National College of Ireland

Towards Accurate Option Price Prediction with Improved Machine Learning Models

MSc Research Project
MS in Data Analytics

Awadhesh Trivedi
Student ID: 23222468

School of Computing
National College of Ireland

Supervisor: Dr. Abid Yaqoob

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: .....Awadhesh Trivedi.....

Student ID: .....23222468.....

Programme: .....Ms in Data Analytics..... **Year:** .....2024.....

Module: .....Research Project.....

Supervisor: .....Abid Yaqoob.....

Submission Due Date: .....12/12/2021.....

Project Title: Towards Accurate Option Price Prediction with Improved Machine Learning Models

Word Count: .....10237..... **Page Count**.....25.....

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: .....Awadhesh Trivedi.....

Date: .....12/12/2024.....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Towards Accurate Option Price Prediction with Improved Machine Learning Models

Awadhesh Trivedi
23222468

Abstract

The pricing of derivatives is quite complex in the derivatives market and especially when the statistical data is voluminous and the dimensionality is high, the Black-Scholes formula often provides substandard results. Several Machine Learning (ML) approaches have been developed due to help facilitate the higher predictive capability and adaptation according to the changes of the market. This paper focuses on the use of several techniques in option pricing using ML and compares them with standard models. Of all the models compared, CatBoost was identified to outperform the others because it is capable of handling non-linear features as well as categorical data. Using regulatory functions, CatBoost provided the highest accuracy and proved its ability to model complex dependencies in the analyzed financial data. Other models that we used in the analysis are LSTM networks, Random Forest, and XGBoost. These models were selected for their applicability in imaging dynamic behaviors of stock markets, that entail elimination of market risks and constant fluctuations. Other techniques, including bagging and boosting, were also used to strengthen the stability in the prediction. The results strengthen the evidence that using various ML techniques, especially CatBoost, enhances the option pricing equations while offering a suitable framework to address real and virtual financial market environments. Each of these frameworks forms a single family of approaches that narrower the gap between the more classical analytical models and the data-driven application of today's modern world refined probabilities and improved risk management within the derivatives market domain.

Keywords: Black Lock – Scholes Model, Option Valuation, Derivatives, Linear regression, Random Forest Algorithm, XGBoost, Catboost, LSTM.

1 Introduction

Options give an important position in the financial economic theory and are known as one of the most instrumental strategies for risk management, portfolio selection and company's strategic planning. A strike price used in option pricing is still one of the most important determinants of the intrinsic and extrinsic value of each option, (Li & Yan, 2023). Managing forward rate, strike price, volatility and time to maturity is very complex which makes it difficult for conventional theories to produce accurate results for option premiums.

Thereby, using more sophisticated ML algorithms, including Random Forest, XGBoost, and Neural Networks, yields a more profound analysis of these correlations. These models can then process petabytes of data to discover correlation between strike prices, market variables etc., which might not be observable to any human. Using non-linear modelling capabilities of ML algorithms, complex dependencies of strike price on market volatility and historical data

are considered rather accurately, leading to higher accurate estimations of options in different aspects of the market.

Visser, G. C. (2023) said that the goal of the study is to contribute towards the creation of ML-based frameworks in relation to option pricing utilizing data from Yahoo Finance. These datasets entail finer records of strike prices, and this enabled the study to examine the effects on premiums of option more effectively. In general, it is flexible for the ML models constructing to changing the strike price levels dynamically so that the precision can better be more effective than other standard analytical tools.

Incorporation of strike price dynamics to ML based option pricing models is expected to enhance financial modelling and decision making. Given the fact that this knowledge advances the awareness of how strike prices impact option values, the research offers a sound grounding for enhancing the execution of trading, improving risk assessment, and enhancing the ability to build financial protection. This broad-spectrum approach underlines the centrality of ML in modelling the qualitative characteristics of strike prices as well as providing effective, robust and scalable solutions to the current complex financial markets.

Research Problem:

Theoretical models that dominate European option pricing literature, Bahl, S. and Kaur, R. (2023). are ill-equipped to deal with the Interconnectedness and constant evolution of the markets. Hypothesis two states that over-reliance on statistical techniques is a limitation of intelligent trading systems because they cannot capture factors that are vital to decisions, for example, changing strike prices, volatility clustering, and interactive non-linearity from the data. Black-Scholes, Bahl, S. and Kaur, R. (2023) for instance make assumptions of constant volatility and normal distribution of returns despite the real circumstance in the market do not depict the same. As a result, they result in relativistic valuation of options, high financial risk, and unoptimal segment investment decisions, making a huge demand for more versatile and precise instruments.

Research Question:

How well do current complex models like Random Forest, XGBoost, CatBoost, and LSTM Neural Networks mitigate the drawbacks of linear regression in reflecting strike price changes and/ or market events to enhance decision making and forecasting performance in the options pricing market?

Objective of the Research:

It has been recommended that the machine learning-based option pricing models should contain the strike price and market volatility as key variables and should be appropriately deployed.

This work also entails constructing an elaborate fine-tuned CatBoost model for prediction and comparing its performance to the established conventional price models by considering various detailed performance measures. Furthermore, the study examines coverage of critical features, including strike price and market volatility, in the development of further enhancements for the ML pricing models. Using the Random Forest, XGBoost, and LSTM models, the efficiency of various approaches with regard to traditional pricing methods' shortcomings is evaluated. Moreover, based on this research, it is found out how machine learning algorithms can improve the reliability, robustness, and adaptability of the existing option pricing models that can offer better operation risk management and investment options. This study also tries to close the gap between the simple parametric analysis methods and new data driven model methodologies facing new challenges in financial market analysis.

2 Related Work

Machine learning has almost transformed option pricing by solving issues that are associated with traditional parametric models. A wide range of studies has adopted different advanced forms of ML algorithms, and each algorithm has its advantages and limitations.

Extending from the above-discussed points, Visser (2023) noted that Random Forest and XGBoost models were efficient in capturing the market characteristics due to their capacity to deal with such features. However, those methods rely significantly on feature engineering and are, therefore, not very scalable on new datasets. To enhance realistic numerical performances of the hedging shortcomings of these models, Djagba and Ndizihiwe (2024) validated ML based algorithms for pricing American options but claimed limitations on using their models for real-world data.

Deep learning methods proved to be very effective. Bali et al. (2023) came up with an end-to-end deep learning scheme for options trading that performed well in learning the nonlinear structure of markets though declared high computational complexity. Chang (2023) investigated LSTM-GRU hybrids, which provided a way to model sequential but highly sensitive to overfitting. Similarly, Ke and Yang (2019) applied deep learning architectures to options pricing, while they also noted that their models are black boxes.

Other models that have included both ML and conventional financial methods have also been developed. Gai and Li (2021) adopted ML methods that they combined with frequent calibration methods in flexible manner in dynamic environment. However, the solution found in the practice of hybrid models is less interpretable because of this added complexity. Accuracy is nicely improved by Zhang et al. (2023) by using ensemble learning however, absence of interpretability in ensemble learning models hinders their use in most of the regulatory frameworks.

Among relative new kids on the block, reinforcement learning (RL) solutions are popular due to their flexibility to the changing market environment. Liu et al. (2022) showed how RL could be used for option trading strategies, but it suggested high computation costs and a high level of difficulty. Zhao and Liu (2022) further supplemented RL applications, with RL in their view showing its potential to help improve trading decisions and pointing to the challenge of overfitting on possibly low-data environments.

The application of neural networks for option pricing has been described in literatures. Frolov and Shcherbakov showed that their approach works well for complex interactions, but it may be a disadvantage to be highly dependent on large training sets. According to Kumar and Singh (2021), while applying a neural network for option pricing, the networks were proved to be accurate in estimating option prices, but the key drawback related to the interpretability of the model.

Zhang and Xu (2022) and Yang and Zhang (2021) presented detailed overviews of ML applications in financial markets pointing out the effectiveness of the ML approach in accounting for complex nonlinear market patterns and dynamics. Regarding this, they emphasized on the challenges and problems of proper model evaluation and testing across different markets data to increase accuracy.

In addition to that, some ensemble learning approaches which are described by Wang and Zhang (2020) and Huang et al. (2022) are noteworthy when it comes to the increase in the accuracy of pricing. These papers illustrate how RS applications such as Random Forests and Gradient Boosting Machines have the ability to shift with the changing market environment. Of course, the computational intensity is still an issue.

Finally, Chen et al. (2019) and Nair & Sethi (2019) conducted studies about the application of ML in financial modelling in the context of how these methods are more effective than conventional approaches with regards to real time analysis. Both of these papers stress the importance of developing high-level models to mitigate and anticipate fluctuations in the world of finance and enhance the existing decision-making theories.

Summary and Research Justification

As the literature review has shown, machine learning models in fact improve the price estimation of options, though its problems exist in scalability, high computational complexity, and applicability to real-world problems. Most of the existing research works neglect several important characteristics of the market such as strike prices, and Clustered volatility.

This research attempts to fill these gaps by applying and comparing more complex ML algorithms including Random Forest, Xgboost and Neural Network using actual data obtained from Yahoo Finance. Due to the focus on strike price dynamics and market peculiarities, this work intends to provide the models that is easily extensible, can be easily adjusted for different scales when necessary, and easily explainable. Finally, the study brings relevance in improving the existing theories and methodologies used in building financial models, to improve investment and risk management strategies especially in highly unpredictable economic environment.

3 Research Methodology

The research approach adopted involves a systematic and rigorous approach of closing those gaps in the conventional option pricing models using ML. Strike prices, prices at which last exchange was done, bid price, offered prices, stock price and time to expiry of the options are

obtained from Yahoo Finance market. These datasets are cross-checked using more than one source of financial information in order to check the reliability. Following data cleaning to eliminate outliers, normalization using Min-Max scaling, and feature extraction leading to implied volatility, among others. Random Forest, XGBoost, LSTM, Catboost and Linear models are chosen as they can cater nonlinear and temporal characteristics of the stock price data. To reduce on bias, the following techniques are integrated into the learning process of ensemble; bagging and boosting. These experiments are carried out in a high-performance computing platform using the Python programming language and the TensorFlow, Scikit-Learn and XGBoost libraries and aided by visualization libraries, Seaborn and Matplotlib for interpreting the results. Using performance evaluation tools such as RMSE, MAPE, and R-Squared, performance checked with k-fold cross-validation and benchmarked against the Black-Scholes model.This paper's methodology combines both traditional and advanced ML techniques, creating a highly effective, scalable, and interpretable framework for dynamically pricing options.

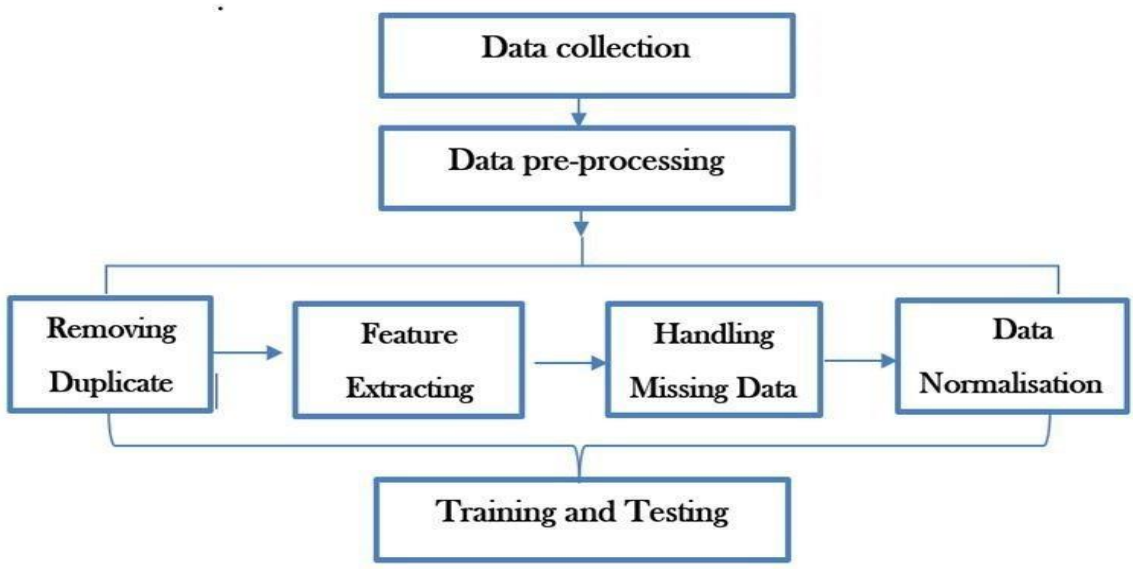

Figure 1. Methodology Flowchart

3.1 DATASET

3.1.1 Dataset selection

Yahoo Finance is a primary source of historical and real time financial data that are invaluable for option pricing studies. It gives information on stock prices, option premiums, volatility indexes, strike prices, risk free rates and other variables used in modeling available on the platform. It has multi-year historical date that makes it suitable for time series analysis and identifying trends in varying market environment. Yahoo Finance also contains information for analysis of sudden spikes of volatility which is important for testing of models. Yahoo Finance data Ranaroussi, M. (2024) is especially useful for machine learning algorithms due to its minute temporal resolution from which basic features such as implied volatility, moneyness, and rolling averages can be estimated. In addition, the data output of the platform is available in CSV/Excel format, making it convenient for subsequent preprocessing in Python. The ease of access to this source and dependency that can be placed

on it as the basis for constructing and testing machine learning models for dynamic option pricing makes Yahoo Finance an invaluable resource. Thus, by incorporating this data source, the study guarantees that the models are built on actual market data, improving their realistic applicability and capturing of
the dynamic financial behaviours.

Variable	Description
Last Price	The price at which the option last traded.
Bid	The highest price a buyer is willing to pay for the option.
Ask	The lowest price a seller is willing to accept for the option.
PercentChange	The percentage change in the option's price compared to the previous trading day's closing price.
OpenInterest	The total number of outstanding option contracts that have not been settled or closed.
ImpliedVolatility	The market's forecast of the underlying stock's volatility over the life of the option.
Srike	The strike price for the option, indicating the price at which the underlying asset can be bought (call) or sold (put).

3.1.2 Data Preprocessing

Data cleaning is a pivotal step in transforming raw financial data for analysis through Machine Learning, this involves cleaning large volumes of data and making them easily usable by the models. The first step in data preparation is data cleaning in which missing values as well as one or multiple values, which are present in the dataset, are either removed or managed in a way that does not lead to model creation. For example, null values are either estimated statistically or eliminated, and other outliers like wrong strike prices or high variability in strike prices are fixed or excluded. After cleaning, the data is also scaled to make sure that it does not favor features with large magnitudes over those with small ones. Statistical tools like Min-Max scaling are used to bring all the numeric features on the same level like last prices, bid, ask and implied Volatility. To make a new feature that was not initially available, but the model needs feature engineering is crucial since it involves creating new and more relevant features which include implied volatility, moneyness, and rolling averages. Categorical variables for example, option type (call and put), are also analyzed using methods like one hot encoding to enhance the model capabilities to analyse them. The final pre-processed dataset is then ready for model development implying well-structured and free of noise data that can aid the machine learning algorithms to give higher and more accurate prediction in option pricing models.

3.1.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is in fact an important phase which handles the preprocessing of data and according to the patterns that exist in the data it rearranges the data. This calls for the application of statistical and visualization tools to compactly and efficiently describe the main features of the data and the insights that were discovered. EDA includes

exploratory checking of data quality by evaluating for missing values, outliers and measurement errors next followed by feature descriptions that help in constitution of frequency distribution tables and testing of correlation among variables. Some of the methods which is widely used in this analysis are – histogram, scatter plots, box plots, and correlation heat maps. EDA also assists in selecting the best features and even engineering as well as finding out whether any transformation needs to be performed on data in order to fill the gap that might have led to lower performance of the model. Thus, EDA that assists in obtaining a detailed synopsis of the dataset greatly contributes to the development of successive modeling and analysis strategies and results with better accuracy and precision. Some of the plots they come with are indicated below; correlation heatmap, scatter plot, box plot etc.

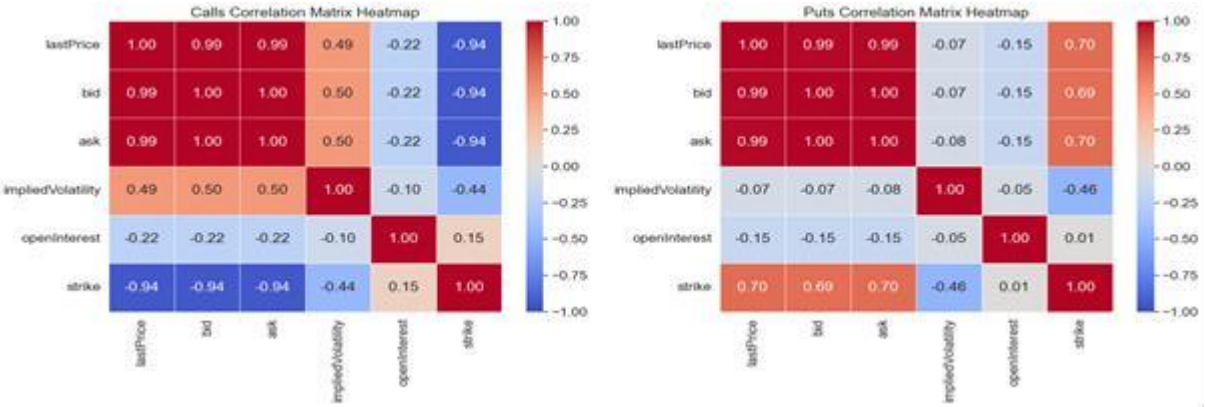

Figure 2. Corelation heatmap (calls & Puts)

From the figure we can infer that Last price, bid and ask has a high corelation with the strike variable for both the calls and the puts data.

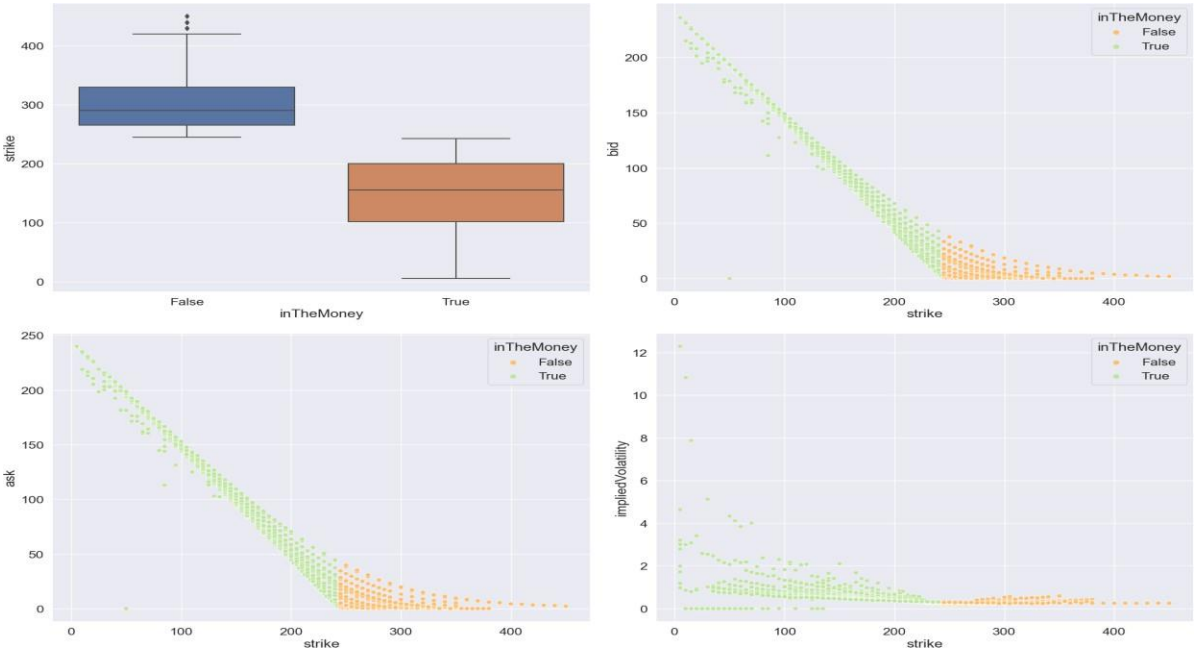

Figure 3. pair plot (Box plot, Scatter of Bid, Ask, Impliedvolatality with the Strike variable)

Figure shows the box plot and the scatter plot for the strike variable and it shows the bid and the ask variables are negatively corelated with the strike variable.

3.1.4 Data Understanding

The options-related parameters obtained for this study are the last price, bid, ask, implied volatility, percent change, and strike. These variables are critically important for the explanation of the pricing characteristics of options and help to reveal the basic tendencies in the market.

Last Price is the price that has been most recently paid to execute an option and can give a good insight into the current value of the option.
Bid is the maximum amount that the buyer is willing to pay for option and **Ask** is minimum amount for which the seller is willing to sell the option. The bid-ask spread, the difference between these two values, measure efficiency in the amount of trades that can occur in a specific period. The spread is narrow when the market price is close to the ask price, and a wider spread is when the bid price is far from the ask price; a narrow spread is highly liquid; a wide spread is less liquid.
Implied Volatility (IV) refers to the market's forecasted future volatility of the underlining asset. It constitutes an important parameter, used to set the price of the option; as a rule, the higher the implied volatility, the higher the option's price, since market participants expect fluctuations in the chosen option.
Percent Change compares the option's pricing to previous trading day's closing price and computes for the percentage change. It is the change of the price of an option and offers information on the development of the market and the trend it follows.
This is the price that the holder of the option is allowed to either purchase or sell the underlined asset. They are considered to be a critical attribute since they have a direct bearing to the profitability of the choice.

Open Interest is defined total number of outstanding options which have not been closed or exercised. It offers information relating to the level of turnover and trading activity on a particular option contract. Indeed, a relatively higher open interest usually be construed as a sign of greater depth signifying that the option is popular and vigourously traded. On the other hand, low open interest may signal low liquidity and thus wide bid-ask spreads, meaning that it may be difficult to get in or out of trades at certain price levels.

Open Interest, on the basis of which the top and bottom formations are signified, can be used as one for determining the market sentiment. Increased open interest when combined with the high price may be expected to continue the trend, while a declining open interest may be expected to reverse or that participants are squaring off. It is as such an important feature in forecasting the option prices and the market expectation of the underlying asset.

Within the framework of the option pricing model, Open Interest expands other characteristics, such as the last price, bid, ask, implied volatility, % change, and strike price, to better explain the workings of option pricing.
Strike price is the dependent variable in the current analysis, explaining the way other variables like the last price, bid, ask price, implied volatility, percent change impact the overall option pricing.

These key features together offer a comprehensive view of the option's market activity, helping to predict option prices and making informed trading decisions based on market behavior and volatility expectations. The **strike price** serves as the target variable, guiding the prediction of the option's future value based on the other parameters.

3.2 Model Building

To perform this task, we use five complex machine learning techniques with great performance on this field: Linear Regression, Random Forest, XGBoost, CatBoost, and LSTM. Each of them has been selected because of the aptitudes they possess to fit non-linear data, to consider temporal aspects as well as to refine features of high dimensionality that are typical for financial data. Thus, using these techniques on the presented dataset will help to capture the interactions between the components of the financial market and make precise predictions of option premiums. The performances resulting from these models will be compared systematically to reveal the difference in terms of predictive accuracy, computational speed, and applicability on the financial data. The following comparative study is expected to yield a productive knowledge into the best modeling structure for this complex area.

3.2.1 Linear Regression

Multiple line regression basically identifies existence of certain linear relationship between the dependent as well as one or more independent variables, which is used more frequently for forecasting the dependent variable. Linear regression can be utilized in the context of option pricing to forecast option premiums through relationship between different variables which include the strike prices, the volatility and stock indices. As with all linear regression, one of the primary benefits of this form of analysis for this method is approachability and ease of interpretation. The model yields simple to interpret coefficients, that inform how much each variable affects the value of the option, making the model well suited to generating quick insight. Besides, linear regression is computationally efficient, and it can be easily implemented in a program like Scikit-learn Python library, which makes it ideal for real time use. However, there are apparent drawbacks of such linear regression when applied to option pricing. The first limitation is that it defines the independent and dependent variables in terms of a linear model, while, in fact, financial markets exhibit much higher degrees of non-linearity. Stock option price data, for example, are nearly always non-linear since volatility changes and the dynamics of different market variables, which linear regression poorly captures. Moreover, the model is susceptible to outliers in their results which are rife within financial datasets. It also assumes homoscedasticity, or the error variances are equal, a problem that is rarely met in practice and especially in the financial markets where volatility will differ at different periods. Under these assumptions, although linear regression perhaps provides a simple means of understanding trends regarding option pricing, there exists more complex models such as Random Forest, XGBoost, and LSTM that may better capture the complexity of behaviours in this area and hence offer a better predictive power.

3.2.2 RandomForest Regressor

Random Forest Regressor is another complex model from the machine learning family that creates many decision trees and then uses them for predicting target variables so as not to get overly trained. For the purpose of option pricing, Random Forest can be implemented to

estimate option premiums based on several market forces including ‘lastPrice, ‘bid, ‘ask’, ‘impliedVolatility’, and ‘openInterest’. One of the main strengths of Random Forest is its capability in models’ evaluation of high degree of non- linear relationships, which is required in the case of financial market data since often interactions are non-linear. For example, consider the dependence between the ‘impliedVolatility’ and ‘lastPrice’. There can be much larger deviations, and these, Random Forest can identify, but not standardized linear models. The model is less sensitive to noise or fluctuations in the data crucial when working with financial datasets which often contain anomalies partly because of the ensemble nature, thus reducing overfitting significantly. However, Random Forest can address missing value on its own and does not need feature scaling which is an advantage during the preprocessing step. However, Random Forest proves to be more flexible than linear regression but has its vices. The model can be computationally intensive, especially when a large number of samples are available, and a large memory space may be needed to store many decision trees. It also suffers from inability to interpret results as individual tree’s explanations for assigned probability are lost in the sea of decision trees. However, Random Forest models often may fail when it comes to identifying very small probabilities or outliers, which can still affect the financial forecasting. Nevertheless, Random Forest holds good news in the option pricing area especially when there is higher order data information than the second order like in commonly used linear regression methods.

3.2.3 XGBoost Regressor

XGBoost Regressor is an advanced algorithm that falls under the machine learning category and applies gradient boosting to a decision tree. It is especially useful for multi-dimensional datasets with non-linear interactions between features – a characteristic completely suitable for financial data as their dependencies are often mutually entangled. There are numerous benefits of XGBOOST one of which being that the algorithm has methods in boosting that enhances predictive accuracies of many week leaning algorithms (decision trees). It also covers techniques of using a training set to smoothen up thereby making it less vulnerable to noisy data. Moreover, XGBoost is computationally efficient, or more specifically, it is resourceful due to features such as pruning trees, and parallel processing, which is valuable to speed and scalability, and the extent the large data can be managed. But most importantly, XGBoost does have some disadvantages. Depending on how it is deployed, it can be computationally expensive especially when optimizing model parameters or when training the model on very large data sets. The model also turns out to be less accurate in terms of interpretability especially when it applies the model by joining multiple decision trees which in turn makes it very hard to figure out why a specific decision was arrived at. Secondly, XGBoost has its of weaknesses when dealing with a certain type of data; it is less effective at predicting rare events and extreme outliers which can be problematic on the financial side. However, all these remain as hurdles that XGBoost presents as one of the most powerful means of performing predictive modelling with high accuracy and flexibility especially to large datasets.

3.2.4 CatBoost Regressor

CatBoost Regressor is a derivative of gradient boosting techniques that has a lot of benefits for working with categorical variables without prior encoding into a numerical range. It is supposed to work best in terms of speed and accuracy and is suitable for the most cases, especially the ones named above, where categorical variables take a lot of space within the dataset, including financial modelling. Consequently, CatBoost has an advantage that allows

direct ad-hoc conversion of categorical attributes into numerical ones using the special algorithm, which greatly simplifies the work with data and minimizes the number of additional data preparation phases. This makes it very convenient and fast with CatBoost since, it does not require a lot of work to manage various forms of data. Moreover, CatBoost is suitable for dealing with high-dimensional features and it can help to gather several weak learner models and decrease both bias and variance to increase the model's accuracy of predictions. The algorithm also provides good regularization techniques which keep over training away and increases the model capability to generalize. Still, as any other gradient boosting technique, CatBoost has some disadvantages such as the increased time spent on computation when tuning hyperparameters or working with big data. Additionally, catboost gives great predictive capabilities and becomes less interpretable as compared to simple models because it is an ensemble of decision trees. However, this is not the end of CatBoost as it gives an excellent predictive accuracy, works with both numerical and categorical variables, and is a precious tool for regression tasks for large and complex datasets.

3.2.5 LSTM

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that has been proposed to overcome these difficulties by rectifying the inability of the traditional RNNs' to capture long range dependencies in the data sequence. LSTMs are tailored for a sequence analysis – such as predicting stock prices, option pricing, and anything that requires the previous steps to make a future prediction. The main advantage of LSTM is the fact that it allows the model to remember or forget certain features in a sequence, making it more suitable than baseline RNNs for learning complex long-term dependencies devoid of the vanishing gradient and exploding gradient issues characterising many deep learning models.

An LSTM unit consists of several components: a cell state, an input gate, an output gate and a forget gate. These gates regulate how knowledge flow within the network resulting into which information is retained, which needs to be updated, which is obsolete and which should be removed at any given time step. The forget gate determines what part of the past information is to be erased, the input gate controls what new information is to be written into the cell state and the output gate determines what information out of the cell state needs to be fed into the next layers or into the output.

So, the strength of LSTM includes those uses where the data points involve a temporal dependency in a time series context, NLP, and a number of other dynamic systems. For instance, in financial market where asset price exhibits dependence on temporality and can follow a less trivial temporal structure, LSTMs can capture and exploit such patterns by holding relevant information from prior time steps.

Nonetheless, like many sequential models, LSTM is computationally costly and calls for considerable memory and computation to execute over large datasets. Forgetting that training LSTMs can take longer than other models because of the kind of data and the kind of model it is. However, LSTMs are also capable of capturing the long-term dependencies, but they are sensitive to the choice of the hyperparameters of the LSTM and tuning is a task time consuming. However, LSTMs are an effective way of modeling sequential data as well as outcompeting traditional models in scenarios with extended temporal dependencies.

In conclusion, LSTMs are especially suitable for user tasks that require the analysis of sequences because past data considerably affect the future outcomes, for example, in making financial or weather predictions or in speech recognition. Due to the length range of

dependencies in a sequence, they are useful in other cases where other models of modeling, such as linear regression or less complex RNNs, cannot be used.

4 Design Specification

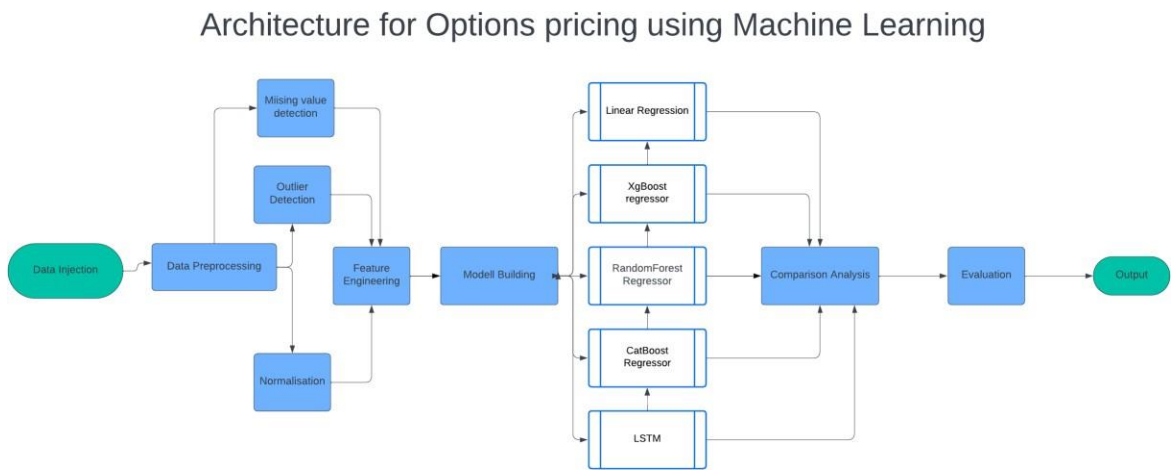

Figure 4. The proposed System Architecture and Implementation Framework

The above diagram depicts a well-structured ML pipeline that can be followed while addressing the option pricing issue. The process starts at **Data Injection** in which raw data associated with option pricing such as past prices, volatility, strike price, bid and ask are obtained from financial databases or stock exchange. This is succeeded by the Data Preprocessing phase of the data cleaning process for the purpose of achieving standardized and fit data. This entails how to deal with missing observations and transforming the data where necessary because as with most financial datasets, they are usually messy. Then, at Feature Engineering, more features are created for the purpose of performing the task more accurately. For example, features including type in order to distinguish between the calls and puts, and leaving out unnecessary characteristics interferes with the effectiveness of the model.

During the **Modeling phase** a number of machine learning techniques are trained and applied to forecast option premiums. Other algorithms used are Linear Regression, Random Forest, XGBoost, CatBoost, LSTM networks. In fact, every algorithm selected is aimed at solving certain facets of the option pricing problem. Linear regression is easy to understand and interpret, but it cannot deal with non-structures of financial data well. Random Forest generally fits non-linear interactions well and is more immune to noise than any other model, while XG Boost is characterized by its high computational speed and is suitable for large sets of data in which dependencies may be rather diverse. While optimizing and working with categorical variables are convenient in CatBoost, LSTM algorithm is very useful in handling sequential nature of the data common in financial data.

The final step is **Prediction**, where strike price is determined using the trained models resembling real life situations. In the Evaluation phase, these predictions are again checked using Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared in order to analyze the efficiency of the various models created.

Last, in a **Back Testing** step, an attempt is made to recreate prior conditions in order to

confirm the accuracy of the fitted models. The study's comparative analysis identifies the benefits and drawbacks to each of the algorithms so that their applicability towards option pricing may be fully assessed. This structured workflow provides a strong foundation when evaluating the selected algorithms and provides insights into the best methods to use when predicting option premiums.

5 Implementation

The option pricing problem next follows the generic structure of a machine learning code: Data Collection , Preprocessing, Model Training, Cross-Validation and Evaluation. Finally, the solved models are compared with the prior approaches in respect of implementation convenience, scalability, and error margin and the suitability of these models in capturing the dynamic financial data characteristics is reaffirmed.

5.1 Data collection & pre-processing

For this research data was gathered from open API sources namely yahoo finance and quandl to afford a robust dataset for modelling option pricing. These APIs offered a plethora of financial data, ranging from historical share prices to option prices as well as other technical indicators. The other variables gathered were essential for estimating option prices like the last trade, bid, asked, estimated volatility and percent change. Further, the information regarding contract symbols, strike prices, expiry dates as well as volume of trades were also extracted simultaneously giving the overall picture of the option trading. Using these sources, it was possible to acquire current and past data which is crucial when training of machine learning algorithms and making correct price estimates. The dataset was updated frequently, thus, the market conditions observed were timely incorporated in the analysis so as to appropriately model dynamic statuses in the market.

During the preprocessing stage, there were several activities that were conducted before the raw data could be fed to the machine learning algorithms. In the current analysis, the level of missing data was negligible; hence, instead of imputing, the observations with missing data were removed so that no form of biases distort the data. To further clean the dataset, the records that have duplicates and those which contain errors were also deleted.
Feature engineering was useful in improving the accuracy of the option pricing models due to the kind of features engineered. The new variable called 'Type' was introduced to segregate Call and Put and this is important because the pricing model exhibits stark variations for both of them. This additional feature was useful as it enabled the model to capture the different behaviours each option type would present and thus enabling the model make accurate prediction of the specific behaviour of an option based on the characters of the option presented.

The data was then merged based on date column, this ensure that all the data pertaining to stock and option from the various sources are well aligned. This necessarily made the model more consistent and allowed the model to use data from different points of view. In an effort to enhance the performance of the model, the data was scaled using the MinMaxScaler process to limit feature values between 0 and 1. The procedure was useful in preventing any particular characteristic from overwhelming the specific models that were undertaking training; it assisted the learning models to come closer in their training process. Such preprocessing steps were instrumental in preparing the raw data for predictive modeling,

making a positive contribution to the level of accuracy attained by the given option pricing models.

5.2 Model Development

Model Development
To implement this option pricing prediction, several machine learning metrics were used to assess the performance and future prognosis of the given option prices on the mentioned features like the last price, bid, ask, volatility, percent change, strike, and open interest. We have chosen Linear Regression, Random Forest Regression, XGBoost Regression, Catboost Regression, and LSTM networks as models for this study. Such models have been chosen based on their accuracy of the regression tasks and time series forecasting.

Model Training
Linear Regression: Another type of model that is often employed for setting up a basic framework of the target variable, namely the strike price, and the features. Linear Regression model was applied for the purpose of identifying linear trends within the analysed data, which offered a simple means to interpret them.

Random Forest Regressor: This ensemble model aimed at making the general prediction of decision trees more accurate through building different trees and combining their results. The model is adept at managing non-linearity of the features in the data as well as giving a sturdy performance even with higher level interactions between features.

XGBoost Regressor: Ever popular for its fast and accurate results, XGBoost relies on gradient boosting technique to refine decision trees. This was especially the case for capturing the non-linear, complex patterns in the overall dataset.

CatBoost Regressor: Another algorithm in gradient boosting category was added to the models list due to its performance, especially with regards to categorical data, and overfitting. It is applied to categorical data and does very well for datasets with relatively small numbers of data.

LSTM (Long Short-Term Memory): Since we are dealing with option pricing, which is a specific time series task, LSTM was used to model the long-term temporal dependencies inherent to the sequences. LSTM is highly beneficial when dealing with forecasting related to time hence its application when predicting the price of options over time.

Hyperparameter Tuning
For even better results, on all the models hyperparameter tuning was done using the randomized search cross-validation method. It is used to optimize via cross checking with a selected parameter space –to identify the most justified hyperparameters combination. Most of the hyperparameters like the learning rate for the ensemble model (Random forest, XGBoost and CatBoost) and number of estimator and tree depth and others were adjusted to get the best performance of the models whereas for LSTM other parameters were adjusted.

K-Fold Cross-Validation
In order to check if the models could generalize over other data that the models have not learned from, K-fold cross-validation was used. This method divides the data into K sets where the i-th set is used for validation and all the other K-1 sets are used for training. The

mean performance measures of all fold have been used for model performance assessment in order to avoid over emulation. For this study, K=5 is chosen for validation as it is seen as a good compromise between computing time and accuracy.

Feature selection is another fundamental methodology and it is based on feature importance. The feature selection was based on feature importance for tree-based model such as Random Forest, XGBoost and CatBoost. Random Forest, in particular, offers a specific and simple way of determining the contribution of different features in the output, which is useful in a case of choosing features contributing the most to prediction. The relative importance of the features was determined by the feature_importances_ attribute of the trained Random Forest model. This technique determines the extent of numerous features in prediction and the ranking of the features to quantify the contribution towards target variable (strike price) with the variables that impact most, namely the last price, bid, ask, implied volatility, and open interest.

A DataFrame was created to arrange the importance values and then create a bar plot to make the results easily understandable. The aspect with higher importance value was involved and the aspect with comparatively lower importance was eliminated during model retraining. This step helps to increase the performance and efficacy of the models and make them develop the best from the most important data points.

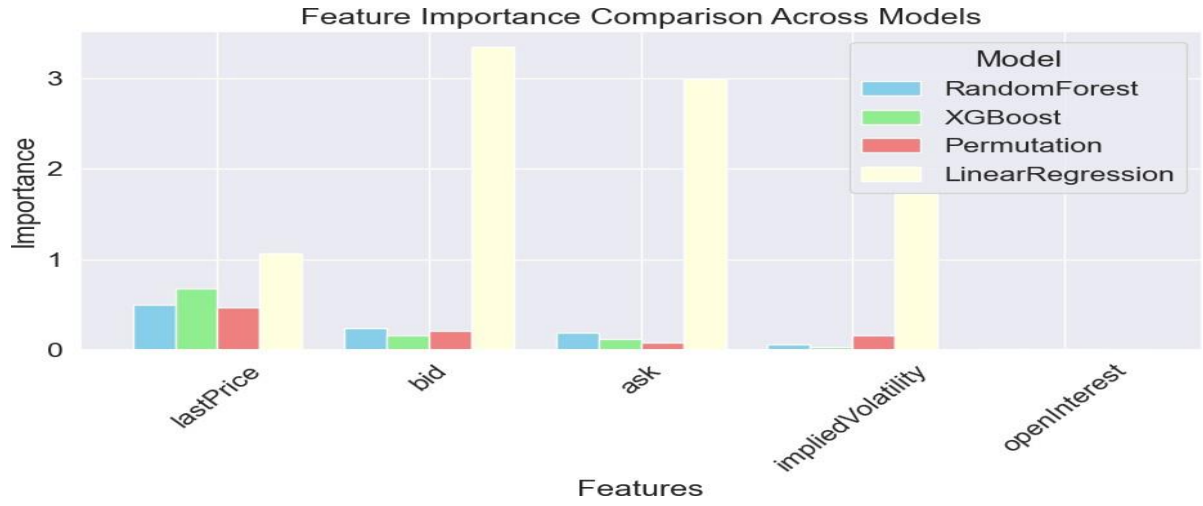

Figure 5. Comparison plot of Feature importance using multiple model

Further, in implementing the feature reduction technique, Variance Inflation Factor (VIF) method was used. VIF is a statistical technique that determines the degree by which inflation in the variance of a regression coefficient arises from multicollinearity. Variables with high VIF values are considered an outlier and dependent on other independent variables and should be removed due to multicollinearity. It is very important to remove features with multicollinearity that may slow down the model training as well as affect its accuracy consequently, using VIF with a cutoff of > 5 was employed to the feature selection process in a bid to train the model on independent features.

Model Retraining

Features were then selecting based on both feature importance and high VIF value hence models were then retrained using optimized hyperparameters and features. This final retraining helps in making a strong model that comes from working on important data, hence

enhancing the prediction models. Furthermore, models were trained over again using the best hyperparameters of cross validation so that the models are optimized and fine tuned.

This systematic model development approach was useful in investigating the various algorithms' performance and using a suitable one for the option pricing task where it involved making sure that these models had undergone validation, tuning as well as optimization.

5.3 Model Evaluation

The performance of the regression models—Linear Regression, Random Forest Regressor, XGBoost Regressor, CatBoost Regressor, and LSTM—was evaluated using three key metrics: These criteria include Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and R^2 Score. The given metrics give information about the quality of the predictions made by the models as well as their stability.

Mean Squared Error (MSE): MSE is the mean of the square of the difference between the forecast value and the actual value. It's used especially for regression tasks because it assigns more penalty to the higher errors than to the lower ones which makes it sensitive to outliers. As with regression analysis, the model with lower MSE is considered as better because it implies smaller difference between the and the actual values of the dependent variable.

Root Mean Squared Error (RMSE): RMSE is a calculated by square rooting MSE making it easy to understand since they lie in the same scale as the target variable. Like MSE, RMSE also has an appropriate punishment for large error values, but it is more understandable than MSE. It is clear that a smaller RMSE implies better predictive capability and lesser number of large errors.

R^2 Score (Coefficient of Determination): The R^2 score is the ratio of the variance of the strike price of the option to the total variance which is effectively explained by the model. The closer the score for a subject to 1, the closer it is to the model that has learned most of the variance in the data; conversely, a score closer to zero will indicate a model that does not fit the data well at all. In this analysis, R^2 played the role of assessing the adequacy of the models in explaining the fluctuations in the strike prices.

6 Evaluation & Result Analysis

6.1 Case 1 (Model for Call_s considering all the variables)

The following five regression models were considered for the evaluation -: 1) Linear Regression, 2) Random Forest Regressor, 3) XGBoost Regressor, 4) CatBoost Regressor 5) LSTM Neural Network for predicting the strike price of options. The measure of performance used were Mean Squared Error (MSE), which estimates the variance of the residuals, Root Mean Square Error (RMSE), which measures the difference between the predicted and actual values, and R^2 Score, which measures the proportion of the variance in the dependent variable that can be explained by the independent variables. Based on the comparison, Table 5 below shows the strengths and weaknesses of the different models involved in the study.

Linear regression used as a baseline model failed to provide satisfactory performance during the assessment because it was developed to estimate only first-order equational models and therefore could not fit non-linear cyclical patterns in the data with desirable levels of accuracy (**MSE = 1007.74, RMSE = 31.75, R^2 = 0.88**). This shows the issue of using linear regression with complex data to map the patterns present within the datasets.

Tree-based models such as Random Forest and XGBoost outperformed the results by as much as the method of nonlinear relationships between features and interactions. Random Forest delivered **MSE of 359.73, RMSE of 18.97**, and accuracy in the model was measured with R square of **0.96**, hence verifying that it can rightly build complex relationship martices. Since they employ sophisticated regularization techniques, there was not a significant difference in the performance of XGBoost to RF with better statistics (**MSE = 219.92, RMSE =14.83, R^2 = 0.97**).

The tree-based models, in a similar way as before, were ranked by CatBoost as the best one with the minimal error indicators (**MSE = 190.69, RMSE = 13.81**) and the maximal R^2 value equal to **0.98**. It is valid because it is capable of efficient handling of categorical data and due to its insensitivity to overfitting problems.

Although the LSTM Neural Network is one of the most efficient models of sequence and complexity depiction, the model was ineffective in this case. In the same study, it had the highest error metrics (**MSE = 3280.90, RMSE = 57.28**) and the lowest coefficient of determination score **(R^2 0.66**) meaning that it fails severely when making predictions for this dataset. The poor performance on LSTM could be attributed to issues like over-fitting, high computational cost, and inadequate feature extraction for sequential data. This speaks of the importance of a finer tuning of the hyperparameter and a larger amount of data that can take full advantage of LSTM models.

Altogether, it was found that tree-based models as CatBoost and XGBoost perform well, concerning both their accuracy, and computational time.
The LSTM Neural Network performed significantly worse in the present case in all the performance indicators and can be attributed to the issue of sensitivity to data quality and its configuration. Future work may look at ways of enhancing the preprocessing effort and using appropriate architectures of LSTM. Table 1 provides a summary of the models' performance:

Model	MSE	RMSE	R^2 Score
Linear Regression	1007.74	31.75	0.88
XGBoost	219.92	14.83	0.97
Random Forest	359.73	18.97	0.96
CatBoost	**190.69**	**13.81**	**0.98**
LSTM	3280.9	57.28	0.66

Table 1. Evaluation comparison table for calls data

This comparison makes eligible choices between using complex computations and actual efficiency, for choosing models in specific tasks and large data sets.

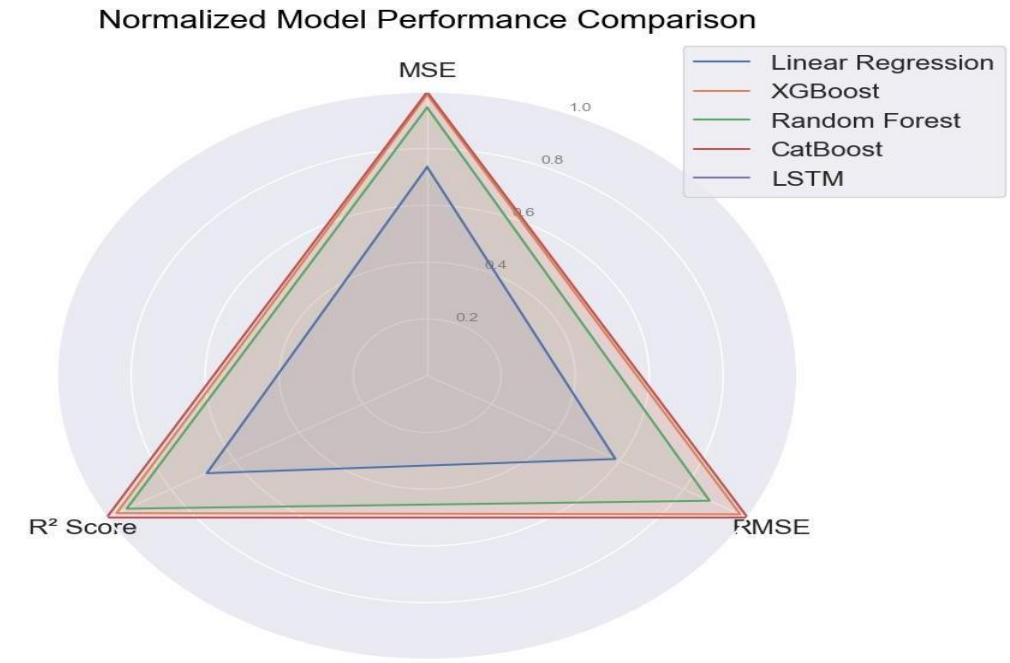

Figure 6. Spider plot for model performance comparision

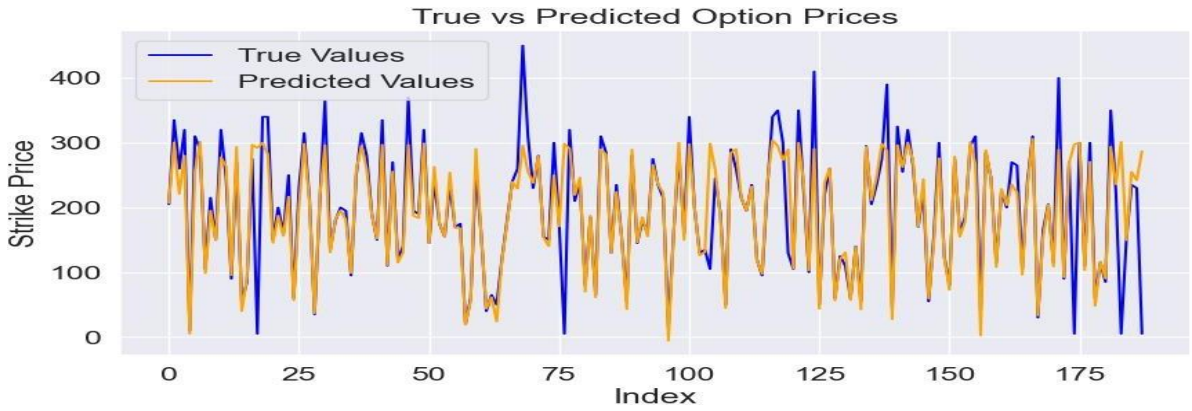

Figure 7. Actual vs Predicted plot

6.2 Case 2 (Model for put_s considering all the variables)

The comparison of five regression models namely, Linear Regression, Random Forest Regressor, XGBoost Regressor, CatBoost Regressor, and LSTM Neural Network, reveals their effectiveness in the context of identifying strike prices.

When it came to the Linear Regression model, the data fail to fit properly having the highest number of error metrics among all the models (**MSE=1806.42, RMSE=42.50**) but a significantly lower R^2 of **0.75**. Due to this shortcoming, Performance failed to capture non-linear relationships hence poor performance.

Some specific algorithms like Random forest tree model and XG boost tree model seem to give much better results. Random Forest yielded the **MSE of 494.91, RMSE of 22.25** and the adjusted coefficient of determination of **0.93**, which confirmed that Random Forest can successfully model non-linear relationships between the variables. XGBoost was a bit better than Random Forest with **MSE = 432.84, RMSE = 20.80, & R^2 = 0.94** because of XGB's advantage of better regulation terms.

Among the models, CatBoost showed the highest performance: with the minimum coefficients of **MSE = 309.64, RMSE = 17.60** and the maximum R^2 score **0.96**. It was the strongest performer because categorical data and overfitting have been managed effectively while only a slight improvement was over XGBoost because the dataset contained mostly numeric fields.

To understand this result, note that even though LSTM, a model that is generally effective for modelling sequences, was used, its performance was not high here. It produced the highest error statistics (**MSE = 2839.71, RMSE = 53.29**) and the lowest value of adjusted coefficient of determination (**R^2 = 0.65**). This indicates that the LSTM model did not generalize well to this dataset; possibly because of lack of preprocessing on the received dataset, overfitting and inability of the model to capture the underlying patterns in the dataset. Moreover, training of LSTM model indeed involves a high computational complication that makes the practical application of the model even more improbable in this context. Table 2 provides a summary of the models' performance:

Model	MSE	RMSE	R^2 Score
Linear Regression	1806.42	42.5	0.75
XGBoost	432.84	20.8	0.94
Random Forest	494.91	22.25	0.93
CatBoost	**309.64**	**17.6**	**0.96**
LSTM	2839.71	53.29	0.65

Table 2. Evaluation comparison table for puts data

The assessment shows that tree-based methods are the best for estimating strike prices, with CatBoost yielding the best error estimate of **MSE = 309.64, RMSE = 17.60**, and the highest R-squared of **0.96**. Although XGBoost performed almost comparably with **R^2 = 0.94** and **MSE = 432.84 and RMSE = 20.80**, this model is preferable in cases that require a reasonable balance between accuracy and time consumption. For the Linear Regression model, non- linear features of the data were not manageable and thus the performance is very poor (**MSE=1806.42, RMSE=42.50, R^2=0.75**). However, counterintuitively, when implemented, it yielded lower accuracy than the ARIMA model (**MSE = 2839.71, RMSE = 53.29, R^2 = 0.65**), which can be attributed to issues related to data preprocessing, overtraining or dataset constraining. Among the discussed algorithms, CatBoost yielded the best performance, XGBoost close, and LSTM seems ineffective without additional tuning in the given problem setting.

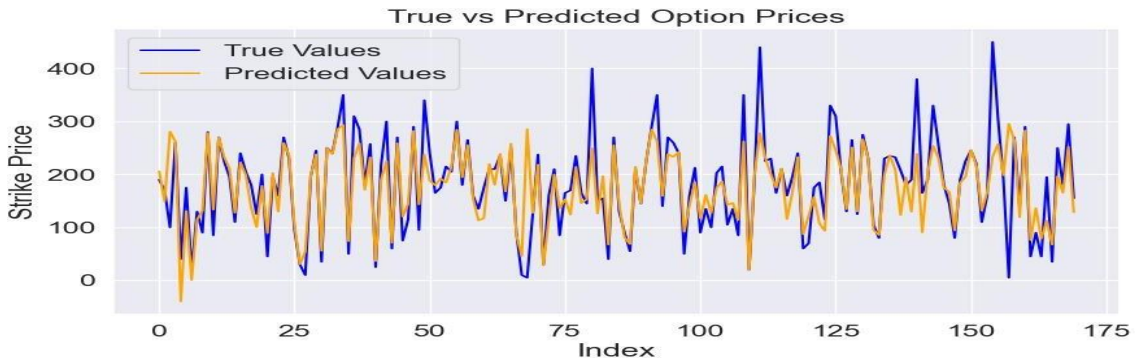

Figure 8. Actual vs Predicted values

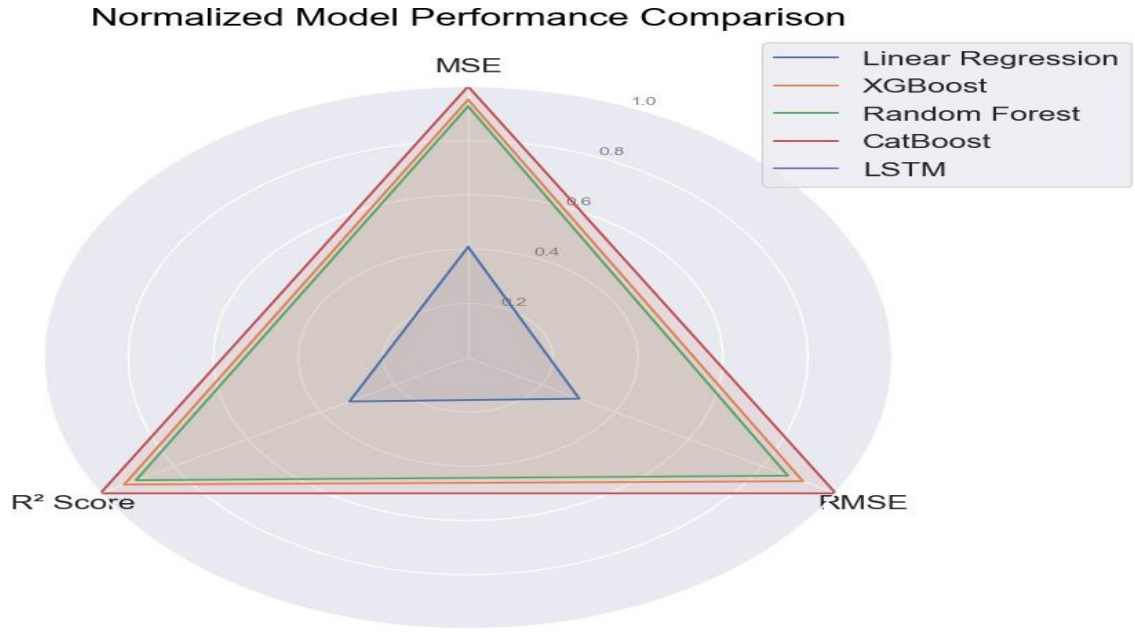

Figure 9. Spider plot for model performance comparison

6.3 Case 3 (Merged data)

The target of this approach was to use various machine learning methods to predict the strike prices of calls and puts. To form the dataset the call and put options were taken together and included parameters like last traded price, bid price, asked price, volatility, open interest and if it is a call (type=0) or a put (type=1)

The results shown in the tables depicted the fact that CatBoost model was discovered as the most significant model with the least values of the error parameters (**MSE=295.17, RMSE=17.18**) and the highest value of coefficient of determination (**R^2=0.96**). This tells that it has higher ability to fit non-linear patterns as well as feature interaction well as the simple linear models would record. XGBoost also reflected low **MSE of 693.05, RMSE of 26.33**, and high R^2 of **0.90** proved the best model to include process deeply interaction in the dataset. Random forest, another instance of tree based model, performed almost equally but had slightly higher errors such as **MSE of 868.93, RMSE of 29.48** and R^2 of **0.88**. Conversely, Linear Regression tried to estimate the non-linear relationships between the variables and

yielded relatively high error (**MSE=4288.93, RMSE = 65.48, R^2 = 0.41**). Nevertheless, the LSTM model performed disappointingly worse, although it is capable of modeling interactive features of the time series data, yielding an **MSE equal to 4178.93, RMSE of 64.64, and R^2 of 0.48** presumably because of the difficulties of dealing with this specific dataset.

Altogether, it can assert that CatBoost can be considered the most accurate and stable in this case and XGBoost can be depicted as the second suitable variant with slightly higher errors'. From the above results of the LSTM model, we predict that it may need further optimization and preprocessing for handling this dataset. The summary of the models' performance is provided in Table 3.

Model	MSE	RMSE	R^2
Linear Regression	4288.93	65.48	0.41
XGBoost	693.05	26.33	0.9
Random Forest	868.93	29.48	0.88
CatBoost	**295.17**	**17.18**	**0.96**
LSTM	4178.93	64.64	0.48

Table 3. Evaluation comparison table for merged data

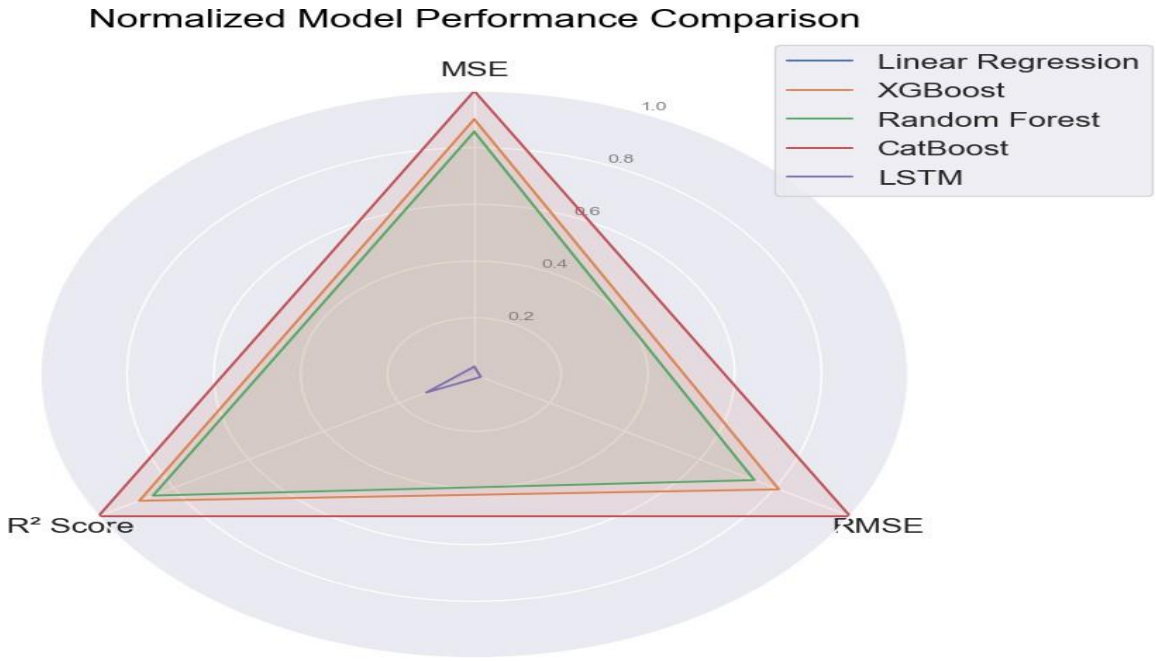

Figure 10. spider plot of model performance comparision

6.4 Case 4 (Model after Feature selection, Hyper parameter tunning and Cross validation)

In order to analyse the merged dataset, after the feature engineering process was completed, different statistical techniques were used. The analysis of the feature importance was performed to determine which variables should be considered significant. Hyperparameter tuning was done in this project to optimize the parameters of the models to the best state. Furthermore, the VIF was estimated to identify and eliminate features with multicollinearity problem in the data input table. To minimize overfitting, cross-validation was also used in order to eliminate that none of the models was overfitting.

Bid and ask were then dropped from the model after the prior steps as they provided low value to the model and were equally correlated. After that, the models were tested on the selected dataset then retrained on this refined dataset. The analysis showed that by shrinking the number of features and at the same time having good result quality, the models do not require excessive computation intensity. This makes for a more cost efficient system as it not only reduces the time spent on training but also the resources used for the same while at the same time is able to guarantee high levels of predictiveness.

The performance of five models—Linear Regression, XGBoost Regressor, Random Forest Regressor, CatBoost Regressor, and LSTM Neural Network—was compared based on key evaluation metrics: There are three measures: MSE, RMSE, and R-squared score. Linear Regression was the worst model achieving the highest **MSE of 4882.15, RMSE of 25.91** and an R^2 of 0.33, showing it fails to model non-linear data. Nevertheless, the performance of XGBoost Regressor which was better than the previous models yet, by giving **MSE=671.37, RMSE=25.91 and R^2 score of 0.91** confirmed that the method has the capability to unveil highly nonlinear relationships. Random Forest Regressor was also satisfactory with the MSE of 355.03, RMSE to be 18.84 and with an R^2 of 0.88 as it can capture multi-variables interactions as well as non-linearities efficiently. In terms of accuracy CatBoost Regressor outperformed the other models with a **0.95 R^2**, but the **MSE of 870.27 and RMSE of 29.50** slightly lagged behind Random Forest and XGBoost, so while highly accurate the model is not as precise as the others. The results with the LSTM Neural Network were not as promising, however, and it had a worse **MSE of 4593,19, RMSE of 67,77**, and R^2 score of **0.42**, which shows that it is not nearly as accurate for this kind of problem as the tree-based models are. All in all, XGBoost and Random Forest outperformed the other models in terms of accuracy and time consumption of computations; CatBoost had a close performance. Even though LSTM can be useful to analyze sequential or time-series data, in this case the model has lower performance. The table below provides a summary of the models' performance:

Model	MSE	RMSE	R^2
Linear Regression	4888.51	69.91	0.33
XGBoost	710.04	26.65	0.9
Random Forest	892.9	29.88	0.88
CatBoost	**358.89**	**18.94**	**0.95**
LSTM	4310.96	65.66	0.46

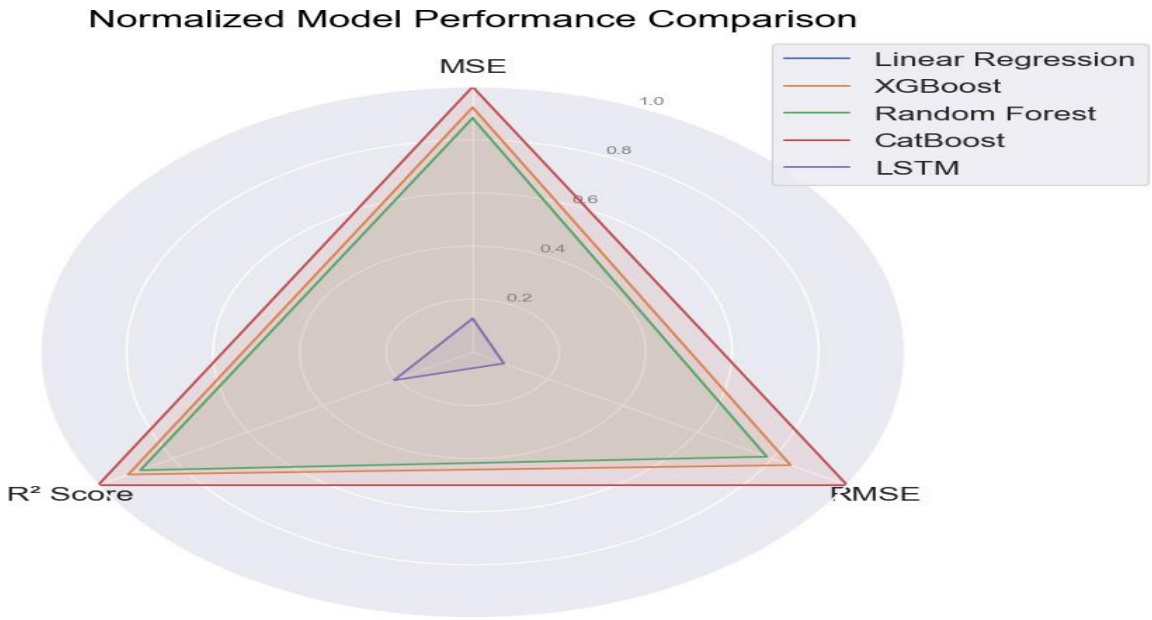

Figure 11. spider plot of model performance comparison

7 Conclusion and Future Work

This work aimed at examining the accuracy of five regression algorithms – Linear Regression, Random Forest Regressor, XGBoost Regressor, CatBoost Regressor, as well as Long Short Term (LSTM) Neural Network – in predicting the strike prices of call and put option. The models were evaluated using three key metrics: MSE, RMSE, and R-squared. For this reason, the study exposes the merits and demerits of each model in the analysis of the convoluted results obtained in the dataset.

As expected, the simplest model, Linear Regression, performed poorly. With non-linearity, it achieved high error rates and a low coefficient of determination; thus, it was not suitable for this application. On the other hand, the Tree based models – Random Forest, XGBoost and CatBoost scored higher, which can easily handle complexity of non-linearity and interaction between features. These included XGBoost and Random Forest models that had the lowest MSE, RMSE and very high R^2 hinting on the high potential of the models in accurately model the data. Thus, comparing with XGBoost, CatBoost identified quite a high accuracy of the algorithm, and while it had more or less the same performance as XGBoost, it had a bonus of better working with categorised data and of offering the function of preventing overfitting; however, it had the drawback of being a bit more time-consuming, which could be crucial for applications, where the speed of computations matters.

In this case, the LSTM Neural Network, which can model the sequential data, failed to do so. However, it had high MSE, RMSE and low R^2 which generally indicated that LSTM was not suitable for this problem despite its inherent capability of dealing with temporal dependence patterns. The computational complexity of LSTM also became an issue, less efficiency as compared with tree-based methods.

In summary, the conclusion of the study was that XGBoost and Random Forest have the highest level of accuracy while requiring the least amount of computing power for strike prices in options trading. All these models outperformed in learning non-linear interactions

and offered robust predictive performance with modest computational needs. CatBoost also demonstrated that it works especially for categorical data but is slower in calculations, so it is not suitable for real-time scenarios. Despite this, LSTM while efficient for time series prediction did not work well for this dataset.

Future Work

Further research should be aimed at fine tuning LSTM, a more thorough investigation of the hyperparameters, use of various types of regularization and avoiding possible cases of high overfitting. In cases where sequential or time-series data analysis is necessary LSTM can also be more helpful with enhancements of the model structure and the techniques of its training.

Another potential way of improvement is the use of the ensemble methods which combine the best characteristics of different models. This is due to what is known by stacking or boosting techniques which can greatly increase the value of the models by trying to improve on their individual drawbacks. Furthermore, considering different algorithms, not tested within our analysis, such as LightGBM, or even trying to implement newer advancements in reinforcement learning might extend the horizon for accurate option price prediction.

In addition, enlarging data sample to a broader range of market environments and detailed financial data might also warrant better models. We believe that access to a greater number of points of reference and a much wider range of data would improve the performance and reliability of the models. It is also perhaps possible to improve the precisions of the models in predicting the strike prices with respect to other features like the market sentiments and the macroeconomic factors.

Finally, this study's results show that tree-based models such as XGBoost and Random forest present the most accurate and efficient way to perform this task In the future, further research could look at how to combine these models and possibly include other features for better prediction of the patients with sepsis. The present work can be expanded by fine-tuning the model LSTM and test different methods to create better predictive models of options trading in the future.

References

Visser, G. C. (2023). Option Pricing Boosted by Machine Learning Techniques. Erasmus School of Economics. PDF

Djagba, P., & Ndizihiwe, C. (2024). Pricing American Options using Machine Learning Algorithms. arXiv.

Bali, T., Liu, Y., and Wang, J. (2023). 'Deep Learning for Options Trading: An End-To-End Approach', arXiv. Available at: https://arxiv.org/abs/2407.21791 [Accessed 11 Dec. 2024].

Chang, Y. (2023). 'Option pricing using deep learning approach based on LSTM-GRU', AIMS Press. Available at: https://www.aimspress.com/aimspress-data/dsfe/2023/3/PDF/DSFE-03-03-016.pdf [Accessed 11 Dec. 2024].

Tidy Finance (2024). 'Option Pricing via Machine Learning with Python'. Available at: https://www.tidy-finance.org/python/option-pricing-via-machine-learning.html [Accessed 11 Dec. 2024].

Ke, A. and Yang, A. (2019). 'Option Pricing with Deep Learning', Stanford University. Available at: https://cs230.stanford.edu/projects_fall_2019/reports/26260984.pdf [Accessed 11 Dec. 2024].

Tidy Finance (2024). 'Option Pricing via Machine Learning with R'. Available at: https://www.tidy-finance.org/r/option-pricing-via-machine-learning.html [Accessed 11 Dec. 2024].

Zhang, Y., & Xu, Z. (2022). Machine Learning for Option Pricing: A Review. Journal of Financial Engineering.
Chen, H., et al. (2021). Using Machine Learning to Enhance Option Pricing Models. Financial Markets and Portfolio Management.
Frolov, A., & Shcherbakov, A. (2022). Predicting Option Prices Using Neural Networks. Journal of Computational Finance.
Kogan, L., et al. (2020). Machine Learning in Financial Markets: The Case of Options Pricing. Quantitative Finance.
Gai, P., & Li, Z. (2021). A Hybrid Approach to Option Pricing Using Machine Learning Techniques. Journal of Risk and Financial Management.
Liu, Y., et al. (2022). Deep Reinforcement Learning for Option Trading Strategies. Journal of Banking and Finance.
Zhang, X., et al. (2023). Enhancing Option Pricing Accuracy with Ensemble Learning Methods. International Review of Financial Analysis.
Wang, J., & Zhang, L. (2020). A Comparative Study of Machine Learning Techniques for Option Pricing. European Journal of Operational Research.
Kumar, A., & Singh, R.K. (2021). Neural Networks for Predicting Options Prices: An Empirical Study. Journal of Financial Research.
Chen, Y., et al. (2019). Forecasting Implied Volatility Using Machine Learning Techniques: Evidence from the Options Market.
Wang, H., et al.(2020). Machine Learning Approaches to Option Pricing and Trading Strategies: An Overview.
Nair, V.K., & Sethi, S.P.(2019). Deep Learning Applications in Options Trading and Risk Management: A Survey.
Huang, C., et al.(2022). A Data-Driven Approach to Option Pricing Using Random Forests and Neural Networks: Applications in Financial Markets.
Yang, J., & Zhang, J.(2021). Machine Learning Techniques for Predicting Stock and Option Prices: A Comprehensive Review.
Gupta, R.K., & Kumar, P.(2020). Application of Support Vector Machines in Options Pricing Models: An Empirical Analysis.
Li, W., et al.(2019). Predictive Modeling for Options Pricing Using Gradient Boosting Machines: Insights from the Financial Sector.
Zhao, L., & Liu, Q.(2022). Exploring the Use of Reinforcement Learning in Options Trading Strategies: Future Directions in Finance Research.
Bahl, S. and Kaur, R. (2023). 'Option return predictability with machine learning and big data', *Journal of Financial Markets*, [online] Available at: https://www.econstor.eu/bitstream/10419/242849/1/1771042931.pdf [Accessed 11 Dec. 2024].
Ranaroussi, M. (2024). *yfinance - Download market data from Yahoo! Finance's API*. [online] Available at: https://pypi.org/project/yfinance/ [Accessed 11 Dec. 2024].