-models, both of which preserve the strength of the traditional BS option pricing model for the moneyness level of ATM performs very well while adapting themselves with new parameters for the moneyness level of OTM and ITM where the BS model fails to capture the proper market dynamics and thus yield large pricing errors. Hybridizing is automatically determined by the ML algorithm to choose parameters to impose better accuracy on real market data, rather than artificially, as made in previous models (see, for example, Das and Padhy, 2017).

Let the hybrid Call option price $C_{H,t}$ be a function of BS Call option price, $C_{BS,t}$, and common option pricing parameters: the underlying asset price on current day S_t, the strike price of the option K, the annualized standard deviation of the continuously compounded return on the stock σ, time to expiration T, and a risk-free interest rate r.

$$C_{H,t} = f(C_{BS,t}, S_t, K, \sigma, r, T) \tag{1}$$

Similarly, hybrid Put option price $P_{H,t}$ be a function of BS Put option price, $P_{BS,t}$, and the common option pricing parameters within the BS model: the underlying asset price on current day S_t, the strike price of the option K, the annualized standard deviation of the continuously compounded return on the stock σ, time to expiration T, and a risk-free interest rate r.

$$P_{H,t} = g(P_{BS,t}, S_t, K, \sigma, r, T) \tag{2}$$

Where $f(\bullet)$ and $g(\bullet)$ are nonlinear functions.

We will present two hybrid NN topologies for (1) and (2). In the first sub-model, all the input parameters, including BS price are included directly in the NN and provide a single option price as we refer $C_{H1,t}$. This is a fully connected network, with a connection between all inputs to the final output node, the option price. In the second sub-model, we took the parameters S_t, K, σ, r, T and estimate separately a nonparametric option price. Then, we used another NN, including BS in the afterward layer, to provide a hybrid single price $C_{H2,t}$. Therefore, the nonparametric option price is the compensation-optimized function to adjust the BS price to the actual market price. Although the two sub-models are quite similar, the second sub-model allows us to analyze the impact of the nonparametric part that "corrects" the BS model and allows us to analyze the parameters using sensitivity analysis. These two sub-models were separately optimized for Call and Put options.

2.1. Sub-model 1 – Hybrid and BS model fully connected network

In this architecture, the BS model is an input parameter in the learning process (meaning the optimal weights in each layer take some weights for the BS model). Because C_{BS} is a closed-end formula that uses the same input parameters, S_t, K, σ, r, T, pre-process step is needed to calculate the BS option price.

This NN architecture learns the interconnections between all input parameters and optimizes $C_{H1,t}$.

Mathematically, a Call option price is given as:

$$C_{H1,t} = f\left(C_{BS,t}, S_t, K, \sigma, r, T | \theta\right) \tag{3}$$

Where θ is the set of matrix weights and vector of biases in each layer:

$$\theta = (W_1, b_1, W_2, b_2, W_3, b_3) \tag{4}$$

The activation function for the hidden layers of each neuron is a sigmoid function which is a value range between 0 and 1. For the output layer, we used a simple linear activation function. By using the linear activation function, we did not bound the weights for the concatenate layer nor the output value, which can be negative and trim the effect of the overpricing in the BS model for some moneyness levels. Fig. 1 shows the topology of the first sub-model.

Fig. 1 shows that we used two hidden layers, each layer consists of 40 neurons. Therefore, the size of the matrix W_1, W_2 is 40X40,and the size of bias vectors b_1, b_2 is 40. We added a dropout of 10 % after each hidden layer to reduce overfitting. The output is a single value, the Call price $C_{H1,t}$.

The same architecture is applied for the $P_{H1,t}$ where the difference in the input parameters is using the Put option for P_{BS}. Note that in the optimization process, the optimal weight for each input parameter might be different for Put and Call options.

2.2. Sub-model 2 – Hybrid and BS models learn separately and hybridize in the final model value

In this architecture, like in sub-model 1, the BS model is an input parameter in the learning process (meaning the optimal weights in each layer take some weight for the BS model). However, we separated the two models into a non-linear model based on ML and concatenated it to the BS model price only in the final and used another NN to set optimal weights for the nonlinear model and the BS model. Therefore, the option price in M sub-model 2 is optimized both in the non-linear model and in the weights for the BS model and nonlinear model.

Mathematically, a Call option price in the second architecture is given as:

$$C_{H2,t} = g\left(\varphi(S_t, K, \sigma, r, T|\theta_1), C_{BS,t}|\theta_2\right) \tag{5}$$

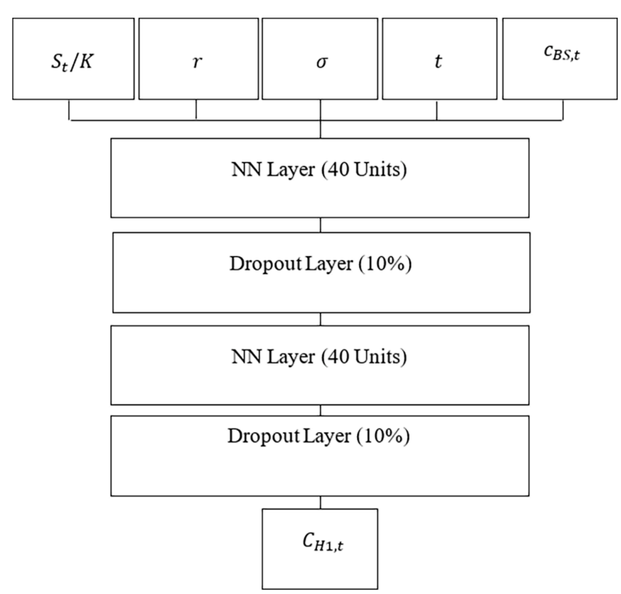


Fig. 1. Topology hybrid sub-model $C_{H1,t}$.

Where $\varphi(\bullet)$ is the nonparametric price, θ_1 is the set of matrix weights W_i and vector of biases b_i in each layer i for the nonparametric price:

$$\theta_1 = (W_1, b_1, W_2, b_2, W_3, b_3) \tag{6}$$

and θ_2 is the set of matrix weights and vector of biases for concatenating between the nonparametric part and the BS model:

$$\theta_2 = (W_4, b_{4,} W_5, b_{5,}) \tag{7}$$

The activation function for the hidden layers for each neuron is a sigmoid function and for the output layer, we used a simple linear function.

For example, hidden layer 4 (after the concatenation) has two equation inputs: the nonparametric model price $\varphi(\bullet)$ and $C_{BS,t}$. By using equation (5) and (7), the output of hidden layer 5, $C_{H2,t}$ is:

$$C_{H2,t} = W_5 \frac{1}{1 + e^{(W_{4,1}\varphi(\bullet) + W_{4,2}C_{BS,t} + b_4)}} + b_5 \tag{8}$$

Fig. 2 shows the architecture of sub-model 2.

The same architecture applied for the $P_{H2,t}$ where the difference in the input parameters is using the Put option for P_{BS}.

Lemma 1. Hybrid option prices $C_{H,t}$ and $P_{H,t}$ is a continuous on σ and have a unique solution when using Neural Network while the last layer output has one neuron.

Proof. *First, we will show that $C_{H,t}$ is a continuous function.*

The first step is to measure if all the inputs are continuous. By definition, $S_t, K, \sigma, T \geq 0$ and r is continuous. Also, by the same definition, we can say that $C_{BS,t}$ is continuous, and that for any given parameter, BS model can produce a single price $C_{BS,t}$.

The second step is to show that the NN is also continuous. We can show that by analyzing the activation function. We used the sigmoid function for the hidden layers and the linear function for the output function, as shown in (8). The sigmoid function is a continuous, monotonically increasing function with a value range from 0 to 1. The second activation we used in the output layer, which is the linear function which, is also continuous and monotonic. Therefore, $C_{H,t}$ by its definition is continuous and has a unique solution when $S_t, K, \sigma, T \geq 0$. Note, that according to the topologic of the model, it cannot assure that $C_{H,t} \geq 0$ (e.g., when $b_5 > 0$ in (8)). This can only be used during the optimization process.

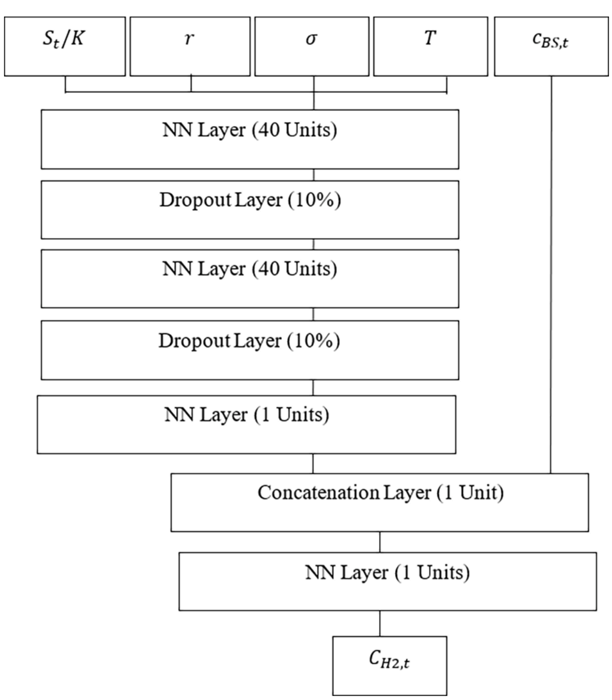


Fig. 2. Topology hybrid sub-model 2 $C_{H2,t}$.

3. Data

We have used historical data between January 1st, 2017, and December 30th, 2022. The Call and Put options data were obtained from CBOE on SPX, the underlying asset of the S&P 500 index. We used the data for a specific time, 15:45 daily time, which represents "real-time" trading with a full picture of the options market as well as the S&P 500 market, including Bid and Ask prices and real market prices. We screened out options with no open interest or no volume and those options in which the mid option price is less than 1/8 of a dollar or with moneyness greater than 1.1 or less than 0.9) and options with days to expiration higher than 120 days. For the risk-free parameter, we used the real-time yield curve for US Government Bonds to match the option expiration day to the risk-free for the same period using Nelson and Siegel (1987) and Svensson (1994) approximation. We used the VIX closing level of the previous day as the standard deviation parameter, following Wang (2019), that found that a higher VIX level has a stronger explanation ability to the international stock markets volatility. Wang's (2019) out-of-sample results indicate that the VIX increases the accuracy of the forecast, especially the large VIX levels. For the market option price, C_{Market}, we used the Bid Ask midpoint price.

A total number of 1,481,446 Call options and 1,725,136 Put options were sampled. We divided them chronologically as follows: 80 % of sampled data (sorted by date) will be used to train the model and calibrate the parameters of the options using real market option prices, and the remaining 20 % will be used to test the model and show the accuracy error compared to real market option prices. Table 1 shows the dataset analysis.

In Table 1, we define the moneyness level as OTM options when $S_t/K < 0.97$, ATM options when $0.97 \leq S_t/K < 1.03$ and ITM options when $S_t/K \geq 1.03$.

Table 1 shows that as the expiration days are close, more Call and Put options are traded. Also, most Call and Put options traded are ATM options within less than 30 days to expiration. Also, Table 1 shows that more OTM options are traded than the ITM options for both Call and Put options since OTM options are usually traded more actively due to the demand for hedging, as OTM options are nearly always less costly than ITM options.

3.1. Models calibration

The training process aims to learn the optimal weights and biases $(W_1, b_1, W_2, b_2, \cdots, W_n, b_n)$ to make the model option function as small as possible to the actual option price in the market using a backpropagation gradient decent method. The optimization function is defined as:

Table 1
Dataset summary.

Number of options	OTM	ATM	ITM	**Total**
Call				
$t \leq 30$	251,901	567,551	121,666	**941,118**
$30 < t \leq 60$	145,443	180,806	36,459	**362,708**
$t > 60$	78,217	78,071	21,332	**177,620**
Total Call	475,561	826,428	179,457	**1,481,446**
Put				
$t \leq 30$	531,237	525,328	86,313	**1,142,878**
$30 < t \leq 60$	198,173	174,094	24,762	**397,029**
$t > 60$	84,795	82,071	18,363	**185,229**
Total Put	814,205	781,493	129,438	**1,725,136**
Total options				
$t \leq 30$	783,138	1,092,879	207,979	2,083,996
$30 < t \leq 60$	343,616	354,900	61,221	759,737
$t > 60$	163,012	160,142	39,695	362,849
Total	**1,289,766**	**1,607,921**	**308,895**	**3,206,582**

$$J = \underset{\theta}{\text{argmin}} L(C_{H,t}, C_{Market,t}) \tag{9}$$

The optimization algorithms start with initial values for θ and move in the direction the loss function decreases. We have used a mean absolute error (MAE) function as loss function L for the error between model price and market price, and with ADAM (Adaptive Moment Estimation) as the optimizer for the weights. In the training process, we use the Bayesian optimization algorithm to find the optimal hyperparameters on the training set. We used 1,000 epochs, a batch size of 512, and a learning rate of 0.001. We divided the training dataset for 75 % of data for training and 25 % of data for cross-validation to reduce overfitting. We divided the training dataset for 75 % of data for training and 25 % of data for cross-validation to reduce overfitting. The optimization problem finds a local minimum. Therefore, in order not to prevent one input feature from obtaining a higher weight due to its relative absolute value, we have scaled the option price of the BS model by the strike price. We also scaled the actual option price in the market by the strike price. Then, to compute the hybrid price model (and not the scaled), we have multiplied the model output by the strike price. This is commonly used in option pricing algorithms (e.g., Raberto et al., 2000; Bennell and Sutcliffe, 2004). Therefore, (9) is optimized as:

$$\begin{aligned} J &= \underset{\theta}{\text{argmin}} L\left(\frac{C_{H,t}}{K}, \frac{C_{Market,t}}{K}\right) \\ &= \underset{\theta}{\text{argmin}} L\left(\frac{C_{H,t}(S_t/K, \sigma, r, T, C_{BS,t}/K)}{K}, \frac{C_{Market,t}}{K}\right) \end{aligned} \tag{10}$$

where (10) is the general optimization problem for both sub-models.

3.2. In-sample analysis

For a given option from the trading system, we evaluate the performance of the models using standard statistical metrics: RMSE and MAE to measure the difference between the theoretical price obtained by each model and the actual market price (see also Bhat and Arekar, 2016; Figueroa-López and Mancini, 2019). The definition of these performance metrics is:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(C_{H,i} - C_{Market,i}\right)^2}{N}} \tag{11}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|C_{H,i} - C_{Market,i}\right| \tag{12}$$

where N is the total number of Call options (N = 1,185,157) in the training set. The same calculation was also applied for Put options (N = 1,380,109).

Smaller values of RMSE and MAE indicate better predictive performance. Similar to previous empirical studies (e.g., Akyildirim et al., 2021), we made comparisons between model prices and actual prices for the same database, where the actual market prices were calculated as the mid-prices between the Bid price and the Ask price. The lower the (post hoc) value of the MSE, the greater the proximity of model prices to actual prices. Table 2 shows the accuracy results by the models for different moneyness levels. As we trained two different models, we will show the accuracy of those models using the common root mean squared error (RMSE) and mean absolute error (MAE) of each model from its actual market price. We used the BS model as a peer model.

In Table 2, we define the moneyness level as OTM options when $S_t/K < 0.97$, ATM options when $0.97 \le S_t/K < 1.03$ and ITM options when $S_t/K \ge 1.03$.

Table 2 also shows that both sub-models had notably lower error values for every moneyness level compared with the BS model. The improvement for both models lies in the optimization process, the optimal weights and biases parameters calibrated to achieve higher accuracy in predicting market prices during the in-sample process. Also, Table 2 shows that the error for the BS model decreases with moneyness for Call options while increasing with moneyness for Put options, while the error of the hybrid sub-models remains relatively consistent for all moneyness levels. The increasing error for the BS model with moneyness relates to the constant volatility assumption in the BS model, which in practice does not hold and causes overpricing for OTM options.

3.3. Hybridization parameters

After the weights and biases optimization process, we analyze the values of the matrix and biases in the second architecture model to get more sense of the results for Call and Put options. More specifically, we will discuss (7) and (8) for both Call and Put options assuming we have two values: $\varphi(\bullet)$ for nonlinear (ML) part and $C_{BS,t}$ for the Call option and $P_{BS,t}$ for the Put option. Table 3 shows the values in (7) and (8).

Table 3 shows that nonparametric part $W_{6,1}$ has a positive weight on the option price (Call and Put) and the BS model has a positive weight for the Call option and negative weight for the Put option. Interestingly, within the last layer (the output layer), the signs of W_5 and b_5 are opposite for the Call and Put Option, meaning that with an increase in the input parameters $\varphi(\bullet)$ and $C_{BS,t}$, the Call option and Put options will move in the opposite which makes perfect sense financially. From Table 3 we can reconstruct (8) using the parameters. Note that we scaled the Option price and BS model price, by dividing them by the strike price. Equations (13) and (14) show the Call and Put option prices using the parameters.

$$\frac{C_{H2,t}}{K} = \frac{-1.15}{1 + e^{\left(0.248\varphi(\bullet) + 1.987\frac{C_{BS,t}}{K} - 0.621\right)}} + 0.757 \tag{13}$$

Table 3
In-sample optimized parameters from the concatenation layers in $C_{H2,t}$ and $P_{H2,t}$.

	$W_{4,1}$	$W_{4,2}$	b_4	W_5	b_5
Call option	0.248	1.987	−0.6208	−1.15	0.757
Put option	0.06	−1.673	−0.144	0.898	−0.201

Table 2
In-sample accuracy results by the models for different moneyness levels.

	RMSE				MAE			
	OTM	ATM	ITM	Total	OTM	ATM	ITM	Total
Call Options								
Hybrid Model 1	0.00197	0.00213	0.00247	0.00213	0.00090	0.00128	0.00133	0.00117
Hybrid Model 2	0.00234	0.00223	0.00250	0.00230	0.00167	0.00156	0.00173	0.00162
BS Model	0.00688	0.00502	0.00268	0.00544	0.00485	0.00393	0.00130	0.00388
Put Options								
Hybrid Model 1	0.00190	0.00251	0.00447	0.00240	0.00091	0.00148	0.00295	0.00129
Hybrid Model 2	0.00186	0.00257	0.00417	0.00239	0.00091	0.00159	0.00293	0.00134
BS Model	0.00274	0.00393	0.00691	0.00367	0.00172	0.00275	0.00424	0.00233

$$\frac{P_{H2,t}}{K} = \frac{0.898}{1 + e^{\left(0.06\varphi(\bullet) - 1.673\frac{P_{BS,t}}{K} - 0.464\right)}} - 0.201 \tag{14}$$

where, by definition, $C_{BS,t}, P_{BS,t} \geq 0, K > 0$.

3.4. Input parameters weights importance

To understand the contribution of each input parameter, we used the Shapley additive explanations (SHAP) values for our set of inputs (Van den Broeck et al., 2022). The advantage of the SHAP method lies in its ability to explain the output of machine learning models. The SHAP values show how much each predictor contributes (positively or negatively) to the output. Another benefit of the SHAP approach is its local interpretability, where each observation is assigned its own set of SHAP values. First, we will estimate the SHAP values across the training data sets for the first model by taking the mean absolute value of each feature.

Fig. 3 displays the SHAP summary plots for the hybrid sub-model 1 that explain how each input contributes to the output and in which direction. This plot utilizes the training data and ranks features vertically in descending order according to the magnitude of their SHAP values. For all observations, their location along the x-axis shows whether the impact is associated with a higher or lower prediction. What stands out in Fig. 3 is the strong dominance of the BS price. This indicates that the hybrid sub-model 1 indeed contributes to modifying the dominance of the BS to obtain a more accurate market price while the other input parameters have a less average net impact on the option price.

3.5. Numerical experiments and discussions

While the hybrid sub-models use a significant number of parameters within the hidden layers, as shown in Fig. 1 & Fig. 2, we can easily demonstrate the sensitivity of the sub-models to different parameters and the importance of the non-parametric part to the option price. First, we will show the sensitivity of the option prices for each parameter $S_t/K, \sigma, r, T$ for Call options. For the numerical example, we will use the following parameters: $T = 1/52, r = 0.02, S_t = 100, \sigma_t = 0.2, K = 100$. For the specific example the Call option prices are: $C_{H1,t} = 0.79$, $C_{H2,t} = 0.89$, $C_{BS,t} = 1.125$. Fig. 4 shows the option price sensitivity of the Call option price parameters.

Fig. 4 shows that the two hybrid sub-models show significantly similar option prices for any given input parameter. This is a result of the optimization process each hybrid sub-model uses, as shown in Table 4. While both hybrid sub-models are similar, there are notable differences from the BS model for any given input parameters. Fig. 4a shows that when the strike is not deep out of money (when the option value is close to zero) nor deep in the money (when the option price is $S - K$), the hybrid sub-models had a lower price than the BS model. This gap in option price between hybrid sub-models and BS increases to the maximum when the option is at-the-money.

Fig. 4b and Fig. 4c show that hybrid sub-models are less sensitive (lower slope) with a lower absolute price to the volatility of the underlying asset price and the time to expiration than the BS model. This underpricing effect is captured within the nonlinear model of the NN, and the hybridization of both models allows this higher accuracy. Fig. 4d shows that the sensitivity of the BS model price to the interest rate is higher than the hybrid sub-models. We attribute these results to the fact that the sample data for the training set was in a period when the interest rate was lower. Thus, the impact of the interest rate parameter is lower, as shown in the feature importance analysis.

We now show the analysis for the Put options. For the specific example made for the Call price when $T = 1/52, r = 0.02, S_t = 100, \sigma_t = 0.2, K = 100$, the Put option prices are: $P_{H1,t} = 0.85$, $P_{H2,t} = 0.87$ $P_{BS,t} = 1.09$. Fig. 5 shows the option price sensitivity of the Put option price parameters.

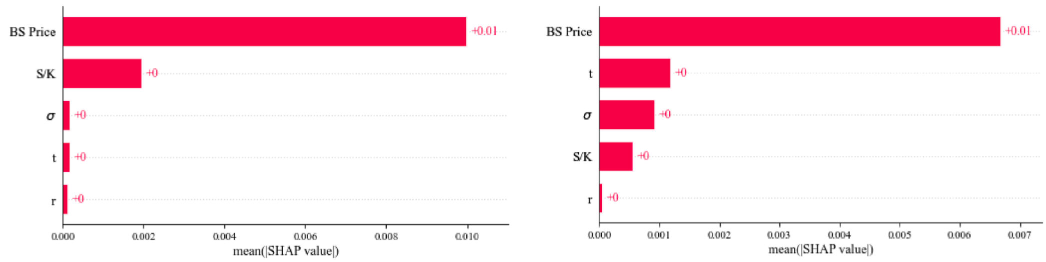
Fig. 5 shows that both hybrid sub-models show significantly similar option prices for any given input parameter, like the sensitivity analysis made for the Call price, because of the optimization process each hybrid sub-model uses, as shown in Table 4. While both hybrid sub-models are similar, there are notable differences from the BS model for any given input parameters. Fig. 5a shows that when the options are deep in the money, the hybrid sub-models prices were less sensitive and had a lower price than the BS model. This result does not correspond with the sensitivity analysis made for the Call options. We relate this result for the Put option to the relatively lower samples of ITM option data, which could be insufficient to learn the option price importance for ITM options.

Fig. 5b and Fig. 5c show that the hybrid sub-models are less sensitive (lower slope) with a lower absolute price to the volatility of the underlying asset price and the time to expiration than the BS model, which is similar to the results for the Call price. This underpricing effect is captured within the nonlinear model of the NN, and the hybridization of both models allows this higher accuracy. Fig. 5d shows that the sensitivity of the BS model price to the interest rate is higher than that of the hybrid sub-models. We attribute these results to the fact that the sample data for the training set was in a period when the interest rate was lower. Thus, the impact of the interest rate parameter is lower, as shown in the feature importance analysis.

In conclusion, the in-sample analysis shows that the hybrid sub-models prices have more accurate and stable results to market price compared with the BS model for all moneyness levels. We also found that BS price, as one of the input features for the hybrid sub-models, had substantial dominance in the input feature weights while the other input parameters had less importance to price sensitivity. We also found that for the hybrid model, the nonparametric part has a negative impact on the option price (Call and Put), and the BS model has a positive impact, meaning the nonparametric part is trimming the BS price (if the nonparametric part has a positive sign). Furthermore, in the last layer (the output layer), the sign of weights and bias are opposite for the Call and Put Option, meaning that with an increase in the input parameters $\varphi(\bullet)$ and $C_{BS,t}$, the Call option and Put options will move in the opposite direction (e.g., Call option price increase and Put option price decrease).

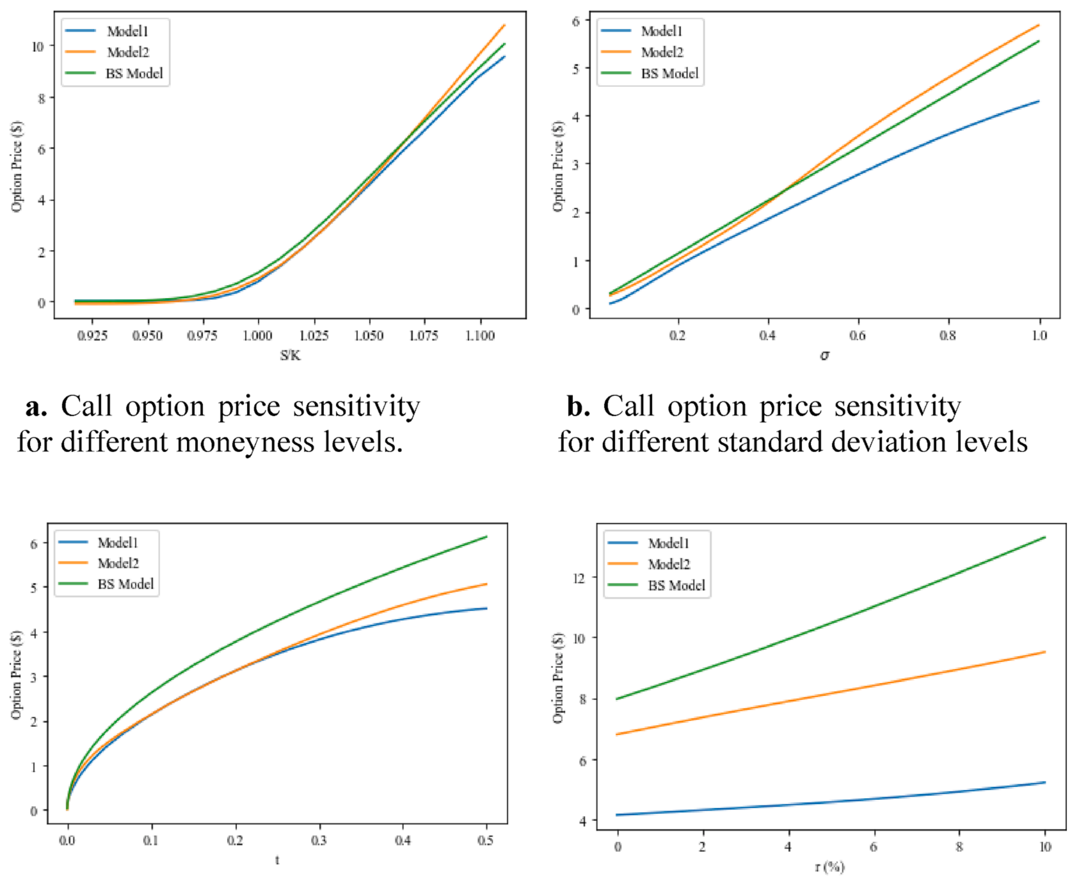
a. SHAP values for the Call option price **b.** SHAP values for the Put option price

Fig. 3a. SHAP values for the Call option price.



a. SHAP values for the Call option price **b.** SHAP values for the Put option price

Fig. 3b. SHAP values for the Put option price.



a. Call option price sensitivity for different moneyness levels.

b. Call option price sensitivity for different standard deviation levels

c. Call option price sensitivity for time to expiration (years)

d. Call option price sensitivity for different interest rate levels (%).

Fig. 4. Call options sensitivity analysis for the hybrid sub-models price and BS price for input parameters. For the numerical example, the following parameters: $T = 1/52, r = 0.02, S_t = 100, \sigma_t = 0.2, K = 100$.

4. Out of sample analysis

In the out-of-sample analysis, we will use the remaining 20 % of the dataset and the optimized parameters calibrated in the in-sample learning. Furthermore, to also use two additional algorithms: XGBoost and Decision trees, as proposed by Ivascu (2021), and Support Vector regression (SVR) purposed by Das and Padhy (2017) that showed superior results over the BS model. These models take the same input as the hybrid sub-models. Table 4 shows the Out-of-sample accuracy results of the models for different moneyness levels.

In Table 4, we define the moneyness level as OTM options when $S_t/K < 0.97$, ATM options when $0.97 \leq S_t/K < 1.03$ and ITM options when $S_t/K \geq 1.03$.

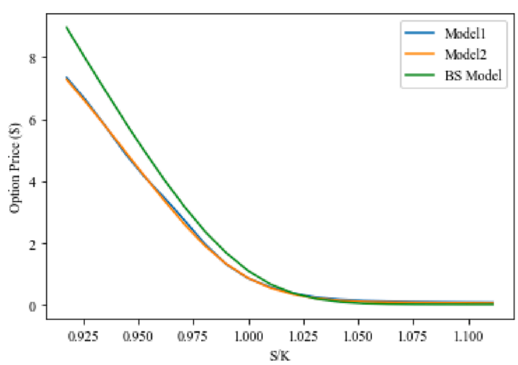
Table 4 shows that hybrid sub-models shows overall better accuracy compared with all given models for both RMSE and MAE. Table 4 also shows that XGBoost had relatively close results to our hybrid sub-models, while SVR had poorly out of sample errors for all moneyness levels and for Put and Call options. Both hybrid sub-models also showed relatively high accuracy to market price, especially for ATM options for both Call and Put options. Therefore, we can conclude that the hybrid sub-models indeed increase the accuracy of the BS model in the ATM while improving the accuracy for OTM options as well as ITM options.

4.1. Robustness test results

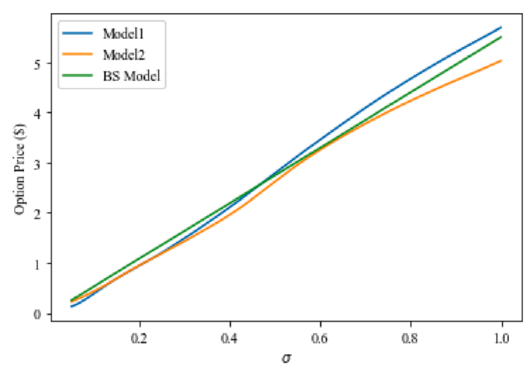
Several methods to compute the robustness quantification of neural

Table 4
Out-of-Sample accuracy results by the models for different moneyness levels.

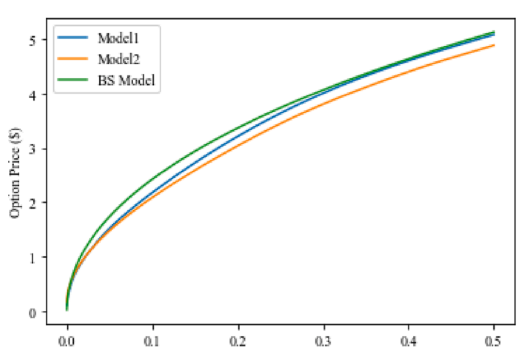
	RMSE				MAE			
	OTM	ATM	ITM	Total	OTM	ATM	ITM	Total
Call Options								
Hybrid Model 1	0.00223	0.00256	0.00205	**0.00238**	0.00155	0.00194	0.00127	**0.00170**
Hybrid Model 2	0.00279	0.00261	0.00283	**0.00271**	0.00206	0.00197	0.00219	**0.00203**
XGBoost	0.00280	0.00336	0.00269	**0.00307**	0.00179	0.00247	0.00191	**0.00212**
SVR	0.08999	0.07499	0.03867	**0.07858**	0.08975	0.07372	0.03472	**0.07591**
Decision trees	0.00307	0.00444	0.00358	**0.00383**	0.00194	0.00323	0.00221	**0.00258**
BS Model	0.00561	0.00419	0.00198	**0.00466**	0.00408	0.00325	0.00124	**0.00337**
Put Options								
Hybrid Model 1	0.00166	0.00281	0.00395	**0.00263**	0.00117	0.00219	0.00334	**0.00193**
Hybrid Model 2	0.00192	0.00272	0.00279	**0.00243**	0.00134	0.00208	0.00218	**0.00179**
XGBoost	0.00252	0.00315	0.00301	**0.00288**	0.00174	0.00234	0.00201	**0.00204**
SVR	0.07760	0.06463	0.03058	**0.06683**	0.07710	0.06336	0.02658	**0.06366**
Decision trees	0.00408	0.00460	0.00313	**0.00420**	0.00270	0.00337	0.00185	**0.00287**
BS Model	0.00218	0.00322	0.00443	**0.00306**	0.00145	0.00237	0.00282	**0.00205**



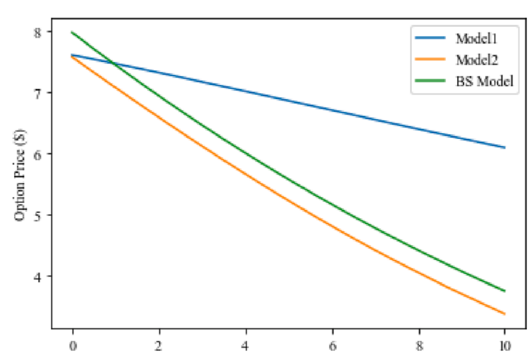
a. Put option price sensitivity for different moneyness levels.



b. Put option price sensitivity for different standard deviation levels



c. Put option price sensitivity for time to expiration (years)



d. Put option price sensitivity for different interest rate levels (%).

Fig. 5. Put options sensitivity analysis for the hybrid sub-models price and BS price for input parameters. For the numerical example, the following parameters: $T = 1/52, r = 0.02, S_t = 100, \sigma_t = 0.2, K = 100$.

networks for Out-of-Sample data were presented in the literature (See, for example, Deng et al., 2020; Ko et al., 2019). To check robustness, we randomly shuffle the Out-of-Sample actual option price data (Actual option price). Thus, the robustness is measured by the distortion between successful adversarial examples and the original ones. By shuffling the data, we can ensure that the results are not arbitrary. If the RMSE between the model and the actual option price is similar to the RMSE of the model to a random shuffle actual price, we can conclude that the model is not robust. Table 5 shows the robustness test RMSE results for the three models.

Table 5 presents the results of the robustness test. The consistent RMSE values for Call and Put options across most models indicate a robust performance in generating similar pricing errors under adversarial conditions and are significantly higher than true Out-of-Sample RMSE, as shown in Table 4. However, the SVR model had a similar RMSE as in true Out-of-Sample RMSE, which might indicate the overfitting of this model.

Table 5
Robustness test RMSE for shuffle data.

	RMSE – Call options	RMSE –Put options
Hybrid Model 1	0.0291	0.0294
Hybrid Model 2	0.0292	0.0295
XGBoost	0.0292	0.0292
SVR	0.0786	0.0668
Decision trees	0.0291	0.0294
BS Model	0.0294	0.0309

5. Put-Call parity

The Put-Call parity refers to the relationship between the price of a European Call and that of its Put counterpart with the same underlying price, strike price, interest rate, volatility, and time to expiration. The original Put-Call parity was discovered by Stoll (1969) and later extended and modified by Merton (1973), as shown in (3).

$$C + Ke^{-rT} = P + S_t \tag{15}$$

Unfortunately, in market practice, such a simple and elegant relationship never strictly holds as a lot of empirical evidence suggests that deviations from Put-Call parity are mainly caused by 1) the information about future returns (Cremers and Weinbaum, 2010); 2) short-sell constraints (Ofek et al., 2004). One of the advantages of the proposed hybrid model is to address such a deviation in the Put-Call parity and thus capture market behavior that has already taken future returns and asymmetry in shorts into consideration. Let us use the hybrid sub-model 2 to explain this by showing how the deviation from the Put-Call parity is addressed by the nonparametric part of our model.

We start introducing a new term $\varepsilon_{p,c}$ in (15) representing the deviation from the Put-Call Parity and then plugging (3) and (4) in the revised (15) to yield

$$f\left(C_{BS,t}, S_t, K, \sigma, T, t\right) + Ke^{-rT} = g\left(P_{BS,t}, S_t, K, \sigma, r, T\right) + S_t + \varepsilon_{p,c} \tag{16}$$

We now introduce a weight α (β) which represents the proportion of the BS model Call (Put) price in the hybrid model price, and $1-\alpha$ $(1-\beta)$ representing the non-parametric Call (Put) price. Then, we can rewrite (16) as follows:

$$\begin{aligned}\varepsilon_{p,c} = {} & \alpha C_{BS,t} + (1-\alpha) f(S_t, K, \sigma, r, T) + Ke^{-rT} - \beta P_{BS,t} \\ & -(1-\beta) g(S_t, K, \sigma, r, T) - S_t\end{aligned} \tag{17}$$

In the special case of $\alpha = \beta$, (17) can be rewritten as:

$$\varepsilon_{p,c} = \alpha C_{BS,t} + (1-\alpha) f(\bullet) + \alpha Ke^{-rT} + (1-\alpha) Ke^{-rT} - \alpha P_{BS,t} - (1-\alpha) g(\bullet) - \alpha S_t - (1-\alpha) S_t \tag{18}$$

After further simplification, we arrive at equation (19) below, which quantifies the deviation from the Put-Call Parity obtained from the nonparametric model.

$$\varepsilon_{p,c} = (1-\alpha)\left(f(\bullet) + Ke^{-rT} - g(\bullet) - S_t\right) \tag{19}$$

A similar explanation can be constructed for sub-model 1 as well.

Now, we verify the above theoretical explanation empirically. Fig. 6 displays the variation of $\varepsilon_{p,c}$ as a function of the moneyness based on the dataset discussed in Section 4. Clearly, both sub-models have captured market deviation from the Put-Call parity as a strictly "parity" would mean a horizontal line with $\varepsilon_{p,c} = 0$. However, sub-model 2 performs better than sub-model 1 as it is much closer to the deviation displayed by the market data, although there are still rooms to be improved. In terms of moneyness level, Fig. 6 shows that two sub-models indicate lower deviations for OTM options. This further confirms the two main reasons for the deviation from the Put-Call parity after the nonparametric part has been demonstrated to capture market behavior.

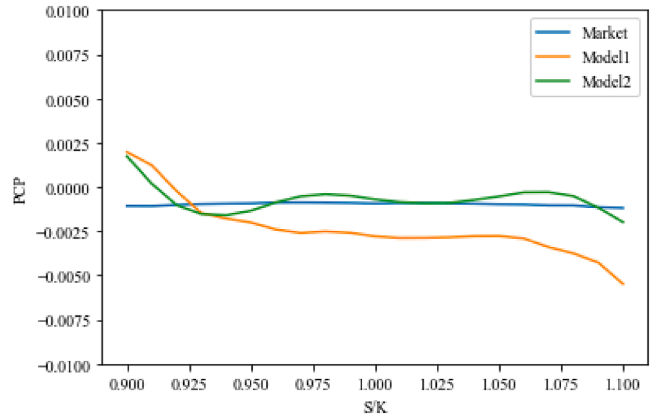


Fig. 6. Put-call parity of market price and hybrid sub-models for different moneyness levels.

A more interesting discussion should be around the variation of the deviation in terms of the time to expiry. Fig. 7 shows an example of such a variation. As the time to expiry approaches zero, the intrinsic option value is dominant, and thus, the nonparametric part is very small. In other words, option prices have to converge to their payoffs unless someone tries to lose money. As the time to expiry increases, the deviation and volatility from the Put-Call parity level increase. We relate that to different views of market players on future returns as reflected in the Call and Put option prices. This volatility on the Put-Call parity is displayed in Fig. 7. As time to expiry decreases, the Put-Call parity converges to almost zero because of the decreasing uncertainty as the option approaches expiry. Clearly, the hybrid model captures the negative values of the deviation. However, sub-model 1 consistently performs better than sub-model 2 across all the time to expiry.

6. Conclusions

6.1. Summary

This paper deals with the key question of the real-time pricing of stock options, a question of interest to researchers and practitioners alike. While the nonparametric approach had better empirical results than parametric models, ML algorithms are still considered a "black box". This paper took a different approach: keeping the BS model with the current assumption and simplicity, but the areas where the BS models have some pricing inaccuracy, calibrate, and modify the model using the same parameters with NN. The NN optimization will set the values to add the features to adjust the model. This paper brings the

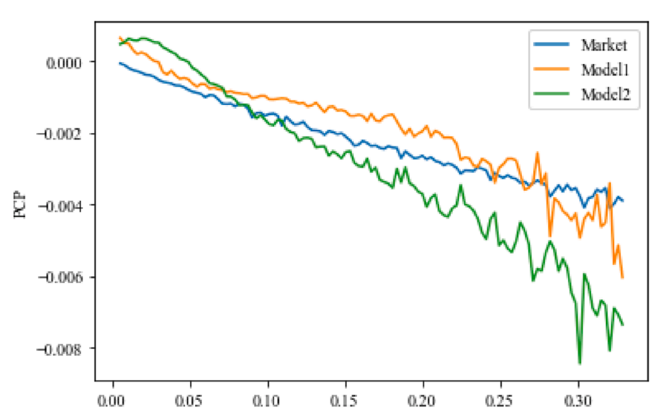


Fig. 7. Put-Call parity of market price and hybrid sub-models for different time to expiration.

following contribution to the literature. Firstly, we demonstrate the improvement in the accuracy of out-of-sample forecasting using proposed new hybrid option pricing models as opposed to benchmark models, especially when parametric models have deficiencies. We also expand our model analysis for both Call option and Put option pricing and discuss the Put-Call parity in the hybrid model. Finally, by using feature importance, we show the sensitivity analysis and the contribution of a parametric and non-parametric part to the option price.

6.2. Advantages of the model and implications

In-sample analysis shows that the hybrid sub-models prices have more accurate and stable results to market price compared with the BS model for all moneyness levels. We also found that BS price, as one of the input features for the hybrid sub-models, had large dominance in the feature importance while the other input parameters had less importance to price sensitivity. We also found that for the hybrid model, the nonparametric part has a negative impact on the option price (Call and Put), and the BS model has a positive impact, meaning the nonparametric part is trimming the BS price (if the nonparametric part has a positive sign). Furthermore, in the last layer (the output layer), the signs of Matrix and Bias are opposite for the Call and Put Option, meaning that with an increase in the input parameters $\varphi(\bullet)$ and $C_{BS,t}$, the Call option and Put options will move in the opposite direction (e.g., Call option price increase and Put option price decrease).

In the Out-of-Sample, this paper demonstrates the improvement in accuracy of out-of-sample forecasting using proposed new hybrid option pricing models as opposed to benchmark models, especially when parametric models have deficiencies. We also expanded our model analysis for both call option and put option pricing. Furthermore, we calculated the Put-Call parity in the hybrid sub-models and explained the deviation from Put-Call parity, which results in options far from expiration day or OTM options.

Our model provides market practitioners with a new framework for real-time options pricing with higher accuracy than existing models and to hedge better the exposure to the underlying asset price risk. It also helps to set new boundaries for Put-Call parity even in real-time market, when Put-Call parity does not hold and benefit from deviation these boundaries.

Our model was tested using S&P500 options. Therefore, it would be interesting to examine the validity of the model to individual stock options, other stock indices options, or commodities options.

CRediT authorship contribution statement

Yossi Shvimer: Conceptualization, Methodology, Software, Writing – original draft. **Song-Ping Zhu:** Writing – review & editing, Investigation, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Akyildirim, E., Cepni, O., Corbet, S., & Uddin, G. S. (2021). Forecasting mid-price movement of Bitcoin futures using machine learning. *Annals of Operations Research*, 1–32.

Almeida, C., Freire, G., Azevedo, R., & Ardison, K. (2022). Nonparametric Option Pricing with Generalized Entropic Estimators. *Journal of Business & Economic Statistics*, 1–15.

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

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

Bennell, J., & Sutcliffe, C. (2004). Black-Scholes versus artificial neural networks in pricing FTSE 100 options. *Intelligent Systems in Accounting, Finance & Management: International Journal, 12*(4), 243–260.

Bhat, A., & Arekar, K. (2016). Empirical performance of Black-Scholes and GARCH option pricing models during turbulent times: The Indian evidence. *International Journal of Economics and Finance, 8*(3), 123–136.

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

Cao, Y., Liu, X., & Zhai, J. (2021). Option valuation under no-arbitrage constraints with neural networks. *European Journal of Operational Research, 293*(1), 361–374.

Carr, P., & Wu, L. (2004). Time-changed Lévy processes and option pricing. *Journal of Financial economics, 71*(1), 113–141.

Corrado, C. J., & Su, T. (1996). Skewness and kurtosis in S&P 500 index returns implied by option prices. *Journal of Financial research, 19*(2), 175–192.

Cremers, M., & Weinbaum, D. (2010). Deviations from put-call parity and stock return predictability. *Journal of Financial and Quantitative Analysis, 45*(2), 335–367.

Das, S. P., & Padhy, S. (2017). A new hybrid parametric and machine learning model with homogeneity hint for European-style index option pricing. *Neural Computing and Applications, 28*(12), 4061–4077.

Deng, M., Meng, T., Cao, J., Wang, S., Zhang, J., & Fan, H. (2020). Heart sound classification based on improved MFCC features and convolutional recurrent neural networks. *Neural Networks, 130*, 22–32.

Figueroa-López, J. E., & Mancini, C. (2019). Optimum thresholding using mean and conditional mean squared error. *Journal of Econometrics, 208*(1), 179–210.

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

Gradojevic, N., & Kukolj, D. (2022). Unlocking the black box: Non-parametric option pricing before and during COVID-19. *Annals of Operations Research*, 1–24.

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

Hull, J., & White, A. (1987). The pricing of options on assets with stochastic volatilities. *The Journal of Finance, 42*(2), 281–300.

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

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

Jang, J. H., Yoon, J., Kim, J., Gu, J., & Kim, H. Y. (2021). DeepOption: A novel option pricing framework based on deep learning with fused distilled data from multiple parametric methods. *Information Fusion, 70*, 43–59.

Jarrow, R., & Rudd, A. (1982). Approximate option valuation for arbitrary stochastic processes. *Journal of financial Economics, 10*(3), 347–369.

Liang, X., Zhang, H., Xiao, J., & Chen, Y. (2009). Improving option price forecasts with neural networks and support vector regressions. *Neurocomputing, 72*(13–15), 3055–3065.

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

Madan, D. B., Carr, P. P., & Chang, E. C. (1998). The variance gamma process and option pricing. *Review of Finance, 2*(1), 79–105.

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

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of financial economics, 3*(1–2), 125–144.

Nelson, C. R., & Siegel, A. F. (1987). Parsimonious modeling of yield curves. *Journal of business*, 473–489.

Ofek, E., Richardson, M., & Whitelaw, R. F. (2004). Limited arbitrage and short sales restrictions: Evidence from the options markets. *Journal of Financial Economics, 74*(2), 305–342.

Park, H., Kim, N., & Lee, J. (2014). Parametric models and non-parametric machine learning models for predicting option prices: Empirical comparison study over KOSPI 200 Index options. *Expert Systems with Applications, 41*(11), 5227–5237.

Raberto, M., Cuniberti, G., Riani, M., Scales, E., Mainardi, F., & Servizi, G. (2000). Learning short-option valuation in the presence of rare events. *International Journal of Theoretical and Applied Finance, 3*(03), 563–564.

Schroder, M. (1989). Computing the constant elasticity of variance option pricing formula. *the. Journal of Finance, 44*(1), 211–219.

Stoll, H. R. (1969). The relationship between put and call option prices. *The Journal of Finance, 24*(5), 801–824.

Svensson, L. E. (1994). Estimating and interpreting forward interest rates: Sweden 1992-1994.

Van den Broeck, G., Lykov, A., Schleich, M., & Suciu, D. (2022). On the tractability of SHAP explanations. *Journal of Artificial Intelligence Research, 74*, 851–886.

Wang, H. (2019). VIX and volatility forecasting: A new insight. *Physica A: Statistical Mechanics and its Applications, 533*, Article 121951.

Yao, J., Li, Y., & Tan, C. L. (2000). Option price forecasting using neural networks. *Omega, 28*(4), 455–466.