Black-Scholes Option Pricing using Machine Learning

Shreyan Sood[1[0000-0002-6048-7565]], Tanmay Jain[1[0000-0002-5633-4507]], Nishant Batra[1[0000-0003-4360-2864]] and H.C. Taneja[1[0000-0002-5545-0987]]

[1] Department of Applied Mathematics, Delhi Technological University, Delhi, India
shreyansood@gmail.com, tanmayjain_2k18mc116@dtu.ac.in, nishantbatra2000@gmail.com, hctaneja@dce.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 have employed the following four machine learning models - Support Vector Machine, Extreme Gradient Boosting, Multilayer Perceptron and Long Short-Term Memory, trained using two different set of input features, to predict option premiums based on the S&P 500 Apple stock option chain historical data from 2018 and 2019. Statistical analysis of the results show that Long Short-Term Memory is the best out of the chosen models for pricing both Call and Put options. Further analysis of the predictions based on moneyness and maturity demonstrated consistency of results with the expected behaviour that validated the effectiveness of the prediction model.

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

1 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 attempted 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 was 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-Scholes 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 model is finding much applications in the field of finance. In our approach to provide a solution for predicting option premiums accurately, we have implemented 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 have applied this approach. We have compared and explored the behaviours and performance of different models with the benchmark predictions obtained from the Black-Scholes Model. In addition to commonly used regression metrics, we have also calculated the Kullback-Leibler Divergence [1], also known as Relative Entropy, to quantify the statistical difference between the probability distributions of the predicted values and the actual values. A comprehensive comparative statistical analysis of the best obtained results has been performed separately for call and put options based on the moneyness and maturity values.

2 Literature Review

Drucker et al [2] introduced a new regression technique, SVR, based on Vapnik's concept of support vectors. Hochreiter and Schmidhuber [3] established a new type 3 gated RNN known as LSTM that was efficient at learning to store information over extended time periods. Chen and Guestrin [4] described 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 [5], utilized 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 [6] showed how mispricing in the Black-Scholes option prices was greater for deeper out-of-the-money options compared to near out-of-the-money options when options are grouped by moneyness. Their research indicated that the mispricing worsened with increase in volatility. They concluded that the Black-Scholes Model was not an optimal tool for pricing options with high volatility while feed forward neural networks had a lot more success in such situations. Gençay et al [7] dove deep into Modular Neural Networks (MNNs) and provided an insight of how they could overcome the shortcomings of Black-Scholes Model, MNN was used to decompose the data into modules as per moneyness and maturity and each module was estimated independently. Palmer [8] implemented a neural network that used a novel hybrid evolutionary algorithm based on particle swarm optimization and differential evolution to solve the problem of derivative pricing. Culkin and Das [9] provided an overview on neural networks, their basic structure and their use in the field of finance, specifically for option pricing. Shuaiqiang et al [10] utilized an artificial neural network (ANN) solver for pricing options and computing volatilities with the aim of accelerating corresponding numerical methods. Ruf and Wang [11] deeply looked at the literature on option pricing using neural networks and discussed the appropriate performance measures and input features. One key takeaway from their paper was the reason why chronological partitioning of the option dataset was better than random partitioning which might lead to data leakage.

The paper is formulated as follows. Section 2 gives the literature review. In Section 3 we describe the dataset employed, Black-Scholes model and the four machine learning models. Section 4 and 5 give respectively the analysis and results obtained. Section 6 gives the conclusion. Section 7 and 8 list the references and figures respectively.

3 Materials and Methods

3.1 Dataset and Features

We have used the S&P 500 Apple (AAPL) stock option chain historical data from 2018 and 2019. After the necessary data cleaning and pre-processing, we compiled 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 contained 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 served as our output ground truth values whereas the rest served as our input features. The four machine learning models were trained using two different sets of input features:

1. Set 1 which excluded Option Greeks i.e., it contained only four features - S, K, t and σ_I.
2. Set 2 that included Option Greeks i.e., it contained all nine features - S, K, t, σ_I, δ, γ, θ, ν and ρ.

Since the corresponding information regarding the risk-free interest rates (r) was unavailable, it was not used as an input feature. For the purpose of calculating the Black-Scholes option premiums, the average value of the U.S. one year treasury rate across 2018 and 2019 was taken as R. Both call and put datasets were split into a 70:30 ratio chronologically for the purpose of generating training and testing datasets.

3.2 Black-Scholes Model (BSM)

The Black-Scholes 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 generated implied volatility, instead of the annualized volatility, hence formed an ideal benchmark as the only error occurs 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:

$$C = S\,N(d_1) - K\,e^{-rt}\,N(d_2) \tag{1}$$

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

$$P = K\,e^{-rt}\,N(-d_2) - S\,N(-d_1) \tag{2}$$

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
and d_1, d_2 are calculated as follows:

$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)t}{\sigma\sqrt{t}} \tag{3}$$

$$d_2 = d_1 - \sigma\sqrt{t}\ ; \tag{4}$$

here σ is the annualized volatility of the stock

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 better fit the training data and accurately capture the underlying relation.

3.3 Machine Learning Models

Support Vector Machine (SVM). These 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 used SVR with radial basis function kernel for predictions. SVR was chosen because of its high accuracy of generalization of high dimensional data and high robustness to outliers even though the model underperformed in cases of large datasets.

Standard normalization of data was performed prior to training and hyperparameter tuning and regularization was carried out using the Grid Search Method to minimize overfitting.

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 was chosen because of its high speed and flexibility with which it handled large, complex datasets even though it was sensitive to outliers. Regularization and early stopping techniques were utilized to minimize overfitting and hyperparameter tuning was carried out using the grid search method.

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 used a sequential MLP with five fully connected hidden layers that used a variety of different activation functions and optimized hyperparameters. To minimize overfitting, we utilized dropout and batch normalization layers throughout the dense layers and also monitored test dataset metrics for early stopping.

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 three 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 implemented dropouts, batch normalization and early stopping that monitored testing dataset loss. Activation functions for every layer and other hyperparameters were efficiently tuned in order to minimize training loss.

4 Analysis

We have analyzed the predictions made by the BSM on both call and put option datasets and have compared it with four regression models, each trained on the two sets of input parameters, which gave a total of nine models for comparison. Out of these nine models, we picked the model with the best results and further broke down its performance based on the following two parameters:

4.1 Moneyness

Moneyness describes the inherent monetary value of an option's premium in the market. For our purposes we divided the underlying stock price (S) by the strike price (K) and categorized the options as being in-the-money, out-of-the-money or at-the-money. The following table shows how moneyness for different options was defined quantitatively.

Table 1. Call Option Test Dataset Metrics.

S. No.	Moneyness Type	Value of S/K (Call)	Value of S/K (Put)
1	In-the-Money	*>1.05*	< 0.95
2	Out-of-the-Money	<0.95	*>1.05*
3	At-the-Money	$0.95 \leq S/K \leq 1.05$	$0.95 \leq S/K \leq 1.05$

4.2 Maturity

Maturity represents the time to expiration in years for the option to be exercised (t). We considered options with $t < 0.1$ to be short-term, $0.1 \leq t \leq 0.5$ to be medium-term and $t > 0.5$ to be long-term.

5 Results and Discussions

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

1. Mean Absolute Error (MAE):

$$MAE = \frac{1}{N} \sum_{i=1}^{N} | x_i - y_i | \tag{5}$$

2. Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2} \tag{6}$$

3. Kullback–Leibler Divergence (KL):

$$KL(p||q) = \sum_{i=1}^{N} p(x_i) log \frac{p(x_i)}{q(y_i)} \tag{7}$$

where
N: number of observations
x_i: actual value
y_i: predicted value
$p(x_i)$: probability of x_i
$q(y_i)$: probability of y_i

A combination of the metrics was taken as the main loss function for training and testing purposes. Following two tables represent the results for the metrics obtained from the testing dataset predictions of call options and put options respectively.

To calculate the KL, the actual values and the predicted values were normalized around their sums to produce probability distributions.

Table 2. Call Option Test Dataset Metrics.

S. No.	Model	MAE	RMSE	KL
1	BSM	2.42	5.32	0.0048
2	SVM	6.25	9.04	0.0539
3	SVM_Greeks	9.58	11.81	0.1464
4	XGB	8.49	15.37	0.0804
5	XGB_Greeks	5.08	11.81	0.0146
6	MLP	2.74	4.52	0.0124
7	MLP_Greeks	3.38	5.35	0.0112
8	**LSTM**	**2.02**	**3.74**	**0.0045**
9	LSTM_Greeks	3.31	4.98	0.0104

In call option test dataset predictions, LSTM performed the best and gave the minimum MAE, RMSE as well as KL. Both LSTM and MLP outperformed the benchmark BSM in terms of MAE and RMSE.

Table 3. Put Option Test Dataset Metrics.

S. No.	Model	MAE	RMSE	KL
1	BSM	1.96	5.52	0.0164
2	SVM	3.76	6.30	0.1301
3	SVM_Greeks	5.21	8.72	0.2277
4	XGB	4.74	8.50	0.1937
5	XGB_Greeks	4.54	8.54	0.1737
6	MLP	1.68	2.95	0.0312
7	MLP_Greeks	4.59	6.71	0.0521
8	**LSTM**	**1.40**	**2.77**	**0.0147**
9	LSTM_Greeks	1.99	3.65	0.0317

In put option test dataset predictions, LSTM performed the best and gave the minimum MAE, RMSE as well as KL. Both LSTM and MLP outperformed the benchmark BSM in terms of MAE and RMSE.

The results from Tables 2 and 3 show that LSTM is the best model for option pricing for both call and put options. SVM and XGB performed worse than BSM, that is, these models underfit and were unable to sufficiently capture the underlying relation for option premium predictions. Results also show that Option Greeks when included as additional input features actually produced worse final test metrics in every single model and introduced a degree of data leakage in the models. Along with LSTM, MLP also outperformed BSM. The LSTM model also shows the minimum KL divergence values for both Call and Put Options indicating that the probability distribution of the predicted results from LSTM most closely matches the probability distribution of the actual values. The metrics show similar distribution and trends for both types of options which implies that the training of the models was impartial to option type.

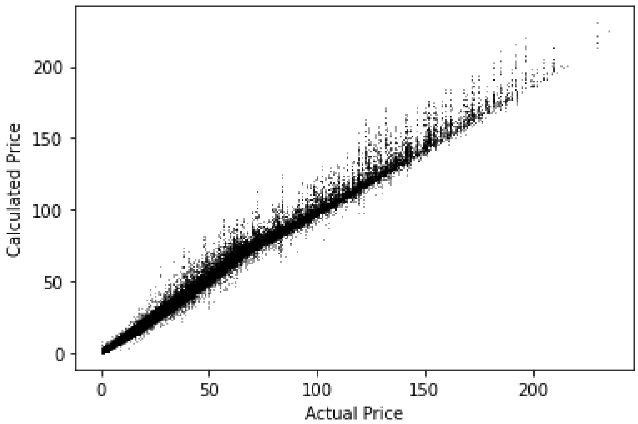


Fig. 1. Calculated (Predicted) Price vs Actual Price Plot for LSTM Call Option Test Dataset Predictions.

Fig. 1. shows the accuracy of LSTM in predicting the actual Call Option Prices. The plot is highly linear which demonstrates high similarity between the predicted and the actual prices.

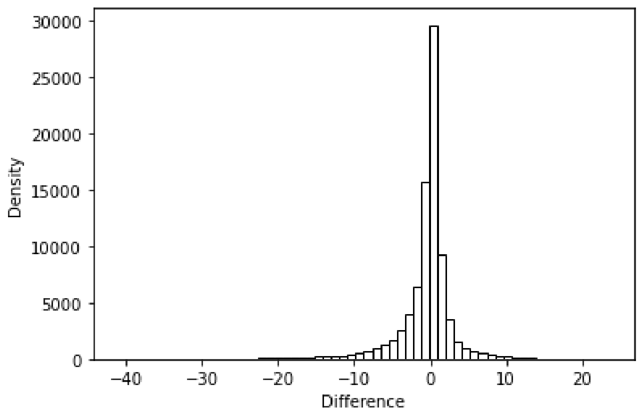


Fig. 2. Error Density Plot for LSTM Call Option Test Dataset Predictions.

Fig. 2. shows that the majority of predicted Call Option Prices give a lesser magnitude of error in LSTM. This demonstrates the high similarity between the predicted and the actual prices.

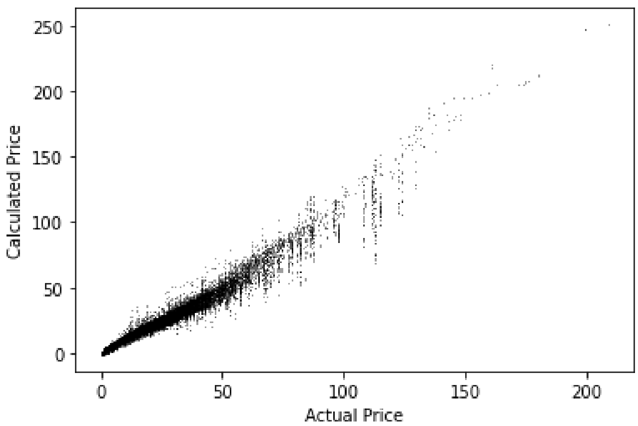


Fig. 3. Calculated (Predicted) Price vs Actual Price Plot for LSTM Put Option Test Dataset Predictions.

Fig. 3. shows the accuracy of LSTM in predicting the actual Put Option Prices. The plot is highly linear which demonstrates high similarity between the predicted and the actual prices.

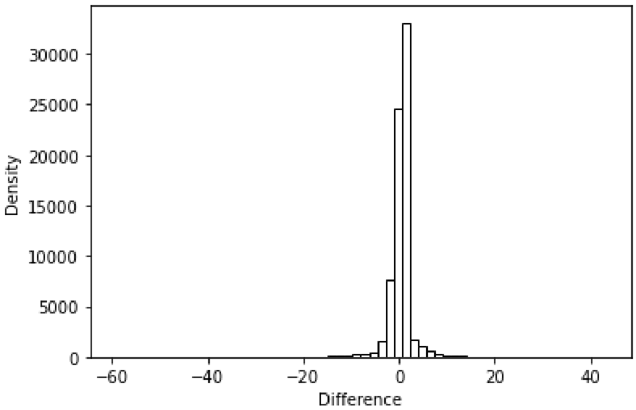


Fig. 4. Error Density Plot for LSTM Put Option Test Dataset Predictions.

Fig. 4. shows that the majority of predicted Put Option Prices give a lesser magnitude of error in LSTM. This demonstrates the high similarity between the predicted and the actual prices.

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 4. LSTM Test Dataset Metrics Grouped by Moneyness.

S. No.	Option Type	Moneyness Type	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

The results from Table 4 show that when grouped by Moneyness, Out-of-the-Money options gave 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 In-the-Money to Out-of-the-Money options, so does the magnitude of our error metrics.

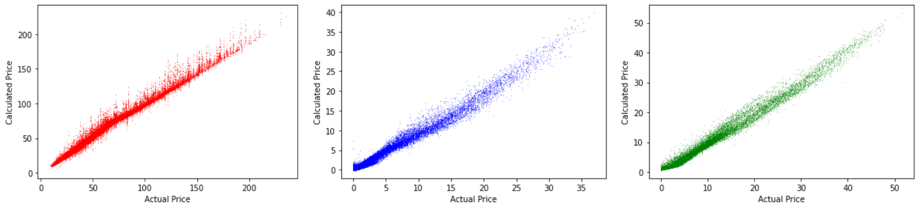


Fig. 5. Calculated Price vs Actual Price Plots for LSTM Call Option Test Predictions Grouped by Moneyness.

Fig. 5. shows that Out-of-the-Money Stock Option Prices illustrates the highest similarity between predicted and actual prices for Call Options in LSTM.

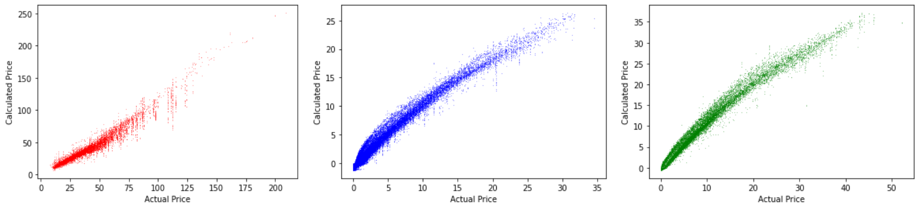


Fig. 6. Calculated Price vs Actual Price Plots for LSTM Put Option Test Predictions Grouped by Moneyness.

Fig. 6. shows that Out-of-the-Money Stock Option Prices illustrates the highest similarity between predicted and actual prices for Put Options in LSTM.

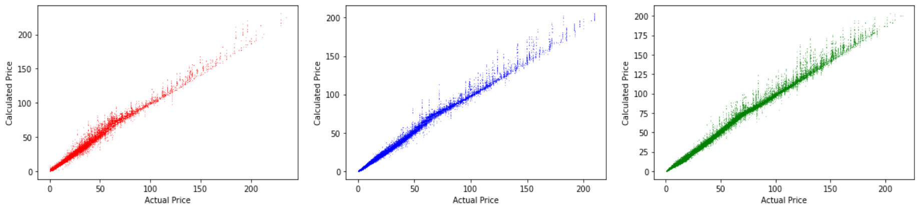
The following table represents the results of the LSTM when the options are grouped by maturity.

Table 5. LSTM Test Dataset Metrics Grouped by Maturity.

S. No.	Option Type	Moneyness Type	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 gave the minimum MAE as well as the minimum RMSE, followed by medium-term and then long-term options.

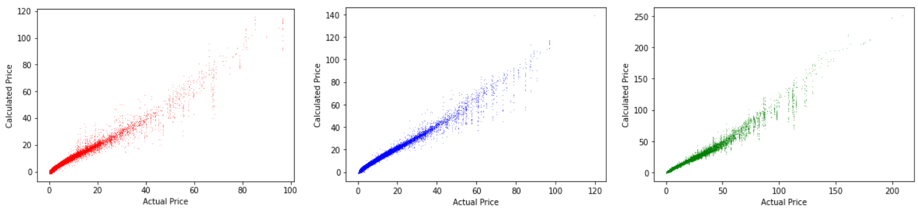
The results from Table 5 show that when grouped by maturity, short-term options gave better metrics followed by medium-term and then long-term options. This behaviour is also to be expected as a lower maturity directly lowers the option premiums. Hence, the magnitude of our error metrics decreases as we move from long-term to short-term options.



a) Short-Term b) Medium-Term c) Long-Term

Fig. 7. Calculated Price vs Actual Price Plots for LSTM Call Option Test Predictions Grouped by Moneyness.

Fig. 7. shows that Short-Term Stock Option Prices illustrates the highest similarity between predicted and actual prices for Call Options in LSTM.



a) Short-Term b) Medium-Term c) Long-Term

Fig. 8. Calculated Price vs Actual Price Plots for LSTM Put Option Test Predictions Grouped by Moneyness.

Fig. 8. shows that Short-Term Stock Option Prices illustrates the highest similarity between predicted and actual prices for Put Options in LSTM.

6 Conclusions

We conclude that LSTM was the best performing model when it came to predicting stock option prices. We found that the performance of both LSTM and MLP was superior to the benchmark Black-Scholes Model in terms of MAE and RMSE whereas SVM and XGBoost failed to outperform the benchmark. Calculation of KL for different models showed that the probability distribution of the predicted results from LSTM had the highest similarity with the probability distribution of the actual values.

We also demonstrated the redundancy of utilizing Option Greeks as input features for the prediction models. Option Greeks introduced a certain degree of data leakage on the models which often reduced the overall performance. This was a new finding as most of the literature about using Option Greeks as input features for option pricing mostly reported an improvement in performance.

The analysis of the option predictions on the basis of moneyness and maturity validated that the LSTM prediction model was able to efficiently and accurately predict option prices in a real-world market scenario.

7 References

1. Kullback., Leibler. "On Information and Sufficiency." Ann. Math. Statist. 22 (1) 79 - 86, March, 1951.
2. Drucker, H., Christopher B., Linda K., Alex S. and Vladimir NV.: “Support Vector Regression Machines.” NIPS. (1996).
3. Hochreiter S. and Jürgen S.: “Long Short-Term Memory.” Neural Computation 9, 1735-1780. (1997).
4. Chen, T. and Carlos G.: “XGBoost: A Scalable Tree Boosting System.” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Association for Computing Machinery, New York, NY, USA (2016).
5. James MH., Andrew WL. and Tomaso P.: A nonparametric approach to pricing and hedging derivative securities via learning networks. The Journal of Finance, 49(3):851–889. (1994).
6. Ramazan G. and Aslihan S.: “Degree of mispricing with the Black-Scholes model and nonparametric cures”, Annals of Economics and Finance, Vol. 4, pp. 73-101. (2003).
7. Nikola G., Ramazan G. and Dragan K.: Option pricing with modular neural networks. IEEE transactions on neural networks, 20(4):626–637. (2009).
8. Palmer, S.: “Evolutionary algorithms and computational methods for derivatives pricing.” (2019).
9. Robert C. and Sanjiv D.: “Machine learning in finance: the case of deep learning for option pricing”. Journal of Investment Management, 15(4):92–100. (2017).
10. Liu S, Oosterlee CW, Bohte SM.: Pricing Options and Computing Implied Volatilities using Neural Networks. Risks. 2019; 7(1):16.
11. Ruf, J. and Weiguan W.: “Neural networks for option pricing and hedging: a literature review.” Journal of Computational Finance (2020).