Black-Scholes Option Pricing using Machine Learning

Shreyan Sood
Department of Applied Mathematics
Delhi Technological University
New Delhi, India
shreyansood_2k18mc110@dtu.ac.in

Tanmay Jain
Department of Applied Mathematics
Delhi Technological University
New Delhi, India
tanmayjain_2k18mc116@dtu.ac.in

Nishant Batra
Department of Applied Mathematics
Delhi Technological University
New Delhi, India
nishantbatra_2k18mc073@dtu.ac.in

***Abstract*—The main objective of this paper is to explore the effectiveness of machine learning models in predicting stock option prices benchmarked by the Black-Scholes Model. We employ the following 4 machine learning models - Support Vector Machine (SVM), Extreme Gradient Boosting (XGB), Multilayer Perceptron (MLP) and Long Short Term Memory (LSTM), trained using 2 different set of input features, to predict option premiums based on the S&P 500 Apple Stock (APPL) option chain historical data from 2018 and 2019. Statistical analysis of the results show that LSTM is the best out of the chosen model for pricing both Call and Put options. Further analysis of the predictions based on Moneyness and Maturity show consistency of results with the expected behaviour that validates the effectiveness of the prediction model.**

Keywords—Option Pricing, Option Greeks, Volatility, Black-Scholes Model, Machine Learning, LSTM, Moneyness

I. Introduction

The Black-Scholes Model is one of the most fundamental and widely used financial models for pricing stock option premiums. However, due to the standard limitations and assumptions of the model, it is considered to be just a useful approximation tool or a robust framework for other models to build upon. Most research studies that attempt to discern the relevance of the Black-Scholes Model in real world scenarios conclude that the assumption of constant underlying volatility over the life of the derivative is the biggest contributing factor for the empirical inaccuracy of the model. Modifications based on the concepts of Stochastic Volatility and Jump Diffusion are widely implemented in the field of Financial Mathematics to correct the shortcomings of the Black-Scoles Model.

The high dimensionality and flexibility of factors upon which option premiums depend makes the task of accurately predicting them extremely complex. Recently, the concept of Machine Learning, specifically, time-series forecasting using predictive models is finding much application in the field of Finance. In our approach to provide a solution for predicting option premiums accurately, we implement certain Machine Learning Models designed with the intent to effectively build upon and outperform the Black-Scholes Model while using the same set of input parameters and subsequently calculated option greeks. This approach of using option greeks as training input for option pricing prediction models is relatively unexplored and our research contributes to the scarce amount of existing literature which utilizes it. We compare and explore the behaviours and performance of different models with the benchmark predictions obtained from the Black-Scholes Model. We perform a comprehensive comparative statistical analysis of the best obtained results separately for call and put based on the moneyness and maturity values.

II. Literature Review

Drucker et al [1] introduce a new regression technique, SVR, based on Vapnik's concept of support vectors. S. Hochreiter and Schmidhuber [2] establish a new type 3 gated RNN known as LSTM that is efficient at learning to store information over extended time periods. Chen and Guestrin [3] describe an optimized tree boosting system known as XGBoost, a sophisticated machine learning algorithm that is popularly used to achieve state-of-the-art results for different problem statements. Hutchinson et al [4], utilize a non-parametric approach to option pricing using an ANN for the first time as a more accurate and computationally efficient alternative. Gencay and Salih [5] show how mispricing in the Black-Scholes option prices is greater for deeper out-of-the-money options compared to near out-of-the-money options when options are grouped by Moneyness. Their research indicates that the mispricing worsens with increase in volatility. They present the conclusion that the Black-Scholes Model is not an optimal tool for pricing options with high volatility while feed forward neural networks have a lot more success in such situations. Gençay, Gradojevic and Selçuk [6] dive deep into Modular Neural Networks (MNNs) and provide an insight of how they can overcome the shortcomings of Black-Scholes Model, MNN is used to decompose the data into modules as per Moneyness and Maturity and each module is estimated independently. Palmer [7] implements a neural network that uses a novel hybrid evolutionary algorithm based on particle swarm optimization and differential evolution to solve the problem of derivative pricing. Culkin & Das [8] provide an overview on Neural Networks, their basic structure and their use in the field of finance, specifically for option pricing. Shuaiqiang, Oosterlee and Bohté [9] utilize an ANN solver for pricing options and computing volatilities with the aim of accelerating corresponding numerical methods. Ruf and Wang [10] deeply look at the literature on option pricing using neural networks and discuss the appropriate performance measures and input features. One key takeaway from their paper is the reason why chronological partitioning of the option dataset is better than random partitioning which might lead to data leakage.

III. Methods

A. Dataset and Features

We use the S&P 500 Apple (AAPL) stock option chain historical data from 2018 and 2019. After the necessary data cleaning and preprocessing, we compile 2 separate chronological datasets for Call and Put Options with the Call Option data containing about 2,77,000 rows and the Put Option data containing about 2,42,000 rows. Each row of data contains the closing values of the Option Premium (C for Call, P for Put), Underlying Stock Price (S), Strike Price

(K), Implied Volatility (σ_I) and Time to Expiration in years (t), as well as the values of the following Option Greeks - Delta (δ), Gamma (γ), Theta (θ), Vega (ν) and Rho (ρ). The Option Premiums serve as our output ground truth values whereas the rest serve as our input features. The 4 machine learning models are trained using 2 different sets of input features:

1. Set 1 which excludes option greeks i.e. it contains only 4 features - S, K, t and σ_I.
2. Set 2 that includes option greeks i.e. it contains all 9 features - S, K, t, σ_I, δ, γ, θ, ν and ρ.

Since the corresponding information regarding the Risk-Free Interest Rates (r) is unavailable, it is not used as an input feature. For the purpose of calculating the Black-Scholes Option Premiums, the average value of the U.S. 1 Year Treasury Rate across 2018 and 2019 is taken as R. Both Call and Put datasets are split into a 70:30 ratio chronologically for the purpose of generating training and testing datasets.

B. Black-Scholes Model

The Black-Scholes-Merton (BSM) Model was introduced in 1973, and provided a straight closed-form solution for pricing European Options. However, the model is based on certain assumptions that do not hold water in real market scenarios, for instance the assumption that the underlying asset price follows a Geometric Brownian Motion and that volatility of underlying prices is constant. For our purposes, the premiums calculated using the BSM used Implied Volatility instead of the Annualized Volatility and hence form an ideal benchmark as the only error incurred arises from the unavailability of the exact Risk Free Interest Rate values for each of the individual options.

Using the Black-Scholes equation, the premium for a call option can be calculated as:

$$C = S\,N(d1) \; - \; K\,e^{-rt}\,N(d2)$$

Similarly, the premium for a put option can be calculated as:

$$P = K\,e^{-rt}\,N(-\,d2) \; - \; S\,N(-\,d1)$$

where

S: underlying stock price

t: time until option exercise (in years)

K: strike price of the option contract

r: risk-free interest rate available in the market

N: cumulative probability function for standard normal distribution

e exponential term

and d1, d2 are calculated as follows:

$$d_1 = \frac{ln(\frac{S}{K})+(r+\frac{\sigma^2}{2})t}{\sigma\sqrt{t}}$$

$$d_2 = d_1 - \sigma\sqrt{t}$$

where

σ: annualized volatility of the stock

ln: natural log

Option Greeks are used to label various kinds of risks involved in options. Each "Greek" is the result of a flawed assumption of the option's relation with a specific underlying variable. They are commonly used by options traders to comprehend how their Profit and Loss will behave, as prices vary. In this paper, we have used five option greeks, namely Delta (δ), Gamma (γ), Theta (θ), Vega (ν) and Rho (ρ). Delta (δ) is the price sensitivity of the option, relative to the underlying asset. Theta (θ) is the time sensitivity or the rate of change in the option price, with time. It is also known as option's time decay. It demonstrates the amount with which the price of an option would reduce, with diminishing time to expiry. Gamma (γ), also known as the second derivative price sensitivity, is the rate of change of option's Delta with respect to the underlying asset's price. Vega (ν) gives the option's sensitivity to volatility. It shows the rate of change of option's value with that of the underlying asset's implied volatility. Rho (ρ) represents the sensitivity to the interest rate. It is the rate of change of the option's value with respect to that of 1% change in interest rate. In essence, the Greeks represent gradient information for the Black-Scholes Model and when fed as additional input parameters, can help the Machine Learning Models to better fit the training data and accurately capture the underlying relation.

C. Machine Learning Models

1. Support Vector Machine (SVM):

SVMs are supervised learning models based on the margin maximization principle and risk minimization by finding the optimal fit hyperplane. They are typically used for non-probabilistic linear classification but can be used for regression as well. Specifically, Support Vector Regression (SVR) is the method based on SVMs that is used for high dimensional nonlinear regression analysis. We use a SVR with Radial Basis Function kernel for predictions. SVR is chosen because of its high accuracy of generalization of high dimensional data and high robustness to outliers even though the model underperforms in cases of large datasets. Standard normalization of data is performed prior to training and hyperparameter tuning and regularization is carried out using the Grid Search Method to minimize overfitting.

2. Extreme Gradient Boosting (XGB):

XGB is a software library which provides an optimized distributed gradient boosting framework that is designed to be highly efficient. Gradient Boosting is a technique that produces a single strong prediction model in the form of an ensemble of weaker iterative models such as decision trees. XGB is chosen because of its high speed and flexibility with which it handles large, complex datasets even though it is sensitive to outliers. Regularization and Early Stopping techniques are utilized to minimize overfitting and hyperparameter tuning is carried out using the Grid Search Method.

3. Multilayer Perceptron (MLP):

MLPs are categorised as feedforward Artificial Neural Networks (ANNs) that utilizes the back-propagation technique for training. MLPs are universal function

approximators and hence are ideal for generalizing mathematical models by regression analysis. In our implementation, we use a sequential MLP with 5 fully connected hidden layers that use a variety of different activation functions and optimized hyperparameters. To minimize overfitting, we utilize dropout and batch normalization layers throughout the dense layers and also monitor test dataset metrics for early stopping.

4. Long Short Term Memory (LSTM):

LSTMs are categorised as Recurrent Neural Network (RNN) architectures which, unlike standard feedforward neural networks, have feedback connections. A common LSTM unit consists of 3 gates - an input gate, a forget gate, and an output gate, which are used in tandem to regulate the flow of information to the LSTM cell which remembers the values over arbitrary time intervals. LSTMs are optimally used to process time-series data, as time-lags of unknown duration can be obtained between important events in a time-series. To minimize overfitting, we implement dropouts, batch normalization and early stopping that monitors testing dataset loss. Activation functions for every layer and other hyperparameters were efficiently tuned in order to minimize training loss.

D. Analysis

We analyze the predictions of the BS Model on both Call and Put Option datasets and compare it with our 4 regression models, each trained on the 2 sets of input parameters, which gives a total of 9 models for comparison. Out of these 9 models, we pick the model with the best results and further breakdown its performance based on the following 2 parameters:

1. Moneyness:

Moneyness describes the inherent monetary value of an option's premium in the market. For our purposes we divide the Underlying Stock Price (S) by the Strike Price (K) and categorize the options as being In-the-Money, Out-of-the-Money or At-the-money. The following table shows moneyness for different options is defined quantitatively.

TABLE I. MONEYNESS DISTRIBUTION

S. No.	*Moneyness Type*	*Value of S/K*	
		Call	***Put***
1	In-the-Money	*>1.05*	< 0.95
2	Out-of-the-Money	<0.95	≥ *1.05*
3	At-the-Money	0.95 ≤ *S/K* ≤ 1.05	0.95 ≤ *S/K* ≤ 1.05

2. Maturity:

Maturity represents the time to expiration in years for the option to be exercised (t). We consider options with t < 0.1 to be Short-Term, 0.1 ≤ t ≤ 0.5 to be Medium-Term and t > 0.5 to be Long-Term.

IV. RESULTS

The following regression metrics are used in the evaluation of models:

1. Mean Absolute Error (MAE):

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |x - y|$$

2. Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (x - y)^2}$$

where

N: number of observations

x: actual value

y: predicted value

A combination of the metrics is taken as the main loss function for training and testing purposes. The following 2 tables represent the results for the metrics obtained from the testing dataset predictions of call options and put options respectively.

TABLE II. CALL OPTION TEST DATASET METRICS

S. No.	*Model*	*Metrics*	
		MAE	***RMSE***
1	BS	2.42	5.32
2	SVM	6.25	9.04
3	SVM_Greeks	9.58	11.81
4	XGB	8.49	15.37
5	XGB_Greeks	5.08	11.81
6	MLP	2.74	4.52
7	MLP_Greeks	3.38	5.35
8	**LSTM**	**2.02**	**3.74**
9	LSTM_Greeks	3.31	4.98

TABLE III. PUT OPTION TEST DATASET METRICS

S. No.	*Model*	*Metrics*	
		MAE	***RMSE***
1	BS	1.96	5.52
2	SVM	3.76	6.30
3	SVM_Greeks	5.21	8.72
4	XGB	4.74	8.50
5	XGB_Greeks	4.54	8.54
6	MLP	1.68	2.95
7	MLP_Greeks	4.59	6.71
8	**LSTM**	**1.40**	**2.77**
9	LSTM_Greeks	1.99	3.65

In both Call and Put Option test dataset predictions, LSTM performs the best and gives the minimum MAE as well as the minimum RMSE. The model also outperforms the benchmark BS Model in both the datasets.

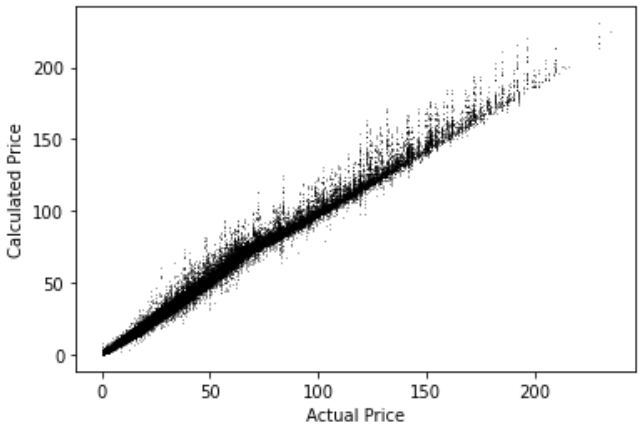


Fig. 1(a). Calculated (Predicted) Price vs Actual Price Plot for LSTM Call Option Test Dataset Predictions

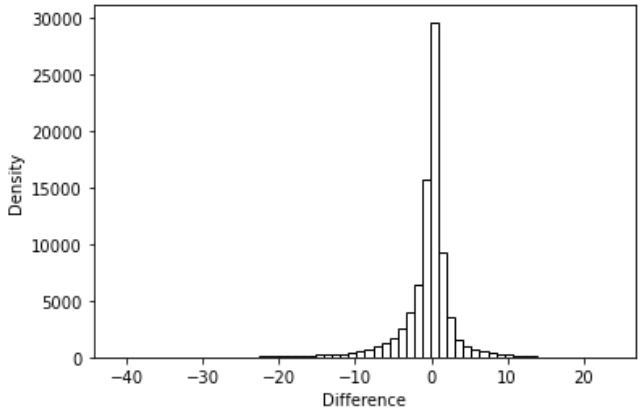


Fig. 1(b). Error Density Plot for LSTM Call Option Test Dataset Predictions

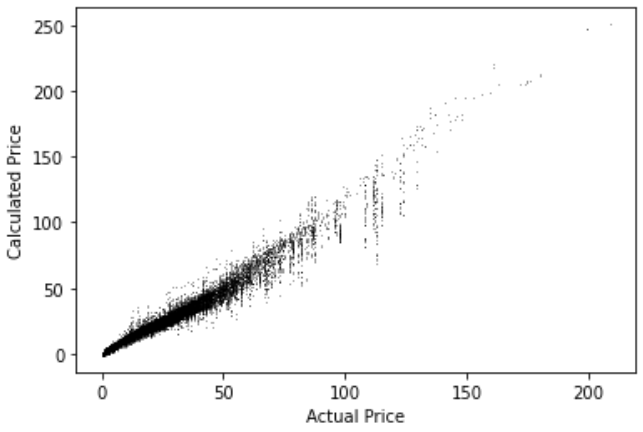


Fig. 2(a). Calculated (Predicted) Price vs Actual Price Plot for LSTM Put Option Test Predictions

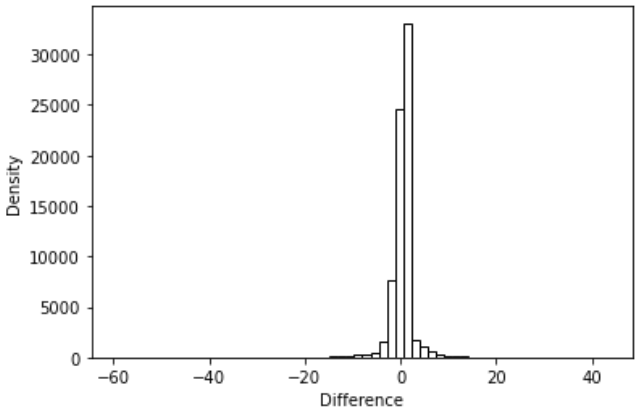


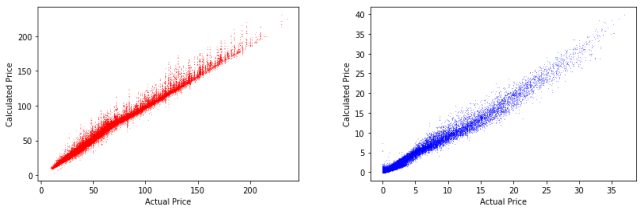
Fig. 2(b). Error Density Plot for LSTM Put Option Test Dataset Predictions

The following table represents the results of the best model obtained (LSTM for both Call and Put) when the options are grouped by moneyness.

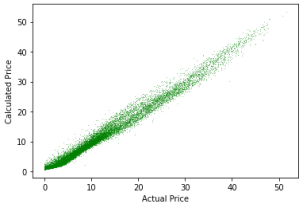
TABLE IV. LSTM Test Dataset Metrics Grouped By Moneyness

S. No.	*Option Type*	*Moneyness Type*	*Metrics*	
			MAE	***RMSE***
1	Call	In-the-Money	3.59	5.43
2	**Call**	**Out-of-the-Money**	**0.55**	**0.87**
3	Call	At-the-Money	1.06	1.37
4	Put	In-the-Money	3.93	6.55
5	**Put**	**Out-of-the-Money**	**0.84**	**1.02**
6	Put	At-the-Money	1.43	2.03

In both Call and Put Option LSTM test dataset predictions, Out-of-the-Money options give the minimum MAE as well as the minimum RMSE, followed by At-the-Money and then In-the-Money options.

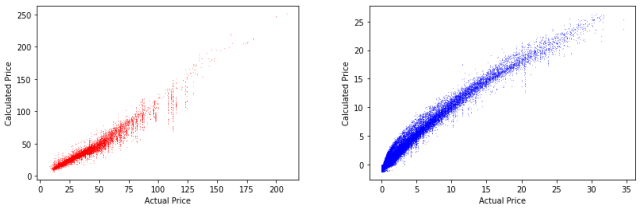


a) In-the-Money b) Out-of-the-Money

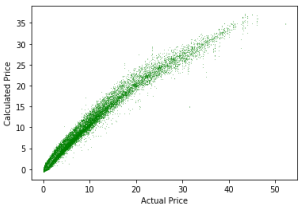


c) At-the-Money

Fig. 3(a). Calculated Price vs Actual Price Plots for LSTM Call Option Test Predictions Grouped by Moneyness



a) In-the-Money b) Out-of-the-Money



c) At-the-Money

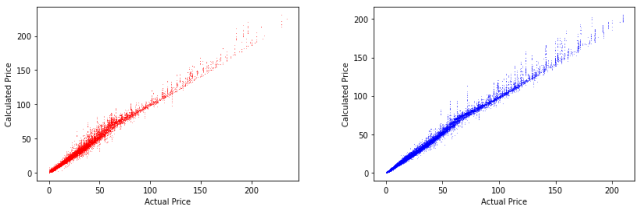
Fig. 3(b). Calculated Price vs Actual Price Plots for LSTM Put Option Test Predictions Grouped by Moneyness

The following table represents the results of the LSTM when the options are grouped by maturity.

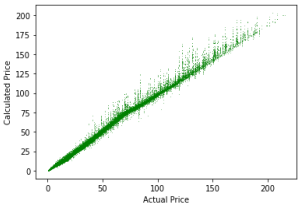
TABLE V. LSTM Test Dataset Metrics Grouped By Maturity

S. No.	*Option Type*	*Moneyness Type*	*Metrics*	
			MAE	***RMSE***
1	**Call**	**Short-Term**	**1.73**	**3.57**
2	Call	Medium-Term	1.87	3.50
3	Call	Long-Term	2.39	4.06
4	**Put**	**Short-Term**	**1.16**	**1.99**
5	Put	Medium-Term	1.33	2.26
6	Put	Long-Term	1.68	3.67

In both Call and Put Option LSTM test dataset predictions, Short-Term options give the minimum MAE as well as the minimum RMSE, followed by Medium-Term and then Long-Term options.

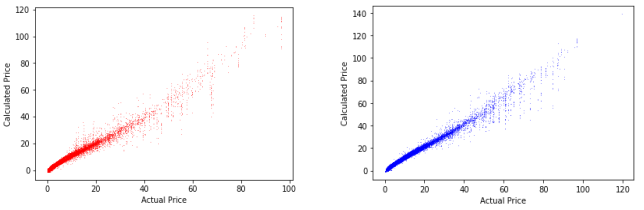


a) Short-Term b) Medium-Term

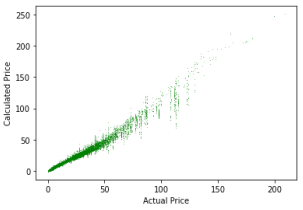


c) Long-Term

Fig. 4(a). Calculated Price vs Actual Price Plots for LSTM Call Option Test Predictions Grouped by Maturity



a) Short-Term b) Medium-Term



c) Long-Term

Fig. 4(b). Calculated Price vs Actual Price Plots for LSTM Put Option Test Predictions Grouped by Maturity

V. Conclusions

The results from Tables II and III show that LSTM is the best model for option pricing for both Call and Put Options. SVM and XGB perform worse than BS i.e. these models undefit and are unable to sufficiently capture the underlying relation for option premium predictions. Results also show that Option Greeks when included as additional input features actually produce worse final test metrics in every single model even though they improved the training metrics in some cases. This is a new finding as most of the literature about using option greeks as input features for option pricing mostly report an improvement in performance. Along with LSTM, MLP also outperforms BS. The metrics show similar distribution and trends for both types of options which implies that the training of the models is impartial to Option Type.

The results from Table IV show that when grouped by Moneyness, Out-of-the-Money options give significantly better metrics followed by At-the-Money and then In-the-Money samples. This behaviour is to be expected as lower Moneyness directly lowers the value of the option premium. Hence, since the magnitude of premium value decreases as we move from Into-the-Money to Out-of-the-Money options, so does the magnitude of our error metrics.

Similarly, the results from Table V show that when grouped by Maturity, Short-Term options give better metrics followed by Medium-Term and then Long Term options. This behaviour is also to be expected as a lower Maturity directly lowers the option premiums. Hence, the magnitude of our error metrics decreases as we move from Long-Term to Short-Term options.

The analysis of the option predictions on the basis of Moneyness and Maturity validate that the LSTM prediction model is able to efficiently and accurately predict option prices in a real world market scenario.

References

[1] Drucker, Harris, Christopher J. C. Burges, Linda Kaufman, Alex Smola and Vladimir Naumovich Vapnik. “Support Vector Regression Machines.” NIPS. (1996).

[2] Hochreiter, Sepp and Jürgen Schmidhuber. “Long Short-Term Memory.” Neural Computation 9, 1735-1780. (1997).

[3] Chen, Tianqi and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System.” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016).

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

[5] Ramazan Gencay and Aslihan Salih, “Degree of mispricing with the Black-Scholes model and nonparametric cures”, Annals of Economics andFinance, Vol. 4, pp. 73-101. (2003).

[6] Nikola Gradojevic, Ramazan Gencay, and Dragan Kukolj. Option pricing with modular neural networks. IEEE transactions on neural networks, 20(4):626–637. (2009).

[7] Palmer, Samuel. “Evolutionary algorithms and computational methods for derivatives pricing.” (2019).

[8] Robert Culkin and Sanjiv R Das. “Machine learning in finance: the case of deep learning for option pricing”. Journal of Investment Management, 15(4):92–100. (2017).

[9] Liu, Shuaiqiang, Cornelis W. Oosterlee and Sander M. Bohté. “Pricing options and computing implied volatilities using neural networks.” (2019).

[10] Ruf, Johannes and Weiguan Wang. “Neural networks for option pricing and hedging: a literature review.” Journal of Computational Finance (2020).