CNN-LSTM vs ANN: Option Pricing Theory

Edward Chang
Faculty of Engineering
Western University
London, Canada
echang49@uwo.ca

***Abstract* – The modern derivatives market has been steadily growing since the development of the first accurate option pricing model by Fischer Black, Robert Merton, and Myron Scholes. Since then, there have been many different approaches to more accurately price options like the binomial option pricing model and approaches using technology such as machine learning. There are many different research papers on option pricing with artificial neural networks ("ANN") but not many with other neural network types. We contribute to the existing literature by developing a convolutional neural network – long short-term memory ("CNN-LSTM") model to price options and compare it to an ANN model. The results from this paper show that the CNN-LSTM model performed much better than the ANN model for pricing options. It is proposed that the model is improved to reach a greater accuracy when predicting option prices.**

***Index Terms* – Option Pricing Theory, Convolutional Neural Network, Long Short-Term Memory, Artificial Neural Network, Black-Scholes Model**

I. INTRODUCTION

In today's financial markets, commodities are traded between the buyers and sellers in many different forms. The rudimentary commodity that is traded between buyers and sellers in the financial markets are stock while an example of an advanced commodity is options. An option is a contract between the buyer and seller which gives the buyer the right to buy or sell a certain stock at a given price. European options allow the option to be exercised on a certain date whilst American options allow the option to be exercised on or before a certain date. For a buyer to gain the right to buy or sell a certain stock at a certain price, the buyer must pay a premium to purchase the contract. The option premium must be priced accordingly to eliminate possibilities of arbitrage. Pricing option premiums based on the calculated probability that the contract will finish in the money at expiration is called 'Option Pricing Theory' [1].

While there are many different models to price options, most of them follow a mathematical formula to price options. With the recent developments in the field of artificial intelligence, machine learning can be used to determine option prices. Artificial neural networks (ANN) could be used to determine an accurate result for option premiums but may not be as good compared to other types of neural networks due because an ANN cannot associate data with relation to time. Creating a convolutional neural network – long short-term memory (CNN-LSTM) model allows for prediction using feature extraction with respect to time-series data.

II. RELATED WORK

There have been other research papers that delve into how artificial intelligence can be used to help price options. As the Black-Scholes option pricing model (BSOPM) is widely regarded as accurate option pricing formula, models use it to compare their accuracy.

A. *Stock Option Pricing Using Bayes Filters (2004) [2]*

Bayes filters take a probabilistic approach to estimate an unknown probability density over time. Liao's technical report predicts stock option prices by using Bayes filters with an Expectation-Maximization algorithm to calculate the implied volatility without a return variance for option pricing using the BSOPM.

The report found that the Bayes filters performed better to predict option prices than the generalized autoregressive conditional heteroskedasticity (GARCH) model as well as the provided implied volatilities for pricing the option.

B. *A new application of fuzzy set theory to the Black-Scholes option pricing model (2005) [3]*

Fuzzy classifiers take a non-binary approach to decision making compared to the traditional binary 'true' or 'false'. Lee *et al.* predicts option prices in the publication article by adding fuzzy classifier decision making to the BSOPM as the original version has assume the riskless interest rate and the volatility to be constant. The report found that ignoring fuzzy classifiers result in overestimations.

C. *A Comparative Study of Support Vector Machine and Artificial Neural Net for Option Price Prediction (2021) [4]*

Support vector machines (SVM) classify data by creating a line or hyperplane to separate the data into different classes. ANNs calculate data with feed-forward layers such as the input layer, the hidden layer, and the output layer. Madhu *et al*. compared the accuracy of option pricing between SVMs and ANNs by comparing the two based on the SPY option price. The researchers concluded that the ANN was able to predict option prices more accurately compared to the SVM.

D. *Nonparametric Machine Learning Methods for Equity Option Pricing (2021) [5]*

Jiang's technical report predicts American and European stock option prices using an ANN. The technical report concluded that the ANN built in the report performed better than the BSOPM and binomial option pricing model (BOPM) for puts but performed worse for calls. Jiang also noted that the main discrepancy between ANNs built for different option types were the activation functions.

E. *Pricing Brazilian Fixed Income Options with Feedforward and Recurrent Neural Networks (2014) [6]*

Recurrent neural networks (RNN) are like ANNs except nodes in an RNN remember inputs from prior computations. Maciel's research compares the accuracy of computing the one-day interbank deposit option prices in the Brazilian market between the Black model, Vasicek model, CIR model, ANN model, Elman RNN (ERNN) model, and the Jordan RNN (JRNN) model. The research results show that all three neural networks perform better than the Black model, Vasicek mode, and CIR model. Additionally, the results indicate that the RNNs were a little more accurate compared to the ANN.

F. *A CNN-LSTM-Based Model to Forecast Stock Prices (2020) [7]*

A CNN-LSTM model has a convolutional neural network (CNN) first for feature extraction with a long short-term memory (LSTM) later applied to make predictions based on previous time series data. Lu et al. compared the accuracy of stock pricing between an ANN, CNN, RNN, LSTM, CNN-RNN, and CNN-LSTM. The researchers suggest that a CNN-LSTM model be used for stock price prediction because a CNN can predict time-series data but not accurately as its primary function is image processing. The results of the research in order of most accurate to least is CNN-LSTM, CNN-RNN, LSTM, RNN, CNN, ANN.

G. *Research Gap*

The related literature identifies different usages of artificial intelligence that can result in better accuracy than the BSOPM. There is a lot of research completed on the option pricing theory with respect to artificial neural networks. Additionally, there is research that provides better alternatives to option pricing compared to artificial neural networks.

The CNN-LSTM was proven to be the best model type for use in stock prediction. Despite the data used for stock price prediction and option pricing being similar, there is no research completed on the viability of a CNN-LSTM model for option pricing. This project seeks to compare the accuracy of option pricing for a CNN-LSTM model compared to an ANN model with the BSOPM used as a baseline.

III. RESEARCH OBJECTIVES

This project has the following research objectives:

1. Build a CNN-LSTM model to accurately price options.
2. Measure the accuracy of the CNN-LSTM model against European options and American options for calls, puts, and both.
3. Compare the accuracy of the CNN-LSTM model against an ANN model and a baseline model which is the BSOPM.

IV. DATA

A. Data Collection

The problem to be solved by this project is if an accurate CNN-LSTM model can be built for option pricing. The option pricing datasets used in the research is hosted by the Wharton Research Data Services from the University of Pennsylvania. From the OptionMetrics vendor, the datasets on option pricing were taken from the Ivy DB US's Index Dividend Yield, Zero Coupon Yield Curve, Standardized Options, and Security Prices [8].

TABLE I
DATA DICTIONARY

Variable	**Variable Explanation**
Stock Price	The stock's closing price of the previous day.
Strike Price	Price of the stock at option maturity.
Time to Expiration	The number of days for the option contract to mature.
Implied Volatility	Percentage of the likelihood for changes in the stock price.
Dividend Yield	Ratio of cash dividends distributed compared to the stock price.
Zero Coupon Yield	Minimum guaranteed rate of return if the money were invested elsewhere.
Premium	Price of the option contract.

The S&P 100 (OEX) was used for training and testing the models for the American style options while the S&P 100 Index European (XEO) was used for training and testing the models for the European style options. The dataset used for this project only includes data from February 12, 2012, to December 31, 2021. The OEX dataset had 51,716 entries while the XEO dataset had dataset had 51,756 entries allowing for equal training between the different models. The variable dictionary is shown in *Table 1*.

B. Data Preprocessing

1) *Dataset Conversion*

The raw datasets contained unnecessary fields. To fix this, the necessary fields were taken from each of the Ivy DB US datasets and added to a single formatted dataset. The date, time to expiration, strike price, premium, and implied volatility were taken from the Standardized Options dataset. The dividend yield and closing stock price was added to the data rows corresponding by the date. The zero-coupon yield was added to the data rows corresponding by the date and closest option maturity value.

2) *Noise Removal*

Some of the parsed data in the dataset were incomplete and would mislead the neural networks. Rows removed included a blank implied volatility, an option premium of 0, and a strike price of 0.

3) *Data Pruning*

To ensure more accurate results in training the models with the datasets, outliers were pruned. 2.5% was pruned from each side of the dataset resulting in an overall 5% loss of data. The metric used for pruning was the stock price to option price (SP-OP) ratio denoted by the following formula:

$$SP - OP\ Ratio = \frac{Stock\ Price\ (Closing)}{Option\ Price} \tag{1}$$

Before pruning, the minimum SP-OP ratio for the OEX dataset was 3.2163 while the maximum SO-OP ratio for the OEX dataset was 312.5078. The minimum SP-OP ratio for the XEO dataset was 5.6802 before pruning while the maximum SP-OP ratio for the XEO dataset was 318.7282. *Figure 1* and *Figure 2* depicts the frequency in ratios in buckets of

10 before pruning where the first bucket is ratios from 0 to 9 while the next is from 10 to 19 continuing that pattern for the rest of the buckets.

FIGURE I

```
xeoRatioBuckets = [
  3044, 18301, 11631, 6973, 3650, 2379,
  1518,  1369,  1031,  744,  584,  541,
   508,   480,   377,  329,  250,  206,
   145,   107,    60,   60,   46,   48,
    34,    24,    18,    8,    6,    6,
     0,     3
]
```

Frequency of ratios in buckets for OEX

FIGURE II

```
oexRatioBuckets = [
  3431, 18316, 11646, 6974, 3574, 2273,
  1569,  1344,   959,  716,  581,  560,
   544,   449,   368,  302,  222,  157,
   130,    70,    56,   52,   44,   33,
    25,    15,    12,    4,    2,    1,
     5,     2
]
```

Frequency of ratios in buckets for XEO

C. *Exploratory Data Analysis*

To have a deeper understanding of the preprocessed dataset, metrics were extracted from the dataset such as call frequency, put frequency, time to maturity, and option price distribution. These metrics can help understand how the neural network models will train and if there will be any biases.

Statistics of the dataset can be found in Table 2. While there are more puts than calls for the OEX and XEO datasets after pruning, the amounts are similar and will not result in training one type of options more than the other. Defining a short-term option contract as having a maturity date of less than or equal to 60 days, only 25.9% of the options are short-term in the OEX and XEO datasets. This indicates that the machine learning models may be less accurate when determining the price of a short-term option. Lastly, comparing the minimum SP-OP ratio and maximum SP-OP ratio of the dataset before and after pruning, it shows the dataset is more normalized. For example, the maximum SP-OP ratio decreased for the OEX from 312.5078 to 154.8459 and from 318.7282 to 160.6989 for the XEO. While the pruned dataset would result in a worse accuracy for the outliers, the overall dataset will result in an increased accuracy for the models.

TABLE II
DATASET STATISTICS

	OEX	XEO
Calls	25783	25734
Puts	25933	26022
Short-Term	13379	13394
Long-Term	38337	38362
Minimum SP-OP Ratio	8.5433	8.7106
Maximum SP-OP Ratio	154.8459	160.6989

A sample of the preprocessed datasets can be found in Table 3 which displays only calls for the OEX. The other datasets are also in this format with the order of the columns displayed in Table 1.

TABLE III
SAMPLE OF PREPROCESSED DATASET

Date	Stock_Price	Strike_Price	Expiry	Volatility	Dividend %	Interest %	Premium
20120213	611.54	594.338694	547	0.218362	2.424426	0.527261	64.826292
20120213	611.54	599.789872	365	0.206942	2.424426	0.492971	50.482618
20120213	611.54	602.628133	273	0.205222	2.424426	0.47113	43.374732
20120213	611.54	605.503654	182	0.191843	2.424426	0.435809	33.191394
20120213	611.54	606.464432	152	0.183932	2.424426	0.424878	29.108571
20120213	611.54	607.433569	122	0.182738	2.424426	0.412491	25.908464
20120213	611.54	608.442994	91	0.180308	2.424426	0.39034	22.076813
20120213	611.54	609.45246	60	0.171892	2.424426	0.357618	17.091265
20120213	611.54	610.450914	30	0.155167	2.424426	0.251891	10.906895
20120214	610.82	593.629552	547	0.221965	2.428058	0.529839	65.77015
20120214	610.82	599.072347	365	0.209284	2.428058	0.494334	50.974851
20120214	610.82	601.912149	273	0.20749	2.428058	0.472824	43.789215
20120214	610.82	604.786064	182	0.196413	2.428058	0.437507	33.924489
20120214	610.82	605.744692	152	0.18915	2.428058	0.425896	29.882287

V. RESEARCH METHODOLOGY

The model used in this paper are the BSOPM, ANN model, and CNN-LSTM model. The mean squared error (MSE) metric was used to determine the accuracy of the models.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$
$$(n, y, \hat{y})$$
$$n = number\ of\ data\ points$$
$$y = given\ values$$
$$\hat{y} = predicted\ values$$

(2)

A. Black-Scholes Option Pricing Model

Developed in 1973 by Fischer Black and Myron Scholes, the BSOPM is a mathematical model determining the price of an option based on the stock price, strike price, stock volatility, time to maturity, and risk-free interest rate [9].

The original model makes the following assumptions:

1. Dividends are not paid out when the option has not yet matured.
2. The option cannot be exercised before maturity.
3. The risk-free interest rate and stock volatility is known and constant.
4. The returns of the stock are normally distributed.
5. The financial markets are random and cannot be predicted.

Later in 1973, Robert Merton extended the Black-Scholes mathematical formula to reflect dividends paid out from the underlying asset before the option has matured [10]. The version of the BSOPM used in this project include Robert Merton's contributions and takes the dividend yield into account.

With the following variables:

$$(x, c, v, r, t, d)$$
$$x = Stock\ Price$$
$$c = Strike\ Price$$
$$t = Time\ to\ Expiration$$
$$r = Risk - Free\ Interest\ Rate$$
$$d = Dividend\ Yield$$
$$v = Annualized\ Stock\ Volatility$$

The formulas to price option calls and puts are

$$Call = xe^{-dt}N(d_1) - ce^{-rt}N(d_2)$$

(3)

$$Put = ce^{-rt}N(-d_2) - xe^{-dt}N(-d_1)$$

(4)

where

$$d_1 = \frac{\ln\left(\frac{x}{c}\right) + (r - d + \frac{v^2}{2})t}{v\sqrt{t}}$$

(5)

$$d_2 = \frac{\ln\left(\frac{x}{c}\right) + (r - d - \frac{v^2}{2})t}{v\sqrt{t}} = d_1 - v\sqrt{t}$$

(6)

From the datasets in the format of Table 3 above, some processing is completed to conform the input parameters to the proper scale of the formula. The dividend yield and zero-coupon yield were converted from percent form to decimal form while the time to expiration was converted from days to years.

The BSOPM was used to determine the calculated option price for American style calls, American style puts, European style calls, and European style puts. With the calculated prices, the calculated option price was compared to the given option price finding the MSE for the above categories.

As the formula to calculate option prices for calls and puts are different, two different methods were used to calculate the MSE of the BSOPM with respect to American style calls and puts, and European style calls and puts.

The "Calls and Puts (Random)" method receives a dataset with a similarly equal number of calls and puts in a random order. During dataset iteration, rows with an even index were calculated with the call formula while rows with an odd index were calculated with the put formula. The MSE of the "Calls and Puts (Random)" method is then determined with the calculated and given option prices.

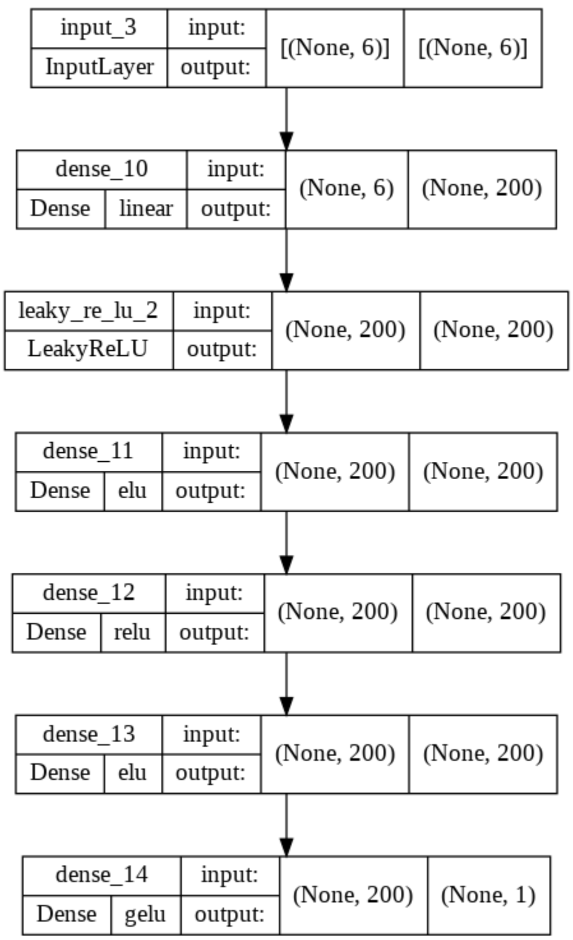
The "Calls and Puts (Calculated)" method calculates the MSE of calls and puts with the MSE for calls only and the MSE for puts only in the following equation.

$$MSE = \frac{n_c MSE_c + n_p MSE_p}{n_c + n_p}$$
$$(n_c, n_p, MSE_c, MSE_p)$$
$$n_c = number\ of\ calls$$
$$n_p = number\ of\ puts$$
$$MSE_c = MSE\ of\ calls\ only$$
$$MSE_p = MSE\ of\ puts\ only$$

(7)

B. Artificial Neural Network Option Pricing Model

FIGURE III



Model architecture for the ANN

As the ANN model is being used for comparison against the CNN-LSTM model, a well-performing model architecture is needed to properly compare the two models. In the feed-forward ANN model used in [5], it was observed that the model was more accurate than the BSOPM and BOPM for some categories while being less accurate in others. Due to the high performance from the feed-forward ANN model in [5], this project's model architecture was based off it.

The model had an input layer of size 6, an output layer of size 1, and 4 hidden layers with 200 neurons each. The inputs used was stock price, strike price, time to expiration, implied volatility, dividend yield, and the zero-coupon rate. The activation functions used in the feed-forward ANN in order were LeakyReLu, ELU, ReLu, ELU, and GELU. The model used a mean squared logarithmic error (MSLE) metric for the loss function while the Adam algorithm was used as the optimizer. Two identical ANN models with different batch sizes and epochs were compared. The first ANN model was trained with 100 epochs with a batch size of 32 while the second ANN model was trained with 200 epochs with a batch size of 128. The second model performed ~20% more accurate than the first. As such, the MSE in the results section for the ANN depict the results for 200 epochs with a batch size of 128.

C. Convolutional Neural Network – Long Short-Term Memory Option Pricing Model

CNNs are used for its ability to extract features from a dataset and while they are normally used in a 2-dimensional space for image processing, they could be used in a 1-dimensional space for time series. CNNs are feedforward and perform worse for predicting time-based data compared to other model types such as RNNs and LSTMs [9]. LSTMs are used for analysis or forecasting due to its ability to back-propagate with no gradient explosions or disappearance which can occur in RNNs.

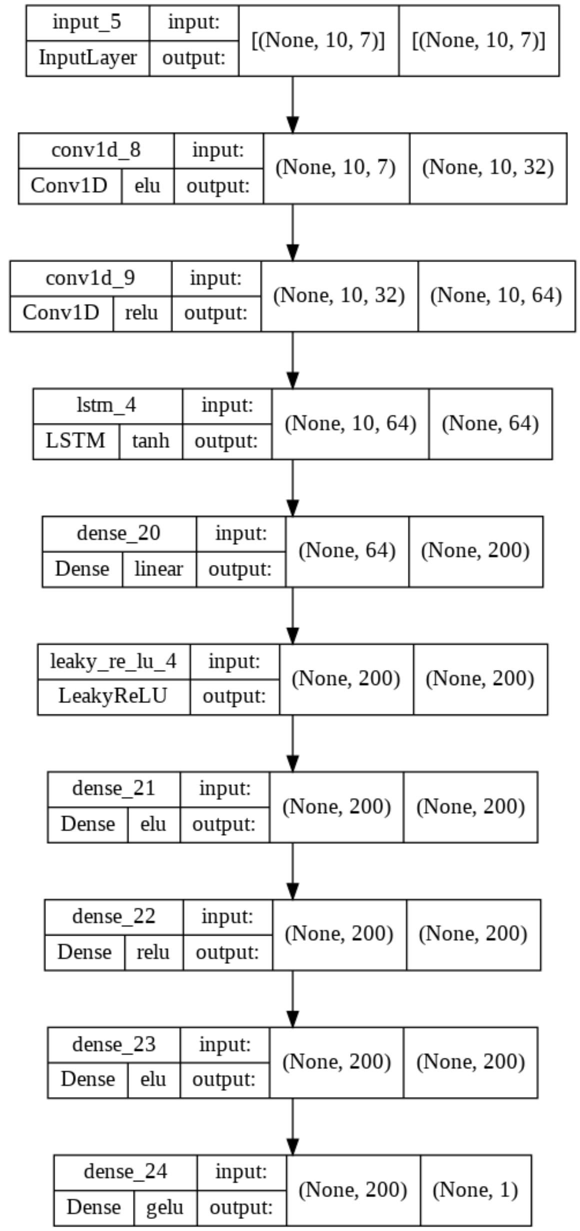
The general architecture for a CNN-LSTM model in order is input layer, convolutional layer(s), long short-term memory layer(s), dense layer(s), and output layer. The CNN-LSTM model architecture for this project was inspired by the CNN-LSTM model in [7] and the ANN model created for this project. Though the research completed in [7] is for stock price forecasting instead of option price forecasting, the data structures are similar. The input layer of the model accepts a dataset with 10 timesteps and 7 parameters in each timestep. The input parameters used are stock price, strike price, time to expiration, implied volatility, dividend yield, the zero-coupon rate, and the option premium for previous timesteps. The convolution section of the model included two Conv1D layers. The first Conv1D layer had 32 filters, a kernel size of 10, and a dilation rate of 4 with causal padding. The second Conv1D layer had the same parameters as above but with 64 filters instead. The LSTM section included one LSTM layer that did not output a sequence and had 64 units. The dense section of the CNN-LSTM model is the same as the ANN model having 4 hidden layers with 200 neurons each. The activation functions used in the CNN-LSTM in order were ELU, ReLu, tanh, LeakyReLu, ELU, ReLu, ELU, and GELU. The model used a MSE metric for the loss function while the Adam algorithm was used as the optimizer. The CNN-LSTM model was trained with a batch size of 128 for 200 epochs.

The final dataset was normalized with the Z-score normalization formula:

$$Dataset = \frac{Dataset - Mean}{Standard\ Deviation} \tag{8}$$

While the data did not need to be processed further for the ANN model after preprocessing, the CNN-LSTM model required the datasets to be converted to time series. With Table 3 as an example, the datasets are sorted by ascending date first and then descending time to expiration second. The data inside the datasets were filtered into sub-datasets sorted by ascending date with the data in each sub-dataset sharing the same time to expiration and option type. Afterwards, each sub-dataset was converted into a timeseries dataset with a sequence length of 10 and sequence stride of 1. The timeseries datasets were then combined to form a single dataset with the last timestep of every row having the option price removed as that is label. This processing was done to ensure the timesteps were in order and on different days.

FIGURE IV



Model architecture for the CNN-LSTM

V. RESULTS

With 20% of the dataset reserved for testing, the MSEs for the BOSPM, ANN model, and CNN-LSTM model are displayed in Table 4, Table 5, and Table 6 respectively. Overall, the rankings for accuracy in order are the BSOPM, CNN-LSTM model, and ANN model. The BSOPM was found to be the most accurate model to predict options in every category. The CNN-

LSTM model was more accurate than the ANN model in every category except for the American "calls and puts".

TABLE IV
BLACK-SCHOLES MODEL ACCURACY

	OEX	XEO
Calls	1.083	0.0033
Puts	0.3442	0.0058
Calls and Puts (Calculated)	0.7125	0.0046
Calls and Puts (Random)	0.7168	0.0044

TABLE V
ANN MODEL ACCURACY

	OEX	XEO
Calls	12.6310	9.7236
Puts	13.3724	8.9890
Calls and Puts	8.4397	5.5702

TABLE VI
CNN-LSTM MODEL ACCURACY

	OEX	XEO
Calls	10.4393	4.2537
Puts	10.6472	3.8264
Calls and Puts	9.3283	3.5539

TABLE VII
CNN-LSTM VS ANN MODEL PERFORMANCE

	OEX	XEO
Calls	17.35%	56.25%
Puts	20.38%	57.43%
Calls and Puts	-9.53%	36.20%

Table 7 compares model accuracies between the CNN-LSTM and ANN models with positive percentages indicating better performance from the CNN-LSTM. Analysing the data from Table 7, the CNN-LSTM performed better than the ANN for American options while the CNN-LSTM performed much better than the ANN for European options. The exception to this is the ANN performing slightly better than the CNN-LSTM for the American "calls and puts".

The leading prediction for why the ANN performed better than the CNN-LSTM in that category is due to the CNN-LSTM finding similarities and differences between calls and puts more difficult than the ANN. The MSE results show that the CNN-LSTM is better than the ANN at predicting option prices for only calls or only puts because the CNN-LSTM can analyze past data and extract features. This implies that if the CNN-LSTM performs worse when calls and puts are together, it is because the CNN-LSTM has more difficulty finding similarities and differences between calls and puts. It was found in both models that training for both calls and puts resulted in a higher accuracy than training only calls or puts. Referring to the mathematical Black-Scholes formula to price call options in Equation 3 and put options in Equation 4, the formulas are similar with the only differences being changes in variable positivity. The datasets for "calls only" and "puts only" were derived by separating the types from the combined dataset. Therefore, the "calls and puts" model was given double the amount of data to learn from compared to the "calls only" and "puts only" models allowing it to better determine correlations resulting in higher accuracies. However, the ANN learned more compared to the CNN-LSTM when training for both calls and puts. Table 8 compares the "Calls and Puts" MSE of the ANN and CNN-LSTM to the lower MSE between "Calls" and "Puts" with positive percentages indicating better performance from the "Calls and Puts" category.

TABLE VIII
CALLS AND PUTS VS CALLS OR PUTS PERFORMANCE

	OEX	XEO
ANN	33.18%	38.03%
CNN-LSTM	10.64%	7.12%

The predicted reason for why the CNN-LSTM had more difficulty finding trends between calls and puts than the ANN resulting in a lower percentage of improvement is because the CNN-LSTM is more complex than the ANN. The input layer of the CNN-LSTM takes a more complex input with a shape of

(None, 10, 7) while the ANN takes an input with a shape of (None, 6). After passing the data through the convolutional layers and LSTM layer, the data inputted into the dense section of the CNN-LSTM model is in the shape of (None, 64). The CNN-LSTM is almost identical to the ANN with 2 added convolutional 1D layers, a long short-term memory layer, and a different input shape for the first dense layer. The difference in model architecture suggests that the cause for a lower percentage of improvement compared to the ANN is either because of trends between calls and puts being less apparent after going through the CNN layers and LSTM layer, or the dense layers being less able to determine trends between calls and puts when the input shape of the first dense layer is (None, 64). The potential solutions to allow a CNN-LSTM to better distinguish calls and puts are either a more efficient model architecture, different hyperparameters, or more training with lower batch sizes and higher epochs to provide the CNN-LSTM more opportunities to find the trends.

TABLE IX
AMERICAN VS EUROPEAN OPTION PERFORMANCE

	ANN	CNN-LSTM
Calls	-23.02%	-59.25%
Puts	-32.78%	-64.06%
Calls and Puts	-34.00%	-61.90%

Table 9 compares model accuracies between American and European style options with positive percentages indicating better performance from the American style options. The percentages show that the CNN-LSTM and ANN can predict European style options more accurately than American style options for every category tested. This is due to the ANN and CNN-LSTM models being more optimized to price European options than American options. While the research in [5] had different model designs for American and European style options, the ANN model and CNN-LSTM model in this project had the same design which was inspired from the European model in [5] for standardization. While the research from [5] found that the best accuracy results for American and European style options both have 4 dense layers comprised of 200 neurons each, the activation functions were different. The model to price American options had the following activation functions in order: ELU, ELU, ELU, ELU, ReLu. The model to price European options had the following activation functions in order: LeakyReLu, ELU, ReLu, ELU, exponential. For comparison, the ANN and CNN-LSTM's dense section have the following activation functions in order: LeakyReLu, ELU, ReLu, ELU, GELU. Additionally, the percentages from Table 9 show the CNN-LSTM had a much greater percentage of improvement compared to the ANN. This is expected as a CNN-LSTM can extract features from a timeseries while an ANN cannot.

TABLE X
BSOPM VS ANN MODEL PERFORMANCE

	OEX	XEO
Calls	91.43%	99.97%
Puts	97.43%	99.94%
Calls and Puts (Calculated)	91.56%	99.92%
Calls and Puts (Random)	91.51%	99.92%

TABLE XI
BSOPM VS CNN-LSTM MODEL PERFORMANCE

	OEX	XEO
Calls	89.63%	99.92%
Puts	96.77%	99.85%
Calls and Puts (Calculated)	92.36%	99.87%
Calls and Puts (Random)	92.32%	99.88%

Table 10 compares model accuracies between the BSOPM model and ANN model while Table 11 compares model accuracies between the BSOPM model and the CNN-LSTM model. Positive percentages indicate better performance from the BSOPM model. Referring to the research completed in the related works section, every article that compared an application of artificial intelligence to price options to the BSOPM found the artificial intelligence model to be more accurate than the BSOPM. Specifically, the research completed in [5] found that ANNs can be more accurate than the BSOPM in some categories and is similar for the others. While the absolute difference in values is small for the MSEs between the BSOPM, ANN, and CNN-LSTM, the BSOPM performs significantly better than the ANN and CNN-LSTM created in this project.

VI. IMPROVEMENTS

Analysing the results from Table 10 and Table 11, the BSOPM is a much better choice for option pricing compared to the ANN and CNN-LSTM. However, previous research has proven that ANNs can be more accurate than the BSOPM and that CNN-LSTMs are more accurate than ANNs for time series data. Therefore, improvements must be made to the current neural networks created for this project to be more accurate.

A. Binomial Option Pricing Model

The BSOPM is currently used in this project as a baseline to measure the accuracy of the neural network models. While the BSOPM is a good baseline model to measure the accuracy of European options, the formula assumes that the option cannot be exercised before maturity and therefore does not support American options. The BOPM is a good, well-known alternative for a baseline model to measure the accuracy of American options. The BOPM is preferred to the BSOPM for pricing American options because it considers an option contract being exercised before maturity by calculating a range of possible results for different periods in a multi-period model.

B. Simulated Data

The dataset used in this project is real market data of the OEX and XEO from February 12, 2012, to December 31, 2021. The dataset provides ~25,000 rows of data to the models predicting call option prices or put option prices separately and ~50,000 rows of data to the models predicting call and put option prices together. Currently, the ANN and CNN-LSTM models are not pretrained before being trained with real market data. This requires the neural networks to determine the best fit through trial and error.

With a train set of 60%, validation set of 20%, and test set of 20%, the neural networks can only be trained with a maximum of ~30,000 rows of data. From Table 11, the CNN-LSTM is required to improve its accuracy by 99.92% to have the same accuracy as the BSOPM. With ~30,000 rows of data, the ANN and CNN-LSTM models would overfit before achieving a similar accuracy to the BSOPM. This could be fixed by gathering a lot more market data, either from simulation or collecting more real data. For reference, the ANNs built in [5] were first pretrained with ~10,000,000 rows of simulated data before being trained with real market data.

Simulating market data is preferred over collecting more real-world market data as it requires preprocessing of the dataset to eliminate noise or any outliers that could decrease the accuracy of a model. Using simulated data, it could be controlled and designed to ensure no outliers or noise would be present. The BOPM would be used to simulate American option data while the BSOPM would be used to simulate European option data. The ANN and CNN-LSTM models would then be pretrained with the simulated market data before being trained with real market data allowing the machine learning models to already have an accurate base to start training from.

C. Longer Training

The ANN and CNN-LSTM created in this project was trained with 200 epochs and a batch size of 128. After 200 epochs, both models were not overfitting and could have been further trained. The decision for not training until the local minima was reached is because the project aims to determine the viability of a CNN-LSTM in option pricing by comparing it to an ANN. However, a comparison could also be made with the most accurate ANN and CNN-LSTM regardless of if their number of epochs and batch sizes are different. To train the models for further accuracy, the models could iterate through more epochs or have smaller batch sizes at the cost of training time.

D. More/Better Data Processing

After preprocessing the datasets, additional processing is done to ensure a better fit with the models. One of these processes is data normalization as non-normalized data could have outliers that mislead the neural network during optimization. Currently, the ANN and CNN-LSTM are normalized with Z-score normalization which is calculated with a mean of the dataset and standard deviation. As the

CNN-LSTM accepts time series data as an input, normalization with a mean of the entire dataset does not address short-term fluctuations. The CNN-LSTM could utilize a moving average for normalization instead which calculates the mean of each subset. This would flatten short-term fluctuations allowing the CNN-LSTM to easier detect long-term trends.

Another step that could be improved for data processing is how labels are scrubbed from the input data in the CNN-LSTM. The CNN-LSTM is given the same parameters as the ANN in the timesteps with an added option price added for past timesteps. As the input shape for the CNN-LSTM is (10, 7), the label would by default be included in the option price of the 'current' timestep. This issue is currently resolved by setting the values to -1 after the dataset is normalized with the expectation that the CNN-LSTM will understand that the field is always an outlier and will learn to ignore it. While this may work, the neural network could also be misled and optimize itself incorrectly. A better solution is to set the values as blank allowing the neural networks to predict its value with the observed relationships between the data. Another better solution would be setting the weight of that field to 0 so the model is explicitly instructed to ignore the field.

E. Different Models

The ANN and CNN-LSTM models created in this project use the same architecture between American and European options. This results in some or all the models being less accurate than a model designed to fit that singular problem. As the current models are designed for European options, the models used to predict American option prices should be redesigned. The American option models should have similar activation functions to the models used for pricing American options in [5]. As previously mentioned, the activation functions used in order are ELU, ELU, ELU, and ReLu.

The ANN model architecture is based off the best performing models in [5] while the CNN-LSTM model architecture is based off the research from [7]. The best-performing ANN models in [5] were proven to either perform better than the BSOPM and BOPM in some categories. Contrarily, the CNN-LSTM in [7] only proved that it performed better than an ANN but did not have a comparison to the BSOPM or BOPM. This could indicate that the architecture design for the CNN-LSTM may not be the most accurate and should therefore be tweaked to determine if there are more accurate designs.

F. Learning Rate

Neural networks have a learning rate which determines how much is learned during optimization. A learning rate too small results in slower convergence while a learning rate too big results in divergence. The neural network models created in this project use the Adam optimizer which has a default learning rate of 0.001. For more accurate results, the learning rate should be tailored to the model and dataset which can be achieved with algorithms. Two approaches that are recommended for further analysis is learning rate decay and the low memory Broyden – Fletcher – Goldfarb – Shanno (L-BFGS) algorithm.

Learning rate decay is a method where the model starts with a large learning rate which is reduced in later training iterations. This results in convergence with less oscillation and results in more accurately reaching the minima. The models in [5] used a learning rate which started at 0.001 and was reduced by 90% after 3 epochs of unimproved training loss.

The L-BFGS algorithm is an optimization algorithm that approximates the Broyden – Fletcher – Goldfarb – Shanno (BFGS) algorithm. This algorithm is designed for larger datasets as determining the optimal value for a hyperparameter with larger datasets are more memory intensive. The L-BFGS algorithm determines the local minima through an estimation of the inverse Hessian matrix represented by a few vectors. While this algorithm is suggested to optimize the learning rate, it could also be used to determine optimal values for other hyperparameters [11].

VII. FUTURE WORK

While this project proved the potential of using a CNN-LSTM for option pricing by performing more accurately than an ANN, more research on the topic of option pricing with CNN-LSTMs are needed. The most important extension of this report is getting the CNN-LSTM to be more accurate than the BSOPM for European options and the BOPM for American options. Additionally, the CNN-LSTM was only compared against an ANN. While the CNN-LSTM is predicted be more accurate than other model types for option pricing, it should be confirmed with a comparison against RNNs, LSTMs, CNN-RNNs, and CNNs. The CNN-LSTM could also be used to predict the Delta, Gamma, Theta, and Vega for options which would allow for trading strategies such as delta hedging. Lastly, an ANN should be tested with the same time-series data as the CNN-LSTM to ensure the CNN-LSTM's higher accuracy is due to the differences in model architecture and not a more complex dataset.

VIII. CONCLUSIONS

The objective of this paper is to build a CNN-LSTM model to accurately price options and determine if it is a viable neural network type by comparing its accuracy against an ANN model. In this paper, we have shown that a CNN-LSTM model can more accurately estimate the price of an option compared to an ANN model.

While the BSOPM performs better than both the ANN and CNN-LSTM created in this project, the results indicate that a CNN-LSTM is a better choice for option pricing than an ANN and should be further researched to be more accurate. It was discovered that while the CNN-LSTM performed better than the ANN in every category except one, the ANN was able to better distinguish the features between calls and puts in the same dataset than the CNN-LSTM model.

REFERENCES

[1] A. Ganti, "Option Pricing Theory," Investopedia, 29 September 2021. [Online]. Available: https://www.investopedia.com/terms/o/optionpricingtheory.asp. [Accessed 13 August 2022].

[2] L. Liao, "Stock Option Pricing Using Bayes Filters," Technical Report, Seattle, 2004.

[3] C.-F. Lee, G.-H. Tzeng and S.-Y. Wang, "A new application of fuzzy set theory to the Black–Scholes," *Expert Systems with Applications,* pp. 330-342, 28 April 2005.

[4] B. Madhu, A. Rahman, A. Mukherjee, Z. Islam, R. Roy and L. E. Ali, "A Comparative Study of Support Vector Machine and Artificial Neural Network for Option Price Prediction," *Journal of Computer and Communications,* vol. 9, no. 5, p. 14, 2021.

[5] B. Jiang, "Nonparametric Machine Learning Methods for Equity," Western University, London, 2021.

[6] L. D. S. Maciel, "Pricing Brazilian Fixed Income Options with Feedforward and Recurrent Neural Networks," in *Soft-Computing in Capital Market*, Boca Raton, BrownWalker Press, 2014, pp. 93-107.

[7] W. Lu, J. Li, Y. Li, A. Sun and J. Wang, "A CNN-LSTM-Based Model to Forecast Stock Prices," *Complexity,* vol. 2020, p. 10, 2020.

[8] Wharton Research Data Services, "Ivy DB US," Wharton Research Data Services, [Online]. Available: https://wrds-www.wharton.upenn.edu/pages/get-data/optionmetrics/ivy-db-us/. [Accessed 13 August 2022].

[9] F. Black and M. Scholes, "The Pricing of Options and Corporate Liabilities," *Journal of Political Economy,* vol. 81, no. 3, pp. 637-654, 1973.

[10] R. C. Merton, "Theory of rational option pricing," *The Bell Journal of Economics and Management Science,* vol. 4, no. 1, pp. 141-183, 1973.

[11] D. C. Liu and J. Nocedal, "On the Limited Memory BFGS Method for Large Scale Optimization," *Mathematical Programming,* vol. 45, no. 1-3, pp. 503-528, 1989.