

Applying Informer for Option Pricing: A Transformer-Based Approach

Feliks Bańka^a and Jarosław A. Chudziak^b

The Faculty of Electronics and Information Technology, Warsaw University of Technology, Poland

Keywords: Option Pricing, Transformers, Neural Networks, Time Series Forecasting, Deep Learning.

Abstract: Accurate option pricing is essential for effective trading and risk management in financial markets, yet it remains challenging due to market volatility and the limitations of traditional models like Black-Scholes. In this paper, we investigate the application of the Informer neural network for option pricing, leveraging its ability to capture long-term dependencies and dynamically adjust to market fluctuations. This research contributes to the field of financial forecasting by introducing Informer's efficient architecture to enhance prediction accuracy and provide a more adaptable and resilient framework compared to existing methods. Our results demonstrate that Informer outperforms traditional approaches in option pricing, advancing the capabilities of data-driven financial forecasting in this domain.

1 INTRODUCTION

Option pricing is a cornerstone of modern finance, essential for developing trading strategies and managing risk. Options enable traders and investors to hedge against potential losses or speculate on price movements. A call (put) option grants the holder the right, but not the obligation, to buy (sell) an asset at a specified price before the contract expires. Accurate option pricing models shape critical decisions in hedging and risk management, directly affecting trading portfolio profitability and stability.

Early theoretical frameworks, such as the Black-Scholes (Black and Scholes, 1973; Merton, 1973) and the Heston (Heston, 1993) models, offered valuable mathematical foundations but often rely on simplifying assumptions (e.g., constant volatility). These assumptions do not always hold in real-world markets, where sudden shifts in macroeconomic conditions or sentiment can lead to rapid changes in asset prices (Bollerslev, 1986). Over the past few decades, machine learning techniques—such as LSTM-based neural networks (Hochreiter and Schmidhuber, 1997; Yue Liu, 2023; Bao et al., 2017)—have demonstrated improved adaptability by capturing non-linearities and sequential dependencies. Yet, their effectiveness can be limited when handling very long time sequences, which demand more efficient and robust architectures.

Transformer-based models, originally devised for natural language processing (A. Vaswani and Polo-

sukhin, 2017), have shown promise in overcoming these challenges by leveraging self-attention mechanisms that allow for parallelized long-sequence processing. Recent advances, such as the Informer model (H. Zhou and Zhang, 2021), have introduced more efficient attention mechanisms geared toward time-series data. However, their application within option pricing remains underexplored, motivating the present study to investigate whether Informer's long-horizon capability and computational efficiency can produce more accurate predictions in option pricing tasks.

This paper contributes to the field of financial modeling by evaluating the application of the Informer architecture for predicting option prices, leveraging its efficient attention mechanism and long-sequence modeling capabilities to enhance prediction accuracy and adaptability to market fluctuations. Informer's ability to handle long-term dependencies makes it an ideal candidate for modeling complex financial data, offering a more advanced approach compared to traditional models like Black-Scholes (Black and Scholes, 1973; Merton, 1973) and Heston (Heston, 1993), as well as existing machine learning models such as LSTM (Hochreiter and Schmidhuber, 1997; Yue Liu, 2023). The contributions of this paper are as follows:

- We apply the Informer architecture to option pricing, leveraging its long-sequence modeling capabilities and self-attention mechanisms to enhance prediction accuracy.
- We benchmark the model against traditional and machine learning-based approaches, evaluating

^a <https://orcid.org/0009-0005-1973-5861>

^b <https://orcid.org/0000-0003-4534-8652>

its performance in high-volatility scenarios.

- We present an analysis of Informer’s predictive accuracy and trading profitability on historical data.

The remainder of this paper is organized as follows: Section 2 discusses related work, focusing on traditional and machine learning approaches to option pricing and the emerging role of Transformers in finance. Section 3 outlines the Informer-based methodology applied to option pricing. Section 4 presents the experimental setup and results, and Section 5 concludes with a summary and potential directions for future research.

2 RELATED WORK

The foundational models for option pricing, such as the Black-Scholes model (Black and Scholes, 1973; Merton, 1973) and the binomial model (Cox et al., 1979), have been pivotal in shaping early financial derivatives pricing. These models introduced critical concepts such as risk-neutral valuation but often rest on simplifying assumptions, such as constant volatility, which do not align with real-world market conditions. The introduction of stochastic volatility models, such as the Heston model (Heston, 1993), offered more flexibility by allowing volatility to vary as a stochastic process.

Despite improvements like stochastic volatility in the Heston model (Heston, 1993), traditional models remain limited in capturing the rapid shifts and complex dependencies of modern financial markets (Jones, 2019; Assaf Eisdorfer and Zhdanov, 2022). This has motivated the exploration of adaptive machine-learning approaches capable of modeling intricate relationships and dynamic patterns in financial data (Gatheral, 2006; Christoffersen, 2009).

Recurrent architectures, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRU), became popular due to their ability to capture temporal dependencies in sequential data (Mintarya et al., 2023; Hochreiter and Schmidhuber, 1997; Yue Liu, 2023). However, these models encounter scalability challenges when dealing with long-term dependencies or high-frequency data, often leading to computational inefficiencies (Mikolaj Binkowski and Donnat, 2018; Bryan Lim and Roberts, 2019). While modular and hybrid neural networks have been employed to integrate financial indicators and better capture non-linearities, issues of scalability and interpretability persist (Amilon, 2003; N. Gradojevic and Kukulj, 2009).

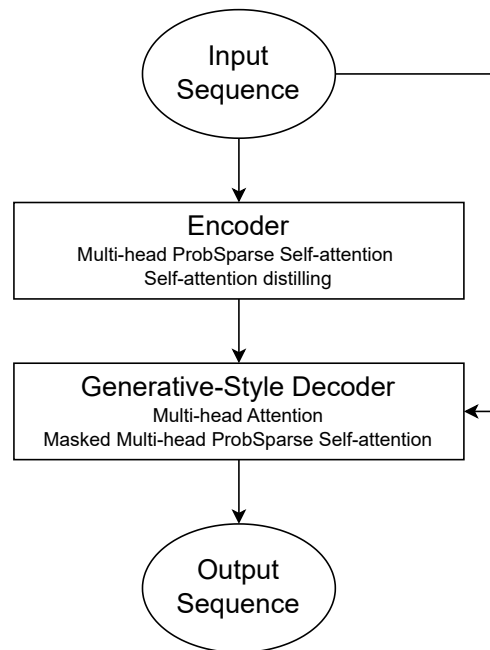


Figure 1: Informer model - conceptual overview. Based on (Szydłowski and Chudziak, 2024a).

Transformers, initially developed for natural language processing (A. Vaswani and Polosukhin, 2017), introduced self-attention mechanisms that bypass the limitations of recurrent models, allowing for the capture of long-term dependencies without the vanishing gradient problem. Szydłowski (Szydłowski and Chudziak, 2024b; Wawer et al., 2024) applied the Hidformer model to stock market prediction, demonstrating its effectiveness in handling long sequences and capturing complex market patterns. Informer, introduced by Zhou et al. (H. Zhou and Zhang, 2021) and illustrated in Figure 2, marked a significant advancement for time-series analysis with its ProbSparse self-attention mechanism, reducing the time and memory complexity of processing long sequences to $O(L \log L)$ for input length L . Wang et al. (C. Wang and Zhang, 2022) demonstrated Informer’s application in predicting stock market indices, showcasing its ability to outperform traditional deep learning models (e.g., CNN, RNN, LSTM) by effectively capturing relevant information while filtering out noise—a common challenge in financial time series. Informer’s robust multi-head attention mechanism allowed for the extraction of key features, leading to significantly higher prediction accuracy, particularly in short-term forecasting.

While studies have applied Transformer-based architectures to option pricing, including the generic Transformer model used by Guo and Tian (Guo and Tian, 2022) and Sagen’s investigation of the Temporal

Fusion Transformer (TFT) (Sagen, 2024), the application of Informer has not been explored in this domain. Given Informer’s strengths in long-sequence modeling and handling high-dimensional data efficiently, this paper seeks to evaluate its potential for enhancing predictive accuracy and computational efficiency in the complex landscape of option pricing.

3 MODEL ARCHITECTURE

In this section, we outline the architecture of the Informer-based model employed for option pricing. The Informer model is chosen for its ability to handle long sequences efficiently and capture dependencies over varying time scales through its unique attention mechanisms and architectural optimizations (H. Zhou and Zhang, 2021; C. Wang and Zhang, 2022). This is essential in financial applications where complex temporal relationships can influence outcomes significantly.

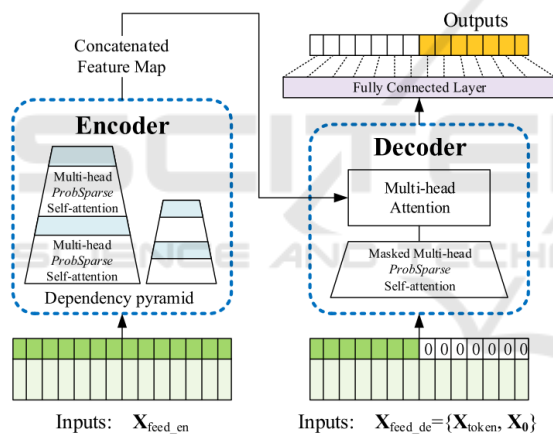


Figure 2: Informer model overview. Copied from(H. Zhou and Zhang, 2021).

3.1 Data and Feature Engineering

For effective model training, we select input features known to be crucial for option pricing. These features include the underlying asset price, implied volatility, time to maturity, strike price, and an indicator for the type of option (call or put). Each of these factors provides valuable insights into how option prices respond to market conditions. Volatility measures indicate market uncertainty (Hull, 2006; Kolm and Ritter, 2019), while time to maturity and strike price are fundamental in assessing the intrinsic and extrinsic value of the option (Jones, 2019; Black and Scholes, 1973; Merton, 1973). Normalization is applied to standard-

ize the data, ensuring all features are on a comparable scale:

$$x_t^{norm} = \frac{x_t - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where x_t^{norm} represents the normalized feature value at time t , and x_{max} and x_{min} denote the maximum and minimum feature values, respectively. This approach keeps all features within the range $[0, 1]$, aiding in model stability and faster convergence during training.

The input data is structured as a time-series sequence with a moving window approach, where T_x past data points are used to predict T_y future option prices or metrics. This sequential setup helps capture dependencies over different time horizons and enables the model to account for short-term fluctuations as well as long-term trends.

3.2 Proposed Model Architecture

The Informer-based model extends the standard Transformer architecture by incorporating enhancements tailored to the challenges of time-series forecasting in financial applications. It consists of two main components - the encoder and the decoder, which exchange information through self-attention mechanisms and encoder-decoder attention modules, as we can see in Figure 2. This section details each of these components and the overall data flow and token construction procedure.

3.2.1 Encoder

The encoder is responsible for extracting meaningful temporal dependencies from the input sequence. It includes an embedding layer, a ProbSparse self-attention mechanism, a feedforward sub-layer, and a self-attention distilling step to reduce computational overhead.

Embedding Layer. Each time step in the raw data is represented as a *token*, which is a set of features (e.g., strike price, time to maturity). The embedding layer projects these tokens into a dense vector space of fixed dimension, enabling the network to learn hidden interactions across features.

ProbSparse Self-Attention Mechanism. This attention mechanism aims to identify and focus on the most informative queries in the attention calculation, as illustrated in Figure 3. Instead of computing attention scores for all L queries and keys, it selects a subset of queries based on the *Kullback-Leibler divergence* (KLD) between the query distribution and a

predefined sparse distribution. Formally:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{Q^\top K}{\sqrt{d_k}}\right)V \quad (2)$$

where Q, K, V are the query, key, and value matrices, and d_k is the dimension of the keys. By selecting only the top- U queries (with $U \ll L$), complexity is reduced from $O(L^2)$ to approximately $O(L \log L)$, making the model scalable for long sequences.

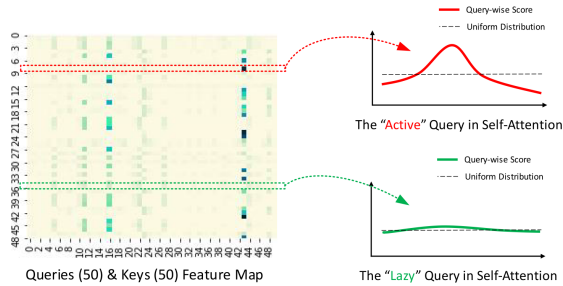


Figure 3: Illustration of the ProbSparse Attention mechanism. Adapted from(H. Zhou and Zhang, 2021).

Feedforward Sub-Layer. The output of the attention sub-layer is passed through a fully connected feedforward network with a hidden dimensionality D_{FF} :

$$\text{FeedForward}(x) = \text{ReLU}(W_1 x + b_1)W_2 + b_2, \quad (3)$$

where W_1, W_2 are weight matrices, and b_1, b_2 are biases. A larger dimension D_{FF} allows the model to capture intricate patterns.

Self-Attention Distilling. To improve efficiency, the Informer applies a distilling mechanism at the end of each layer, pooling or downsampling the sequence to retain only the most critical tokens. Mathematically,

$$\mathbf{Z}^l = \text{Pooling}(\mathbf{X}^l), \quad \mathbf{X}^{l+1} = \text{SelfAttention}(\mathbf{Z}^l),$$

where \mathbf{X}^l is the layer- l input. This process concentrates the model's capacity on dominant features, enhancing performance while mitigating overfitting.

Encoder Output. The final encoder output, denoted by \mathbf{E}_t , is a contextually enriched representation of the input tokens and will be passed to the decoder for generating forecasts.

3.2.2 Decoder

The decoder produces the target sequence by leveraging both the encoder output and partially known future labels. It features a self-attention sub-layer, encoder-decoder attention, and a feedforward network. Unlike the traditional approach, which decodes one step at a time, the Informer employs a

generative-style approach to predict all future steps simultaneously.

Decoder Input Preparation. To provide the decoder with partial knowledge of the future horizon, the model concatenates the most recent T_{label} observed values with placeholder zeros for the T_y unknown time steps. This can be expressed as:

$$\mathbf{D}_t = [y_{t-T_{\text{label}}+1}, \dots, y_t, 0, \dots, 0].$$

During training, the first portion corresponds to known labels or ground truth values, while zeros mark positions to be predicted.

Attention Modules and Feedforward Sub-Layer.

In the decoder, self-attention accounts for dependencies among known and future positions in \mathbf{D}_t , while encoder-decoder attention utilizes \mathbf{E}_t (the encoder output) as keys and values to incorporate previously extracted temporal structure. It also applies a feedforward sub-layer similar to that in the encoder.

Generative-Style Decoding. Finally, the decoder produces the entire predicted sequence in one forward pass:

$$\hat{\mathbf{Y}}_t = \text{Decoder}(\mathbf{E}_t, \mathbf{D}_t).$$

This approach improves inference speed compared to autoregressive decoding, which is advantageous for time-sensitive financial applications.

Decoder Output. The vector $\hat{\mathbf{Y}}_t$ constitutes the model's forecast for the T_y future time steps.

3.3 Model Workflow

The overall workflow begins by converting each time step into a token that bundles relevant features. These tokens are then passed to the embedding layer, which maps them into a continuous space of dimension d_{model} . The encoder applies ProbSparse self-attention, feedforward transformations, and self-attention distilling to capture critical dependencies with reduced computational overhead. Its final output \mathbf{E}_t , enriched with temporal context, is transferred to the decoder.

In parallel, the decoder constructs its input \mathbf{D}_t by combining partially known labels from the prediction window with placeholder zeros. Self-attention in the decoder identifies dependencies among these elements, while encoder-decoder attention integrates signals from \mathbf{E}_t . The generative-style decoding step then yields a full multi-step forecast in a single pass, producing $\hat{\mathbf{Y}}_t$. This hierarchical design is especially suited to financial time-series forecasting, where long-range dependencies and efficient computation are both critical.

4 EXPERIMENTS

The experiments conducted aim to evaluate the robustness and predictive power of the proposed Informer-based model in the context of option pricing. A thorough comparison is established using baseline models that encompass traditional and machine learning-based methods.

4.1 Dataset and Data Preparation

We use a dataset comprising eight years of historical option contracts for Apple Inc. (AAPL), sourced from publicly available financial databases, covering the period from January 4, 2016, to March 31, 2023. The dataset includes both call and put options with varying strike prices, expiration dates, and moneyness levels, providing a diverse and comprehensive foundation for analysis.

To improve data quality and ensure relevance, the preprocessing stage included the application of strict selection criteria. Options with a time-to-maturity (TTM) below 30 days were excluded, as such short-term contracts are typically highly volatile and speculative (Heston, 1993). Furthermore, only options with a moneyness ratio (the ratio of the underlying asset's price to the strike price) between 0.6 and 1.3 were included, as near-the-money options are more liquid and exhibit more reliable pricing (Bakshi et al., 2000). Contracts with insufficient data points or low trading volume were also removed to maintain robustness and integrity. The dataset is split into training, validation, and test sets, with 70% of the data allocated for training, 15% for validation, and the remaining 15% for testing (Matsunaga and Suzumura, 2019). This split ensures that the model is evaluated on unseen data, simulating real-world conditions where future predictions depend on past training.

4.2 Model Configuration and Training Strategy

The Informer model is configured to handle complex time-series data with the following parameters: the input sequence length is set to 30 days ($T_x = 30$), and predictions are made over a 30-day horizon ($T_y = 30$). The architecture includes one encoder layer and two decoder layers with a label length of 5 days, each featuring three attention heads. The embedding dimension (D_{MODEL}) is set to 32, balancing computational efficiency and model expressiveness. The feed-forward network dimension is set to 8, with a dropout rate of 0.06 to prevent overfitting. The model employs full attention with a factor of 3, suitable for capturing

temporal patterns effectively in financial time-series data. The training process employs a batch size of 64 and utilizes the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 0.0001. Training proceeds over 300 epochs, with early stopping applied based on validation loss, using a patience of 30 epochs. A weighted mean squared error (MSE) loss function is used, prioritizing accuracy across the entire 30-day prediction horizon. Hyperparameters, including the number of layers, attention heads, embedding dimension, learning rate, and dropout rate, were fine-tuned via random search.

4.3 Evaluation Metrics

The performance of the Informer model is evaluated using a comprehensive set of metrics to ensure a robust evaluation (Ruf and Wang, 2020):

Prediction Accuracy: The model's outputs are compared with the ground truth on the validation set to evaluate the prediction accuracy. Two commonly used indicators are employed: Mean Absolute Error (MAE), which measures the average magnitude of prediction errors, and Root Mean Squared Error (RMSE), which emphasizes larger errors to capture prediction variance. Lower values of both metrics indicate better model performance.

Final-Day Evaluation: We focus on final-day evaluation because it highlights the model's ability to make accurate long-term predictions, which is crucial for strategic financial decision-making (Kristoufek, 2012). To measure this, we use Direction Accuracy (DA), which measures the percentage of sequences where the predicted and actual price changes have the same direction, and Final-Day MAE, which calculates the MAE between predicted and actual prices specifically on the last day.

Return Calculation: The trading effectiveness of the model is evaluated using a simple strategy based on the predicted price at the end of each sequence. For a given sequence, if the predicted price (\hat{y}_{t+30}) is higher than the starting price (y_t), a long position is taken; otherwise, a short position is assumed. The return for the sequence is calculated as:

$$R = \ln \frac{y_{t+30}}{y_t} \times \text{sign}(\hat{y}_{t+30} - y_t) \quad (4)$$

where y_{t+30} is the true price at the prediction horizon, y_t is the starting price, and \hat{y}_{t+30} is the predicted price.

The cumulative net value (NV) aggregates returns across all sequences in the dataset, starting from an initial value of 1:

$$NV = 1 + \sum_{i=1}^N R_i \quad (5)$$

where N is the total number of sequences.

By combining predictive accuracy metrics (MAE and RMSE) with trading performance (NV), this evaluation framework captures both the statistical precision and the practical utility of the model in financial applications.

To benchmark the performance of the Informer-based model, we compare it against several established baseline models, including the Black-Scholes model, the Heston model, and the simple LSTM-based model. These models, ranging from traditional finance to advanced machine learning, help evaluate how the Informer performs in option pricing, highlighting its strengths and areas for improvement.

4.4 Results and Analysis

The results of the experiments demonstrate that the Informer model consistently outperforms all other models, both in terms of prediction accuracy and final-day evaluation metrics.

Table 1: Overall prediction metrics for all models.

Model	MAE	RMSE
Informer	2.7145	3.6766
LSTM	3.9343	5.0373
Black-Scholes	4.1765	5.3840
Heston	4.1282	5.3565

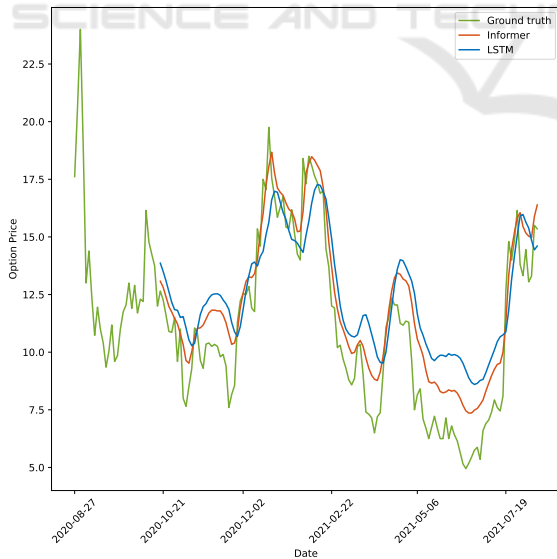


Figure 4: Comparison of Informer and LSTM predictions on a longer period for an option contract.

Table 1 highlights the overall prediction metrics, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The Informer model achieves the low-

est MAE (2.7145) and RMSE (3.6766) among all models, demonstrating its superior ability to predict option prices with high accuracy. The LSTM model, while a competitive machine-learning approach, exhibits a significantly higher MAE (3.9343) and RMSE (5.0373). Traditional models like Black-Scholes and Heston, despite their widespread use in finance, perform worse than the machine-learning-based methods. The Black-Scholes model has a slightly lower MAE (4.1765) compared to the Heston model (4.1282), but both models fail to capture complex market dynamics as effectively as the Informer. Figure 4 further illustrates the comparative performance of the Informer and LSTM models on a longer prediction period, highlighting the Informer’s ability to track trends more closely.

Table 2: Final-day evaluation metrics for all models.

Model	DA (%)	Final-Day MAE
Informer	54.43	2.9709
LSTM	52.19	4.0900
Black-Scholes	52.53	4.6880
Heston	51.74	4.6861

Table 2 presents the final-day evaluation metrics, including Direction Accuracy (DA) and Final-Day MAE. The Informer achieves the highest DA (54.43%) and the lowest Final-Day MAE (2.9709), showcasing its ability to predict both the direction and final value of option prices with superior precision. The LSTM model, while demonstrating a reasonable DA (52.19%), exhibits a higher Final-Day MAE (4.0900), indicating less reliability in final price predictions. Among the traditional models, Black-Scholes performs slightly better than Heston, achieving a DA of 52.53% compared to 51.74%, but both models have significantly higher Final-Day MAE values, exceeding 4.68.

Table 3: Performance of the trading strategy for Apple options based on final cumulative net value.

Model	Net Value
Informer	1.30
LSTM	1.21
Heston	1.15
Black-Scholes	1.14

In trading performance, the Informer achieved the highest cumulative net value (NV), outperforming all models, as shown in Table 3. With a final NV of 1.30, the Informer model demonstrates its superior ability to generate profitable trading strategies by accurately predicting directional movements over a 30-day horizon. The LSTM model follows with an NV of

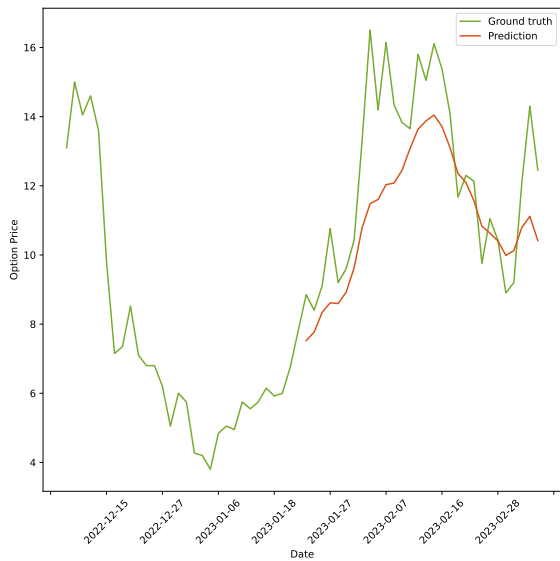


Figure 5: Example of upward trend Informer prediction for one option contract.

1.21, while the traditional models, Heston and Black-Scholes, lag slightly behind with NVs of 1.15 and 1.14, respectively.

As we can see on Figures 5 to 7 the Informer’s predictions remain stable across different trend types—upward, downward, and mixed. This stability highlights the potential of the Informer model as a valuable tool for investors, providing reliable insights to navigate diverse market conditions effectively.

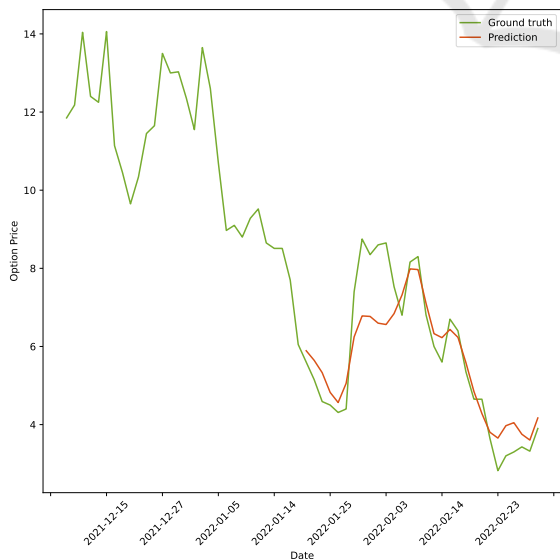


Figure 6: Example of downward trend Informer prediction for one option contract.

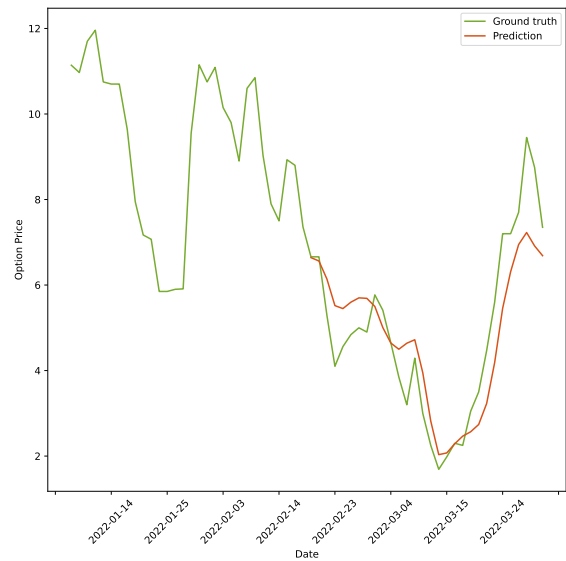


Figure 7: Example of mixed trend Informer prediction for one option contract.

5 CONCLUSION AND FUTURE WORK

Our research demonstrates that the Informer model, with its specialized attention mechanisms and generative-style decoder, outperforms traditional models like Black-Scholes and Heston, as well as recurrent neural networks such as LSTM, in predicting option prices and capturing long-term dependencies in financial data. The Informer not only achieved the lowest MAE and RMSE across all tested models but also generated the highest cumulative net value in trading evaluations, outperforming all other models and demonstrating its practical value in optimizing trading strategies.

This paper contributes to the field of option pricing by implementing the Informer model for option trading and evaluating its performance against other established models.

This study demonstrates the potential of the Informer model in enhancing option pricing predictions, yet there are several avenues for further exploration. Future work could involve incorporating reinforcement learning (RL) strategies to dynamically adjust trading decisions based on model predictions (Szydlowski and Chudziak, 2024a), improving adaptability in real-time trading environments. Additionally, applying the Informer architecture within a broader portfolio management framework could reveal insights into its effectiveness in balancing risk and return across diverse financial instruments. An-

other promising direction would be to test and refine trading strategies based on model outputs, such as mean-reversion or momentum-based approaches, to assess the practical profitability and robustness of Informer in real-world trading applications (Chudziak and Wawer, 2024).

REFERENCES

- A. Vaswani, N. Shazeer, N. P. J. U. L. J. A. G. L. K. and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- Amilon, H. (2003). A neural network versus black-scholes: A comparison of pricing and hedging performances. *Journal of Forecasting*, 22(4):317–335.
- Assaf Eisdorfer, R. S. and Zhdanov, A. (2022). Maturity driven mispricing of options. *Journal of Financial and Quantitative Analysis*, 57(2):514–542.
- Bakshi, G., Cao, C., and Chen, Z. (2000). Pricing and hedging long-term options. *Journal of Econometrics*, 94(1):277–318.
- Bao, W., Yue, J., and Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLOS ONE*, 12(7):1–24.
- Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. In *Journal of Political Economy*. Journal of Political Economy.
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroscedasticity. *Journal of Econometrics*, 31(3):307–327.
- Bryan Lim, S. Z. and Roberts, S. (2019). Enhancing time-series momentum strategies using deep neural networks. *The Journal of Financial Data Science*, 1(4):19–38.
- C. Wang, Y. Chen, S. Z. and Zhang, Q. (2022). Stock market index prediction using deep transformer model. *Expert Systems with Applications*, 208:118128.
- Christoffersen, P. F. (2009). *Elements of Financial Risk Management*. Academic Press, San Diego, CA.
- Chudziak, J. A. and Wawer, M. (2024). Elliottagents: A natural language-driven multi-agent system for stock market analysis and prediction. In *Proceedings of the 38th Pacific Asia Conference on Language, Information and Computation*, Tokyo, Japan, (in press).
- Cox, J. C., Ross, S. A., and Rubinstein, M. (1979). Option pricing: A simplified approach. *Journal of Financial Economics*, 7:229–263.
- Gatheral, J. (2006). *The Volatility Surface: A Practitioner's Guide*. Wiley, Hoboken, NJ.
- Guo, T. and Tian, B. (2022). The study of option pricing problems based on transformer model. In *Proceedings of the IEEE Conference*. IEEE.
- H. Zhou, S. Zhang, J. P. S. Z. J. L. H. X. and Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115.
- Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6(2):327–343.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9:1735–1780.
- Hull, J. C. (2006). *Options, Futures, and Other Derivatives*. Pearson Prentice Hall.
- Jones, C. M. (2019). Volatility estimation and financial markets. *Journal of Financial Markets*, 42:12–36.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv*, 1412.6980.
- Kolm, P. N. and Ritter, G. (2019). Dynamic replication and hedging: A reinforcement learning approach. *The Journal of Financial Data Science*, 1(1):159–171.
- Kristoufek, L. (2012). Fractal markets hypothesis and the global financial crisis: Scaling, investment horizons and liquidity. *Advances in Complex Systems*, 15(06):1250065.
- Matsunaga, D. and Suzumura, T. (2019). Long-term rolling window for stock market predictions. *arXiv preprint*, 1911.05009.
- Merton, R. C. (1973). Theory of rational option pricing. *The Bell Journal of Economics and Management Science*, 4(1):141–183.
- Mikolaj Binkowski, G. M. and Donnat, P. (2018). Autoregressive convolutional neural networks for asynchronous time series. In *International Conference on Machine Learning*, pages 580–589. PMLR.
- Mintarya, L. N., Halim, J. N., Angie, C., Achmad, S., and Kurniawan, A. (2023). Machine learning approaches in stock market prediction: A systematic literature review. *Procedia Computer Science*, 216:96–102. 7th International Conference on Computer Science and Computational Intelligence 2022.
- N. Gradojevic, B. G. and Kukolj, S. (2009). Option pricing with modular neural networks. *Neural Networks*, 22(5):716–723.
- Ruf, J. and Wang, W. (2020). Neural networks for option pricing and hedging: a literature review.
- Sagen, L. K. (2024). Applied option pricing using transformers. Master's thesis, Norwegian University of Science and Technology (NTNU).
- Szydlowski, K. L. and Chudziak, J. A. (2024a). Toward predictive stock trading with hidformer integrated into reinforcement learning strategy. In *Proceedings of the 36th International Conference on Tools for Artificial Intelligence (ICTAI 2024)*, Herndon, VA, USA.
- Szydlowski, K. L. and Chudziak, J. A. (2024b). Transformer-style neural network in stock price forecasting. In *Proceedings of the 21th International Conference on Modeling Decisions for Artificial Intelligence (MDAI 2024)*, Tokyo, Japan.
- Wawer, M., Chudziak, J. A., and Niewiadomska-Szynkiewicz, E. (2024). Large language models and the elliott wave principle: A multi-agent deep learning approach to big data analysis in financial markets. *Applied Sciences*, 14(24).
- Yue Liu, X. Z. (2023). Option pricing using lstm: A perspective of realized skewness. *Mathematics*, 11(2):314.