Option valuation under no-arbitrage constraints with neural networks

Yi Cao, Xiaoquan Liu, Jia Zhai

University of Nottingham
UK | CHINA | MALAYSIA

University of Nottingham Ningbo China, 199 Taikang East Road, Ningbo,
315100, China

First published 2020

Option valuation under no-arbitrage constraints with neural networks

Yi Cao[*] Xiaoquan Liu[†] Jia Zhai[‡]

November, 2020

Abstract

In this paper, we start from the no-arbitrage constraints in option pricing and develop a novel hybrid gated neural network (hGNN) based option valuation model. We adopt a multiplicative structure of hidden layers to ensure model differentiability. We also select the slope and weights of input layers to satisfy the no-arbitrage constraints. Meanwhile, a separate neural network is constructed for predicting option-implied volatilities. Using S&P 500 options, our empirical analyses show that the hGNN model substantially outperforms well-established alternative models in the out-of-sample forecasting and hedging exercises. The superior prediction performance stems from our model's ability in describing options on the boundary, and in offering analytical expressions for option Greeks which generate better hedging results.

JEL code: C63, F47

Keywords: Finance; Artificial neural networks; Implied volatilities; Option Greeks; Hedging.

[*] Management Science and Business Economics Group, Business School, University of Edinburgh, 29 Buccleuch Place, Edinburgh EH8 9JS, UK. Email: jason.caoyi@gmail.com.

[†] Corresponding author. University of Nottingham Business School China, University of Nottingham Ningbo, Ningbo 315100, P. R. China. Email: xiaoquan.liu@nottingham.edu.cn. Phone: +86 574 88180000 ext 8207.

[‡] Business School, Xian Jiaotong-Liverpool University, Suzhou, Jiangsu Province, China 215123. Email: jia.zhai@xjtlu.edu.cn

1 Introduction

Option valuation is a much devoted area in the financial economics literature. Since the seminal work of Black and Scholes (1973), many pricing models have been developed that extend restrictive assumptions of the Black and Scholes (BS) model and advance our understanding of options and the market in which they are traded. Well-known parametric models include the stochastic volatility model of Heston (1993), the jump diffusion models of Merton (1976), the stochastic volatility with jump model of Bates (1996) and Bakshi et al. (1997), and the double-jump model of Eraker et al. (2003). In parallel, nonparametric specifications such as the GARCH option pricing of Duan (1995), the spline based method of Bliss and Panigirtzoglou (2002), and quadratic approximation of Jackwerth and Rubinstein (1996) are formulated. From different perspectives, these models try to accommodate the stylized volatility smile with different levels of empirical success.

Option valuation is also a topical issue in the data science literature as sophisticated numerical methods make it possible to achieve great pricing and forecasting precision (see Dugas et al., 2001; Garcia and Gencay, 2000; Gradojevic et al., 2009; Liu et al., 2019, for example). As a popular data science method, the neural network is gaining popularity and increasingly being adopted in option valuation and hedging since the early effort of Hutchinson et al. (1994) (see Buehler et al., 2019; Culkin and Das, 2017; Dugas et al., 2009; Liang et al., 2009; Yao et al., 2000, for example). Ruf and Wang (2020) offer an excellent and updated literature review.

Motivated by the above strands of the literature, in this paper we propose an economically meaningful hybrid gated neural network (hGNN) based option valuation model. We start from both the neural network architecture and six necessary and sufficient no-arbitrage constraints in the option valuation theory. We select the *softplus* function as the activation function in hidden neurons and construct a multiplicative structure to maintain its differentiability. Meanwhile, the slope and weights of the linear function in input layers must satisfy the no-arbitrage constraints. We finalize the structure with a pre-processing module before the input layer. This pre-processing module also meets economic constraints and is added to enhance the input-output mapping capability. Furthermore, we build a separate neural network for modeling and predicting latent option-implied volatilities, an essential input in option pricing models. Hence, our model is called a hybrid gated neural network (hGNN) based model.

The model design contributes to the financial economics literature as well as the data science literature. Our first contribution is that by taking into account the essential no-arbitrage constraints, our model attaches economic meaning to the neural network architecture. Quite often, the data science methods start with specifically designed architectures, i.e., a number of neuron layers or infinite dimensional hyperplane, and consider option pricing as a nonlinear and complex regression problem. The mapping relationship between the input and the output is learned from a large amount of data. These models usually aim at building complex neural network algorithms to enhance the learning capability for improved mapping performance (Hutchinson et al., 1994; Gradojevic et al., 2009). Hence, they are often compared to a *black box*, indicating that the economic interpretability and internal functionality of these models are opaque to most users (Knight, 2017a,b; McNelis, 2005).

More recently, a burgeoning theme emerges in this area as studies implement an economic hint or select particular activation functions to satisfy economic constraints, thus offering economic interpretability (see Gilpin et al., 2018; Guidotti et al., 2018; Mudrakarta et al., 2018; Ribeiro et al., 2016; Yang et al., 2017, for example). Our paper builds upon and extends this strand of the literature as we try to introduce economic intuition into neural network models, a traditionally technical and data-driven approach. As an important aspect of this contribution, our proposed hGNN model allows us to obtain analytical expressions for European option Greeks, and we believe our paper is the first to do so. These analytical Greek letters would facilitate option traders to better design and implement trading and hedging strategies and underscore the practicality of our model.

We also contribute to the literature by offering comprehensive empirical evidence that our hGNN model outperforms popular neutral network based models as well as economically motivated models such as those featuring stochastic volatility and jumps in underlying asset returns. They include a deep neural network (dNN) model, the best-performing specification in Andreou et al. (2008, 2010) (AnNN), a stochastic volatility model, a stochastic volatility model with stochastic interest, and a stochastic volatility model with jumps. These three stochastic volatility based models come from the seminal work of Bakshi et al. (1997). Our sample includes more than 2 million S&P 500 options over 952 trading days from 22 May, 2014, to 2 March, 2018. We implement a rolling scheme to produce 7- and 30-day ahead option price forecasts, which are evaluated by two loss functions. Furthermore, we perform a delta hedging exercise based on 7- and 30-day re-balancing frequencies. Our empirical results show that the hGNN model performs better in generating more precise option price forecasts and smaller errors in the hedging exercise. This is the case regardless of forecasting horizons, put or call option type, option moneyness, time to maturity, or the loss function.

What drives this superior performance? To address this question, we form two groups of options. The first group includes the most mispriced options according to the dNN model, and the second group contains a randomly selected sample of the remaining options. We observe that options in Group 1, which the dNN model has a hard time predicting, tend to be short-dated ones with extreme moneyness. Our hGNN model is constructed based on no-arbitrage conditions, including option boundary conditions, and is trained by synthetic prices for thinly traded options, it thus substantially outperforms the other two neural network based models in predicting options in Group 1. This underscores the empirical prowess of our model.

The rest of the paper is structured as follows. In Section 2, we provide the methodological and empirical considerations that motivate our specific model design. Section 3 discusses the model structure, introduces no-arbitrage constraints in the option pricing literature, proves that our model satisfies these constraints, and constructs a separate neural network for modeling option-implied volatilities. In Section 4, we derive analytical expressions for European option Greeks based on the hGNN model, and outline a delta-hedging strategy. Section 5 conducts empirical analyses and robustness tests using S&P 500 options. Finally, Section 6 concludes.

2 Design motivation

The architecture and algorithm of our hGNN model are motivated by the following considerations.

1. Empirically, Gradojevic et al. (2009) show that neural network based models usually perform poorly for deep OTM and very long- and/or short-dated options. The paper addresses this issue by grouping options according to their moneyness and maturity, and constructing separate models for each group. Although this leads to improved empirical performance, the grouping is static and cannot adapt to changing market conditions. Furthermore, the paper does not cover very deep out-of-money call options and the algorithm is computationally cumbersome. More recently, Yang et al. (2017) design an architecture of neural networks with selected no-arbitrage theories for European call options but fail to consider European put options or important boundary conditions. This leaves a gap in the literature.

2. As a key input for option valuation, the volatility is a latent variable that needs to be proxied. In an unreported empirical examination, we follow the hybrid neural network model in Andreou et al. (2008) and find that using the BS implied volatilities instead of realized volatilities

as the volatility proxy significantly improves option pricing accuracy. This motivates us to build a separate structure for predicting option-implied volatilities.

3. To the best of our knowledge, analytic expressions for European option Greeks, i.e., the partial derivatives of option prices with respect to underlying asset price, volatility, strike price, and time to maturity, have not been derived for neural network based models. Hence the *black box criticism* towards this type of models (Knight, 2017a,b).

4. Most importantly, the economic interpretation of option pricing models represents the biggest gap between theory motivated models in the finance literature and data science based models. The latter may fit the data very well due to their data-driven and data-intensive nature but falls short of offering economic intuition. Our paper represents a step towards bridging this gap.

3 Model construction

In this section, we first briefly outline the non-arbitrage constraints for European options. We then develop our hGNN model and show that it fully satisfies the constraints. Finally, a separate neural network is constructed for modeling and predicting implied volatilities.

3.1 No-arbitrage constraints

In the financial economics theory, there exists a risk-neutral probability measure Q under which the discounted asset price is a martingale (Delbaen and Schachermayer, 1994; Cochrane, 2001). This allows us to write the call option price as follows:

$$C(K, S_t, \tau, \sigma_t, r) = e^{-r\tau} \int_K^\infty (S_T - K)^+ f(S_T|S_t, \tau) dQ_\tau(S_T). \tag{1}$$

Likewise for put options we have the following:

$$P(K, S_t, \tau, \sigma_t, r) = e^{-r\tau} \int_0^K (K - S_T)^+ f(S_T|S_t, \tau) dQ_\tau(S_T), \tag{2}$$

where K is the strike price, S_t is the underlying asset price at time t, T is the expiry date, $\tau = T - t$ is time to maturity, σ_t is the volatility at time t, r_t is the risk-free interest rate, and C and P are the call and put option prices, respectively.

We follow Theorem 2.1 in Roper (2010) and the call option surface constraints in Fengler and Hin (2015) to estimate a continuous function of option prices. Thus, option prices C and P and the variables $(K, S_t, \tau, \sigma_t, r)$ observed at time t are subject to the following constraints:

(c1) *Convexity in K*

Both C and P are convex across K for $\tau \geq 0$. C is monotonically non-increasing with K, whereas P is monotonically non-decreasing with K. Hence, $\frac{\partial C}{\partial K} \leq 0$ and $\frac{\partial P}{\partial K} \geq 0$.

(c2) *Monotonicity in τ*

Both C and P are non-decreasing with $K > 0$. Hence, $\frac{\partial C}{\partial \tau} \geq 0$ and $\frac{\partial P}{\partial \tau} \geq 0$.

(c3) *Strike limit*

When $\tau > 0$, $\lim_{K\to\infty} C = 0$ for call options, and $\lim_{K\to 0} P = 0$ for put options.

(c4) *Boundary conditions*

$(S_t - K)^+ \leq C \leq S_t$ for call options, and $(K - S_t)^+ \leq P \leq Ke^{-r\tau}$ for put options.

(c5) *Expiry value*

When $\tau = 0$, $C = (S_t - K)^+$ and $P = (K - S_t)^+$.

(c6) Constraints (c1), (c3), and (c4) imply that option prices are twice differentiable with respect to K for all $\tau > 0$. Hence, $\frac{\partial^2 C}{\partial K^2} \geq 0$ and $\frac{\partial^2 P}{\partial K^2} \geq 0$.

The proof for (c1)-(c6) follows Theorem 2.1 in Ait-Sahalia and Duarte (2003).

3.2 Model design under constraints

Existing neural network based option valuation models usually adopt the traditional three-layer architecture: an input layer with N input variables, a hidden layer with H neurons, and an output layer with a single neuron. Each hidden neuron includes a certain type of activation function: either a *sigmoid* function as in Gradojevic et al. (2009), or a *hyperbolic tangent* function as in Andreou et al. (2008, 2010). These models achieve nice empirical performance by utilizing the input-output mapping capability of a neural network, or a stack of neural networks, but pay little attention to the no-arbitrage constraints central to option valuation. Therefore, they perform poorly for options with extreme moneyness and long-/short-dated maturity.

Our proposed model improves upon these. It also has a three-layered structure but before the input layer, we add a division module with three inputs (K, S_t, Ψ) and one output of option moneyness:

$$m = (K/S_t)^{\Psi}, \tag{3}$$

where Ψ is a call/put indicator: it is 1 for call options and -1 for put options. This design accommodates both call and put options with $m = K/S_t$ and $m = S_t/K$, respectively. The input layer has $N_i = 4$ input variables (m, τ, σ_t, r), which are called *features* in the data science literature. Contrary to Andreou et al. (2010) and Das and Padhy (2017), we use these variables as the input of the Black-Scholes model for two reasons. First, our model aims to solve the canonical option valuation problem and our approach is the same as those in the traditional finance literature. Second, using these variables as the input makes it possible for our proposed model to provide analytical expressions for the Greeks.

The hidden layer of our model consists N_h neurons, and each neuron contains a *gated network architecture* activation function following the architecture in Memisevic (2013) and Sigaud et al. (2015). This gated structure is an extension of the deep learning building block in Bengio (2013) and LeCun et al. (2015). It is particularly well-suited for multiplicative interactions between the input and output as in our case, and selected to maintain the model's first- and second-order differentiability so that it satisfies no-arbitrage constraints (c1)-(c6). Finally, the output layer contains an additive linear function with one output variable called the *target*.

Based on the above considerations, our proposed GNN option valuation model $y(m, \tau, \sigma_t, r)$, illustrated in Figure 1, is expressed as follows:

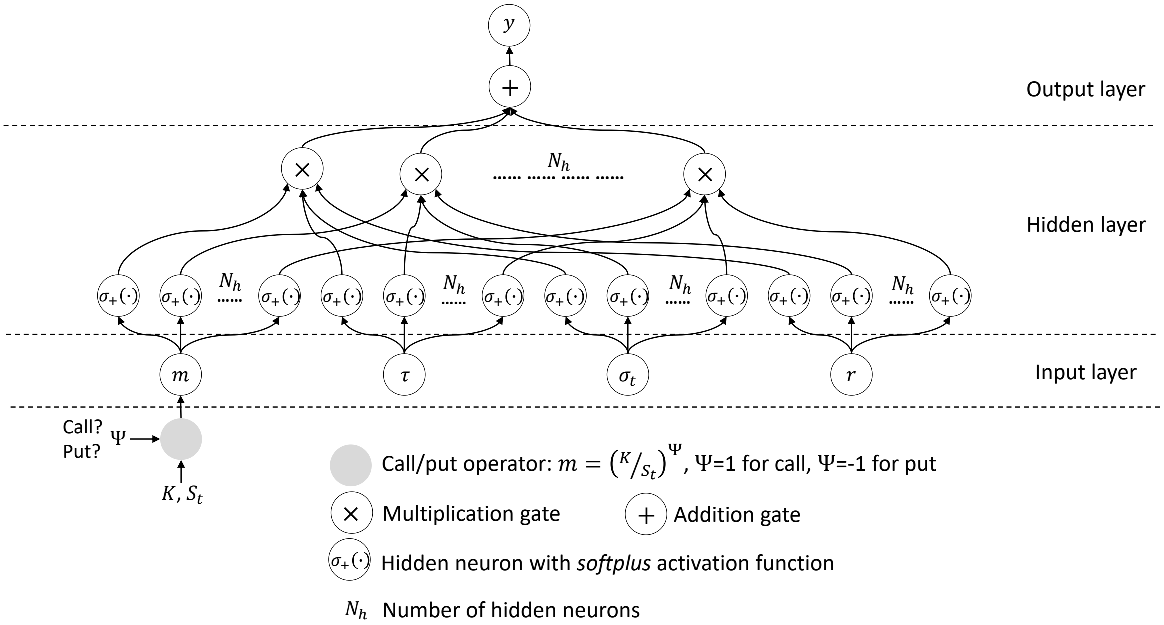
$$y(m, \tau, \sigma_t, r) = \sum_{j=1}^{N_h} \left[\sigma_+(b_j^m - m e^{w_j^m})\right] \left[\sigma_+(b_j^\tau + \tau e^{w_j^\tau})\right] \left[\sigma_+(b_j^r \pm r e^{w_j^r})\right] \left[\sigma_+(b_j^{\sigma_t} + \sigma_t e^{w_j^{\sigma_t}})\right], \tag{4}$$

where $\sigma_+()$ is the *softplus* function $\sigma_+(x) = \log(1 + e^x)$. The weights ($w_j^m$, w_j^τ, w_j^r, $w_j^{\sigma_t}$) and biases (b_j^m, b_j^τ, b_j^r, $b_j^{\sigma_t}$) are parameters to be estimated. The $+$ and $-$ in $(b_j^r \pm r e^{w_j^r})$ are for call and put options, respectively. The sign in each $\sigma_+()$ function is designed according to specific constraints.

In the architecture of Figure 1, the model has four input variables: moneyness m, time to maturity τ, volatility σ_t, and interest rate r. Each input variable is directly connected to N_h numbers of *softplus* activation functions, and their outputs are directly connected to N_h multiplication gates. Thus, each multiplication gate has inputs from four *softplus* activation functions corresponding to four input variables. The outputs of N_h multiplication gates are aggregated into an addition gate for generating the final output. Hence, the output $y(m, \tau, \sigma_t, r)$ can be expressed in Eq.(4) as the

Figure 1. The architecture of the gated neural network (GNN) model

This figure shows the architecture of the gated neural network. The variables (m, τ, σ_t, r) are the inputs of the hidden layer, $\sigma_+(.)$ is the *softplus* function, and $\times$ is the multiplication gate.



y
+
Output layer
×
N_h
Hidden layer
$\sigma_+(\cdot)$
N_h
Input layer
m
τ
σ_t
r
Call?
Put?
Ψ
K, S_t
Call/put operator: $m = (K/S_t)^\Psi$, Ψ=1 for call, Ψ=-1 for put
Multiplication gate
Addition gate
Hidden neuron with *softplus* activation function
N_h Number of hidden neurons

cumulative form of four outputs from the activation function based on the inputs.

Our proposed model exhibits two clear advantages compared with existing models in the literature. First and very importantly, from the option pricing perspective, our model integrates the no-arbitrage constraints as the prior to support the logic of the option valuation. Hence, the model goes beyond a large scaled connection of the neurons and is able to reflect option pricing theories. From the data science perspective, incorporating external knowledge facilitates the model's learning of input features and enhances the overall performance (Hu et al., 2016; Kursuncu et al., 2019).[1]

Proposition 1. The architecture of the activation function in equation (4) satisfies constraints (c1), (c2), (c3), and (c6).

Proof: Constraint (c1)

The derivative of a *softplus* function $\sigma_+(x)$ can be obtained as follows:

$$\frac{d\log(1+e^x)}{dx} = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}. \tag{5}$$

The function $\frac{1}{1+e^{-x}}$ is called the *sigmoid*, which can also be used as an activation function. We represent it as $\sigma_s = \frac{1}{1+e^{-x}}$ thus $\sigma'_+(x) = \sigma_s(x)$. In this way, constraint (c1) can be written as follows:

$$\frac{\partial y}{\partial m} = \sum_{j=1}^{N_h} \left[-e^{w_j^m}\sigma_s(b_j^m - me^{w_j^m})\right]\left[\sigma_+(b_j^\tau - \tau e^{w_j^\tau})\right]\left[\sigma_+(b_j^r \pm re^{w_j^r})\right]\left[\sigma_+(b_j^{\sigma_t} + \sigma_t e^{w_j^{\sigma_t}})\right]. \tag{6}$$

Hence, $\frac{\partial y}{\partial m} \leq 0$. Consider the definition of moneyness, we have the following for call options:

$$\frac{\partial y}{\partial K} = \frac{\partial y}{\partial m}\frac{\partial m}{\partial K} = \frac{\partial y}{\partial m}\frac{1}{S_t} \leq 0. \tag{7}$$

Likewise for put options:

$$\frac{\partial y}{\partial K} = \frac{\partial y}{\partial m}\frac{\partial m}{\partial K} = -\frac{\partial y}{\partial m}\frac{S_t}{K^2} \geq 0. \tag{8}$$

Proof: Constraint (c2)

[1] It is important to note that, unlikely the traditional *divide and conquer* approach such as that adopted in Gradojevic et al. (2009) in which options are categorized into groups and different neural networks are trained separately for each group, we use all the training data *simultaneously* on our model. We only group options according to moneyness and maturity when summarizing the results.

Similarly, we can express the constraint (c2) for call and put options as follows:

$$\frac{\partial y}{\partial \tau} = \sum_{j=1}^{N_h} \left[\sigma_+(b_j^m - me^{w_j^m})\right] \left[e^{w_j^\tau}\sigma_s(b_j^\tau + \tau e^{w_j^\tau})\right] \left[\sigma_+(b_j^r \pm re^{w_j^r})\right] \left[\sigma_+(b_j^{\sigma_t} + \sigma_t e^{w_j^{\sigma_t}})\right] \geq 0. \tag{9}$$

Proof: Constraint (c3)

We have $\lim_{K\to\infty} m = \lim_{K\to\infty} \frac{K}{S_t} = \infty$ for call options, and $\lim_{K\to 0} m = \lim_{K\to 0} \frac{S_t}{K} = \infty$ for put options. Furthermore,

$$\lim_{m\to\infty} \sigma_+(b_j^m - me^{w_j^m}) = \lim_{m\to\infty} \log\left(1 + e^{(b_j^m - me^{w_j^m})}\right) = \log\left(1 + e^{-\infty}\right) = 0. \tag{10}$$

Thus, $\lim_{K\to\infty} C = 0$ for calls and $\lim_{K\to 0} P = 0$ for puts.

Proof: Constraint (c6)

Based on the proof for constraint (c1), we have the following:

$$\frac{\partial^2 y}{\partial m^2} = \sum_{j=1}^{N_h} \left[e^{2w_j^m}\sigma_s'(b_j^m - me^{w_j^m})\right] \left[\sigma_+(b_j^\tau - \tau e^{w_j^\tau})\right] \left[\sigma_+(b_j^r \pm re^{w_j^r})\right] \left[\sigma_+(b_j^{\sigma_t} + \sigma_t e^{w_j^{\sigma_t}})\right], \tag{11}$$

where $\sigma_s'(x) = \sigma_s(x)(1 - \sigma_s(x)) \geq 0$, thus $\frac{\partial^2 y}{\partial m^2} \geq 0$ and we have the following:

$$\frac{\partial^2 y}{\partial K^2} = \frac{\partial^2 y}{\partial m \partial K} = \frac{\partial^2 y}{\partial m^2}\left(\frac{\partial m}{\partial K}\right)^2 + \frac{\partial y}{\partial m}\frac{\partial^2 m}{\partial K^2}. \tag{12}$$

For call options, $\frac{\partial^2 m}{\partial K^2} = 0$, thus:

$$\frac{\partial^2 y}{\partial K^2} = \frac{\partial^2 y}{\partial m^2}\frac{1}{S_t^2} \geq 0. \tag{13}$$

Likewise for put options:

$$\frac{\partial^2 y}{\partial K^2} = \frac{\partial^2 y}{\partial m^2}\frac{S_t^2}{K^4} + \frac{\partial y}{\partial m}\frac{2S_t}{K^3}. \tag{14}$$

Since dividing a positive constant on both sides of an equation does not change the sign of the equation, we divide $\frac{S_t}{K^3}$ on both side of equation (14) and let $\mathcal{F}(y, m) = m\frac{\partial^2 y}{\partial m^2} + 2\frac{\partial y}{\partial m}$ for $K > 0$ and $S_t > 0$. To determine the value of $\mathcal{F}(y, m)$, we approximate it by the second-order Taylor expansion and obtain the following:

$$\mathcal{F}(y, m) \quad = \quad 2y' + my'' \approx f'(a)(x - a) + \frac{f''(a)}{2!}(x - a)^2$$

$$= \quad y'(a)2 + y''(a)m \approx f'(a)(m-a) + f''(a)\frac{(m-a)^2}{2}. \tag{15}$$

Finally, to approximate the value of $\mathcal{F}(y, m)$, we solve two equations $m - a = 2$ and $m = \frac{(m-a)^2}{2}$ and obtain $a = 0$ and $m = 2$. Therefore, $\mathcal{F}(y, m) \approx y(2) \geq 0$. This completes the proof of constraints (c1), (c2), (c3) and (c6).

The output layer contains one neuron for the final estimated option price $\hat{C}$ and $\hat{P}$ as follows:

$$y(m, \tau, \sigma_t, r) = \begin{cases} \hat{C}, & \text{if } \Psi = 1 \\ \hat{P}, & \text{if } \Psi = -1 \end{cases} \tag{16}$$

where $y(m, \tau, \sigma_t, r)$ is the output value of the GNN.

To summarize, our GNN is able to map four features, i.e., the moneyness m, maturity time τ, volatility σ_t, and the interest rate r, to the target option price $y(m, \tau, \sigma_t, r)$, while ensuring that important no-arbitrage constraints are satisfied.

3.3 Boundary conditions

Usually the output bound of regression-type applications of a neural network is achieved by scaling or normalizing the target variables rather than modifying the model structure, as the structure is determined by data. The downside of this data-driven approach is manifestoed in the poor pricing performance for deep OTM and extreme short- or long-dated options (Andreou et al., 2008, 2010; Gradojevic et al., 2009). This underlines the importance of satisfying the option boundary constraints (c4) and (c5).

The options on the boundary are those with strike prices approaching zero, S_t, or infinite, and those very close to maturity. These options are thinly traded in the market, and this lack of data undermines data science models. To cope with this, we synthesize prices for these options based on available market data and use synthesized prices as *hints* for our model. This useful approach is first developed in Abu-Mostafa (1993, 1994) and becomes a popular approach to compensating an imbalanced dataset in the literature (Barua et al., 2014; Chawla et al., 2002; Galar et al., 2012; Khoshgoftaar et al., 2011). Abu-Mostafa (1995), Cao et al. (2015), Cao et al. (2016), and Garcia and Gencay (2000), in particular, implement the method with financial data.

Prices for options with $K = 0$ and $\tau = 0$ are straightforward to calculate via constraints (c4) and (c5): $C = S_t$ and $P = 0$ when $K = 0$, and $C = (S_t - K)^+$ and $P = (K - S_t)^+$ when $\tau = 0$.

For every τ, we synthesize virtual option prices with $K = 0$ and for every K, we synthesize virtual option prices with $\tau = 0$. For ATM options with $K = S_t$, we synthesize virtual calls that are slightly ITM, and obtain put prices via the put-call parity. This follows Song and Xiu (2016), which compensate low trading volume of ITM calls by obtaining call prices from OTM puts.

Options with $K = \infty$ do not exist. We synthesize prices for these options using the Black-Scholes model with an almost zero call option price. We take advantage of the precision of modern computers and take 2^{-126}, the number closest to zero under the single-precision floating-point format (IS Committee, 2008), as the option price for iteratively calculating the strike price. For example, to synthesize a call option with $\tau = 15$, we obtain strike price K by solving $2^{-126} = BS\,(K, S_t = 4973.07, \tau = 15, r = 0.02, \sigma_t = 0.296)$, where the implied volatility $\sigma_t = 0.296$ corresponds to strike price $K = 5700$. Due to the convexity constraint, we solve this by the traditional Newton Raphson method, and obtain $K = 11087.52$ and $C = 2.003 \times 10^{-39}$. Likewise, for each τ, we synthesize a call option with almost zero price and infinite strike price. All alternative models in this paper are trained with the same dataset that includes market data and the prices for these synthesized options on the boundary.

To train our hGNN model, we use a powerful stochastic gradient descent optimization algorithm in Kingma and Ba (2017) with the mean absolute percentage error (MAPE) as the loss function. This *R* package *GradDescent* is obtained from Wijaya et al. (2018). We follow the empirical study of Zhou et al. (2016) and set the optimal values of number of hidden neurons $N_h = 100$, the number of epochs to be 100, batch size as 100, and learning rate as 10^{-3} (see also Cheng et al., 2020; Yu et al., 2019).

3.4 Modeling implied volatilities

Volatility is an essential input to option pricing models. In the literature, modeling option-implied volatilities follows two main directions. For deterministic volatility functions (DVF), implied volatilities are usually expressed as a function of option moneyness, time to maturity, and lagged implied volatilities (Andreou et al., 2010, 2014; Chalamandaris and Tsekrekos, 2014). Meanwhile, Dunis et al. (2013) and Konstantinidi et al. (2008) incorporate economic variables, such as the yield curve slope, interest rate, and stock index returns, in a vector autoregression (VAR) to obtain volatility forecasts. In these studies, the determinants m, τ and $IV_{t-l}^{m,\tau}$ are either regressed linearly by AR or VAR models (Dunis et al., 2013), or *nonlinearly* as m^2, τ^2, or $m\tau$ enter the DVF. The implicit assumption is that implied volatilities are well described by these determinants via a pre-defined relation.

We extend these studies in two ways. First, data-wise, we implement the principal component analysis (PCA) to extract the most relevant information from the determinants while reducing the computational complexity; second, methodologically, we adopt a deep neural network for a data-defined nonlinear model with two modules. We assume that implied volatilities $IV_t^{m,\tau}$ are determined by moneyness m, time to maturity τ, and lagged volatilities $IV_{t-l}^{m,\tau}$, $l \in 1, ..., L$, where $L = 10$ is considered the optimal VAR lag in Dunis et al. (2013). Hence, we construct a vector of 12 dimensions $\boldsymbol{X}_t^{m_1,\tau_1} = \left[m_1, \tau_1, IV_{t-1}^{m_1,\tau_1}, \ldots, IV_{t-10}^{m_1,\tau_1}\right]^T$ for implied volatilities $\boldsymbol{Y}_t^{m_1,\tau_1} = IV_t^{m_1,\tau_1}$ of m_1 and τ_1 on day t. We collect implied volatilities $\boldsymbol{Y}_t^{m_i,\tau_j}$ of all available m_i and τ_j ($i = 1, \ldots, I$ and $j = 1, \ldots, J$) on day t to construct a matrix of input data for day t, and the final data consist N such matrices from day t to $t - N + 1$.

Our implied volatility model includes two modules. For the first module, we implement the PCA to extract five most important components, PC^p, $p = 1, \ldots, 5$, that are able to explain most of the variation in volatility (Kolanovic and Krishnamachari, 2017). The corresponding principal component scores PS_p, $p = 1, \ldots, 5$ are used to predict implied volatilities. The second module is a multi-layer feed-forward artificial neural network, also termed the deep neural network (Fischer and Krauss, 2018; Krauss et al., 2017). It has one input layer with five neurons corresponding to five principal components, two hidden layers with 30 and 10 neurons, respectively, and an output layer with a linear transfer function. Each neuron in the hidden layers contains a hyperbolic tangent sigmoid activation function. To control overfitting, we perform regularization by an input-dropout ratio of 0.2 and a hidden dropout ratio of 0.5 (Hinton et al., 2012; H2O, 2018). We train this model with 200 epochs, which indicate the number of passes to carry out over the training dataset.

To forecast one-day ahead implied volatilities on day t+1 from m_i and τ_j, we construct the $[1 \times 12]$ input vector $\boldsymbol{X}_{t+1}^{m_i,\tau_j} = \left[m_i, \tau_j, IV_t^{m_i,\tau_j}, \ldots, IV_{t-9}^{m_i,\tau_j}\right]^T$, and multiply it with the $[12 \times 5]$ principal component coefficient matrix to obtain the pricinpal component scores $\mathrm{PS}_{t+1} = \boldsymbol{X}_{t+1}^{m_i,\tau_j} \times C_{coef}$. The PS_{t+1} are subsequently fed into the model to forecast implied volatilities $\boldsymbol{Y}_{t+1}^{m_i,\tau_j} = IV_{t+1}^{m_i,\tau_j}$. To predict N-day ahead implied volatilities, a daily rolling scheme is performed. We first predict implied volatilities on day $t+2$ with data from $t-8$ to $t+1$, whereby volatilities on day $t+1$ are forecasted; we then predict implied volatilities on day $t+3$ from data from $t-7$ to $t+2$, whereby volatilities on day $t+1$ and day $t+2$ are forecasted; and so forth. If $N > 10$, the forecasts are based entirely on forecasted implied volatilities. Although this rolling scheme potentially deteriorates the precision of long-term volatility forecasts, our empirical results exhibit strong performance.

To summarize, we first obtain option-implied volatility $IV_t^{m,\tau}$ for day t from moneyness m, option time to maturity τ, the interest rate r, and implied volatilities $IV_{t-n}^{m,\tau}$, $n = 1, ..., 10$ from the previous 10 days. Once we know $IV_t^{m,\tau}$, it is used together with m, τ, r to forecast option prices on day t.

4 Option Greeks

A main motivation for this paper is to develop the hGNN model in such a way that it offers analytical expressions for European option Greeks. This allows the model to lend itself readily in hedging strategies, and represents a key contribution of our paper. Given the structure of our model in equation (4), we first derive the option Greeks in this section. We then outline how they are used in hedging exercises.

4.1 Analytical option Greeks

Option Δ

European call option Δ is expressed as follows:

$$\begin{aligned}
\Delta_{\hat{C}} &= \frac{\partial \hat{C}}{\partial S} \\
&= \frac{\partial \hat{C}}{\partial y}\frac{\partial y}{\partial m}\frac{\partial m}{\partial S_t} \\
&= \sum_{j=1}^{N_h}\left[-e^{w_j^m}\sigma_s(b_j^m - me^{w_j^m})\right]\left[\sigma_+(b_j^\tau + \tau e^{w_j^\tau})\right]\left[\sigma_+(b_j^r + re^{w_j^r})\right]\left[\sigma_+(b_j^{\sigma_t} + \sigma_t e^{w_j^{\sigma_t}})\right]\left(-\frac{K}{S_t^2}\right) \\
&= \frac{m}{S_t}\sum_{j=1}^{N_h}\left[e^{w_j^m}\sigma_s(b_j^m - me^{w_j^m})\right]\left[\sigma_+(b_j^\tau + \tau e^{w_j^\tau})\right]\left[\sigma_+(b_j^r + re^{w_j^r})\right]\left[\sigma_+(b_j^{\sigma_t} + \sigma_t e^{w_j^{\sigma_t}})\right], \qquad (17)
\end{aligned}$$

where $m = \frac{K}{S_t}$, $\sigma_+(x) = \log(1 + e^x)$, and $\sigma_s = \frac{1}{1+e^{-x}}$. Likewise, the Δ for European put option can be written as follows:

$$\begin{aligned}
\Delta_{\hat{P}} &= \frac{\partial \hat{P}}{\partial S} \\
&= \frac{\partial \hat{P}}{\partial y}\frac{\partial y}{\partial m}\frac{\partial m}{\partial S_t} \\
&= \sum_{j=1}^{N_h}\left[-e^{w_j^m}\sigma_s(b_j^m - me^{w_j^m})\right]\left[\sigma_+(b_j^\tau + \tau e^{w_j^\tau})\right]\left[\sigma_+(b_j^r - re^{w_j^r})\right]\left[\sigma_+(b_j^{\sigma_t} + \sigma_t e^{w_j^{\sigma_t}})\right]\frac{1}{K}
\end{aligned}$$

$$= -\frac{1}{K}\sum_{j=1}^{N_h}\left[e^{w_j^m}\sigma_s(b_j^m - me^{w_j^m})\right]\left[\sigma_+(b_j^\tau + \tau e^{w_j^\tau})\right]\left[\sigma_+(b_j^r - re^{w_j^r})\right]\left[\sigma_+(b_j^{\sigma_t} + \sigma_t e^{w_j^{\sigma_t}})\right] \tag{18}$$

Option ν

European option $\nu_{\hat{C}}$ is given as follows:

$$\begin{aligned}
\nu_{\hat{C}} &= \frac{\partial \hat{C}}{\partial \sigma_t} \\
&= \frac{\partial \hat{C}}{\partial y}\frac{\partial y}{\partial \sigma} = \frac{\partial \hat{P}}{\partial y} \\
&= e^{w_j^{\sigma_t}}\sum_{j=1}^{N_h}\left[\sigma_+(b_j^m - me^{w_j^m})\right]\left[\sigma_+(b_j^\tau + \tau e^{w_j^\tau})\right]\left[\sigma_+(b_j^r + re^{w_j^r})\right]\left[\sigma_s(b_j^{\sigma_t} + \sigma_t e^{w_j^{\sigma_t}})\right] \qquad (19)
\end{aligned}$$

where $\sigma_s = \frac{1}{1+e^{-x}}$.

Option Θ

The option $\Theta_{\hat{C}}$ can be expressed as follows:

$$\begin{aligned}
\Theta_{\hat{C}} &= \frac{\partial \hat{C}}{\partial \tau} \\
&= \frac{\partial \hat{C}}{\partial y}\frac{\partial y}{\partial \tau} \\
&= e^{w_j^\tau}\sum_{j=1}^{N_h}\left[\sigma_+(b_j^m - me^{w_j^m})\right]\left[\sigma_s(b_j^\tau + \tau e^{w_j^\tau})\right]\left[\sigma_+(b_j^r \pm re^{w_j^r})\right]\left[\sigma_+(b_j^{\sigma_t} + \sigma_t e^{w_j^{\sigma_t}})\right]. \qquad (20)
\end{aligned}$$

Option ρ

The $\rho_{\hat{C}}$ of European call option can be written as follows:

$$\begin{aligned}
\rho_{\hat{C}} &= \frac{\partial \hat{C}}{\partial r} \\
&= \frac{\partial \hat{C}}{\partial y}\frac{\partial y}{\partial r} \\
&= e^{w_j^r}\sum_{j=1}^{N_h}\left[\sigma_+(b_j^m - me^{w_j^m})\right]\left[\sigma_+(b_j^\tau + \tau e^{w_j^\tau})\right]\left[\sigma_s(b_j^r + re^{w_j^r})\right]\left[\sigma_+(b_j^{\sigma_t} + \sigma_t e^{w_j^{\sigma_t}})\right]. \qquad (21)
\end{aligned}$$

Likewise for put options:

$$\rho_{\hat{P}} = \frac{\partial \hat{P}}{\partial r}$$

$$
\begin{aligned}
&= \frac{\partial \hat{P}}{\partial y}\frac{\partial y}{\partial r} \\
&= -e^{w_j^r}\sum_{j=1}^{N_h}\left[\sigma_+(b_j^m - me^{w_j^m})\right]\left[\sigma_+(b_j^\tau + \tau e^{w_j^\tau})\right]\left[\sigma_s(b_j^r - re^{w_j^r})\right]\left[\sigma_+(b_j^{\sigma_t} + \sigma_t e^{w_j^{\sigma_t}})\right]. \quad (22)
\end{aligned}
$$

4.2 Hedging exercises

We are interested in the hedging performance of our hGNN model as options are an essential risk management tool for investors. We implement a conventional delta-neutral hedge following Bakshi et al. (1997): For the SV model, we hedge both the price and volatility risks with positions in the underlying asset and in a second option contract; for the SVSI model, we involve a bond for hedging the interest rate risk in addition to the hedging strategy of the SV model; for the SVJ model, due to the difficulty associated with stochastic jump sizes (Bates, 1996; Merton, 1976), we implement a partial hedge for which only the diffusion risks are neutralized but the jump risk is unhedged. For the hGNN model, we hedge the risks in the underlying price, volatility, and interest rate with positions in the underlying asset, a second option contract, and a bond.

Suppose we sell one call option with time to maturity τ and strike price K. We need to hedge the price risk with a position in $V_{S,t}$ shares of the underlying asset, the interest rate risk with a position in $V_{B,t}$ units of τ-period discount bond, and the volatility risk with a position in $V_{C,t}$ units of a second call option with the same maturity τ but different strike price $\bar{K}$. The overall portfolio value at time t can be expressed as $V_{0,t} + V_{S,t}S_t + V_{B,t}B_{t,\tau} + V_{C,t}C_{t,\tau,\bar{K}}$, where $V_{0,t}$ represents the initial cash position. The derivation of $V_{S,t}$, $V_{B,t}$, and $V_{C,t}$ is outlined in Bakshi et al. (1997) as follows:

$$
\begin{aligned}
V_{C,t} &= \frac{\Delta_{V,t,\tau,K}}{\Delta_{V,t,\tau,\bar{K}}}, && (23) \\
V_{S,t} &= \Delta_{S,t;\tau,K} - \Delta_{S,t,\tau;\bar{K}}V_{C,t}, && (24) \\
V_{B,t} &= \frac{1}{B_{t,\tau}\varrho_\tau}\left(\Delta_{R,t,\tau;\bar{K}}V_{C,t} - \Delta_{R,t,\tau;K}\right), && (25) \\
V_{0,t} &= C_{t,\tau;K} - V_{S,t}S_t - V_{C,t,\tau;\bar{K}} - V_{B,t}B_{t,\tau}. && (26)
\end{aligned}
$$

For the hGNN model, Δ_S, Δ_V, and Δ_R are equivalent to the Greek letters $\Delta_{\hat{C}}$, $\nu_{\hat{C}}$, and $\rho_{\hat{C}}$ specified in equations (17), (19), and (21), respectively.

The hedged portfolio thus constructed is updated at each time interval Δt when it is re-balanced.

Hence, the hedging error can be written as follows:

$$\epsilon_{t+\Delta t} = V_{0,t}e^{r\Delta t} + V_{S,t}S_{t+\Delta t} + V_{B,t}B_{t+\Delta t,\tau-\Delta t} + V_{C,t}C_{t+\Delta t,\tau-\Delta t;\bar{K}} - C_{t+\Delta t,\tau-\Delta t;K}. \tag{27}$$

In the empirical analysis, we choose Δt=7 and 30 days following Bakshi and Madan (2000) and summarize the average hedging error. We train the model with the first 60% of data and use the remaining 40% to evaluate the hedging performance without re-training the model.

5 Data and empirical analysis

In this section, we first describe the options data used for the empirical analysis. We then compare the performance of the hGNN model with that of well-established models in the literature in terms of forecasting option prices and hedging.

5.1 Data

We use options and futures written on the S&P 500 index. As both contracts share the same underlying asset and maturity dates, index options can be considered as if they are written on the futures thus dividends are not considered. The underlying index level, strike prices, time to maturity, settlement prices, and trading volume are obtained from the data vendor IVolatility.com and contain 2,061,308 European options over 952 trading days from 22 May, 2014, to 2 March, 2018. Similar information is obtained for S&P index futures over the same sample period. Table 1 summarizes our dataset. We observe that on average put options are more frequently traded with larger volume and open interest than calls. Options have around two months to maturity on average. We also note that there are a reasonable number of different strikes per day for both calls (124) and puts (177).

Table 1. Summary statistics for S&P 500 options

This table summarizes the S&P 500 options that we use in the empirical analysis, including the number of trading days, the mean, maximum and minimum values of the trading volume, open interest, time to maturity (in days), number of different strikes per day, and moneyness. VDOTM, DOTM, and DITM stand for very deep OTM, deep OTM and deep ITM options. The sample period is from 22/5/2014 to 02/03/2018.

		Call		Put	
Trading days		952		952	
Trading volume	Mean	14		26	
	Max	49376		62384	
	Min	0		0	
Open Interest	Mean	76		160	
	Max	41527		59190	
	Min	0		0	
Time to maturity	Mean	60		60	
	Max	183		183	
	Min	0		0	
Strikes per day	Mean	124		177	
	Max	337		357	
	Min	10		26	
Moneyness (K/F)	Mean	1.0101		0.8653	
	Max	1.3304		1.5682	
	Min	0.0352		0.3285	
	DITM	<=0.80	21357		
	ITM	(0.8, 0.99]	242123	>1.01	114829
	ATM	(0.99, 1.01]	94256	(0.99, 1.01]	92617
	OTM	(1.01, 1.2]	397712	(0.8, 0.99]	729690
	DOTM	>1.2	5747	(0.6, 0.8]	309316
	VDOTM			<=0.6	53661
Total obs.			761195		1300113

We categorize all options into one of five moneyness groups, as the literature shows that the neural network based option valuation models are better at pricing some moneyness groups than others. In terms of option maturity, short-, medium- and long-term options have fewer than 90 days, between 90 to 180 days, and more than 180 days to maturity, respectively. We evaluate forecasting performance with two loss functions: the mean absolute percentage errors (MAPE) and the root mean square errors (RMSE). The statistical significance of forecasting error differences is gauged via the popular pairwise comparison developed in Diebold and Mariano (1995).

We compare the out-of-sample forecasting performance of the proposed hGNN model with two neural network based models: the best-performing deep neural network (dNN) model in Krauss et al. (2017), and the best-performing configuration in Andreou et al. (2008) (AnNN). They are implemented by the *H2O*, an open-source data science platform (Phan et al., 2017) in the *R* package *h2o* (LeDell, 2018). We also include three popular traditional option pricing models specified in Bakshi et al. (1997): the stochastic volatility (SV) model, the stochastic volatility and stochastic interest rate (SVSI) model, and the stochastic volatility with random jumps (SVJ) model. We follow exactly the specifications in (Bakshi et al., 1997) for these three models.

The models are used to generate out-of-sample option prices from 19 March, 2015, to 2 March, 2018, in a rolling scheme. We set the training and validation window to be 300 days and the trained models are tested in the following 7 or 30 days. Afterwards the training and validation window rolls forward for 7 or 30 days and used for the second testing. This goes on until the end of the sample period.

5.2 Empirical results

Implied volatility prediction

Table 2 summarizes the MAPE and RMSE of forecasted implied volatilities over 7- and 30-day horizons for call (Panel A) and put (Panel B) options. We find that overall our model performs well in generating option-implied volatility forecasts with a lot of precision. For example, regardless of the option type or the prediction horizon, the worst MAPE is only 1.69% for 30-day long-term call options, while the best is 0.91% for 7-day long-term calls. We note that, not surprisingly, the 7-day ahead forecasts tend to be more accurate than 30-day ones. Furthermore, volatility forecasts for calls are slightly better on average than those for puts across loss functions, and forecasts for deep ITM options are better than those for deep OTM options across both forecast horizons and loss functions.

Table 2. Forecasting performance of S&P implied volatilities

This table summarizes the forecasting performance of implied volatilities for S&P 500 options measured by the mean absolute percentage error (MAPE) and root mean squared error (RMSE). The first training window is 300 days from 22/05/2014 to 18/03/2015, and the first testing window is from 19/03/2015 to 26/03/2015 for the 7-day window, and from 19/03/2015 to 20/04/2015 for the 30-day rolling window. The training and testing windows are rolled 7 or 30 days forward until 02/03/2018. Short-, medium- and long-term options have less than 90 days, between 90 to 180 days, and more than 180 days to maturity, respectively. See also notes to Table 1 for option moneyness categories.

		Short-term options		Medium-term options		Long-term options	
		MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
Panel A. Call options							
7-day ahead	DITM	0.9853	0.0797	1.1904	0.0573	0.9061	0.0754
	ITM	1.1105	0.0681	1.0660	0.0634	1.1045	0.0651
	ATM	1.1989	0.0688	1.1529	0.0616	1.1976	0.0627
	OTM	1.2276	0.0688	1.2182	0.0605	1.2370	0.0623
	DOTM	1.2405	0.0795	1.2331	0.0669	1.2856	0.0630
30-day ahead	DITM	1.5380	0.0791	1.6023	0.0569	1.6924	0.0748
	ITM	1.5079	0.0674	1.4452	0.0629	1.3732	0.0650
	ATM	1.6157	0.0682	1.5684	0.0611	1.5123	0.0625
	OTM	1.6404	0.0682	1.6289	0.0600	1.5848	0.0621
	DOTM	1.6668	0.0795	1.6477	0.0664	1.6402	0.0628
Panel B. Put options							
7-day ahead	ITM	0.9905	0.8189	0.9264	0.7648	0.9569	0.8548
	ATM	1.0540	0.3302	1.0540	0.3302	0.9625	0.7800
	OTM	1.1483	0.3548	0.9962	0.5986	0.9833	0.6789
	DOTM	1.2119	0.5127	1.1381	0.5874	1.0988	0.6250
	VDOTM	1.2253	0.7088	1.2195	0.6615	1.2040	0.6715
30-day ahead	ITM	1.3082	0.8095	1.1627	0.7614	1.1809	0.8525
	ATM	1.3629	0.3280	1.3629	0.3280	1.1592	0.7783
	OTM	1.5018	0.3528	1.2404	0.5955	1.2362	0.6770
	DOTM	1.6042	0.5101	1.4795	0.5839	1.3820	0.6232
	VDOTM	1.6351	0.7043	1.6193	0.6574	1.5991	0.6688

Table 3. The 7-day ahead forecasting performance of alternative option valuation models

This table reports the 7-day ahead forecasting performance of the proposed hybrid Gated Neural Network (hGNN), the deep Neural Network (dNN), and Andreou et al. Neural Network (AnNN), SVJ, SVSI, and SV models for S&P 500 options. See also notes to Tables 1 and 2 for option moneyness and time-to-maturity categories.

		MAPE						RMSE					
		hGNN	dNN	AnNN	SVJ	SVSI	SV	hGNN	dNN	AnNN	SVJ	SVSI	SV
Panel A. Call options													
Short-term	DITM	0.9068	4.7082	7.7459	8.5967	8.1560	14.427	14.470	111.01	187.00	187.86	187.71	191.58
	ITM	1.4380	5.3417	6.4899	6.8020	7.8699	8.2488	2.8624	11.126	15.206	15.752	15.847	21.334
	ATM	1.4949	14.142	18.370	18.366	19.281	18.910	0.5959	4.4155	6.1233	6.1062	6.1942	7.0094
	OTM	1.4952	23.961	34.928	30.588	35.529	36.173	0.1684	1.5617	2.3448	1.1951	5.1573	4.0795
	DOTM	1.5015	26.785	40.960	38.816	42.715	43.621	0.0121	0.2198	0.3571	0.2783	3.1528	1.8341
Medium-term	DITM	1.4382	11.395	14.084	14.549	15.740	17.414	13.920	127.70	204.10	204.89	204.58	211.28
	ITM	1.4234	3.4555	4.5304	4.5983	5.2669	11.835	3.5713	10.086	13.041	13.836	15.101	19.440
	ATM	1.5009	6.2692	7.4090	7.3818	9.4576	10.944	1.4096	6.6432	8.6026	8.5779	10.931	10.388
	OTM	1.5017	19.351	27.716	23.150	29.874	28.118	0.4046	2.8234	3.6683	2.2458	4.6349	5.2668
	DOTM	1.5020	27.695	40.730	34.329	42.691	40.773	0.0108	0.1695	0.3182	0.1582	3.0418	3.0173
Long-term	DITM	0.7664	12.258	10.259	10.631	10.823	16.861	3.4092	51.384	43.658	43.988	43.818	43.986
	ITM	1.5732	4.5289	5.8295	5.9598	8.1337	6.6131	4.1677	15.281	19.582	20.408	21.916	24.904
	ATM	1.7651	8.0328	9.0606	9.0476	9.2289	12.733	1.9879	10.033	12.761	12.742	12.847	15.701
	OTM	1.4663	21.861	32.236	28.984	32.465	33.788	0.5136	4.9001	6.3626	5.7121	7.4535	8.8917
	DOTM	1.6415	26.935	45.679	40.798	48.116	47.093	0.0132	0.1886	0.2798	0.5022	2.3312	0.7599
Panel B. Put options													
Short-term	ITM	1.0300	2.3728	3.6747	4.4018	3.9382	8.5679	4.0006	19.377	37.152	37.289	38.594	40.297
	ATM	1.2255	4.3285	5.9002	6.3042	8.7863	12.205	0.4406	1.3171	1.8434	2.6990	2.2552	7.7955
	OTM	1.4453	10.491	13.916	13.909	16.157	17.997	0.1460	0.6522	0.8543	0.8244	2.4273	2.6853
	DOTM	1.5025	16.735	22.808	15.824	23.163	23.447	0.0257	0.2562	0.4256	0.3252	2.4540	2.6174
	VDOTM	1.4968	22.898	33.878	30.808	34.492	35.872	0.0093	0.1532	0.2648	0.2566	2.5461	2.5967
Medium-term	ITM	0.6312	1.4410	2.8226	3.0859	5.4275	9.8405	1.4533	8.6912	13.267	14.132	14.307	19.514
	ATM	1.2204	4.3368	5.9128	6.2270	7.7942	6.1552	0.4285	1.3172	1.8422	1.9906	2.8081	7.4748
	OTM	1.0526	2.1472	3.2850	3.2755	5.8862	7.8656	0.3839	0.8739	1.2428	1.2254	1.8100	6.0545
	DOTM	1.5116	6.0621	6.9096	3.0653	9.8328	8.0143	0.0958	0.4076	0.4889	0.4064	1.5006	1.6148
	VDOTM	1.5024	17.436	23.186	19.727	25.054	23.973	0.0147	0.1723	0.2488	0.2383	3.2091	0.5432
Long-term	ITM	0.9337	1.6884	2.8824	3.3583	4.3139	9.2650	1.7487	4.1576	5.3781	5.7307	8.3144	11.765
	ATM	0.7864	1.5756	2.7518	3.6772	5.3466	4.1328	0.9397	1.8656	3.0347	3.6323	5.2289	8.2881
	OTM	1.0330	1.7011	2.9022	2.8981	4.7031	5.1092	0.4890	0.9234	1.5192	1.4919	3.1351	5.5123
	DOTM	1.5893	4.0484	4.8505	2.3919	7.8489	6.9475	0.1784	0.3701	0.4663	0.3127	2.7136	0.8474
	VDOTM	1.4360	14.137	16.652	13.762	19.117	18.580	0.0302	0.2479	0.2638	0.2618	0.6910	1.6563

Option price prediction

In Table 3, we report the baseline out-of-sample prediction results for the 7-day ahead horizon between our proposed hGNN model and alternative models. We summarize the forecasting performance for call (Panel A) and put (Panel B) options across different time to maturity and moneyness over two loss functions.

We find that the hGNN model consistently produces lower forecasting errors compared with the dNN and AnNN. For deep ITM calls that traditional neural network based models have a hard time predicting, the MAPE and RMSE for the hGNN are a mere 0.9% and 14.5, respectively, and they compare favorably with the other two models whereby the MAPE and RMSE are 4.7% and 111 for the dNN, and 7.8% and 187 for the AnNN, respectively. This superior forecasting performance is also evident for put options. For VDOTM puts, the forecasting errors measured by the MAPE and RMSE for the proposed model is 1.44% and 0.03, respectively, for long-term options, whereas they are 14% and 0.25 for the dNN, and 17% and 0.26 for the AnNN, respectively.[2]

Furthermore, we show that the hGNN model exhibits significantly improved performance compared with traditional SVJ, SVSI, and SV models with much smaller MAPE and RMSE. This is the case regardless of the option type, time to maturity, or option moneyness, and highlights the computational prowess of neural network based option valuation models.

For robustness, we conduct the same exercises over 30-day ahead forecasting horizon. We further determine the statistical significance of the pricing error differences for 7- and 30-day ahead forecasting errors between all six models via the Diebold and Mariano (1995) and Giacomini and White (2006) tests. These robustness tests show that the hGNN model continues to outperform the other five models with more precise forecasts, and is always the preferred model statistically to alternative models according to statistical inferences based on Diebold and Mariano (1995) and Giacomini and White (2006) tests.[3]

What drives this significantly improved forecasting performance by the hGNN model? To address this question, we form two groups of options. Group 1 includes 29,771 options, which are among the top 1% of our sample with the largest absolute percentage errors (APE) according to the dNN model, whereas Group 2 contains 294,795 options that are randomly selected from the rest of the sample. By taking a closer look at the characteristics and pricing performance of these two groups of options, we hope to better understand the driving factor behind the success of the hGNN model relative to the dNN and AnNN models.

In Table 4, we provide a simple summary of these two groups of options along the moneyness (K/F) and maturity dimensions. We note that overall options in Group 1 tend to be OTM in terms of moneyness, and much more short-dated in option maturity. In Figure 2, we illustrate the MAPE differences between the dNN and hGNN models in orange bars and between the AnNN and hGNN models in grey bars across forecasting horizons, option moneyness, and time to maturity. It is striking that for Group 1 options on the left of each panel, the MAPE differences are substantial: moving from DITM to DOTM options, the error differences increase dramatically within the same time-to-maturity group. For example, for 30-day ahead forecasts for call options, the MAPE differences are around 40% for DITM options but they go beyond 100% for DOTM options, indicating

[2] With the same volatility inputs, we show that the Black and Scholes model produces much less accurate forecasts of option prices. These results are shown below in Table A1.

[3] The results for robustness tests, unreported to conserve space, are available in the online appendix.

Table 4. Summary statistics for S&P 500 options in Groups 1 and 2

This table reports summary statistics of moneyness (K/F) and annualized time to maturity (TTM) for S&P 500 options in Groups 1 and 2: Group 1 contains options with the largest 1% of absolute percentage error from the dNN model, whereas Group 2 contains other options.

		7-day ahead calls		30-day ahead calls		7-day ahead puts		30-day ahead puts	
		Group 1	Group 2	Group 1	Group 2	Group 1	Group 2	Group 1	Group 2
K/F	Mean	1.1169	0.4922	1.1189	0.4911	0.8471	0.3627	0.8628	0.3695
	Median	1.1275	0.4971	1.1110	0.5058	0.8693	0.3814	0.8758	0.3894
	Max	1.3401	1.2357	1.3105	1.2165	1.4684	1.4581	1.4953	1.4392
	Min	0.4414	0.0047	0.2788	0.0012	0.3624	0.0004	0.3937	0.0009
TTM	Mean	0.0815	0.6862	0.0722	0.6889	0.0460	0.6790	0.0383	0.6862
	Median	0.0700	0.6849	0.0495	0.6758	0.0368	0.6512	0.0326	0.6540
	Max	0.2165	0.9826	0.2047	0.9899	0.2028	0.9914	0.2141	0.9863
	Min	0.0000	0.4897	0.0000	0.4959	0.0000	0.4819	0.0000	0.4901

that hGNN is able to generate prediction error that is 100% smaller than that generated by the dNN or the AnNN models for DOTM option group.[4] This pattern is also observed in the other panels for Group 1 options, highlighting the methodologically contribution of our model that it is exactly for these options on the moneyness boundary that our hGNN model massively outperforms the dNN and AnNN.

Meanwhile, the error differences for Group 2 options also increase gradually from DITM to DOTM options but with a much smaller magnitude. For DOTM options, the biggest MAPE differences tend to take place for short-term options at around 40-50%. This indicates that although the hGNN model generates more accurate out-of-sample option price predictions, the improvement tends to be milder.

In Figure 3, we further visualize the forecasting errors generated by the three models for individual options in Group 1. It is evident that the black dots, representing pricing errors generated by the hGNN model, lies below the blue dots, i.e. pricing errors from the dNN model, whereas the green dots, i.e., pricing errors generated by the AnNN model, are all over the place.

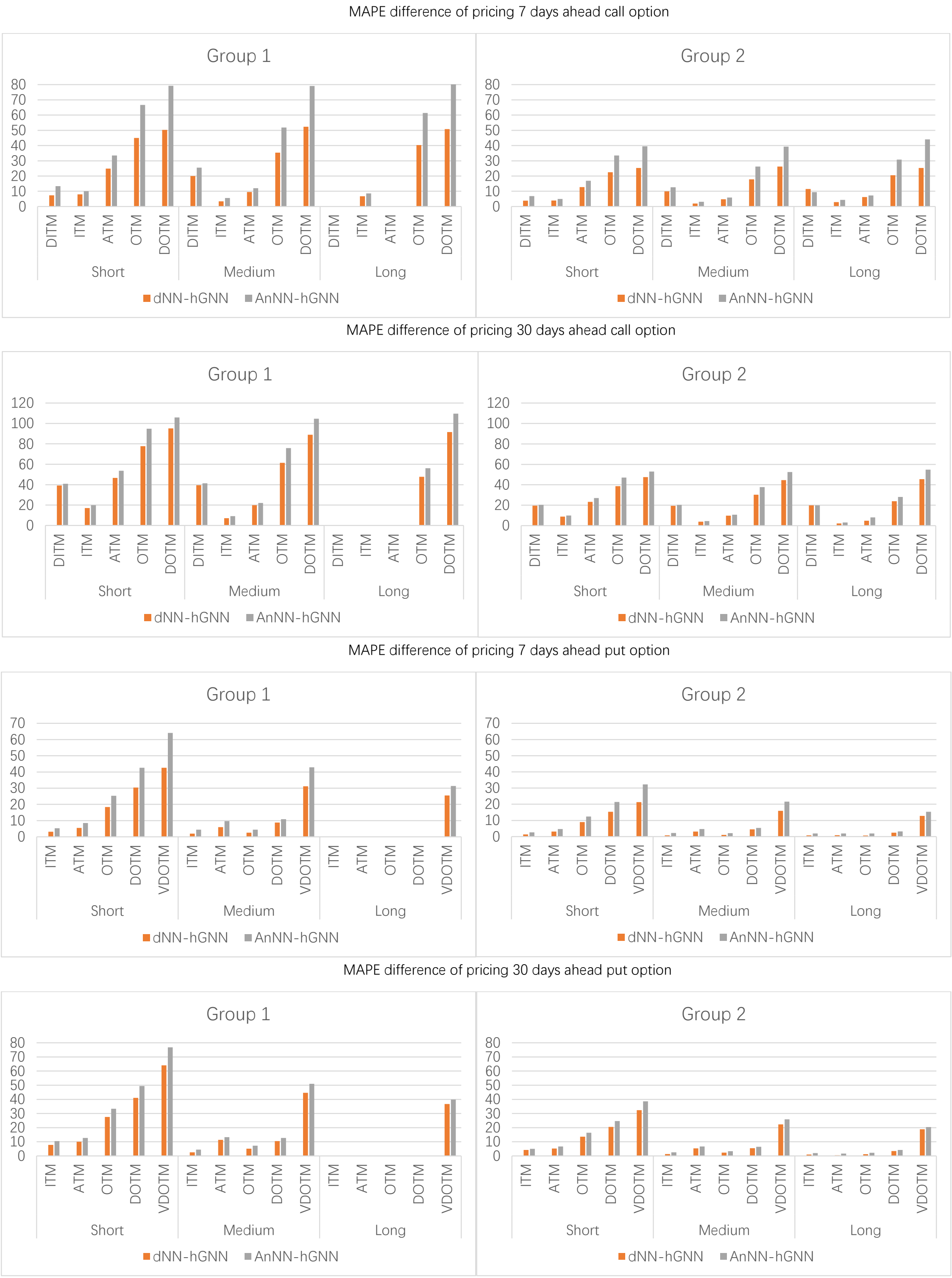
Hedging performance

Finally, we focus on the hedging performance of alternative models with 7- and 30-day rebalancing. Figure 4 shows the option Greeks δ, ν, Θ, and ρ for call options across strike prices and time-to-maturity. The average hedging errors for options with different time-to-maturity and moneyness are reported in Table 5. We note that across the board, the hGNN model consistently generates smaller average hedging error than the alternative models except one case whereby its

[4] In some cases, as option prices are smaller than one, the MAPE can be greater than 100%.

Figure 2. Comparison of forecasting performance of alternative option valuation models

This figure shows the difference in the mean absolute percentage error (MAPE) for the deep Neural Network (dNN) and the Andreou et al. Neural Network (AnNN) when compared to the hybrid Gated Neural Network (hGNN) model for options in Groups 1 and 2. The Y-axis is in percent. See also notes to Tables 1 and 2 for option moneyness and time-to-maturity categories.



MAPE difference of pricing 7 days ahead call option

MAPE difference of pricing 30 days ahead call option

MAPE difference of pricing 7 days ahead put option

MAPE difference of pricing 30 days ahead put option

Figure 3. Forecasting performance of alternative neural network based option valuation models for individual options in Group 1

This figure shows the absolute percentage error (APE) of the hybrid Gated Neural Network (hGNN), the deep Neural Network (dNN), and the Andreou et al. Neural Network (AnNN) models for Groups 1 options which lie in the top 1% of absolute percentage error (APE) according to the dNN model. See also notes to Tables 1 and 2 for option moneyness and time-to-maturity categories.

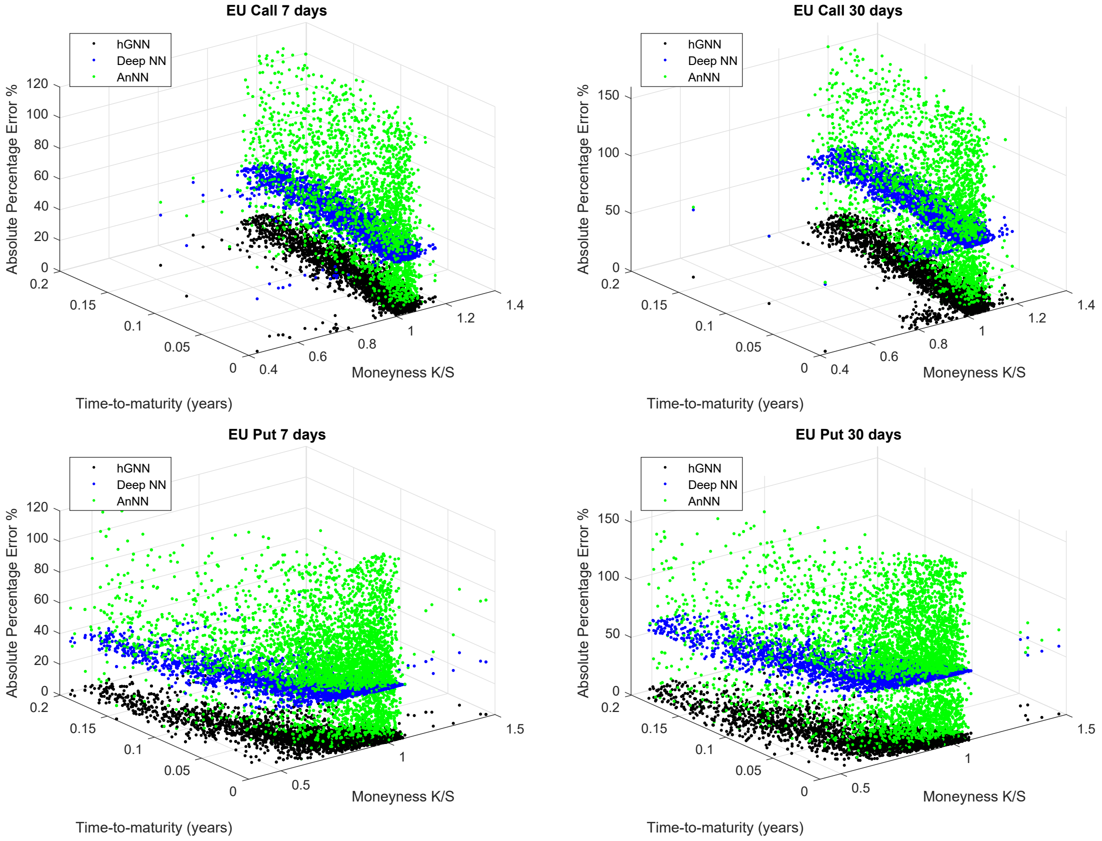
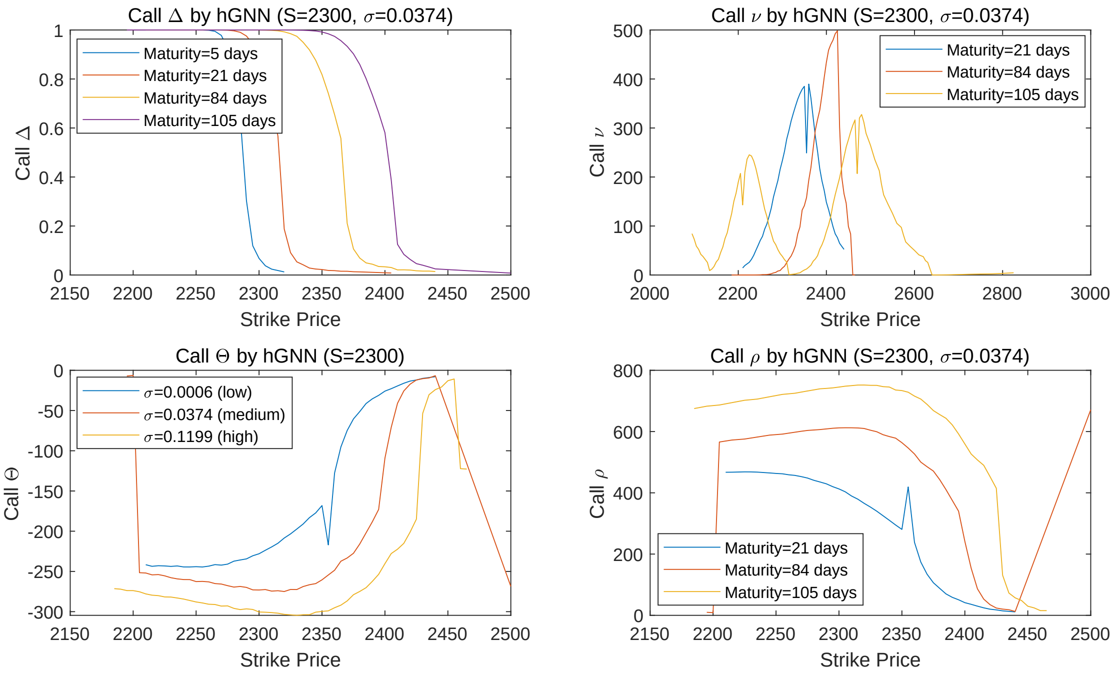


Figure 4. Greek letters for call options obtained via the hGNN model

This figure shows the option price with respect to the underlying asset (δ), volatility (ν), time-to-maturity (Θ), and interest rate (ρ) across different strike prices obtained via the hGNN model.



hedging error is slightly bigger. As the hedging performance is based on trading strategies involving options, the underlying asset, and bonds, smaller hedging errors represent the true economic value that the hGNN model generates for market participants relative to the other models.

To summarize, the empirical analyses conducted in this section show that the proposed hGNN option valuation model outperforms the other neural network based models as well as traditional models both in generating significantly smaller option price predictions and in offering smaller error on average in hedged portfolios using options. The results are robust with respect to different option moneyness, time to maturity, forecasting horizons, and the put/call type.

Robustness check

In addition to the baseline results, we have further examined the prediction performance of these six option pricing models over 30-day ahead forecasting horizon. Using the Diebold and Mariano (1995) and Giacomini and White (2006) tests, we show that the forecasting errors from our proposed model is significantly smaller than those from the other models over both 7- and 30-day ahead horizons. This is also the case for grouped options.

Table 5. The hedging performance of alternative option valuation models

This table reports the average hedging error when portfolios are re-balanced every 7 or 30 days for the hGNN, dNN, AnNN, SVJ, SVSI, and SV models for S&P 500 call options. The forecasting period is from 29/08/2016 to 02/03/2018. See also notes to Tables 1 and 2 for option moneyness and time-to-maturity categories.

		7-day rebalance						30-day rebalance					
		hGNN	dNN	AnNN	SVJ	SVSI	SV	hGNN	dNN	AnNN	SVJ	SVSI	SV
Short-term	DITM	-0.0582	-0.0939	-0.0862	-0.1162	-0.1062	-0.1662	-0.1898	-0.2387	-0.2698	-0.2998	-0.2898	-0.3498
	ITM	-0.0675	-0.0866	-0.0873	-0.1173	-0.1073	-0.1673	-0.1411	-0.2124	-0.2224	-0.2524	-0.2424	-0.3024
	ATM	-0.0550	-0.0739	-0.0924	-0.1224	-0.1124	-0.1724	-0.1347	-0.2027	-0.2447	-0.2747	-0.2647	-0.3247
	OTM	-0.0402	-0.0722	-0.1073	-0.1373	-0.1273	-0.1873	-0.0973	-0.1995	-0.2835	-0.3135	-0.3035	-0.3635
	DOTM	-0.0412	-0.0715	-0.0999	-0.1299	-0.1199	-0.1799	-0.1041	-0.1928	-0.2833	-0.3133	-0.3033	-0.3633
Medium-term	DITM	-0.0625	-0.0949	-0.1442	-0.1742	-0.1642	-0.2242	-0.1894	-0.2727	-0.3126	-0.3426	-0.3326	-0.3926
	ITM	-0.0475	-0.0810	-0.0950	-0.1250	-0.1150	-0.1750	-0.1615	-0.2050	-0.2782	-0.3082	-0.2982	-0.3582
	ATM	-0.0526	-0.0661	-0.0792	-0.1092	-0.0992	-0.1592	-0.1087	-0.2000	-0.1974	-0.2274	-0.2174	-0.2774
	OTM	-0.0324	-0.0680	-0.0974	-0.1274	-0.1174	-0.1774	-0.0769	-0.1972	-0.2752	-0.3052	-0.2952	-0.3552
	DOTM	-0.0140	-0.0432	-0.0789	-0.1089	-0.0989	-0.1589	-0.0331	-0.1279	-0.2185	-0.2485	-0.2385	-0.2985
Long-term	DITM	-0.1043	-0.1016	-0.1198	-0.1498	-0.1398	-0.1998	-0.2170	-0.2635	-0.3127	-0.3427	-0.3327	-0.3927
	ITM	-0.0525	-0.0843	-0.1174	-0.1474	-0.1374	-0.1974	-0.1506	-0.2085	-0.2935	-0.3235	-0.3135	-0.3735
	ATM	-0.0315	-0.0797	-0.1131	-0.1431	-0.1331	-0.1931	-0.1101	-0.2026	-0.2767	-0.3067	-0.2967	-0.3567
	OTM	-0.0126	-0.0630	-0.0972	-0.1272	-0.1172	-0.1772	-0.0429	-0.1590	-0.2841	-0.3141	-0.3041	-0.3641
	DOTM	-0.0226	-0.0515	-0.0909	-0.1209	-0.1109	-0.1709	-0.0424	-0.1527	-0.2510	-0.2810	-0.2710	-0.3310

6 Conclusion

The recent literature has seen substantial advancement in a number of new option pricing models based on data science methods that generate more precise option price forecasts. However, a major issue for this kind of models is the lack of economic intuition and interpretation. Hence, they are considered a *black box* to the mainstream finance industry.

This study represents a novel approach in addressing this issue. We develop a hybrid gated neutral network (hGNN) model for option valuation that not only produces superior prediction accuracy as many neural network based models do, but also offers analytical expressions for option Greeks which improves its hedging performance. We start from no-arbitrage constraints in the option pricing theory that all models need to meet, construct a multiplicative structure for the hidden neurons to maintain its differentiability, and select the slope and weights in the input layer to satisfy the no-arbitrage constraints. We further train this model with synthesized theoretical values for options on the boundaries thus ensuring that all constraints are satisfied. Furthermore, we construct a separate neural network model for forecasting option-implied volatilities with information from option moneyness, time to maturity, and lagged implied volatilities.

Using daily data from May 2014 to March 2018, we show empirically that our hGNN model consistently and significantly outperforms both neural network based models such as the dNN and AnNN, and traditional models such as the stochastic volatility model with jumps in forecasting

Table A1. The 7-day ahead forecasting performance of the Black and Scholes model

This table reports the 7-day ahead forecasting performance of the Black and Scholes option valuation model for S&P 500 options. The implied volatilities are predicted via the neural network model outlined in Section 3.4. See also notes to Tables 1 and 2 for option moneyness and time-to-maturity categories.

		MAPE	RMSE		MAPE	RMSE
	Call options			*Put options*		
Short-term	DITM	10.661	129.77	ITM	23.356	109.45
	ITM	12.652	36.923	ATM	14.325	10.859
	ATM	7.6431	3.8802	OTM	27.353	6.9819
	OTM	123.94	25.140	DOTM	40.076	7.0761
	DOTM	588.55	101.74	VDOTM	45.565	5.9635
Medium-term	DITM	96.314	689.02	ITM	11.402	19.917
	ITM	35.996	89.599	ATM	13.539	11.802
	ATM	27.852	77.758	OTM	20.180	22.304
	OTM	321.42	66.937	DOTM	33.089	7.4446
	DOTM	1501.9	752.14	VDOTM	42.698	1.5740
Long-term	DITM	76.612	352.95	ITM	11.951	16.194
	ITM	4.1100	20.146	ATM	14.029	25.493
	ATM	9.0711	11.912	OTM	19.256	15.299
	OTM	291.89	912.84	DOTM	29.534	5.9951
	DOTM	1556.2	982.50	VDOTM	39.901	2.9784

S&P 500 option prices across board. It also offers smaller hedging errors in a delta hedging strategy involving stocks and bonds compared with alternative models. These empirical findings substantiate the novelty of our model as a step towards formulating neural network based models with economic insight.

References

Abu-Mostafa, Y. S., 1993. A method for learning from hints. Advances in Neural Information Processing Systems 5, 73–80.

Abu-Mostafa, Y. S., 1994. Learning from hints. Journal of Complexity 10, 165–178.

Abu-Mostafa, Y. S., 1995. Financial applications of learning from hints. Advances in Neural Information Processing Systems 7, 411–418.

Ait-Sahalia, Y., Duarte, J., 2003. Nonparametric option pricing under shape restrictions. Journal of Econometrics 116, 9–47.

Andreou, P. C., Charalambous, C., Martzoukos, S. H., 2008. Pricing and trading European options by combining artificial neural networks and parametric models with implied parameters. European Journal of Operational Research 185, 1415–1433.

Andreou, P. C., Charalambous, C., Martzoukos, S. H., 2010. Generalized parameter functions for option pricing. Journal of Banking & Finance 34, 633–646.

Andreou, P. C., Charalambous, C., Martzoukos, S. H., 2014. Assessing the performance of symmetric and asymmetric implied volatility functions. Review of Quantitative Finance and Accounting 42, 373–397.

Bakshi, G., Cao, C., Chen, Z., 1997. Empirical performance of alternative option pricing models. Journal of Finance 52, 2003–2049.

Bakshi, G., Madan, D., 2000. Spanning and derivative-security valuation. Journal of Financial Ecnomics 55, 205–238.

Barua, S., Islam, M. I., Yao, X., Murase, K., 2014. MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning. IEEE Transactions on Knowledge and Data Engineering 26, 405–425.

Bates, D. S., 1996. Jumps and stochastic volatility: Exchange rate processes implicit in Deutsche Mark options. Review of Financial Studies 9, 69–107.

Bengio, Y., 2013. Deep learning of representation: Looking forward. In: Dediu, A. H., Martin-Vide, C., Mitkov, R., Truthe, B. (Eds.), Statistical Language and Speech Processing. Springer, Berlin, pp. 1–37.

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

Bliss, R. R., Panigirtzoglou, N., 2002. Testing the stability of implied probability density functions. Journal of Banking & Finance 26, 381–422.

Buehler, H., Gonon, L., Teichmann, J., Wood, B., 2019. Deep hedging. Quantitative Finance 19, 1271–1291.

Cao, Y., Li, Y., Coleman, S. A., Belatreche, A., McGinnity, T. M., 2015. Adaptive hidden Markov model with anomaly states for price manipulation detection. IEEE Transactions on Neural Networks and Learning Systems 26, 318–330.

Cao, Y., Li, Y., Coleman, S. A., Belatreche, A., McGinnity, T. M., 2016. Detecting wash trade in financial market using digraphs and dynamic programming. IEEE Transactions on Neural Networks and Learning Systems 27, 2351–2363.

Chalamandaris, G., Tsekrekos, A. E., 2014. Predictability in implied volatility surfaces: Evidence from the Euro OTC FX market. European Journal of Finance 20, 33–58.

Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P., 2002. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357.

Cheng, Z., Xu, Y., Cheng, M., Miao, Y., Pu, S., Niu, Y., Wu, F., 2020. Refined gate: A simple and effective gating mechanism for recurrent units, retrieved from arXiv: https://arxiv.org/abs/2002.11338v2.

Cochrane, J. H., 2001. Asset Pricing (Revised Edition). Princeton University Press, New Jersey.

Culkin, R., Das, S. R., 2017. Machine learning in finance: The case of deep learning for option pricing. Journal of Investment Management 15, 92–100.

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.

Delbaen, F., Schachermayer, W., 1994. A general vresion of the fundamental theorem of asset pricing. In: The Mathematics of Arbitrage. Springer Finance. Springer, Berlin, pp. 463–520.

Diebold, F. X., Mariano, R. S., 1995. Comparing predictive accuracy. Journal of Business & Economic Statistics 13, 253–263.

Duan, J.-C., 1995. The GARCH option pricing model. Mathematical Finance 5, 13–32.

Dugas, C., Bengio, Y., Belisle, F., Nadeau, C., Garcia, R., 2001. Incorporating second-order functional knowledge for better option pricing. Advances in Neural Information Processing Systems 13, 472–478.

Dugas, C., Bengio, Y., Belisle, F., Nadeau, C., Garcia, R., 2009. Incorporating functional knowledge in neural networks. Journal of Machine Learning Research 10, 1239–1262.

Dunis, C., Kellard, N. M., Snaith, S., 2013. Forecasting EUR-USD implied volatility: The case of intraday data. Journal of Banking & Finance 37, 4943–4957.

Eraker, B. E., Johannes, M., Polson, N., 2003. The impact of jumps in volatility and returns. Journal of Finance 58, 1269–1300.

Fengler, M. R., Hin, L.-Y., 2015. Semi-nonparametric estimation of the call-option price surface under strike and time-to-expiry no-arbitrage constraints. Journal of Econometrics 184, 242–261.

Fischer, T., Krauss, C., 2018. Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research 270, 654–669.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F., 2012. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42, 463–484.

Garcia, R., Gencay, R., 2000. Pricing and hedging derivative securities with neural networks and a homogeneity hint. Journal of Econometrics 94, 93–115.

Giacomini, R., White, H., 2006. Tests of conditional predictive ability. Econometrica 74, 1545–1578.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., Kagal, L., 2018. Explaining explanations: An approach to evaluating interpretability of machine learning. arXiv preprint arXiv:1806.00069.

Gradojevic, N., Genay, R., Kukolj, D., 2009. Option pricing with modular neural networks. IEEE Transactions on Neural Networks 20, 626–637.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Predreschi, D., 2018. A survey of methods for explaining Black Box models. ACM Computing Surveys 51, 93–135.

H2O, 2018. H2O documentation. `http://docs.h2o.ai/`, [Online; accessed 30-August-2018].

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

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R., 2012. Improving neural networks by preventing co-adaptation of feature detectors, https://arxiv.org/abs/1207.0580v1.

Hu, Z., Ma, X., Liu, Z., Hovy, E., Xing, E., 2016. Harnessing deep neural networks with logic rules. arXiv preprint arXiv:1603.06318.

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

IS Committee, 2008. 754-2008 IEEE standard for floating-point arithmetic. IEEE Computer Society Std 2008.

Jackwerth, J., Rubinstein, M., 1996. Recovering probability distribution from option prices. Journal of Finance 51, 1611–1631.

Khoshgoftaar, T. M., Van Hulse, J., Napolitano, A., 2011. Comparing boosting and bagging techniques with noisy and imbalanced data. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 41, 552–568.

Kingma, D. P., Ba, J., 2017. Adam: A method for stochastic optimization, arXiv:1412.6980.

Knight, W., 2017a. The dark secret at the heart of AI, MIT Technology Review.

Knight, W., 2017b. The financial world wants to open AI's black boxes, MIT Technology Review.

Kolanovic, M., Krishnamachari, R. T., 2017. Big data and AI strategie - Machine learning and alternative data approach to investing, JP Morgan Global Quantitative and Derivatives Strategy Team.

Konstantinidi, E., Skiadopoulos, G., Tzagkaraki, E., 2008. Can the evolution of implied volatility be forecasted? Evidence from European and US implied volatility indices. Journal of Banking & Finance 32, 2401–2411.

Krauss, C., Do, X. A., Huck, N., 2017. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. European Journal of Operational Research 259, 689–702.

Kursuncu, U., Gaur, M., Sheth, A., 2019. Knowledge infused learning (k-il): Towards deep incorporation of knowledge in deep learning. arXiv preprint arXiv:1912.00512.

LeCun, Y., Bengio, J., Hinton, G. E., 2015. Deep learning. Nature 521, 436–444.

LeDell, E., 2018. h2o: R interface for 'h2o'. https://cran.r-project.org/package=h2o, [Online; accessed 31-August-2018].

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

Liu, X., Cao, Y., Ma, C., Shen, L., 2019. Wavelet-based option pricing: An empirical study. European Journal of Operational Research 272, 1132–1142.

McNelis, P. D., 2005. Neural Networks in Finance: Gaining Predictive Edge in the Market. Elsevier Academic Press, USA.

Memisevic, R., 2013. Learning to relate images. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1829–1846.

Merton, R. C., 1976. Option pricing when underlying stock returns are discontinuous. Journal of Financial Economics 3, 125–144.

Mudrakarta, P. K., Taly, A., Sundararajan, M., Dhamdhere, K., 2018. Did the model understand the question? arXiv preprint arXiv:1805.05492.

Phan, W., Stensmo, M., Dymczyk, M., Candel, A., Kou, Q., 2017. Deep learning with deep water. www.h2o.ai/wp-content/uploads/2018/01/DeepWater-BOOKLET.pdf, [Online; accessed 31-August-2018].

Ribeiro, M. T., Singh, S., Guestrin, C., 2016. Why should I trust you? Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp. 1135–1144.

Roper, M., 2010. Arbitrage free implied volatility surfaces, Unpublished manuscript, University of Sydney.

Ruf, J., Wang, W., 2020. Neural networks for option pricing and hedging: A literature review, *forthcoming* Journal of Computational Finance.

Sigaud, O., Masson, C., Filliat, D., Stulp, F., 2015. Gated networks: An inventory, arXiv:1512.03201.

Song, Z., Xiu, D., 2016. A tale of two option markets: Pricing kernels and volatility risk. Journal of Econometrics 190, 176–196.

Wijaya, G. P., Handian, D., Nasrulloh, I. F., Riza, L. S., Megasari, R., Junaeti, E., 2018. GradDescent. `https://cran.r-project.org/package=gradDescentR`, [Online; accessed 31-August-2018].

Yang, Y., Zheng, Y., Hospedales, T. M., 2017. Gated neural networks for option pricing: Rationality by design, In: 31st AAAI Conference on Artificial Intelligence.

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

Yu, Y., Si, X., Hu, C., Zhang, J., 2019. A review of recurrent neural networks: LSTM cells and network architectures. Neural Computation 31, 1235–1270.

Zhou, G.-B., Wu, J., Zhang, C.-L., Zhou, Z.-H., 2016. Minimal gated unit for recurrent neural networks, retrieved from arXiv: https://arxiv.org/abs/1603.09420v1.