# Advanced Machine Learning in Finance - COMP0162 Coursework

**Avlonitis Ektor**
23165499
ektor.avlonitis.23@ucl.ac.uk

## Abstract

This coursework explores the application of machine learning techniques to predict the implied volatility skew in financial markets. Focusing on Long Short-Term Memory (LSTM) networks, a model is developed that outperforms traditional volatility forecasting methods. By analyzing various options within the Euro STOXX 50 index, the study demonstrates how machine learning can provide more accurate predictions of market dynamics, offering significant improvements over the Black-Scholes model.

## 1 Introduction

The Black-Scholes option pricing model (BSOPM), formulated by Black and Scholes in 1973, is still a fundamental tool in the financial markets, despite its assumptions, such as constant volatility, being widely criticized. This criticism has led to the development of numerous other option pricing models with the goal of outperforming the BSOPM in terms of market predictions. (Bakshi et al., 1997)

The Black-Scholes formula for a Call option, $C_t$, at time t is described as follows:

$$C_t = S_t N(d_1) - Ke^{-r(T-t)} N(d_2) \tag{1}$$

where $S_t$ represents the spot price of the underlying asset, $K$ the strike price, $r$ the risk-free interest rate, and $T - t$ the time to maturity, with $N(d)$ indicating the cumulative distribution function of the normal distribution and $\sigma$ the volatility of the asset. The formulas for $d_1$ and $d_2$ integrate these variables, enabling the calculation of the option's market value based on these inputs.

Research has increasingly focused on implied volatility rather than direct option pricing. Even with its drawbacks, the Black-Scholes model's implied volatility, which frequently captures information not represented in historical volatility, is considered a crucial signal for option traders. Implied volatility is considered a forward-looking metric that reflects the market's expectations regarding the future volatility of the underlying asset. (Jiang and Tian, 2005)

Implied volatility is important because it can be used instead of the market price in options quoting. Instead of just looking at the current price of an option, traders use implied volatility to get a better idea of how the market expects the asset to move. This helps them decide how to price options and develop trading strategies. So implied volatility is used by traders to predict future market changes and make more informed decisions. The vega of an option, which shows how the price of an option is expected to change with volatility, captures the relationship between an option's price and its volatility. Vega is always positive and thus this relation implies that an increase in implied volatility results in options with higher prices and vice versa. (Carr and Wu, 2011)

### 1.1 Advances in Volatility Modeling and Prediction

Implied volatility research goes beyond simply predicting option prices. It rather concentrates on taking advantage of the dynamics of the implied volatility surface to improve option pricing and hedging tactics. This approach reflects a shift from traditional models, emphasizing the significance of understanding and modeling implied volatility for financial decision-making. A notable improvement is the Heston model, which introduces stochastic volatility to better mirror market dynamics. This model allows for a more realistic representation of how volatility changes over time, addressing a major drawback of the constant volatility assumption in the Black-Scholes model. (Heston and Nandi, 2000)

The Black-Scholes option pricing model, which assumes volatility is constant, has been criticized for not reflecting real market conditions. This led to a focus on the volatility skew, showing that implied volatility varies with option strike prices, a fact that became especially clear after the 1986 market crash. Options with lower strike prices (out-of-the-money puts) tend to have higher implied volatilities, partly because investors use them to protect against potential losses. This concept, known as the volatility skew, demonstrates that the market expects different levels of price movement depending on the option's strike price, contradicting the constant volatility idea in the original model. (Chen and Zhang, n.d.)

Understanding the volatility skew provides a deeper insight into market expectations than the flat volatility landscape of the Black-Scholes model. The real value of the volatility smile, which is the curve showing how implied volatility changes across options with different strike prices, comes from its ability to improve future option pricing. Traders can use insights from the volatility smile to more precisely price upcoming options, aiding in the development of option strategies. Understanding tomorrow's volatility patterns allows for better hedging, spotting undervalued options, and overall smarter strategic choices.

There has been a growing application of machine learning techniques to further refine the accuracy of volatility predictions and option pricing. These cutting-edge methods are becoming key in navigating the financial market's complexities, offering fresh perspectives and increasing the precision of trading strategies.

### 1.2 Machine Learning Innovations in Implied Volatility Prediction

Recent focus in financial markets, especially options trading, has highlighted the importance of predicting Implied Volatility (IV) for better investment decisions and risk management. IV represents the market's prediction of how widely an asset's price might vary in the future. However, forecasting the entire IV surface, which varies across different strike prices and expiration times, is challenging due to its complex, high-dimensional nature. (Chen and Zhang, n.d.)

Historically, models such as GARCH (Generalized Autoregressive Conditional Heteroskedasticity) and HAR (Heterogeneous Autoregressive model) have been used to forecast realized volatility based on historical price data. These models are primarily designed for predicting realized volatility, which is calculated from past price movements of the asset (like standard deviation). These models fall short with implied volatility, which reflects future market expectations and is derived from options prices, not historical data. This gap underscores the difficulty of predicting the IV surface due to its intricate and multidimensional characteristics, leading to limited research in this area. (Chen and Zhang, n.d.) To overcome these challenges, recent research has turned to machine learning (ML) techniques, which are well-suited for analyzing the complex, high-dimensional IV surface. Techniques such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks have proven to be particularly effective, outperforming traditional forecasting models by capturing complex data patterns in IV prediction. (Vrontos et al., 2021) The evolution of forecasting methodologies has introduced hybrid models that merge neural networks' analytical power with the structural benefits of GARCH models. These hybrids specifically tackle the IV skew issue, offering more accurate options pricing and risk management by reflecting the IV surface's variations with strike prices. (Medvedev, n.d.)

### 1.3 Uniqueness of This Project

Unique to this project is the innovative approach adopted for predicting the volatility skew. Instead of treating the IV surface as a tensor, implied volatilities for different strike prices are organized into separate columns within the dataset (e.g., IV_4000, IV_4500, etc.). This structure allows for the creation of distinct datasets for each option while still incorporating the implied volatilities of other options. This methodology diverges from traditional approaches by assuming that the implied volatility of one option is influenced not only by its historical data but also by the historical volatilities of other options across different strike prices. The uniqueness of this approach is also discussed in the data processing part of the project.

This project's methodology stands out for its segmented and comprehensive approach. It keeps separate datasets for every option, divided into groups according to the strike price, and incorporates volatilities from a variety of other strike prices to integrate the larger market context. This strategy enables a more detailed analysis, acknowledging that each segment of the volatility skew is potentially influenced by movements in other segments. This comprehensive approach facilitates understanding of how different market segments affect each other, offering valuable insights into the dynamics of the volatility skew.

## 2 Methodology

### 2.1 RNN

Recurrent Neural Networks (RNNs) are a class of neural networks that are specialized for processing sequential data. Unlike traditional feed-forward neural networks, RNNs possess the unique feature of maintaining internal memory, known as hidden states, which helps them understand context and patterns over time.

At the heart of RNNs is the hidden layer, which updates its state at each step by looking at the new input and the previously received information. This is mathematically represented as follows:

Let $x_t$ be the input vector at time step t and let $h_t$ represent the hidden state at the same time step. The update of the hidden state can be expressed by the equation:

$$h_t = \phi(Wx_t + Uh_{t-1} + b) \tag{2}$$

In this formula, $W$ and $U$ are weight matrices associated with the input and the previous hidden state, respectively, while b is a bias vector. The function $\phi$ represents the activation function, typically a non-linear function like tanh or ReLU, applied element-wise to the linear combination of inputs and previous hidden states. RNNs are great for tasks where understanding the order and context of data is crucial, such as voice recognition or language translation. However, they can struggle to remember information from far back in the sequence, a problem known as vanishing or exploding gradients. Despite these challenges, RNNs have introduced the way for more complex models like LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) networks, which are better at handling long-distance relationships in data. (Zhang et al., n.d.)

### 2.2 LSTM

Long Short-Term Memory networks (LSTMs), created by Hochreiter & Schmidhuber in 1997, tackle RNNs' shortcoming of maintaining information over long periods. LSTMs are complex, with four key layers that manage memory efficiently, proving useful in language modeling and time series prediction by holding onto relevant information and ignoring the rest. (Zhang et al., n.d.)

As seen in Figure 1, each LSTM cell is composed of a cell state and three distinct gates, which regulate the flow of information, making them particularly adept at managing memory in sequence learning tasks:

Input Gate (it): This gate controls the extent to which a new input is allowed to modify the cell's internal state. It decides whether or not to let new information in. Mathematically, this is represented as:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

Here, $\sigma$ is the sigmoid activation function, making the gate's output range between 0 and 1. $W_i$ are the weights, $h_{t-1}$ is the previous hidden state, $x_t$ is the current input, and $b_i$ is the bias. The result of this gate tells the cell how much of the new information to add.

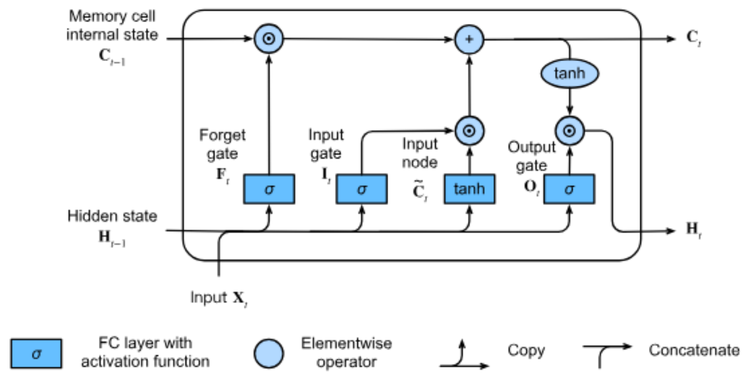

Figure 1: LSTM cell structure with gates and operations for handling sequences. (Zhang et al., n.d.)

Forget Gate ($f_t$): Determines the amount of the previous cell state to retain or discard. If the forget gate outputs a value close to 0, it means "forget almost everything," while a value close to 1 means "retain everything." The forget gate allows the LSTM to discard irrelevant parts of the earlier state for long-term efficiency and is calculated as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{4}$$

In this equation, $W_f$ is the weight matrix and $b_f$ is the bias corresponding to the forget gate.

Output Gate ($o_t$): This gate controls the extent to which the internal state affects the output. It decides what next hidden state ht should be. The output gate's function is similar to a filter, determining which parts of the cell state are passed to the output:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

Then, the cell updates its internal state $C_t$ in a two-step process: first, it decides what to forget and what new information to add, and then it updates the old cell state $C_{t-1}$ into the new cell state $C_t$, combining the information allowed through the forget and input gates:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{6}$$

Here,

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{7}$$

represents the candidate values to be added to the state, as seen in the Figure below. Finally, the cell decides its new output:

$$h_t = o_t * \tanh(C_t) \tag{8}$$

These mechanisms allow LSTMs to remember important information over long sequences, enhancing their performance in tasks like natural language processing and time series analysis. (Zhang et al., n.d.)

This research focuses on using LSTMs for their proficiency in handling long-term dependencies in sequential data. However, a comparative study with an Autoregressive (AR) model which will be used as a baseline model, will provide insight into the LSTM's capabilities versus traditional statistical methods in time series forecasting.

### 2.3 Autoregressive Model

Autoregressive (AR) models are statistical tools used for analyzing time series data, where predictions about future points are made based on past values. These models are particularly useful for understanding and forecasting data that changes over time. An autoregressive model of order p, called AR(p), is defined by the equation:

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \varepsilon_t \tag{9}$$

where $X_t$ is the value of the time series at time t, c is a constant, $\phi_1$, $\phi_2$, ... , $\phi_p$ are the parameters of the model, and $\varepsilon_t$ is white noise. The "order" (p) tells you how many past values the model considers. For instance, AR(1) looks one step back, AR(2) looks two steps back, and so on. AR models are widely used in various sectors like economics and environmental science because they offer a straightforward way to capture the dependency of current data on its history. Unlike complex neural networks such as RNNs and LSTMs, which analyze sequences by maintaining hidden states, AR models simply use past data points in the series without remembering earlier inputs beyond what's included in the model. This simplicity makes AR models easier to use and understand, though they may not capture complex patterns as effectively as some neural network models. Variations of AR models like Autoregressive Integrated Moving Average (ARIMA) and Seasonal ARIMA (SARIMA) models add functionality to AR models, allowing them to capture trends, seasonality, and external influences, and thus broaden their applicability in forecasting. (Zhang et al., n.d.)

## 2.4 Data Description

In this report, a carefully selected dataset is examined comprising historical data on eleven put options from the Euro STOXX 50 index, all being European style. This means these options can only be exercised on their expiration date, which is on April 19, 2024. The dataset behind this study was sourced from Refinitiv Eikon, which is a platform known for its comprehensive financial data and analytical tools. The selection of options extends over a variety of strike prices. Strike prices are the pre-determined prices at which the underlying asset can be bought or sold when the option is exercised. This diversity in strike prices allows us to cover a broad spectrum of the market's view on risk associated with these options. By analyzing the predicted IV across these different strike prices, the project aims to map out a pattern known as the volatility smile or skew. These patterns provide insightful revelations about market sentiment and investors' expectations. The reason all that is done is to forecast what the skew is going to look like and thus help us make informed investment decisions. The put options utilized in the dataset span from out-of-the-money (OTM) to at-the-money (ATM) and in-the-money (ITM). Prioritizing ATM options due to their higher liquidity ensures greater reliability in the analysis. However, to capture valuable insights into market sentiment and expectations, OTM and ITM options are also included, exhibiting distinct implied volatilities reflecting potential price movements. By diversifying the dataset, a broader range of market conditions and responses is encompassed, enhancing the analysis. Deep ITM or OTM options are excluded due to their typically low liquidity and limited information on implied volatility movement.

### 2.4.1 Data processing / cleaning

In the data preprocessing stage for an option pricing model project, several steps are undertaken to clean and prepare the dataset for analysis. The initial dataset comprised multiple CSV files, each representing options data with varying strike prices. The files include columns such as Exchange Date, Close, Net, %Chg, Open, Low, High, Volume, O-C, H-L, %CVol, Bid, Ask, Implied Volatility, Ask Implied Volatility, Bid Implied Volatility, Delta, Theta, Gamma, Open Interest, Rho, Vega, Mid Price, Strike Price, and Days to Maturity.

The preprocessing involves the following steps:

- **Conversion and Cleaning:** 'Net' and '%Chg' columns are dropped due to their irrelevance to the model's objectives. The 'Exchange Date' column is converted to datetime format for accurate temporal analysis. The signs '+' and '%' inside values are removed to normalize numeric fields. Only dates with data across all strike prices in the dataset are retained.
- **Feature Engineering:** A new column **Strike Price** is added representing the strike price of options. Also, the **Days to Maturity** column is added which calculates the difference between the 'Exchange Date' and a fixed maturity date of April 19, 2024. An additional column **Close_Index** is added, incorporating the prices of the underlying asset, the Euro STOXX 50 index. A new column for **Log Forward Moneyness** has been added, which is calculated using the formula $\ln(F/K)$, where $F$ is the forward price of the underlying asset and $K$ is the option's strike price. This forward price is derived based on the Black-Scholes

model assumptions, incorporating the risk-free rate obtained from the Euro area's 1-year maturity Triple-A rated government bond yield curve. (European Central Bank, 2023)

- **NaN Values Handling:** Rows with NaN values in 'Implied Volatility' or 'Close' are removed to ensure model reliability. Other columns' NaN percentages are calculated to assess data completeness, guiding further preprocessing choices. Columns such as 'Open', 'Low', 'High', 'Volume', 'O-C', 'H-L', '%CVol', 'Bid', 'Ask', and 'Mid Price' are dropped due to the high percentage of missing values, significantly exceeding 60% in most cases, making them unreliable for predictive modeling.

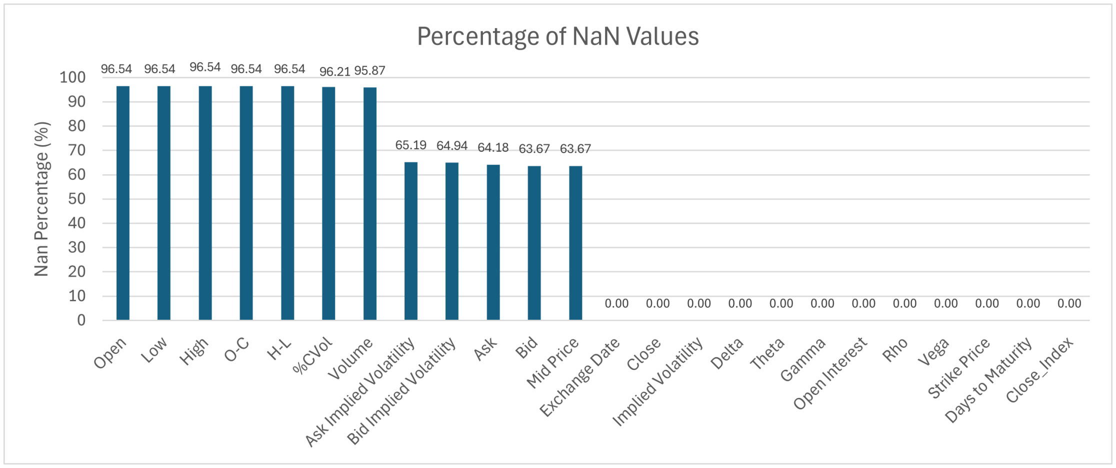

Figure 2: Percentage of NaN values per column

After completing the feature selection step, the dataset is further prepared to ensure it is ready for the model. This project's approach deviates from traditional methodologies outlined in the literature by adopting a more innovative strategy for predicting the implied volatility skew. Unlike the other methods that generate a vector to predict the entire skew collectively, this model focuses on forecasting the implied volatility for each option separately and then aggregates these predictions to build the volatility smile. This technique is executed by utilizing the implied volatilities of the other options (with different strike prices) as features in each model, alongside the specific option's characteristics such as 'Close', 'Delta', 'Gamma', among others. Consequently, each option data is fed into the model with a distinct dataset that not only reflects its own attributes but also incorporates the influence of implied volatilities from other strikes. This unique approach allows us to capture the intricate relationships between different strike prices, offering a good understanding of the factors driving the implied volatility skew.

Next, the 'Exchange Date' and 'Strike Price' columns are dropped because in each dataset the strike price contains only a single value and therefore is redundant for the prediction process. The features used for each prediction are shown with their descriptions in the table below.

Table 1: Description of Data Fields

| Feature | Description |
|---|---|
| Exchange Date | Date on which the option's data was recorded. |
| Days to Maturity | Number of days until the option expires. |
| Close Index | Closing price of Euro STOXX 50 index. |
| IV 4500 to IV 5000 | Implied Volatility (IV) for options at different strike prices (from 4500 to 5000). |
| Close | Closing price of the option on the exchange date. |
| Delta | Option's Delta: Price sensitivity to a change in the price of the underlying asset. |
| Theta | Option's Theta: Rate of change of the option's price with respect to time. |
| Gamma | Option's Gamma: Rate of change of delta over time. |
| Open Interest | Total number of outstanding option contracts that have not been settled. |
| Rho | Option's Rho: Rate of change of the option's price with respect to the interest rate. |
| Vega | Option's Vega: Price sensitivity to changes in the implied volatility of the underlying asset. |
| Strike Price | Price at which the holder of the option can buy/sell the underlying asset. |
| Log Forward Moneyness | Natural logarithm of the ratio of the forward price of the asset to the strike price. |

## 3 Results

### 3.1 Hyperparameter Tuning

In the process of developing the Long Short-Term Memory (LSTM) model for time series forecasting, one crucial step is the implementation of hyperparameter tuning to enhance the model's predictive accuracy and performance. This process involves experimenting with different configurations of model parameters to find the combination that yields the best results based on predefined criteria.

#### 3.1.1 Number of steps

One of the primary hyperparameters tuned is the "Number of Steps" (n_steps). The n_steps parameter determines the length of the input sequences fed into the LSTM model. It specifies how many previous time points should be considered to predict the future value. In sequential data like time series, a sliding window approach is used because it organizes the data into fixed-size sequences, making it suitable for LSTM models. This method helps the model learn from recent trends to predict future values.

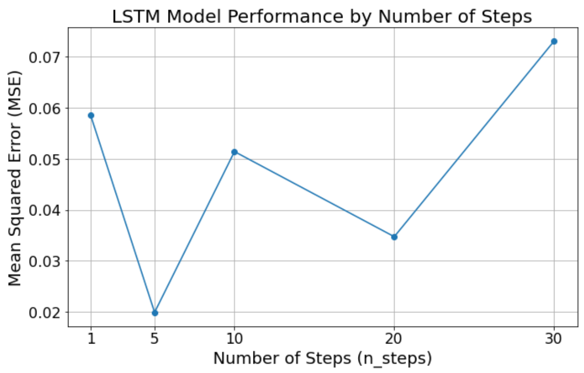

Figure 3: Variation of LSTM model's MSE with different time steps

Plotting the graph with the Mean Squared Error (MSE) metric and the number of steps, the parameter with the lowest MSE is picked as the optimal for the model. As seen in Figure 3, the optimal number of steps is 5. With this number of steps, the model achieves the lowest MSE, indicating the best balance between capturing relevant historical information and avoiding the introduction of noise or irrelevant information that could degrade its performance.

### 3.1.2 Grid Search

Next, in the following table, the hyperparameters that were tuned using a Grid Search approach are outlined. Grid Search is a method which evaluates and compares the performance of different model configurations to identify the combination of hyperparameters that yields the best results based on a specified performance metric, such as validation loss or accuracy. The hyperparameters that are tuned include the number of LSTM layers, the number of units in each LSTM layer, the dropout rate, the activation function, the learning rate of the optimizer, and the number of epochs for training. The following table provides a summary of the resulting optimal hyperparameter settings derived from the Grid Search.

Table 2: Hyperparameters of the LSTM model

| **Hyperparameter** | **Value** | **Description** |
|---|---|---|
| Number of LSTM Layers | 2 | Two LSTM layers are used in the model. |
| Units in LSTM Layer 1 | 100 | The first LSTM layer has 100 units. |
| Units in LSTM Layer 2 | 50 | The second LSTM layer has 50 units. |
| Activation Function | 'relu' | Rectified Linear Unit (ReLU) activation function. |
| Dropout Rate | 0.2 | Applied after the first LSTM layer to prevent overfitting. |
| Optimizer | 'adam' | Adam optimizer is used for adjusting the weights. |
| Number of Epochs | 20 | The model is trained for 20 epochs, representing 20 complete passes of the training data. |

## 3.2 Model Performance

The LSTM model's performance is assessed on a dataset split into training and testing sets, with 70% of the data used for training and the remaining 30% used for testing. The performance of the LSTM model is evaluated using multiple metrics that provide insights into the accuracy and reliability of its predictions. The following table summarizes the average performance metrics across all considered strike prices:

Table 3: Performance Metrics of the Model

| **Metric** | **Value** |
|---|---|
| MSE | 1.204 |
| RMSE | 1.066 |
| MAE | 0.857 |
| MAPE | 5.485% |

These metrics are calculated after unscaling the data to reflect the model's performance in the original scale of implied volatility values, which typically range from 0 to 100. The Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) provide measures of the average squared difference and the square root of this difference, respectively, between the predicted and actual implied volatility values. The MSE metric is equal to 1.204 and the RMSE metric is equal to 1.066. This means that, on average, the model's predictions are close to the actual values, considering the full range of implied volatility.

The Mean Absolute Error (MAE), at 0.857, shows that the average absolute error of predictions is less than one volatility point. The Mean Absolute Percentage Error (MAPE) of 5.485% is also a relative measure of the prediction error compared to the actual values, suggesting that the model's predictions are generally within 5.49% of the actual implied volatility figures.

### 3.2.1 Comparison with Autoregressive Model

These metrics are primarily used for comparison. They provide quantitative means to evaluate and compare the predictive accuracy and error magnitudes of different models under the same conditions. In this report the Autoregressive (AR) model is used as a baseline model for comparison with the machine learning technique. Based on the graph below measuring Mean Squared Error (MSE) for the autoregressive model, the model with lag 8 was chosen because it presents the minimum MSE.

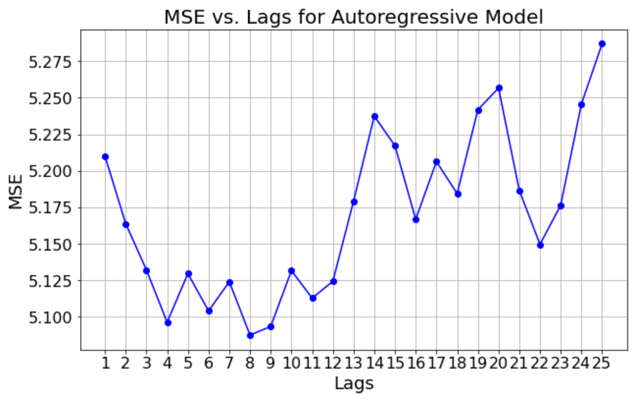

Figure 4: MSE trends across different lags in an autoregressive model.

The Autoregressive (AR) model, a benchmark for time series forecasting, has yielded the following average performance metrics across all considered strike prices:

Table 4: Comparison of LSTM and AR Model Performance Metrics

| **Metric** | **LSTM Model** | **AR Model** |
|---|---|---|
| MSE | 1.204 | 3.089 |
| RMSE | 1.066 | 1.672 |
| MAE | 0.857 | 1.477 |
| MAPE | 5.485% | 9.644% |

Comparing these results with those of the LSTM model, we observe the following: The MSE for the AR model is clearly higher than that of the LSTM model (3.089 compared to 1.204), indicating that, on average, the squared prediction errors of the AR model are larger. This suggests that the LSTM model is more effective at capturing the complex patterns of implied volatility across different strike prices. The RMSE and MAE for the AR model also exceeds those of the LSTM model (1.672 and 1.477 compared to 1.066 and 0.857 respectively), further indicating that, on average, the LSTM model's predictions are closer to the actual values. The MAPE for the AR model stands at 9.644%, significantly higher than the 5.485% achieved by the LSTM model. Thus, the average deviation of the AR model's predictions from the actual values is higher than that of the LSTM model, indicating the AR model is less accurate in relative terms. The LSTM model's performance surpasses that of the traditional AR model. It is superior in predicting implied volatility. The lower MSE, RMSE, MAE, and MAPE values achieved by the LSTM model suggest that it is more capable of capturing the complex patterns inherent in the implied volatility data, leading to more accurate and reliable forecasts. LSTM also has the advantage of making use of additional features beyond just past values of the target variable. By incorporating various relevant inputs such as historical index price data, option Greeks, and market indicators, the LSTM model can leverage a broader context. This allows it to understand and predict the future movements of implied volatility with greater precision. The model's architecture, designed to process sequential data, enables it to effectively learn from the temporal dependencies and non-linear relationships present in the financial markets. Consequently, the LSTM model demonstrates strong predictive power and also shows a strong ability to integrate diverse data sources, significantly enhancing its forecasting accuracy.

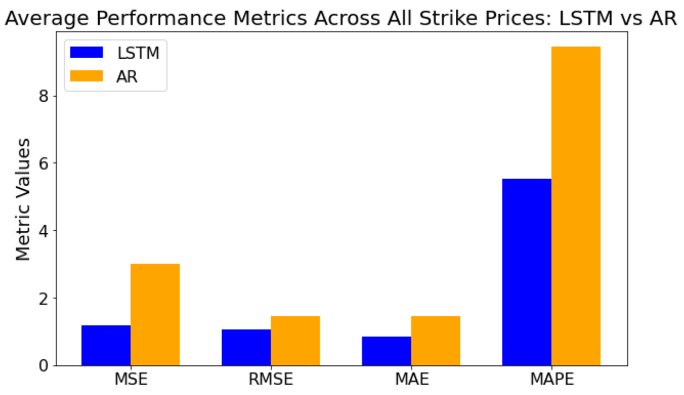

Figure 5: Comparing average performance metrics (MSE, RMSE, MAE, MAPE) for LSTM vs AR.

### 3.2.2 Predictions vs Actual values

In further analysis of the implied volatility (IV) predictions, the Figure 6 below serves as an example specifically for the option with a strike price of 4850. The graph showcases the model's ability to predict IV against the actual IV observed. It's evident from the graph that there's a reasonable level of accuracy in the predictions. The model manages to follow the overall trends in IV quite closely, capturing the general direction in which IV is moving. However, it's also noticeable that the model does not perfectly match the actual IV at every point, particularly in the finer details of the market's volatility peaks and troughs. This level of predictive accuracy suggests that the model possesses predictive power regarding the underlying patterns of IV movements. The model has potential as a forecasting tool, though there is still room to enhance its precision.

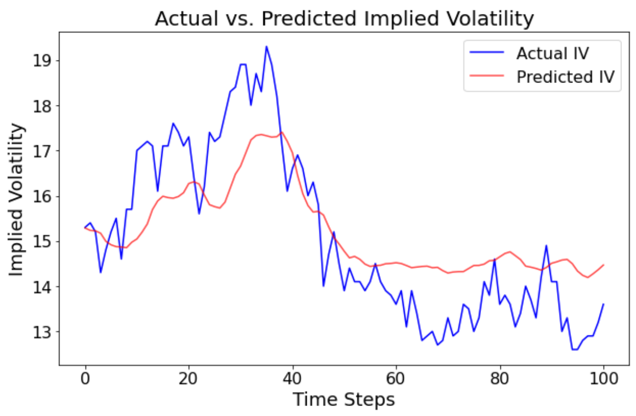

Figure 6: Visualizing actual and predicted implied volatility values.

Also, in the graph below which shows the model's performance for predicting implied volatility surfaces with an LSTM, it is observed that both the training and validation losses drop quickly at the start, which means the model is learning well. The training loss keeps decreasing, as well as the validation loss, indicating the model isn't just memorizing the data. This indicates effective learning without overfitting, supported by the presence of dropout layers.

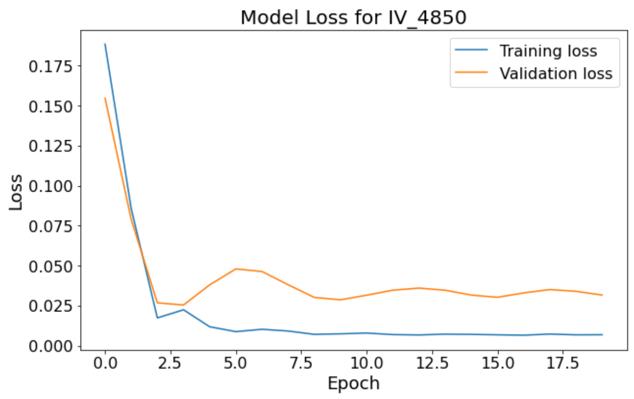

Figure 7: Training and validation losses vs epochs.

### 3.2.3 Next trading day's Implied Volatility predictions

For further analysis, an example is implemented by predicting the next day's trading prices using the discussed model, showcasing its practical application in forecasting market movements. Using the graph below, a trader can observe a volatility smile, indicating higher implied volatilities for options far from the current market price, both OTM and ITM. This pattern suggests greater expected price movements for these options, revealing opportunities for strategic strike price selection. For example, a trader might buy OTM options anticipating significant market movements, leveraging the higher implied volatility's indication of expected larger price changes. Additionally, the deviation from a flat implied volatility curve, as some models predict, to a smile suggests potential mispricing opportunities. Traders could exploit these by engaging in arbitrage strategies that involve buying and selling options with different strike prices but the same expiration, aiming for profits as prices converge.

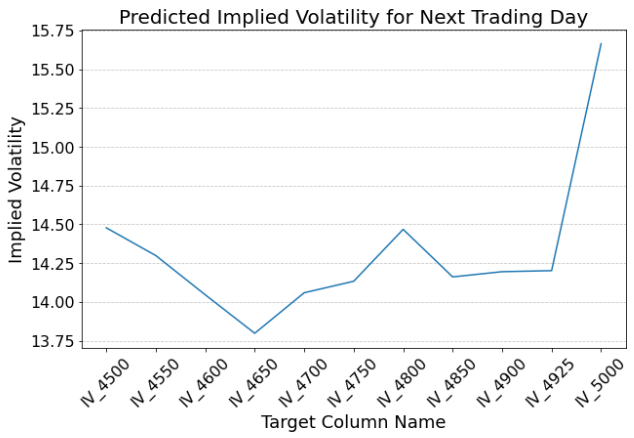

Figure 8: Visualizing the predicted volatility smile for the next trading day.

Moreover, the sharp increase in implied volatility at extreme strike prices, especially on the far right, hints at market anticipation of higher volatility or a potential upward movement. Traders can use this insight for hedging by purchasing OTM call options to protect against sudden market upswings. Understanding how volatility changes with strike price also aids in risk management, allowing traders to assess risks associated with different options positions more accurately.

## 4 Discussion

In this study, the application of Long Short-Term Memory (LSTM) networks for predicting implied volatility (IV) surfaces is explored. The model demonstrated promising results. However, certain improvements could enhance its performance and applicability. Firstly, the model's handling of peaks in the actual versus predicted IV figures could be refined. Advanced data smoothing techniques or incorporating nonlinear layers within the LSTM might improve peak prediction accuracy. Secondly, expanding the dataset to include call options alongside put options could offer a more rounded view of market sentiment and IV fluctuations. Similarly, incorporating options with varying maturity days could enrich the model's understanding of time decay effects on IV. However, these enhancements come with challenges. Introducing additional features and data types would inevitably increase the computational complexity and resource demands. Strategies to mitigate this include employing more efficient data processing techniques or leveraging more powerful computational resources. Furthermore, extending the model to predict IV surfaces for timeframes beyond the next trading day, such as one month ahead, could offer valuable insights for longer-term trading strategies. This requires the model to learn from broader market trends and could introduce additional layers of complexity. Future research could also explore integrating market indicators or macroeconomic factors to improve predictive accuracy. However, this must be done carefully to avoid overfitting by ensuring that the model remains generalizable and robust to market changes. Beyond LSTM, other methods such as attention mechanisms and transformers could be explored for handling sequential data. These approaches have demonstrated remarkable capabilities in capturing long-range dependencies and could offer superior performance in predicting implied volatility surfaces. Transitioning to these models may require additional adjustments and testing to tailor their complex architectures to the specificities of IV prediction. In summary, while the initial results are encouraging, there is significant room for refining the model through the incorporation of additional data, improving prediction of IV peaks, and extending the forecasting horizon. These enhancements could lead to a more comprehensive and effective tool for traders and risk managers.

## References

Bakshi, G., Charles, C., & Chen, Z. (1997). Empirical performance of alternative option pricing models. *Journal of Finance*, *52*(5), 2003–2049. https://doi.org/10.1111/j.1540-6261.1997.tb02749.x

Carr, P., & Wu, L. (2011). *A New Simple Approach for Constructing Implied Volatility Surfaces* (tech. rep.).

Chen, S., & Zhang, Z. (n.d.). *Forecasting Implied Volatility Smile Surface via Deep Learning and Attention Mechanism* (tech. rep.).

European Central Bank. (2023). *Yield curve spot rate, 1-year maturity - government bond, nominal, all issuers whose rating is triple a - euro area* [Accessed on 2023-01-03]. https://shorturl.at/ekmp1

Heston, S. L., & Nandi, S. (2000). *A Closed-Form GARCH Option Valuation Model* (tech. rep. No. 3). https://www.jstor.org/stable/2645997

Jiang, G. J., & Tian, Y. S. (2005, December). The model-free implied volatility and its information content. https://doi.org/10.1093/rfs/hhi027

Medvedev, N. (n.d.). *Multi-Step Forecast of the Implied Volatility Surface Using Deep Multi-Step Forecast of the Implied Volatility Surface Using Deep Learning Learning* (tech. rep.). https://openprairie.sdstate.edu/etd

Vrontos, S. D., Galakis, J., & Vrontos, I. D. (2021). Implied volatility directional forecasting: a machine learning approach. *Quantitative Finance*, *21*(10), 1687–1706. https://doi.org/10.1080/14697688.2021.1905869

Zhang, A., Lipton, Z. C., Li, M. U., & Smola, A. J. (n.d.). *Dive into Deep Learning* (tech. rep.).