Journal of Mathematics and Modeling in Finance (JMMF)
Vol. 4, No. 1, Winter & Spring 2024
Research paper

Calibration of European option pricing model using a hybrid structure based on the optimized artificial neural network and Black-Scholes model

Farshid Mehrdoust[1], **Maryam Noorani**[2]

[1] Department of Applied Mathematics, Faculty of Mathematical Science, University of Guilan
far.mehrdoust@gmail.com

[2] Department of Applied Mathematics, Faculty of Mathematical Science, University of Guilan
maryam.nooraani@gmail.com

Abstract:
This study suggests a novel approach for calibrating European option pricing model by a hybrid model based on the optimized artificial neural network and Black-Scholes model. In this model, the inputs of the artificial neural network are the Black-Scholes equations with different maturity dates and strike prices. The presented calibration process involves training the neural network on historical option prices and adjusting its parameters using the Levenberg-Marquardt optimization algorithm. The resulting hybrid model shows superior accuracy and efficiency in option pricing on both in sample and out of sample dataset.

Keywords: Artificial neural network, Calibration, Levenberg-Marquardt algorithm, Option pricing.
Classification: G13, C45

1 Introduction

The pricing of European options plays a crucial role in financial markets, as it enables investors to assess the value of their investments and make informed decisions. However, accurately pricing these options can be a challenging task, as traditional models often fail to capture the complexity of the market dynamics. In recent years, artificial neural networks (ANNs) have emerged as a powerful tool for option pricing, offering increased flexibility and accuracy compared to traditional models. Calibration of European option pricing model refers to the process of adjusting the parameters of a model in order to accurately price European options. Pricing these options requires the use of mathematical models, such as the Black-Scholes model, which rely on certain input parameters to generate accurate pricing estimates. Calibration involves adjusting these parameters to reflect market conditions and ensure

[1] Corresponding author
Received: 21/03/2024 Accepted: 02/05/2024
https://doi.org/10.22054/JMMF.2024.78910.1128

that the model prices options in line with observed market prices. This process is crucial for traders and investors to make informed decisions about trading options and managing risk in their portfolios.

The importance of estimating and calibrating the parameters of financial models cannot be understated, as it has significant economic implications and has become a focal point for researchers in recent times. Various methods exist for model calibration, including techniques such as nonparametric calibration, least-squares fitting and relative entropy ([4]). However, these methods often necessitate the specification of initial guesses for regularization terms, posing a challenge for accurate calibration. Alternatively, other estimation techniques like maximum likelihood estimation (MLE) and the expectation-maximization (EM) algorithm offer viable options for estimating the parameters of a given process (see [26]; [17]). Bayesian estimation, prioritizing parameter estimation, relies on the Markov chain Monte-Carlo (MCMC) for computational purposes ([24]; [6]). The implementation of the MCMC algorithm typically demands a significant time and simulation commitment. The method of moments, deriving model moments, has gained traction in parameter estimation in recent years. [28] investigated the parameters estimation of Lévy-Ornstein-Uhlenbeck model with similar approaches studied by [30] and [25]. Challenges arise when the density function is required for estimating model parameters using MLE method, especially if the financial model being implemented does not allow for the explicit determination of the density function. When dealing with these situations, it is necessary to take into account an estimation of the function, which introduces a new layer of complexity to the problem.

In this study, we address the classical issue in mathematical finance: the calibration of option pricing models to market data, which has recently gained attention in the financial community in the context of ANN . Previous works, such as [9] and [8], have explored this topic, as well as additional references cited within. In summary, the essence of these works is to optimize the parameters of asset pricing models so that they minimize the discrepancy (measured by a certain norm) between the model-predicted prices and the market-provided prices (refer to [13] and [15] for more details). Specifically in this paper, we focus on option pricing models for clarity. The motivation behind utilizing ANN techniques for this problem stems from the realization that computing option prices using traditional models can be time-consuming, leading to slow calibration processes. The approach introduced in this study breaks down the calibration into two distinct steps. Initially, a slow pricer is approximated by an ANN, which is trained using a set of in-sample data, enabling the weights of the ANN to be determined. Subsequently, in the calibration process, the model pricer is replaced with the trained ANN from the previous step, expediting the overall calibration procedure.

In recent years, machine learning (ML) algorithms have become popular for forecasting time series. These algorithms utilize intricate mathematical models to analyze extensive data sets and uncover patterns and connections that might not be obvious to human analysts. One of the key advantages of using ANNs as an ML algorithm is their ability to effectively approximate non-linear functions and handle multi-dimensional variables. ANNs function as behavioral approximations similar to the human brain and can be trained with the appropriate number of hidden layers and computational resources to approximate complex functions. Researchers have explored using ANNs to estimate parameters for various types of equations, such as stochastic, ordinary, partial, and uncertain differential equations. However, the application of ANN in estimating Black-Scholes parameters for modeling option prices has not been thoroughly investigated. This paper aims to address this gap by employing an optimized ANN to calibrate the Black-Scholes model using the Levenberg-Marquardt (LM) optimization method, with performance evaluation based on numerical results. The LM optimization method combines elements of both the steepest descent and Gauss-Newton algorithms, allowing it to efficiently navigate complex and nonlinear parameter spaces (see [21]). One of the key advantages of using the LM algorithm for calibration is its ability to handle ill-conditioned problems and noisy data, making it robust and reliable for a wide range of applications. Additionally, the algorithm is computationally efficient and can often converge to a solution faster than other optimization methods.

The adventitious of calibration of ANN model parameters by LM optimization offers a powerful and versatile tool for refining mathematical models and achieving a closer alignment between theoretical predictions and real-world data. By iteratively adjusting the parameter values based on the gradient and curvature of the objective function, the algorithm converges towards a set of optimal parameter values that best fit the observed data (see [19]). Calibration algorithms aim to minimize the error criterion, or loss function, by exploring a range of parameters intelligently. Previous studies have discussed this approach (see [16]; [20]). The loss function can be likened to the inverse vega on option market prices, as explored in other works (see [3]; [11]). One common method for calibrating parametric models, such as the Heston model and its variations, is using the least squares method on option prices. When calibrating the proposed ANN model, we aim to minimize the mean square error (MSE) between option market prices and a semi-analytical solution of option prices using the LM method.

2 The framework of ANN-based calibration

The architecture of an ANN typically made up of three layers: the input layer, hidden layer, and output layer, each containing neurons that are spread out throughout the network. Research in this field suggests that there is a possibility for the hidden layer to include multiple layers, rather than just one ([23]), but some theoretical studies have shown that using just one hidden layer is sufficient for estimating complex nonlinear functions. Experimental evidence presented by [7] also supports the idea that using more than one hidden layer can lead to the network getting stuck in local minimums, with no significant improvement in forecasting accuracy.

The ANN framework can be viewed as a sequence of functional advancements. This task involves creating the M linear combinations of the input variables $x_1, x_2, \ldots, x_N$ in the specified form as follows

$$a_j = \sum_{i=1}^{N} w_{ji}^{(1)} x_i + w_{j0}^{(1)}, \quad j = 1, \ldots, M, \tag{1}$$

where the caption (1), it is specified that the parameters pertain to the input layer of the ANN structure. The biases and weight vectors are denoted as $w^{(1)}ji$ and $w^{(1)}j0$, respectively, while a_j represents the activation value. Let $h(\cdot)$ be a differentiable and nonlinear function serving as the activation function, expressed as:

$$z_j = h(a_j). \tag{2}$$

The biases and weight parameters can be modified according to the activation function in this scenario. These parameters are influenced by the output of the activation functions and are assigned to the hidden modules within the structure of the ANN. The function can be represented as such:

$$y(X, W) = f\Big(\sum_{j=1}^{M} w_j \phi_j(X)\Big). \tag{3}$$

The activation functions commonly used to address both non-linear and linear regression problems include the hyperbolic tangentsigmoid (tansig), logsigmoid (logsig), and linear (purelin) functions. To calculate the output module activations, a linear combination of these values is represented as shown in Eq. (2).

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)}, \quad k = 1, \ldots, K, \tag{4}$$

Let K represent the total number of outputs. By applying an appropriate activation function, the outputs generated by the neuron activations are transformed into a set of outputs for the ANN, denoted as y_k. This function is treated as a regression problem, where y_k is equal to a_k. Furthermore, the output neurons are mapped into multiple binary problems using the ReLu function, thereby ensuring that

$$y_k = \sigma(a_k), \tag{5}$$

where

$$\sigma(a) = \max\big(0, a\big), \tag{6}$$

Equivalently, a softmax activation function can be expressed as follows

$$p(C_k \mid X) = \frac{\exp(\exp(a_k))}{\sum_j \exp(\exp(a_j))}. \tag{7}$$

In this study, we consider the inputs of the neural network as an analytical (or semi-analytical) function of the option price written on a particular asset with value S on a given day as follows:

$$x_i \equiv C(\Lambda; S, T_i, K_i), \quad i = 1, 2, \ldots, N,$$

where Λ is the parameters set of the desired option on the ith neuron of the input layer, and strike price K_i and maturity time T_i on the ith neuron are derived from market information.

One way to represent the activation function of an ANN is by using the ReLu function. Here is the expression for it:

$$P_k = \sigma\Big(\sum_{j=1}^{M} w_{kj}^{(2)} h\Big(\sum_{i=1}^{N} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\Big) + w_{k0}^{(2)}\Big), \quad k = 1, \ldots, N. \tag{8}$$

Let $w^{(2)}kj$ represent the weights connecting neurons from the hidden layer to the output layer, $w^{(1)}ji$ represent the weights connecting neurons from the input layer to the hidden layer, and $w^{(1)}j0$ and $w^{(2)}k0$ represent the bias values for the hidden and output layers, respectively. Moreover, h and σ are the output result on the hidden and output layer. Assuming that $W := \{w_{ji}^{(1)}, w_{kj}^{(2)}, w_{j0}^{(1)}, w_{k0}^{(2)}\}_{k,i=1,\ldots,N,j=1,\ldots,M}$, and by denoting $\Theta := \{W, \Lambda\}$ as the set of ANN parameters and the option price function, we provide the estimation of Θ.

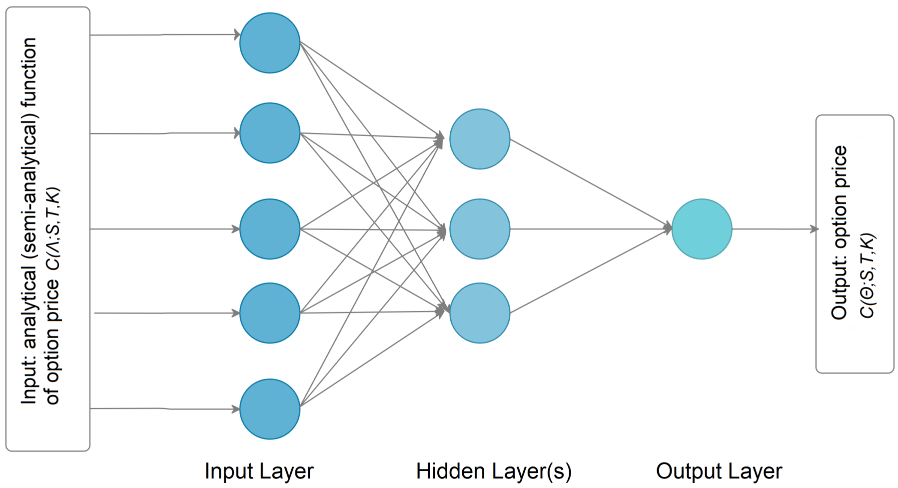

Figure 1: A one-hidden-layer neural network used in this study.

The structure of an ANN is determined by a nonlinear function of a vector of input variables $X := x_i$ to the vector of output variables $P := P_k$, which is controlled by a vector of adjustable parameters W (see [7]). In this research, the identity function is utilized as the activation function connecting the input and hidden layers in this study. Conversely, the ReLU function is implemented as the activation function that links the hidden and output layers. The goal of using an adventious RELu function is to improve the performance of a neural network or deep learning model by customizing the activation function to better fit the data and optimize the learning process. Another advantage is that the ReLU function closely resembles biological reality by disallowing negative values. Consequently, Eq. (8) can be expressed as follows:

$$P_k(\Theta) = \max\Big[0, \sum_{j=1}^{M} w_{kj}^{(2)} \Big(\sum_{i=1}^{N} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \Big) + w_{k0}^{(2)}\Big], \quad k = 1, \ldots, N.$$

To achieve our goal of forecasting option prices, we have structured the ANN as 5 input neurons (related to the option price), 3 hidden neurons (determined through trial and error), and 1 output neuron (predicting the option price for a specific market). This configuration is represented as 5-3-1 in Figure 1.

3 The ANN optimized by LM algorithm

To predict option prices, the dataset described in the previous section is divided into separate parts. There is no universally agreed upon data partition ratio in the literature. Previous research (such as [27]) found that the selection of data for training can impact the final outcome of data prediction. To ensure the reliability of the neural network, it is necessary to implement the algorithm on multiple partitions. In this study, the dataset for option prices is divided into an 80-20 ratio, where the first number represents the percentage of training data and the second number represents the percentage of test data. The primary goal of the LM algorithm is to minimize the value of a given objective function by iteratively updating the parameters of the model in the direction that reduces the error between the predicted and actual values. This is achieved by calculating the gradient of the objective function and determining the step size in the parameter space that will result in the greatest reduction in the error. The LM optimization algorithm is a popular method for performing this parameter update as it combines the strengths of gradient descent and Gauss-Newton methods for efficient and robust optimization (10). This function serves as a generalization of the error function, which helps accelerate convergence. By determining the optimal values for these parameters, the model is ready to predict option prices within the neural network structure.

Once all the necessary parameters have been identified, the next step is to determine a suitable approach. A commonly used method is to identify the optimal parameters that minimize the discrepancy between market prices and model prices. There are various definitions of this discrepancy, with one common approach being to calculate the mean squared error:

$$\bar{\Xi} = \frac{1}{N} \sum_{k=1}^{N} \frac{\mid P_k(\Theta) - Y_k \mid^2}{Y_k}. \tag{9}$$

This objective function measures the relative difference between market prices (denoted by Y_k) and model prices (calculated using a specific set of parameters Θ). The total number of observations selected for estimation is represented by N. However, using mean squared error as the objective function has a drawback it may give disproportionate weight to cheaper options (see [2] and [12]). To address this issue, we opted to use mean squared errors as follows:

$$\Xi = \frac{1}{N} \sum_{k=1}^{N} \mid P_k(\Theta) - Y_k \mid^2 . \tag{10}$$

With this revised approach, the goal is to search the parameter space systematically in order to identify a parameter set that minimizes the mean squared error between model prices and market prices.

Selecting the right optimization method is a key concern in the optimization process. Local minimization, for example, relies heavily on an initial guess that is very close to the true optimal solution. While this approach can be quicker, it often yields unreliable results due to its dependence on the starting point. The objective function (10) may not be convex, leading to the presence of multiple local minima, potentially causing the global minimum to be mistaken for a local minimum. In contrast, a well-designed global optimization method is capable of bypassing local minima and accurately pinpointing the global minimum in a more efficient manner.

The calculation of option prices and empirical studies using the ANN model requires input parameters that are not directly available from the market data. According to a study in [1], the implicit structural parameters can differ significantly from the estimated parameters obtained from sample data in the time series. It is crucial to fine-tune the model parameters to ensure that the model's price aligns with the market prices. This discrepancy between the market data and the parameters further complicates the calibration process. In reality, achieving precise matches of observed prices is unfeasible and lacks significance. As a result, the calibration issue for the model is treated as a nonlinear optimization challenge. The objective is to reduce the discrepancy between the option prices calculated by the model and those traded in the market for a specific set of options. One method to quantify this discord is by calculating the MSE between the market prices and the model, resulting in solutions for the nonlinear least square problem.

Let $Y_i(K_i, T_i)$ denote the market price of a European call option with strike K_i and expiry date T_i, and let $P_i(\Theta, K_i, T_i)$ be the European call option price computed by the ANN model using N inputs, treating Black-Scholes equations with maturity date T_k and strike price K_i for all $i = 1, \ldots, N$. In order to calibrate the model parameters, Θ, the residuals for the N option prices are adjusted as follows:

$$E_i(\Theta) := \frac{1}{N}\sum_{k=1}^{N} \mid P_i(\Theta) - Y_i \mid^2, \quad i = 1, \dots, N.$$

Define the residual vector $\mathbf{E}(\Theta) \in \mathbb{R}^{n\times 1}$ as follows:

$$\mathbf{E}(\Theta) := \big(E_1(\Theta), E_2(\Theta), \dots, E_N(\Theta)\big)^\top.$$

The calibration of the proposed ANN model is treated as a nonlinear least-square problem given by:

$$\min_{\Theta\in\mathbb{R}^{m\times 1}} \Xi(\Theta), \tag{11}$$

where m denotes the total number of parameters, and

$$\Xi(\Theta) = \frac{1}{2}\|\mathbf{E}(\Theta)\|^2 = \frac{1}{2}E^\top(\Theta)E(\Theta). \tag{12}$$

When the number of observations exceeds the number of model parameters, specif-

Algorithm 1 Calibration of designed ANN model by using LM algorithm.

Require: The initial guess Θ_0, initial damping factor $\varpi_0 = \omega \max\{\mathrm{diag}[\mathcal{J}_0]\}$, tolerance level Tol and $\zeta_0 = 2$
  Calculate $\|\mathbf{E}(\Theta_0)\|$ and $\mathcal{J}_0$
  for $i = 0, 1, 2, \dots$ **do**
    Obtain Θ_i values by solving system (14)
    Calculate $\Theta_{i+1} = \Theta_i + \Delta\Theta_i$ and $\|\mathbf{E}(\Theta_{i+1})\|$
    Calculate $\varrho = \Delta\Theta_i^\top\big(\varpi_k\Delta\Theta_i + \mathcal{J}_i\mathbf{E}(\Theta_k)\big)$ and $\varrho' = \|\mathbf{E}(\Theta_k)\| - \|\mathbf{E}(\Theta_{k+1})\|$
    if $\varrho > 0$ and $\varrho' > 0$ **then**
      Calculate $\mathcal{J}_{i+1}, \varpi_{i+1} = \varpi_i, \zeta_{i+1} = \zeta_i$
    else
      Set $\varpi_i = \varpi_i\zeta_i, \zeta_i = 2\zeta_i$ and repeat from line 4
    end if
    if $\|\mathbf{E}(\Theta_i)\| \leq$ Tol **then**
      Break
    end if
  end for

ically when $N \gg m$, we are working with an overdetermined system.

The operator that deals with changes along the parameter vector Θ is denoted as $\nabla = \partial/\partial\Theta$, and we use $\nabla\nabla^\top$ as the Hessian operator.

Let $\mathcal{J} := (\mathcal{J}_{ji})_{\substack{i=1,\dots,N,\\ j=1,\dots,m}} = \nabla\mathbf{E}^\top \in \mathbb{R}^{m\times n}$ be the Jacobian matrix of the residual vector $\mathbf{E}$, such that

$$\mathcal{J}_{ji} = \left[\frac{\partial \mathbf{E}_i}{\partial \Theta_j}\right] = \left[\frac{\partial P\left(\Theta; K_i, T_i\right)}{\partial \Theta_j}\right].$$

Additionally, we define $\mathcal{H}(E_i) := \nabla abla^\top E_i \in \mathbb{R}^{m\times m}$ as the Hessian matrix of each residual E_i with elements $\mathcal{H}_{jk}(E_i) = \left[\frac{\partial^2 E_i}{\partial\Theta_j \partial\Theta_k}\right]$. In accordance with the nonlinear least-square expressions (11) and (12), the gradient and Hessian of Ξ can be expressed as follows

$$\begin{aligned}\nabla\Xi &= \mathcal{J}\mathbf{E},\\ \nabla\nabla^\top\Xi &= \mathcal{J}\mathcal{J}^\top + \sum_{i=1}^{n} E_i\mathcal{H}(E_i).\end{aligned} \tag{13}$$

The LM algorithm works by minimizing a multi-variable function represented as the sum of squares of non-linear real-valued functions, as detailed in [21]. Widely accepted as a standard technique for handling non-linear least-squares problems, the LM algorithm has been extensively utilized and studied across various fields within applied mathematics. Depending on the proximity of the current solution to the optimal one, the algorithm's behavior can mimic either method. When the solution is far from optimal, it operates akin to a steepest descent method, converging slowly. Conversely, when closer to the optimal solution, it transitions into a Gauss-Newton method. The search step within this algorithm is determined by calculating

$$\Delta\Theta = (\mathcal{J}\mathcal{J}^\top + \varpi\mathcal{I})^{-1}\nabla\Xi, \tag{14}$$

where the damping factor ϖ and the identity matrix $\mathcal{I}$ are embedded in the optimization process. The steepest descent and Gauss-Newton methods are employed by adjusting the parameter ϖ, when the target amount in a particular step deviates significantly from the optimal value. To achieve this, a substantial value is incrementally applied to ϖ, ensuring that the Hessian matrix is primarily influenced by a diagonal matrix, like the identity matrix.

$$\nabla\nabla^\top\Xi \approx \varpi\mathcal{I}.$$

When the objective and optimal values are closely clustered at a particular step, the Gauss-Newton approximation assigns a small value to ϖ in order to dominate the Hessian matrix. In such situations, we obtain

$$\nabla\nabla^\top\Xi \approx \mathcal{J}\mathcal{J}^\top, \tag{15}$$

The second part in equation (13) disappears. Eq. (15) provides a reliable approximation when E_i or $\mathcal{H}(E_i)$ is small. This situation arises in two cases: when dealing with a small residual problem, or when Ξ exhibits near-linear behavior. It is important to highlight that this model is expected to yield a minimal residue within the optimal range; otherwise, it is not a suitable model.

4 Empirical studies

In this section, we experimentally evaluate the effectiveness of option pricing model calibration by ANN model , whose inputs are Black-Scholes equations (here we call this structure ANN-Black-Scholes) with different strike prices and maturity times. For this purpose, we consider the European call option of the SPX and VIX markets on trading days 2023-1-27 and 2023-1-31, respectively, with various strike prices and maturity dates. The calibrated parameters involved in the Black-Scholes model along with the error measures as mean square error (MSE) and mean absolute error (MAE) in both in sample and out of sample datasets by two calibration methods based on the Black-Scholes and ANN-Black-Scholes pricing models for the SPX and VIX markets are presented in Tables 1 and 2, respectively. It should be noted that the in sample refers to data that was used to train and develop a model, while out of sample refers to data that was not used during the training phase and is used to test the model's performance on new and unseen data. We can see the graphical representation of these results in Figures 2-7. As can be seen from these tables and figures, the calibration of the option pricing model by the ANN-Black-Scholes structure is more accurate and can be used in practice for the pricing of financial options.

Table 1: Assessing the accuracy of the in sample and out of sample dataset obtained by the Black-Scholes and ANN-Black-Scholes models calibrated from the European option price in SPX market on trading day 2023-1-27.

	In sample		Out of sample		
Maturity date	MSE	MAE	MSE	MAE	Parameter estimation
	Black-Scholes model				
2023-1-31	0.7535	0.6222	0.2439	0.4018	$r = 0.1249, \sigma = 0.1397$
2023-2-9	17.8782	3.7194	117.2197	10.6777	$r = 0.0885, \sigma = 0.0739$
2023-3-3	70.5238	7.7238	350.7596	18.4467	$r = -0.1140, \sigma = 0.3098$
	ANN-Black-Scholes model				
2023-3-3	0.6207	0.5892	0.0991	0.1728	$r = 0.3043, \sigma = 0.1228$
2023-2-9	4.4464	1.7502	4.8920	2.1838	$r = 0.1699, \sigma = 0.1779$
2023-3-3	4.2334	1.5310	19.5738	4.2802	$r = 0.1269, \sigma = 0.0003$

The tables and figures presented clearly demonstrate the superior calibration of the option pricing model using the ANN- Black-Scholes structure. When comparing the predicted option prices to the actual market prices, the ANN-Black-Scholes model consistently outperformed traditional Black-Scholes and other pricing models. This level of accuracy is crucial in the financial industry, where small discrepancies in option pricing can have significant financial implications. By incorporating the ANN technology into the Black-Scholes framework, financial institutions and investors can have more confidence in the prices they are using to value options. This

Table 2: Assessing the accuracy of the in sample and out of sample dataset obtained by the Black-Scholes and ANN-Black-Scholes models calibrated from the European option price in VIX market on trading day 2023-1-31.

	In sample		Out of sample		
Maturity date	MSE	MAE	MSE	MAE	Parameter estimation
	Black-Scholes model				
2023-1-31	0.4245	0.5972	0.2043	0.4519	$r = 0.0013, \sigma = 0.6631$
2023-2-9	0.4450	0.6607	0.2255	0.4740	$r = 0.5133, \sigma = 0.5381$
2023-3-3	0.7162	0.5891	0.4751	0.6892	$r = 0.0599, \sigma = 0.2503$
	ANN-Black-Scholes model				
2023-3-3	0.2069	0.4037	0.0338	0.1522	$r = 0.1802, \sigma = 0.5793$
2023-2-9	0.0880	0.2461	0.0118	0.0880	$r = 0.1251, \sigma = 0.0259$
2023-3-3	0.2221	0.4103	0.2961	0.4453	$r = 0.6178, \sigma = 0.1454$

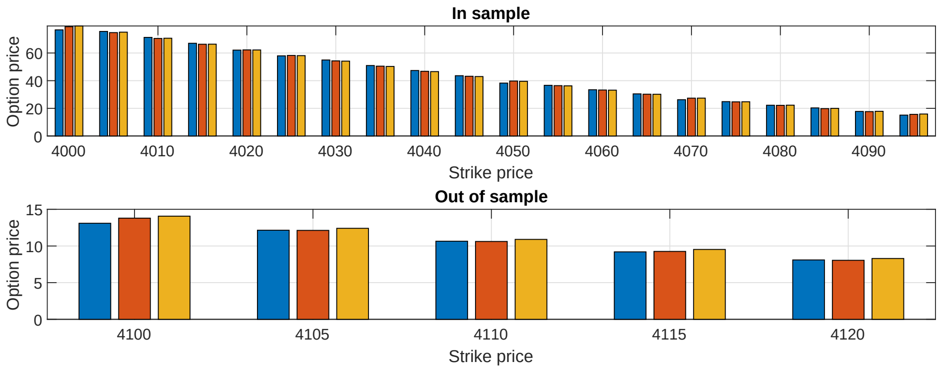

Figure 2: Evaluating accuracy of calibrated Black-Scholes (yellow color) and ANN equipped with Black-Scholes models (orange color) with European option price in SPX market (blue color) from 2023-1-27 with maturity date 2023-1-31.

enhanced accuracy also opens up opportunities for more complex option strategies that may have been previously viewed as too risky due to uncertainties in pricing. With the ANN-Black-Scholes model, traders and investors can have greater certainty in the pricing of these more advanced options, allowing for more sophisticated investment strategies. The results presented in these tables and figures provide strong evidence that the calibration of the option pricing model by the ANN-Black-Scholes structure is not only more accurate but also practical for use in real-world financial applications. This advancement in option pricing technology has the potential to revolutionize the way options are priced and traded in the financial markets.

For American options, which allow for early exercise, our method would need to be modified to account for the additional complexity of determining the optimal exercise strategy. This could potentially be achieved by incorporating decision tree algorithms or reinforcement learning techniques into our hybrid structure. By optimizing the neural network to better capture the dynamics of early exercise, we could improve the accuracy of our calibration for American options. Moreover, for exotic options, which have non-standard payoff structures, our method could be extended by training the neural network on a wider range of exotic option types and payoffs. By incorporating more exotic payoffs into the training data, our neural network could learn to better approximate the pricing of these complex options. Additionally, we could explore the use of Monte Carlo simulations or other numerical methods to better model the behavior of exotic options within our hybrid structure. By continually refining and expanding the capabilities of our method through the incorporation of advanced techniques and data, we believe that our hybrid structure could be extended to effectively calibrate pricing models for a broader range of option types, including American and exotic options.

We believe that in the future, ML models built on recurrent neural network (RNN) or convolutional neural network (CNN) could replace the neural network model used in the current calibration framework. There are several potential benefits to experimenting with different neural network architectures such as RNN or CNNs for calibrating European option pricing models. One benefit is that these architectures may be better suited to capturing non-linear relationships and complex patterns in the data. RNNs, for example, are particularly good at capturing sequential dependencies in time-series data, which could be relevant for modeling the dynamics of option prices over time. CNNs, on the other hand, are adept at capturing spatial patterns in data, which could be useful for identifying spatial correlations in option pricing data. Another benefit is that experimenting with different neural network architectures could potentially improve the accuracy and performance of the pricing model. Each architecture has its own strengths and weaknesses, and by exploring different options, researchers may be able to find a model that better fits the specific characteristics of the data they are working with. Additionally, using a hybrid approach that combines the strengths of different neural network architectures with the Black-Scholes model could lead to a more robust and accurate pricing model. For example, the Black-Scholes model may be good at capturing the underlying fundamentals of option pricing, while a neural network could help to improve the model's ability to capture complex patterns and dynamics in the data. Exploring different neural network architectures for calibrating option pricing models could open up new possibilities for improving the accuracy and performance of these models, and may lead to more reliable pricing estimates in the long run.

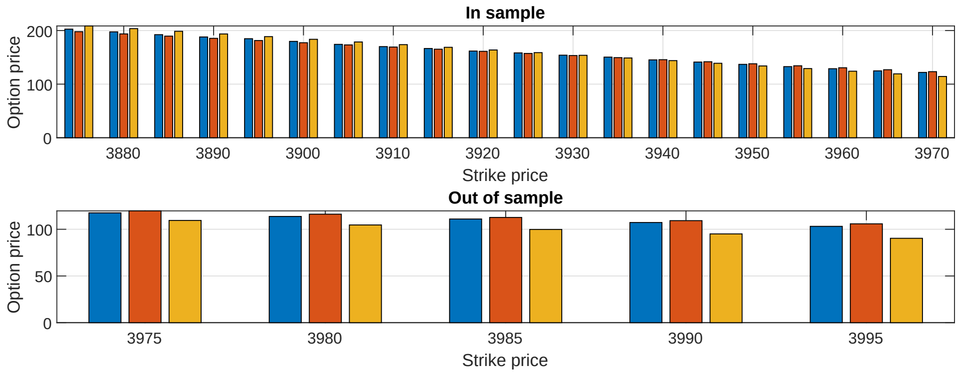

Figure 3: Evaluating accuracy of calibrated Black-Scholes (yellow color) and ANN equipped with Black-Scholes models (orange color) with European option price in SPX market (blue color) from 2023-1-27 with maturity date 2023-2-9.

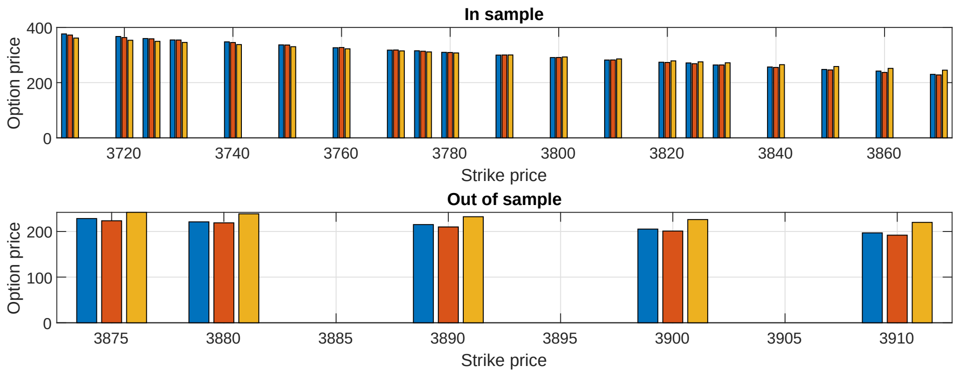

Figure 4: Evaluating accuracy of calibrated Black-Scholes (yellow color) and ANN equipped with Black-Scholes models (orange color) with European option price in SPX market (blue color) from 2023-1-27 with maturity date 2023-3-3.

5 Conclusion

In this study, we have proposed an optimized artificial neural network approach for calibrating European option pricing models when its inputs were Black-Scholes models with various strike prices and maturity dates. By training the neural network on a dataset of option prices using LM optimization algorithm, we were able to effectively capture the complex relationships between the input parameters and the corresponding option prices. The empirical results on European call option of SPX and VIX markets demonstrated that the optimized neural network significantly outperforms traditional calibration methods in terms of accuracy and efficiency. The neural network was able to effectively learn the underlying patterns in the

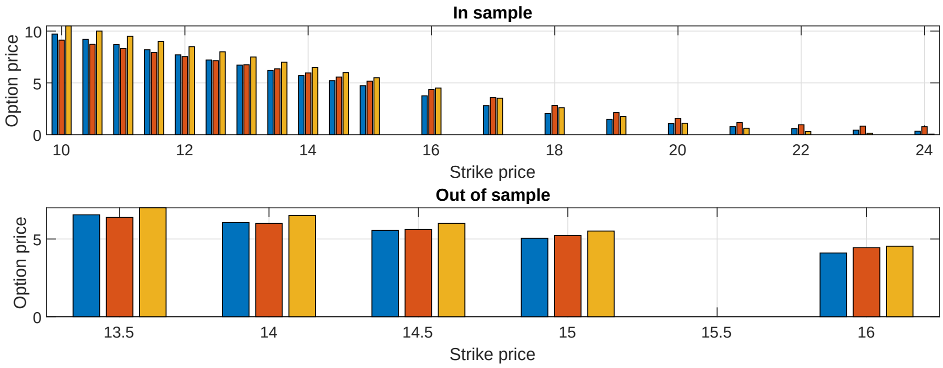

Figure 5: Evaluating accuracy of calibrated Black-Scholes (yellow color) and ANN equipped with Black-Scholes models (orange color) with European option price in VIX market (blue color) from 2023-1-31 with maturity dates 2023-2-8 and 2023-2-15.

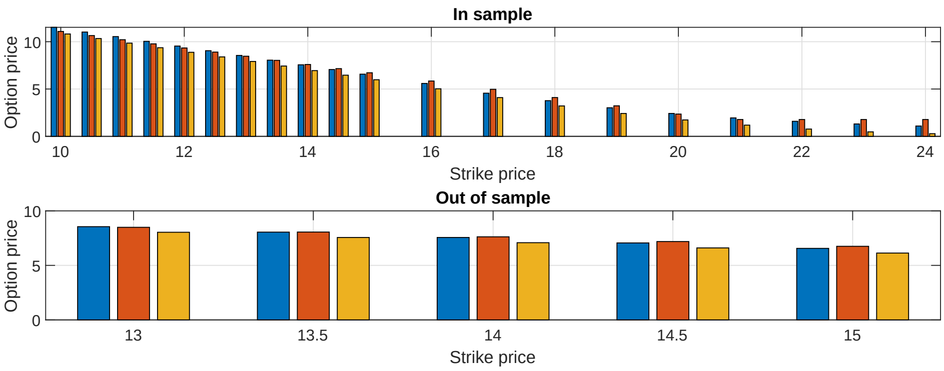

Figure 6: Evaluating accuracy of calibrated Black-Scholes (yellow color) and ANN equipped with Black-Scholes models (orange color) with European option price in VIX market (blue color) on trading day 2023-1-31 with maturity dates 2023-2-22 and 2023-3-1.

Black-Scholes model and accurately predict option prices for a wide range of strike prices and maturity dates. Our findings suggested that artificial neural networks can provide a valuable tool for calibrating option pricing models and improving the accuracy of pricing calculations in the financial industry. This approach has the potential to enhance risk management practices and decision-making processes for investors and traders. Further research could explore the application of neural networks in other areas of options pricing and financial modeling to continue advancing the field.

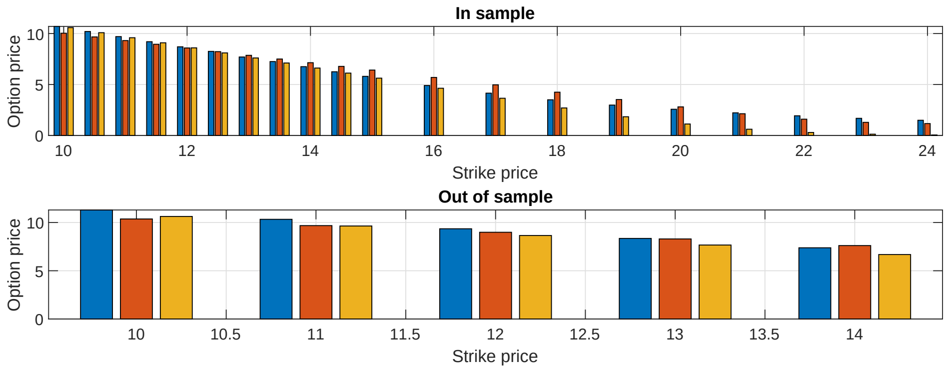

Figure 7: Evaluating accuracy of calibrated Black-Scholes (yellow color) and ANN equipped with Black-Scholes models (orange color) with European option price in VIX market (blue color) from 2023-1-31 with maturity dates 2023-3-22 and 2023-4-19.

Bibliography

[1] Bakshi, G., Cao, C., & Chen, Z. (1997). Empirical performance of alternative option pricing models. *The Journal of finance*, 52(5), 2003-2049.

[2] Christoffersen, P., & Jacobs, K. (2004). Which GARCH model for option valuation?. *Management science*, 50(9), 1204-1221.

[3] Christoffersen, P., Jacobs, K., & Ornthanalai, C. (2012). GARCH option valuation: theory and evidence.

[4] Cont, R., & Tankov, P. (2004). Financial modelling with jump processes (No. 22631). Chapman & Hall; CRC.

[5] Dua, V. (2011). An artificial neural network approximation based decomposition approach for parameter estimation of system of ordinary differential equations. *Computers & Chemical Engineering*, 35(3), 545-553.

[6] Griffin, J. E., & Steel, M. F. (2006). Inference with non-Gaussian Ornstein-Uhlenbeck processes for stochastic volatility. *Journal of Econometrics*, 134(2), 605-644.

[7] Günay, M. E. (2016). Forecasting annual gross electricity demand by artificial neural networks using predicted values of socio-economic indicators and climatic conditions: Case of Turkey. *Energy Policy*, 90, 92-101.

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

[9] Itkin, A. (2019). Deep learning calibration of option pricing models: some pitfalls and solutions. arXiv preprint arXiv:1906.03507.

[10] Jamili, E., & Dua, V. (2021). Parameter estimation of partial differential equations using artificial neural network. *Computers & Chemical Engineering*, 147, 107221.

[11] Kaeck, A., & Alexander, C. (2012). Volatility dynamics for the S&P 500: Further evidence from non-affine, multi-factor jump diffusions. *Journal of Banking & Finance*, 36(11), 3110-3121.

[12] Lim, K. G., & Zhi, D. (2002). Pricing options using implied trees: Evidence from FTSE100 options. *Journal of Futures Markets: Futures, Options, and Other Derivative Products*, 22(7), 601-626.

[13] Mehrdoust, F., & Noorani, I. (2023). Implied higher order moments in the Heston model: a case study of S& P500 index. *Decisions in Economics and Finance*, 46(2), 477-504.

[14] Mehrdoust, F., Noorani, I., & Belhaouari, S. B. (2023). Forecasting Nordic electricity spot price using deep learning networks. *Neural Computing and Applications*, 35(26), 19169-19185.

[15] Mehrdoust, F., Noorani, I., & Kanniainen, J. (2024). Valuation of option price in commodity markets described by a Markov-switching model: A case study of WTI crude oil market. *Mathematics and Computers in Simulation*, 215, 228-269.

[16] Mehrdoust, F., Noorani, I., & Fallah, S. (2022). Markov regime-switching Heston model with CIR model framework and pricing VIX and S&P500 American put options. *Mathematical Reports*, 24(74), 781-806.

[17] Mehrdoust, F., & Noorani, I. (2019). Pricing S& P500 barrier put option of American type under Heston-CIR model with regime-switching. *International Journal of Financial Engineering*, 6(02), 1950014.

[18] Mehrdoust, F., & Noorani, M. (2023). Prediction of cryptocurrency prices by deep learning models: A case study for Bitcoin and Ethereum. *International Journal of Financial Engineering*, 10(04), 2350032.

[19] Mehrdoust, F., Noorani, I., & Hamdi, A. (2023). Two-factor Heston model equipped with regime-switching: American option pricing and model calibration by Levenberg-Marquardt optimization algorithm. *Mathematics and Computers in Simulation*, 204, 660-678.

[20] Mehrdoust, F., & Noorani, I. (2023). Valuation of spark-spread option written on electricity and gas forward contracts under two-factor models with non-Gaussian Lévy processes. *Computational Economics*, 61(2), 807-853.

[21] Moré, J. J. (2006, August). The Levenberg-Marquardt algorithm: implementation and theory. In Numerical analysis: proceedings of the biennial Conference held at Dundee, June 28-July 1, 1977 (pp. 105-116). Berlin, Heidelberg: Springer Berlin Heidelberg.

[22] Noorani, I., & Mehrdoust, F. (2022). Parameter estimation of uncertain differential equation by implementing an optimized artificial neural network. *Chaos, Solitons & Fractals*, 165, 112769.

[23] Platon, R., Dehkordi, V. R., & Martel, J. (2015). Hourly prediction of a building's electricity consumption using case-based reasoning, artificial neural networks and principal component analysis. *Energy and Buildings*, 92, 10-18.

[24] Roberts, G. O., Papaspiliopoulos, O., & Dellaportas, P. (2004). Bayesian inference for non-Gaussian OrnsteinUhlenbeck stochastic volatility processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(2), 369-393.

[25] Spiliopoulos, K. (2009). Method of Moments Estimation of Ornstein-Uhlenbeck Processes Driven by General Lévy Process. *In: Annales de lISUP, Institut de statistique de lUniversitt'e de Paris* 53: 3-18.

[26] Wang, X., He, X., Bao, Y., & Zhao, Y. (2018). Parameter estimates of Heston stochastic volatility model with MLE and consistent EKF algorithm. *Science China Information Sciences*, 61, 1-17.

[27] Witten I. H., & Frank, E. (2005). Data mining: practical machine learning tools and techniques (Data management systems). *San Mateo: Morgan Kaufmann.*

[28] Wu, Y., Hu, J., & Zhang, X. (2019). Moment estimators for the parameters of Ornstein-Uhlenbeck processes driven by compound Poisson processes. *Discrete Event Dynamic Systems*, 29, 57-77.

[29] Xie, Z., Kulasiri, D., Samarasinghe, S., & Rajanayaka, C. (2007). The estimation of parameters for stochastic differential equations using neural networks. *Inverse Problems in Science and Engineering*, 15(6), 629-641.

[30] Zhang, S., Zhang, X., & Sun, S. (2006). Parametric estimation of discretely sampled Gamma-OU processes. *Science in China Series A: Mathematics*, 49, 1231-1257.

How to Cite: Farshid Mehrdoust[1], Maryam Noorani[2], *Calibration of European option pricing model using a hybrid structure based on the optimized artificial neural network and Black-Scholes model*, Journal of Mathematics and Modeling in Finance (JMMF), Vol. 4, No. 1, Pages:67–82, (2024).