Quantitative Finance

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/rquf20

Merged LSTM-MLP for option valuation

Jacob Vinje, Erlend Stegavik Rygg, Cassandra Wu, Morten Risstad, Rita Pimentel, Sjur Westgaard & Christian O. Ewald

To cite this article: Jacob Vinje, Erlend Stegavik Rygg, Cassandra Wu, Morten Risstad, Rita Pimentel, Sjur Westgaard & Christian O. Ewald (08 May 2025): Merged LSTM-MLP for option valuation, Quantitative Finance, DOI: 10.1080/14697688.2025.2493965

To link to this article: https://doi.org/10.1080/14697688.2025.2493965

Published online: 08 May 2025.

Quantitative Finance, 2025
https://doi.org/10.1080/14697688.2025.2493965

Merged LSTM-MLP for option valuation

JACOB VINJE*, ERLEND STEGAVIK RYGG, CASSANDRA WU, MORTEN RISSTAD, RITA PIMENTEL, SJUR WESTGAARD and CHRISTIAN O. EWALD

Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Trondheim, Norway

(*Received 29 July 2024; accepted 26 March 2025*)

Traditional option pricing models rely on estimates of expected volatility. The true volatility is not directly observable and must hence be estimated, inevitably with error. Any measurement errors immediately translate into inaccurate pricing, leading to potential losses for economic agents trading options for hedging or speculative purposes. This paper proposes a novel merged LSTM-MLP model for option pricing that circumvents the need for an explicit volatility estimate, leading to more accurate valuations. Through extensive out-of-sample testing on S&P500 call options data from 2015 to 2022 we document the statistical accuracy and economic benefits of the model when compared to relevant benchmarks. The superior performance is enabled by the combined LSTM-MLP architecture, which simultaneously utilizes both time series data and the cross-section of observed option characteristics in a deep learning neural network that accurately captures the complex price dynamics. The results are consistent over time and robust across option moneyness and time-to-expiry.

1. Introduction

Volatility is the primary determinant of option prices. Option pricing models, such as Black-Scholes-Merton (Black and Scholes 1972, Merton 1973) and Heston (1993), parameterize option premia as functions of implied volatility. Although widely used, this is inherently difficult since volatility is a latent variable that must be estimated with error. The importance of addressing volatility uncertainty has been emphasized since (Lo 1986), which provides a rigorous statistical framework for understanding its impact on pricing discrepancies. This paper proposes a deep learning option pricing model that circumvents the requirement of an explicit volatility estimate for the purpose of valuation. We achieve this by combining two artificial neural networks: a long short-term memory (LSTM) and a multilayer perceptron (MLP). This LSTM-MLP architecture combines time series data with cross-sectional pricing information. First, time series data in the form of historical underlying returns is provided as input to the LSTM network. Subsequently, the output of the LSTM is used as input in the MLP, along with the option features for a given contract, to determine the option price. We coin this model as *merged LSTM-MLP*. This architecture is motivated by the hypothesis that the time series information extracted by the LSTM is more informative for option pricing than the explicit volatility input commonly used. Therefore, we replace the explicit volatility input to the MLP with extracted time series information conveyed via the LSTM outputs.

Through extensive out-of-sample testing, using SPX European call options on the S&P 500 index from November 1, 2011, to December 31, 2022, we find that the merged LSTM-MLP achieves better pricing and trading performance than relevant parametric and machine learning benchmark models. We thereby demonstrate the practical relevance of the model by simulating how a market participant could deploy and retrain the model in a real environment by using sliding windows to train and test the model. We follow Andreou *et al.* (2008)† and Cao *et al.* (2021),‡ which differ from most previous studies that split the entire data set chronologically for training, validation, and testing. Compared with the previous works, we use a different combination of the length of the training, validation, and test sets. Also, we implement more sliding windows over a longer time period.

The application of machine learning methods has attracted the attention of many researchers. The emergence of this field is motivated by the ability of machine learning algorithms to learn non-linear relationships between input and

*Corresponding author. Email: vinje.jacob@gmail.com

† Andreou *et al.* (2008) divide the data set into ten different overlapping training and validation sets, each followed by a non-overlapping test set.
‡ Cao *et al.* (2021) uses 300 days for testing and validation and the following 7 or 30 days for testing. Then, the training and validation window rolls forward 7 or 30 days for the second testing. This continues until the end of the sample period.

output variables (Hornik *et al.* 1989), without necessarily being limited by economic and/or statistical assumptions as the traditional option pricing models (Ivaşcu 2021). The work of Hutchinson *et al.* (1994) is one of the first studies to use neural networks for option pricing. They train an MLP with one hidden layer with four hidden units on simulated data and demonstrate that it is capable of learning the Black-Scholes formula with a high degree of accuracy. The practical relevance of the proposed model is tested by using it to price options on S&P 500 futures, for which it outperforms Black-Scholes. Culkin and Das (2017) follow the approach of Hutchinson *et al.* (1994) by training and testing their model on simulated call option prices. In contrast to Hutchinson *et al.* (1994), they use a deep MLP consisting of four hidden layers with 100 units each and incorporate more input variables. Their results show that the proposed model learns the Black-Scholes option pricing model based on simulated data with high accuracy when testing on the same data. It is worth noting that they do not test on real financial data.

Bali *et al.* (2023) use both tree-based methods and neural networks, but for the task of predicting future option returns instead of estimating the price at the time of prediction. Besides MLPs, improved pricing performance is also demonstrated by other neural network architectures. Liang and Cai (2022) propose a Convolutional Neural Network (CNN) architecture and an LSTM architecture, both of which achieve better pricing performance than the benchmark models Black-Scholes, Merton, and Heston. They attribute the enhanced performance of the pricing models to their ability to use information embedded in the time-series of the underlying return.

Appropriately quantifying volatility remains challenging for traditional and neural network-based option pricing methods. For the latter, the most common volatility measures used in the literature are historical volatility (Hutchinson *et al.* 1994, Amilon 2003, Hsu *et al.* 2018, Ivaşcu 2021), implied volatility (Gradojevic and Kukolj 2022), as well as GARCH and its extended versions, such as exponential GARCH (EGARCH) and threshold GARCH (TGARCH). These extensions address features like asymmetric volatility responses to market shocks, providing more flexibility in modeling financial time series.

Historical volatility measures cover a wide range of time periods. The lowest range identified is seen in Hsu *et al.* (2018), which proposes a deep neural network with historical 1-minute and 5-minute volatility to price monthly and quarterly options. However, no significant improvement in pricing accuracy is achieved by including these very short-term volatilities. Longer historical periods of 10 to 30 days are seen in Amilon (2003), and 60 days in Hutchinson *et al.* (1994) and Ivaşcu (2021). Iltüzer (2022) tests historical volatility over five different time horizons, including 360 days, 30 days, and 10 days for calendar days and 21 and 252 days for trading days. Liang and Cai (2022) input to their proposed LSTM model 5 lags of every input used. Amilon (2003) also provide lagged values of the underlying asset for the last four days as inputs to their MLP pricing model. This is motivated by a similar hypothesis as our research about the potential of the pricing model to learn the volatility structure, or any useful distribution structure, from the historical data. Andreou *et al.* (2008) calculate the implied volatility in the Black-Scholes formula for a specific option on one day and use it to price the same option on the next day. The MLP pricing model that uses this implied volatility appears to be the overall best-performing model when evaluated using RMSE, MAE, and profitability from trading strategies net of transaction costs.

Other neural network-based pricing models use volatility prediction using GARCH as an input to the model. Gençay and Gibson (2007) use a GARCH(1,1) prediction as the volatility input to the proposed MLP pricing model that in addition to the GARCH volatility prediction uses the same input variables as the Black-Scholes model. The proposed model performs better across all maturities and moneyness than benchmark models, including the Black-Scholes model with historical as well as GARCH prediction volatility, stochastic volatility model, and stochastic volatility random jump model.

Other neural network-based pricing models use GARCH-based volatility predictions as inputs. Gençay and Gibson (2007) evaluate an MLP model that incorporates GARCH(1,1) predictions as the volatility input alongside the same variables used in the Black-Scholes model. They compare the MLP to benchmark models, including the Black-Scholes model with both historical volatility and GARCH-predicted volatilities, the stochastic volatility (SV) model, and the stochastic volatility random jump (SVJ) model. Importantly, in these benchmarks, GARCH is used solely as a method for estimating volatility, not as an independent pricing model. Their findings demonstrate that the MLP consistently outperforms these benchmarks across all maturities and moneyness levels, highlighting its robustness in capturing nonnormal return distributions.

In contrast to the aforementioned literature, which predominantly presents option pricing models relying on an explicit volatility input, some studies argue that neural networks can infer volatility information from historical data during training. For instance, Yao *et al.* (2000) suggest that their MLP pricing model does not require volatility as an input when daily prices for the underlying asset are provided to the network. They propose that this setup allows the network to implicitly capture volatility through daily returns, which are the fundamental determinants of volatility. However, given the intimate relationship between an option's price and the volatility of its underlying asset, the omission of volatility as an explicit input raises questions about the interpretation of the resulting option prices. Without an explicit volatility parameter, it may be less clear how the model attributes changes in option prices to underlying market conditions, potentially complicating the economic interpretation of its outputs. In this line, we propose an architecture that combines LSTM and MLP networks and does not need volatility as input. We replace the volatility input to the MLP network with the output from the LSTM network. Given the ability of recurrent neural networks, such as LSTM, to capture state information, this can be more useful to option pricing than historical volatility. This was inspired by the report from Ke and Yang† where they propose a similar architecture but their model fails to outperform the MLP pricing model with historical volatility as an input. In our model, the two networks have a direct connection, and this collaborative design enables training and backpropagation throughout the entire network as a unified entity. This characteristic, along with a comprehensive search of the hyperparameters, implies that the merged LSTM-MLP delivers indeed superior pricing performance when compared to the benchmarks.

Besides determining the pricing performance, we also investigate the economic significance of our model using a trading strategy. To the best of our knowledge, Andreou *et al.* (2008) is the only study of neural network-based option pricing models that implement trading strategies to evaluate model performance. They find that trading strategies based on both neural networks and traditional pricing models are profitable for certain combinations of transaction cost and mispricing margin required to make the trade. The best models yield profits of 77%–82% of the transactions made. The neural network-based models provide the most improvement compared to the traditional methods that use less sophisticated volatility measures, such as 60-day historical volatility. Portfolios are created by buying (selling) undervalued (overvalued) options by the market compared to the model predictions. Additionally, they take a delta hedging position on the underlying asset by which a short (long) position on a call option is hedged using a long (short) position in the underlying asset. A position is held as long as the option is undervalued or overvalued, after which the position is liquidated, the profit or loss is computed, and a new position is entered based on current market conditions. A limitation of this trading approach is the lack of performance measures other than absolute profits, which we include in our study.

In the context of the literature on neural network-based option pricing models, our contribution is threefold. First, we demonstrate that the merged LSTM-MLP achieves better pricing and trading performances than the benchmark models. This reinforces that the time series information extracted by the LSTM network is more informative for option pricing than an explicit volatility input. Second, we simulate how a market participant could deploy and retrain the model in a real environment using sliding windows. Our setup is novel in that we implement more sliding windows over a longer time period. Lastly, we evaluate the practical relevance of the proposed pricing model with a trading strategy and provide a comprehensive analysis of the trading performance. This goes beyond common practice for evaluating model performance in the literature, which is often limited to pricing accuracy measured as the difference between the model and market prices.

The rest of the paper is structured as follows. Section 2 details the implementation of the merged LSTM-MLP option pricing model and its training process. This section also outlines the methods for evaluating model performance in terms of pricing accuracy and risk-adjusted trading returns. Section 3 describes the data set used and data processing. Section 4 presents the results for pricing and trading performances and discusses these compared to benchmark models. Finally, Section 5 summarizes the findings and provides suggestions for future research.

† The report for the course CS230: Deep Learning, Fall 2019, Stanford University, CA, by Ke and Yang untitled *Option Pricing with Deep Learning* is available online https://cs230.stanford.edu/projects_fall_2019/reports/26260984.pdf.

2. Methodology

Estimating the volatility of stock returns is a challenging task. On the one hand, considering the volatility of a stock return as constant over time is simplistic. On the other hand, as the volatility smile shows, the implied volatility changes with the option's moneyness. Therefore, we propose an option pricing model that bypasses the volatility estimation, which we coin as the *merged LSTM-MLP option pricing model*.

2.1. Merged LSTM-MLP option pricing model

This new option pricing model combines an LSTM network with an MLP, as illustrated in Figure 1.‡

A key strength of the merged LSTM-MLP architecture is the direct connection between the LSTM network and the MLP. This collaborative design enables training and backpropagation throughout the entire network as a unified entity. In other words, extracting information from the stock returns is not treated as a separate process from pricing the option but as one integrative task to be solved by the merged LSTM-MLP model. This particular setup intends to align the training of the LSTM network and the MLP to maximize pricing accuracy. Rather than training the LSTM network to extract pre-specified volatility measures from the historical returns, it learns the relevant information for the MLP to price the option.

While the precise nature of the information extracted by the LSTM is uncertain, it is reasonable to assume that it embeds relevant factors influencing option pricing behavior, such as volatility knowledge. Although the merged LSTM-MLP model does not explicitly adhere to theoretical constructs like the risk-neutral framework, it learns from data that naturally reflect these principles through the inclusion of features such as the risk-free rate and historical price behavior. The merged LSTM-MLP is more likely to tailor the volatility measure for specific moneyness and time to maturity since multiple connections between the LSTM and MLP components allow for high dimensional volatility data. This integrative model structure thereby demonstrates a potential for addressing the inherent complexities associated with volatility representation in option pricing while allowing for additional information extraction from the asset returns. Through the training process, the model learns how options markets behave during events like jumps in the underlying asset, making it well-suited to handle such discontinuities.

In the merged LSTM-MLP model, the LSTM part comprises three layers, each including eight nodes. Each node contains memory cells with an internal structure. The MLP component consists of four stacking structures, each involving a dense layer followed by a batch normalization layer.

‡ The Python code is available upon request.

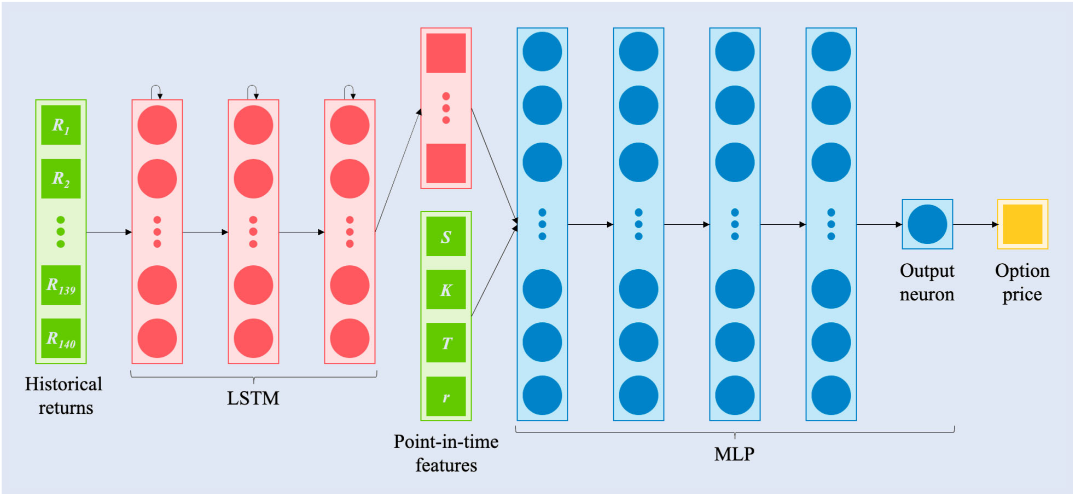

Figure 1. Architecture of the Merged LSTM-MLP option pricing model. R_i is the return on the stock i days before the option pricing date, whilst S is the stock price at this date. K and T represent the option's strike price and time to maturity, respectively, while r denotes the risk-free rate.

The dense layer forms a fully connected network with the nodes of the preceding layer. Batch normalization layers are implemented between each dense layer to normalize the activations of the preceding layer in each batch. This strategy addresses common challenges in deep neural networks, such as vanishing or exploding gradients, curbs overfitting, and accelerates the training process by reducing the number of iterations required for convergence (for further details, see Ioffe and Szegedy 2015). The MLP part uses leaky ReLu as an activation function. Unlike the conventional ReLU function, which sets all negative values to zero, leaky ReLU multiplies negative inputs by a small, fixed parameter. This design decision mitigates the 'dead neuron' problem that can arise with regular ReLu, ensuring that all neurons remain active and continue learning, which eventually leads to a more robust and effective network (see Maas *et al.* 2013 for more details). For the final dense layer, consisting of a single output node, we use the conventional ReLu activation function to ensure it is a non-negative option price. We perform a comprehensive search of the merged LSTM-MLP hyperparameters, which is described in Appendix 1, and the final model configuration is presented in Table A1.

While standalone LSTM models, as presented in Liang and Cai (2022) and Liu and Wei (2022), can capture time series information, they require all input variables to be lagged. This requirement introduces complexity and computational cost. The number of lags is typically determined based on a trade-off between capturing sufficient temporal information and maintaining computational feasibility, with choices often informed by domain expertise or empirical performance during training. Despite not explicitly referring to computational constraints, Liang and Cai (2022) only consider five lags in the LSTM, supplementing it with historical volatility input. This implies that handling an array of lagged variables over extensive time series in large datasets can become computationally burdensome and potentially unfeasible. The merged LSTM-MLP model overcomes this issue by only lagging the stock returns, while other variables such as the strike price, time to maturity, and the risk-free rate are handled directly by the MLP component, which avoids analyzing features without much temporal information as a time-series. The merged LSTM-MLP model overcomes this issue. The LSTM segment captures temporal dependencies within the historical returns, while the MLP component efficiently processes option data at a given time, capturing the complex nonlinear relationships inherent in option characteristics. This combined architecture thus streamlines computational resources and enhances model performance. It is relevant to highlight that the number of historical returns considered in the merged LSTM-MLP model is of central importance. To achieve accurate option prices, the hyperparameter search finds that 140 historical daily returns are recommended.

As shown in Figure 1, the LSTM input is the stock's historical returns over the 140 days before the option pricing date and it outputs information extracted from these data through eight output nodes. The MLP takes as input the values of these eight nodes, the option characteristics, including the stock price at the option pricing date, the strike price, and the time to maturity, besides the risk-free rate. The final output of the merged LSTM-MLP is the option price.†

We train and test the model using sliding windows, following the approaches in Andreou *et al.* (2008) and Cao *et al.* (2021). Compared to their implementation, we use a greater number of sliding windows and cover a longer time period overall. The window slides one month from one model instance to the next one, creating 96 sliding windows in total. Each model instance uses three chronological consecutive sets: three years for training, one month for validation, and another month for testing. We have chosen to retrain the model on a monthly basis due to computation time. However, in a real-life setting, a market participant can do it at a higher frequency, such as weekly or daily.

The implementation of sliding windows supplies several advantages. First, it keeps the training data up-to-date, i.e. it is relevant and representative of the out-of-sample options

† The bid-ask midpoint is used as a proxy for the market price, following Yang *et al.* (2017) and Liang and Cai (2022). This enables comparison with the benchmark models as they do not output bid and ask prices separately.

to be priced. At the same time, we also provide sufficient historical data for the model to learn the past pricing behavior of the market. The model should, therefore, be able to adapt to changing market conditions. This is key given the time-inhomogeneity inherent in financial data (Ruf and Wang 2020), where the data's statistical properties and underlying dynamics can change over time. Second, it reinforces the robustness of our model architecture by testing across multiple periods, which provides a stronger argument for its generalization capability.

The learning process incorporates an exponentially decaying learning rate to optimize parameter tuning. We scale the features using min-max scaling normalization to ensure possible discrepancies do not affect the training. The process begins with randomly generated initial weights for the merged LSTM-MLP model. Then, iteratively, these weights are calibrated using the Adam optimizer (Kingma and Ba 2014), where mean squared error (MSE) is the objective function. Early stopping is implemented and the model stops training if there is no improvement in validation loss for 20 consecutive epochs. We also utilize checkpointing to save the model configuration each time a reduction in validation loss is observed.

To ensure consistency across training periods, we fix the number of option contracts in each training instance to 1 000 000. In the initial windows, this represents almost all the available data. As the number of options in the market grows, we randomly sample 1 000 000 based on a uniform distribution. This provides a balanced and representative sample and facilitates computational efficiency. While the LSTM-MLP model does not explicitly enforce arbitrage-free constraints, it learns pricing patterns from market data, which are generally arbitrage-free due to real-world market dynamics.

2.2. Backtesting trading strategy

To investigate the economic significance of the merged LSTM-MLP model, we implement a trading strategy based on the difference between the model and market prices. Following Andreou *et al.* (2008), the strategy is to buy (sell) undervalued (overvalued) options by the market compared to the model. The trading strategy is based on the following assumptions.

- There is no slippage, i.e. we assume infinite liquidity at every bid and ask price;†
- There is no market impact of trading decisions in the sense that trades on one day do not affect the price on the next day;
- There are no margin requirements.

Even though these assumptions underneath simplifications compared to practice, it is worth highlighting that their effects depend on the transaction size. Also, the portfolio has significant cash at all times, which could act as collateral. The trading strategy does not assume market efficiency but seeks to exploit potential pricing inefficiencies. If the strategy generates consistent profits after accounting for transaction costs, this may reflect temporary or structural deviations from market efficiency. Further investigation is warranted to assess the persistence of these profits in real-world trading conditions.

Next, we describe the details of the trading procedure.

(1) The trading portfolio is initialized with a starting capital of $1 000 000;
(2) Options are bought at the ask price and sold at the bid price;
(3) Options with a mid-price under 50 cents are disregarded to avoid extreme percentage fluctuations.
(4) A transaction cost of 0.5% is charged.‡ This is meant to represent the exchange fees, clearing fees, regulatory costs, and technology and infrastructure a market participant would have to pay when deploying this trading strategy;
(5) A stop-loss threshold is set at 500%. This means that, when holding a short position, if the bid price is 500% higher than the price originally sold for, the option is bought back for the current ask price;§
(6) Trading signals are generated when the difference between the model price and the observed market price is greater than 15%;¶
(7) At the moment of a trading signal, a position (buy or sell) is taken for an amount equivalent to 0.005% of the available cash in the portfolio;‖
(8) If an option is bought (sold) on one day, it cannot be bought (sold) again on subsequent days, so that the portfolio doesn't build large positions in a specific contract;
(9) If an option is bought (sold) on one day, it can be sold (bought) again on subsequent days if the price passes the sell (buy) threshold, i.e. it becomes mispriced in the opposite direction;
(10) The difference in long and short positions cannot exceed 10% of the total number of positions. When it exceeds 10%, only new positions that lower this percentage are permitted. This rule limits the directional exposure in the portfolio, ensuring that profits are primarily generated from market mispricings rather than directional bets.

2.3. Benchmark models

The traditional Black-Scholes (BS) model, first introduced by Black and Scholes (1973), is the foundational benchmark in option pricing. We evaluate its performance using three volatility measures: 30-day historical volatility, based on past returns; GARCH(1,1)-predicted volatility, capturing

† In practice, the liquidity at a specific price may be limited such that the rest of the position is executed at a worse price.

‡ Andreou *et al.* (2008) find that the best strategies retained profitability up to transaction costs of 0.5%.

§ This is not triggered when holding a long position as the maximum loss, in this case, is 100% of the invested amount.

¶ Andreou *et al.* (2008) show that a mispricing margin of 15% yields the highest absolute returns for their trading strategy. They also observe that the P&L increases in a diminishing fashion for higher mispricing thresholds, indicating that there is an optimal threshold for maximizing trading profits.

‖ In practice, this might not be entirely realistic as it requires the options trader to consider a discrete number of options to fit the specified percentage position exactly.

time-varying volatility as suggested by Bollerslev (1986) and in Taylor (1986); and implied volatility, a market-driven measure adopted from Wang *et al.* (2012). For short, we call them BS 30-day, BS GARCH, and BS IV, respectively. Additionally, the Heston stochastic volatility model, proposed by Heston (1993), and renowned for its intertwined asset price and volatility dynamics, is also considered as a benchmark.†

Along with the Black-Scholes and Heston parametric models, a standard MLP model is implemented as a benchmark to isolate the impact of substituting the explicit volatility input with the LSTM component of the merged LSTM-MLP model. The MLP model is developed and trained following a methodology similar to the one used for the merged LSTM-MLP model, but it considers as input the 30-day historical volatility instead of the LSTM output.

2.4. *Performance metrics*

We use the root mean squared error (RMSE) and the mean absolute error (MAE) to evaluate the pricing accuracy of the merged LSTM-MLP model and the benchmark models.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i|,$$

where, for each option i, y_i is the observed price, and $\hat{y}_i$ is the price estimate obtained by the model, whereas m represents the total number of options. These metrics measure the degree of pricing error, with lower values indicating better performance.

To carry out pairwise comparisons of the models, we employ the Diebold-Mariano (DM) test proposed by Diebold and Mariano (1995), using the MSE loss function to assess the similarity of the model predictions.

Besides assessing the pricing performance, we also check the trading performance of the options portfolio obtained with the designed trading strategy. We assess the risk-adjusted performance using several metrics, namely the Sharpe ratio, the Sortino ratio, the returns, and the maximum drawdown (MDD). These metrics are intended to assess both the trading returns and risks.

The Sharpe ratio, introduced by Sharpe (1966), is a measure of risk-adjusted return, taking into account the total volatility of returns. The Sortino ratio, a variant of the Sharpe ratio introduced by Sortino (1994), distinguishes harmful volatility from total volatility by considering only the downside deviation. We compute the Sharpe and Sortino ratios using daily returns and then annualize the measures.

$$\text{Sharperatio} = \frac{R_p - r_f}{\sigma_p}$$

$$\text{Sortinoratio} = \frac{R_p - r_f}{\sigma_d},$$

where R_p is the return of the portfolio, r_f is the risk-free rate, σ_p is the standard deviation of the portfolio returns, σ_d is the standard deviation of the negative portfolio returns.

3. Data

Our data set consists of SPX European call options on the S&P 500 index. We obtain daily closing prices and relevant options attributes, namely underlying price, strike price, and time to maturity, from optionsDX.‡ The sample period is from November 1, 2011, to December 31, 2022. Before January 1, 2015, the data is used exclusively for training and validation for hyperparameter tuning. For the risk-free rate, we use daily Treasury yields from the U.S. Department of the Treasury.§ The interest rates are available only for some maturities, specifically 1, 3, 6, 12, and 24 months; thus, we use the Nelson-Siegel-Svensson method (Svensson 1994) to obtain the yield curves.

Initially, we have 13 108 756 option contracts. We do not consider options with some invalid or missing values regarding their features. Further, we remove options with zero time to maturity, as their price equals their intrinsic value, which can bring noise into the model. To avoid bias due to outliers, we also discard options with time to maturity greater than two years and those within the top and bottom 5% of moneyness (lower than 0.8 and larger than 2.0). Indeed, options with these moneyness and maturity ranges are often less traded and possibly less efficiently priced. The data preprocessing stage removes 13.58% of the initial data set, leading to 11 328 707 option contracts. Table 1 presents summary statistics of option characteristics and risk-free rates (Table A2 in Appendix presents descriptive statistics sectioned per year).

Figure 2 displays the empirical distribution of time to maturity and moneyness, confirming well-established stylized facts. As evident from the left panel, the majority of trades have a relatively short time to maturity. Similarly, the right panel indicates relatively better liquidity for options with strikes not too distant from the current level of the underlying S&P 500 index.

Figure 3 reflects the historical evolution of equity option markets. The annual number of option trades has steadily increased over time, driven by a combination of commercial hedging demand and speculative flow, both increasing the informational content of option prices. The increase in nominal average option prices follow naturally from the gradual increase in the S&P 500 index. This expansion of the information set - by virtue of the number of trades - also motivates our application of a rolling window training strategy.

† Arguably, the random behavior of the underlying S&P500 index could be captured by modeling volatility with additional components, such as jumps in price dynamics. However, this approach requires an a-priori specification of the jump-diffusion model as well as accurate parameter estimates, which has proven notoriously difficult (Andersen *et al.* 2015, Cont and Tankov 2003, He *et al.* 2006). Instead, we note that implied volatility, being a risk-neutral expectation of future volatility, inherently contains market-based information about both the likelihood and magnitude of future jumps. Thus, we implicitly account for jump risk through the BS-IV benchmark model.

‡ https://www.optionsdx.com
§ https://home.treasury.gov

Table 1. Summary statistics of variables of interest. TtM means time to maturity and Rf rate is risk-free rate.

	Option price	S&P 500	Strike	Time to maturity (years)	Interest rate	Moneyness
mean	359.47	3047.01	2784.52	0.25	0.94	1.14
std	393.74	915.48	993.32	0.34	1.1	0.23
min	0.02	1158.67	600	0	− 0.01	0.8
median	235.3	2884.77	2660	0.12	0.29	1.07
max	2405.4	4795.57	5900	2	4.82	2

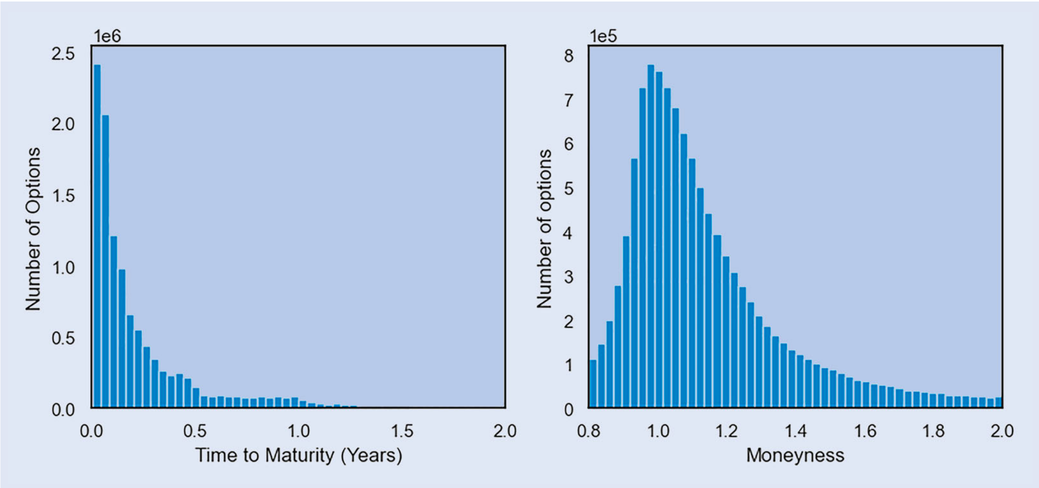

Figure 2. Histograms of the number of options quotes at different levels of maturity and moneyness.

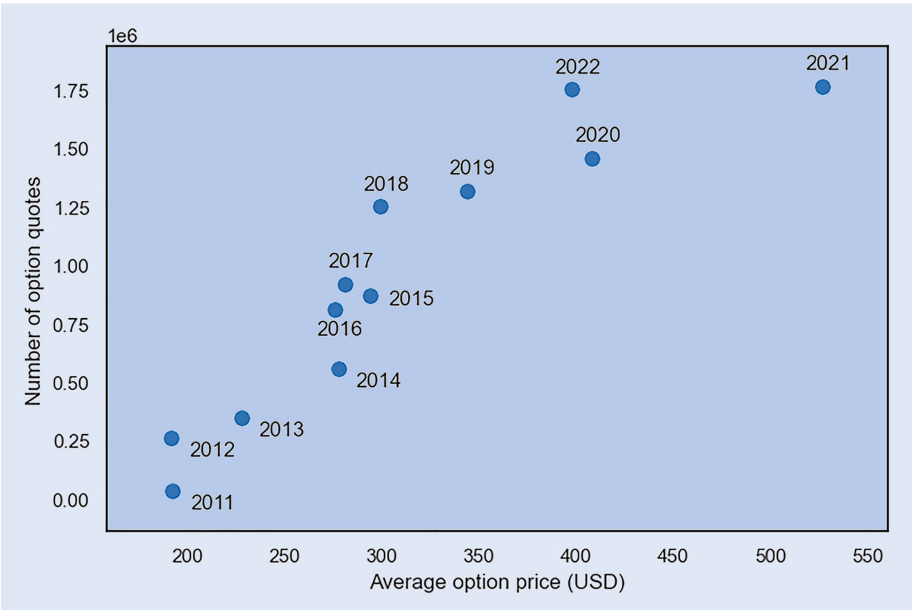

Figure 3. Scatter plot of the number of options versus the average option price for each year.

4. Results and discussion

We test the merged LSTM-MLP model pricing European call options from 2015 to 2022, during which the model is evaluated using 96 sliding windows with a one-month out-of-sample test period each. In total, 10 136 680 options are priced out-of-sample. Table 2 presents the overall pricing performance of the merged LSTM-MLP and benchmark models. The merged LSTM-MLP pricing model outperforms all benchmark models in terms of pricing accuracy, measured by the RMSE and MAE. The Diebold-Mariano test confirms that these pricing differences are statistically significant, as reported in Table 3. The finding that both neural network-based models perform better than BS is supported by multiple previous works in the literature (Hutchinson *et al.* 1994, Amilon 2003, Wang *et al.* 2022, Iltüzer 2022).

In comparison to the benchmark MLP, the increased performance of the LSTM-MLP suggests that the merged LSTM-MLP architecture does enhance pricing accuracy. More

Table 2. RMSE and MAE for each pricing model from 2015 to 2022. The lowest (best) value for each metric is highlighted in bold.

Metric	BS 30-day	BS GARCH	BS IV	Heston	MLP	LSTM-MLP
RMSE	35.92	36.26	20.42	13.15	17.10	**11.84**
MAE	15.48	15.52	10.78	7.47	8.86	**6.61**

Table 3. Diebold-Mariano test statistics for each pair of models. All values are significant at the $p < 0.01$ level.

	BS 30-day	BS GARCH	BS IV	Heston	MLP	LSTM-MLP
BS 30-day		6.27	– 255.91	– 302.42	– 271.42	– 311.73
BS GARCH	– 6.27		– 250.20	– 281.67	– 254.95	– 288.92
BS IV	255.91	250.20		– 325.54	– 143.26	– 368.01
Heston	302.42	281.67	325.54		266.82	– 180.88
MLP	271.42	254.95	143.26	– 266.82		– 344.71
LSTM-MLP	311.73	288.92	368.01	180.88	344.71	

specifically, it demonstrates that an MLP with inputs from an LSTM outperforms an MLP with a single 30-day historical volatility input.

Table 4 displays the RMSE pricing errors for all models per year. In every year, except for 2022, the merged LSTM-MLP model is the top performer, achieving the lowest RMSE compared to the benchmark models. In 2022, however, the Heston model marginally surpasses the merged LSTM-MLP model. This can be attributed to the nature of the Heston model, which explicitly captures stochastic volatility and its correlation with asset prices. During periods of extreme market uncertainty or specific volatility dynamics, such as those observed in 2022, this explicit representation may offer a slight edge over data-driven models that rely on learned patterns. Additionally, the LSTM-MLP model's performance in this year could be influenced by unique market conditions that were less represented in the training data.

Despite its flexibility, the MLP model still falls short of consistently outperforming the Heston model. One possible reason is the reliance of the MLP on the 30-day historical volatility, which might not capture rapid market changes effectively. It is conceivable that the MLP could be improved with alternative volatility measures. However, this raises the crucial point of our research: the assumption-laden process of choosing specific volatility measures. Given this, enabling the model to learn its own volatility representation might be more effective, freeing it from predefined assumptions.

The VIX index measures the market's expectation of future volatility. This makes it a key measure in option pricing, as higher volatility generally leads to higher option prices. We

Table 4. RMSE for each pricing model for each year from 2015 to 2022. The lowest (best) value for each year is highlighted in bold.

Year	BS 30-day	BS GARCH	BS IV	Heston	MLP	LSTM-MLP
2015	11.99	13.99	8.70	6.55	7.10	**3.12**
2016	11.23	12.37	8.72	4.85	5.30	**3.33**
2017	11.79	9.81	9.03	3.81	3.41	**3.20**
2018	18.25	22.75	13.07	7.76	11.46	**5.24**
2019	19.82	18.50	12.51	6.52	7.32	**6.12**
2020	72.90	70.83	31.24	22.79	34.88	**21.41**
2021	36.57	34.66	25.98	16.15	14.39	**12.54**
2022	31.20	37.28	23.84	**13.84**	17.21	14.45

Table 5. RMSE and MAE for each pricing model for each VIX regime. The low regime is defined as VIX values below 14.22 and the high regime as VIX values above 20.48, while the medium regime comprises the VIX values in between. Each regime consists of 1/3 of the total days in the period. The lowest (best) value for each VIX regime and measure is highlighted in bold.

	BS 30-day	BS GARCH	BS IV	Heston	MLP	LSTM-MLP
Low VIX						
RMSE	12.64	11.21	9.84	4.74	4.91	**4.42**
MAE	6.67	6.01	5.61	3.54	3.23	**3.02**
Medium VIX						
RMSE	25.51	25.07	17.33	10.78	10.29	**8.69**
MAE	13.45	12.92	10.07	6.62	6.69	**5.27**
High VIX						
RMSE	51.23	52.25	27.18	18.04	25.28	**16.71**
MAE	23.32	24.33	14.96	10.91	14.62	**10.26**

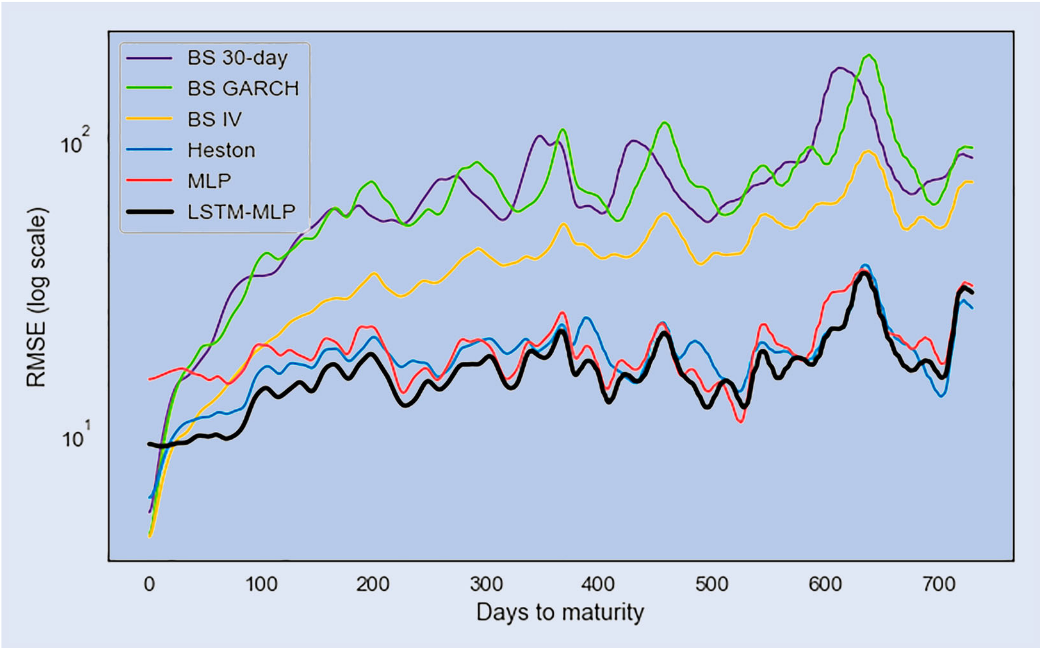

Figure 4. RMSE for each pricing model for different maturities. (the graphs are slightly smoothed for readability).

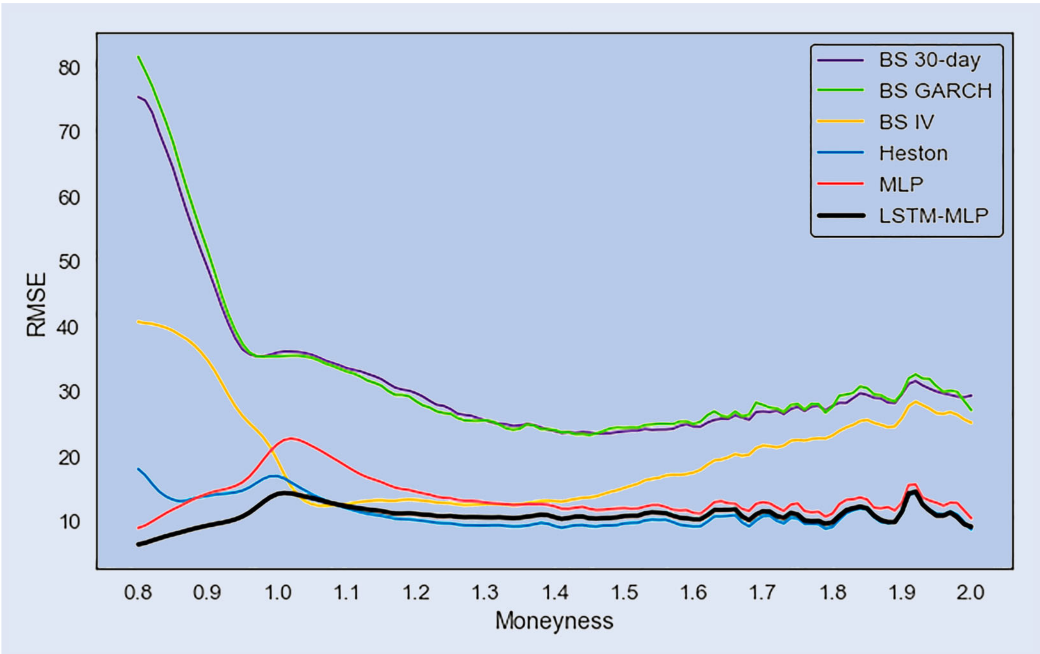

Figure 5. RMSE for each pricing model at different moneyness. (the graphs are slightly smoothed for readability).

explore this aspect further by examining the pricing performance of the models across three distinct VIX† regimes. We divide the VIX range of all days in the test period into three equal parts, resulting in threshold values of 14.22 and 20.48. Table 5 presents the results.

The merged LSTM-MLP model stands out in all scenarios with the lowest RMSE and MAE. This suggests that the model can effectively adjust its pricing behavior to prevailing market volatility conditions due to its LSTM component, which captures the temporal dependencies in the return data for the S&P 500 index. In contrast, the MLP model only surpasses all other benchmark models in the medium VIX regime regarding RMSE and in the low VIX regime in terms of MAE. MLP exhibits a worsening relative performance in the high VIX regime compared to the closest competitors. This means that the MLP model may not be as robust when it comes to adjusting to different market volatility conditions.

Other relevant characteristics in option pricing are maturity and moneyness. Figures 4 and 5 illustrate the pricing performance of each model for different maturities and at varying levels of moneyness, respectively.

From Figure 4, we observe that, for all models, the RMSE generally increases for longer maturities, which is expected given the increased option values for longer maturities. When comparing the merged LSTM-MLP and the MLP models, it

† The daily VIX data is retrieved from Cboe used: https://www.cboe.com/.

Table 6. RMSE for each pricing model, grouped by moneyness and maturity. For moneyness, out-of-the–money is defined as less than 0.97, in-the-money as greater than 1.03, and at-the-money in between. The lowest (best) value for each combination of moneyness and maturity is highlighted in bold.

Moneyness	TtM	BS 30-day	BS GARCH	BS IV	Heston	MLP	LSTM-MLP
out-of-the-money	0-30	12.15	13.65	6.85	7.22	7.33	**4.98**
	31-90	31.20	31.99	18.66	13.04	13.79	**8.52**
	91-300	65.39	68.98	43.57	20.20	20.26	**13.41**
	301-730	104.34	107.84	73.23	23.27	23.61	**17.22**
at-the-money	0-30	15.68	13.43	12.35	12.94	19.26	**11.71**
	31-90	26.74	25.30	17.19	16.99	22.24	**12.83**
	91-300	56.32	56.94	27.24	22.69	25.38	**18.67**
	301-730	100.49	101.99	48.95	23.97	25.08	**21.76**
in-the-money	0-30	8.04	6.61	**4.71**	6.23	16.60	9.11
	31-90	16.62	15.13	**7.74**	9.19	14.57	9.42
	91-300	41.12	40.75	17.17	14.77	16.78	**14.29**
	301-730	72.45	72.59	36.43	21.41	20.51	**20.31**

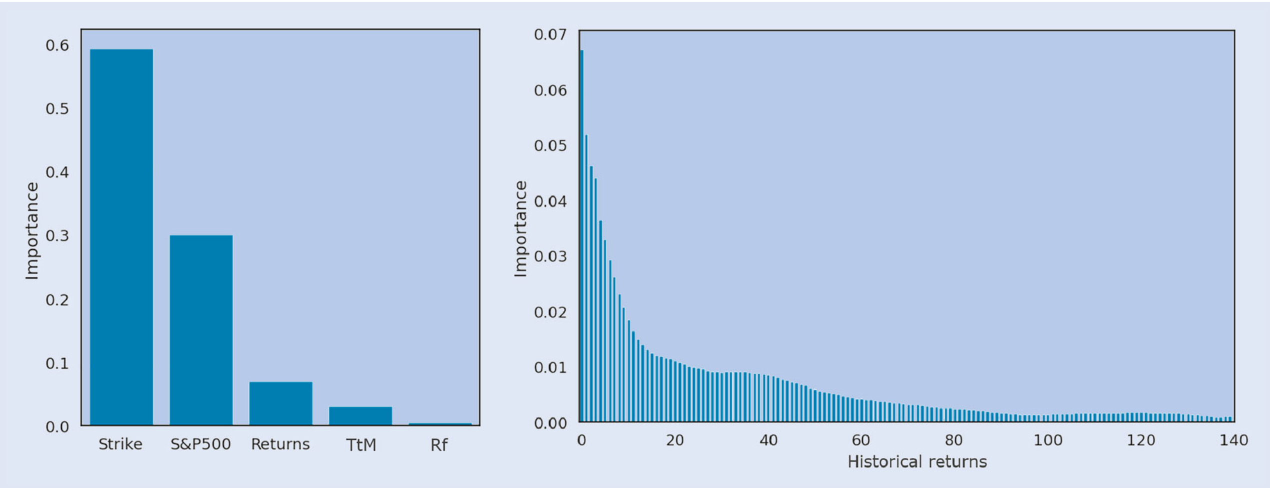

Figure 6. ALE feature importance for the merged LSTM-MLP model averaged across the last model trained each year from 2015 to 2022. The left plot shows the feature importance across the option characteristics with aggregated returns impact. The right plot shows the feature importance of each individual lagged return towards the overall impact of the return series.

is noticeable that the former outperforms the latter for short-maturity options but shows similar performance for long-maturity options. It, therefore, seems that the LSTM input is more significant in extracting the relevant volatility for the short and medium maturities relative to the explicit volatility input given to the MLP model. We also see that the neural network-based models have relatively lower performance for extremely short maturities when compared with the remaining models. This may be attributed to the lower liquidity of these options, which has two direct implications: less data to train the model and a lower signal-to-noise ratio. Also, the behavior of very short-term options is highly sensitive to immediate market movements, which the models have not observed yet.

Figure 5 shows that the merged LSTM-MLP outperforms all benchmark models for out-of-the-money and at-the-money options. For deeper in-the-money options, the merged LSTM-MLP has a lower RMSE than most of the benchmark models, with the exception of the Heston model. The Heston model demonstrates consistently strong performance across moneyness levels due to its ability to capture the volatility smile through its stochastic volatility framework. This allows it to perform comparably to the LSTM-MLP for in-the-money options, particularly where volatility effects are more pronounced. The underperformance of the MLP also lessens for in-the-money options but remains consistently worse than the merged LSTM-MLP. Low moneyness corresponds to a highly leveraged position in the underlying asset. The leverage effect in equities, depicting asymmetric volatility of returns, is well-known. It is plausible that the LSTM-MLP model more accurately captures this, resulting in lower RMSE for low moneyness options.

To have a combined analysis of the maturity and moneyness, Table 6 presents the RMSE for each pricing model at different maturities within a specified level of moneyness. For out-of-the-money and at-the-money options, the merged LSTM-MLP outperforms the benchmark models across maturities. This is also in line with the one-dimensional analysis of pricing performance by the moneyness dimension shown in Figure 5. This suggests that while traditional stochastic volatility models such as Heston can struggle with reproducing steep short-term skews (as noted in Bayer *et al.* 2016), our model's flexible architecture allows it to learn these patterns directly from the data.

The merged LSTM-MLP model is particularly effective at capturing the behavior of OTM options. For ATM options, the

Table 7. Average yearly trading performance for each option pricing model from 2015 to 2022 compared to holding the S&P 500. The best value for each specific metric is highlighted in bold.

Metric	BS 30-day	BS GARCH	BS IV	Heston	MLP	LSTM-MLP	S&P 500
Return (%)	2.86	− 1.87	− 2.31	4.70	9.52	**18.47**	15.6
Sharpe	0.16	− 0.52	− 0.21	0.38	0.83	**1.28**	1.26
MDD (%)	− 10.19	− 8.02	− 9.92	− 12.98	**− 7.74**	− 9.26	− 12.64

Table 8. Alpha and Beta values based on the CAPM for each of the option pricing models studied.

	BS 30-day	BS GARCH	BS IV	Heston	MLP	LSTM-MLP
α (%)	3.33	0.16	0.08	8.30	10.86	21.14
SE α	4.92	3.56	4.03	5.66	4.25	5.35
α p-value	0.4985	0.9639	0.9836	0.1430	0.0107**	8.108e-05***
β	0.0802	0.1510	0.3060	0.0527	− 0.1329	− 0.2564
SE β	0.0164	0.0119	0.0134	0.0189	0.0142	0.0178
β p-value	1.12e-06***	9.98e-36***	4.59e-102***	0.0053***	1.73e-20***	1.33e-44***

Note: The statistical significance of the results is denoted by asterisks: *** indicates significance at the 1% level, ** indicates the 5% level, and * indicates the 10% level. A model with alphas and betas that are statistically significantly different from zero is able to provide distinct predictions of asset pricing.

merged LSTM-MLP model again shows the lowest RMSE across all maturities, but the differences in RMSE between models are smaller compared to OTM. For all models, the performance is better for shorter maturities, which may indicate that the models better capture the implied volatility for short-term ATM options.

The outperformance across models is more mixed for in-the-money options. The merged LSTM-MLP struggles for short and medium-term (lower than 90 days) options, for which BS with implied volatility performs best. This is also seen in Figure 5, where there is a specific moneyness when the BS IV has the best performance measure. If there are volatility measurement errors, they have a smaller impact on the BS framework for shorter than longer maturities. Therefore, the merged LSTM-MLP has the best performance for longer maturities (larger than 90 days) again.

Overall, we conclude the merged LSTM-MLP is the most parsimonious model, in the sense that even if it does not outperform in all cases, it is the one that performs best for most of the situations.

To shed some light on the impact of each feature on the decisions made by the merged LSTM-MLP model, we use the global explainable AI method called Accumulated Local Effects (ALE), as introduced by Apley and Zhu (2020). Figure 6 illustrates the average feature importance of each feature for the last models trained each year from 2015 to 2022. We see that the return series, which is the input to the LSTM network, achieves an importance of 7%, as a whole. Breaking down the impact of the return series, we see the expected behavior of the most recent observations being the most important, but it still considers every historical value.

Figure 7 illustrates the feature importance for the MLP benchmark model when an explicit volatility input is employed. The MLP model assigns low importance to the volatility input at 0.2%. This observation aligns with Liang and Cai (2022) who also identified historical volatility as the least important feature. They reported a comparable importance percentage for the volatility input for both their MLP and LSTM models in pricing S&P 500 options.

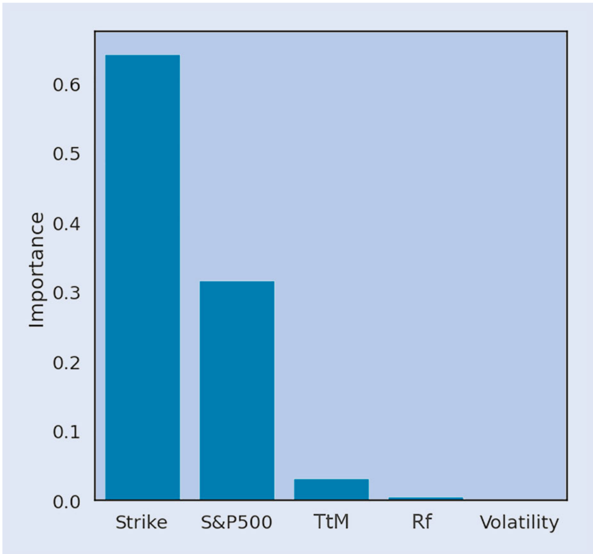

Figure 7. ALE feature importance for the MLP benchmark model averaged across the last model trained each year from 2015 to 2022.

The stark contrast between the importance assigned to the return series in the merged LSTM-MLP model and the volatility input in the MLP model provides additional evidence supporting the efficacy of time series information extraction using an LSTM component in the merged LSTM-MLP model. While the strike and S&P500 prices naturally prove the most important in determining the option price, the 7% feature importance attributed to the return series appears to be the differentiating factor causing the merged LSTM-MLP model to outperform the MLP.

Finally, we evaluate the trading performance of the different models, following the trading strategy presented before. Table 7 shows the average yearly trading performance of each option pricing model across the full out-of-sample test period from 2015 to 2022. We observe that the merged LSTM-MLP model outperforms all benchmark models measured by Sharpe ratio, Sortino ratio, and percentage return. However,

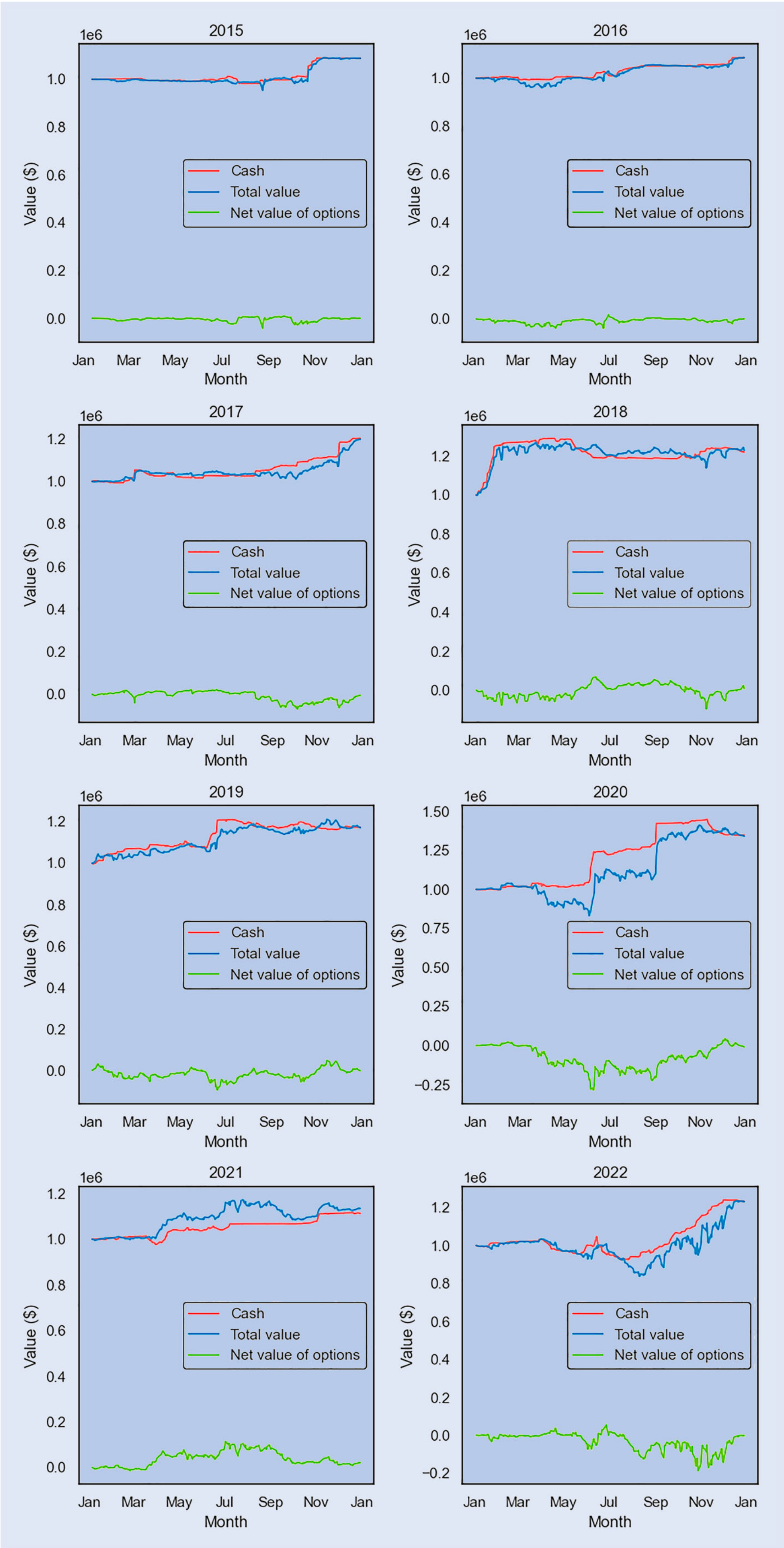

Figure 8. Annual progression of cash balance, net value of the options portfolio, and total portfolio value for the merged LSTM-MLP trading model from 2015 to 2022.

the MLP and BS GARCH perform best in terms of maximum drawdown. The finding that a neural network-based pricing model achieves an overall best trading performance is consistent with Andreou *et al.* (2008).

The combination of better values for returns and Sharpe and Sortino ratios emphasizes the ability of the merged LSTM-MLP to achieve better risk-adjusted returns than all benchmark models. This suggests that the model is better at accurately pricing options in a way that aligns more closely with the true option value.

Table 8 shows the trading results in terms of alpha and beta values using the CAPM. It should be noted that even though the CAPM model is used in this research, it should not be seen as an endorsement of the model for options trading. As stated in Leland (1999), it is not ideal to apply the CAPM to strategies with positively skewed returns, for which strategies that limit downside risk will be mismeasured. Despite these potential shortcomings, CAPM is still utilized in our study as it facilitates a certain degree of performance comparison between the models and because within the hedge fund industry, α is frequently used as the go-to reference model for benchmarking trading strategies (Ewald *et al.* 2022).

The table shows that for all models, only the LSTM-MLP has an alpha significant at the 1% level. This implies strong statistical confidence that the excess return generated by the LSTM-MLP model is not due to chance or by taking on systematic risk, but rather the proficiency of the model. The MLP model is the only benchmark model to achieve a statistically significant alpha at the 5% level. The superior alpha for the LSTM-MLP model provides further support for the improved risk-adjusted performance of the proposed model.

Figure 8 illustrates the development in the cash position and net value of the options portfolio,† and total value‡ for each trading year using trading signals generated by the merged LSTM-MLP. The graphs clearly demonstrate a consistent upward trajectory for the total value, depicted by the blue line. We also observe that in some periods, few trades are made. This is either due to the fact that the model identifies few sufficiently mispriced options at the time or that it only identifies overpriced or underpriced options, thus not being able to create a balanced portfolio satisfying the trading restrictions.

5. Conclusion

The expected volatility of the underlying is indisputably the most important factor determining option premia. Although the true volatility is not directly observable, and hence must be approximated with error, traditional option pricing models, among them Black-Scholes-Merton and Heston, require numerical estimates of volatility. The superior performance of the novel merged LSTM-MLP option pricing model proposed in this paper stems from the convenient property of avoiding explicit estimates of expected volatility. This is enabled through a hybrid deep learning neural network that combines times series information with the forward-looking information contained in the cross-section of option prices.

We consider data from November 1, 2011, to December 31, 2022. This period covers important economic events impacting volatility across global asset classes, such as the 2012–2015 EURO banking and sovereign crises, the 2015 oil price collapse, the 2020 Covid19 outbreak and the Russian invasion in 2022. In addition to covering a relatively long time period, which we consider representative for the problem at hand, we take a number of additional measures to ensure robustness of results. In contrast to the majority of the times series literature using deep learning models we apply a rolling window estimation approach, which, according to our opinion, is closer to how practitioners would employ the model. We also estimate a broad set of benchmark models. Furthermore, we analyze model performance at relatively fine granularity in terms of calendar year, time to maturity and moneyness. We document the usefulness of the proposed merged LSTM-MLP model not only in terms of statistical accuracy, by also using risk-adjusted measures of profitability in a simulated real-world constrained trading portfolio.

The most striking, and very appealing, characteristic of the proposed model is it's highly satisfactory performance across all the dimensions mentioned above. In particular, we note the superior performance of the model during periods of market turbulence. This is particularly interesting, as rapid and temporal changes in the market environment are well-known to cause challenges for most volatility models. As the merged LSTM-MLP does not explicitly estimate volatility, it is less prone to errors of the type caused by biased coefficient values in parametric pricing models.

Given the demonstrated ability of the LSTM component to extract valuable time series information in the option pricing context, it might prove interesting to experiment with different sequential models, such as transformers or alternative recurrent neural network architectures, in combination with an MLP. Building on this capability to extract temporal patterns, another promising direction would be to extend the merged LSTM-MLP model to explicitly distinguish between overnight and intraday returns (Muravyev and Ni 2020), as the LSTM component's ability to selectively retain or forget information could help identify which historical price movements - overnight jumps versus intraday drift - are most informative for option pricing.

Acknowledgments

We are grateful to particpants at the NTNU Fintech, AI in Finance and Banking Conference in Trondheim, 23–24 May 2023, for helpful comments.

Disclosure statement

No potential conflict of interest was reported by the author(s).

† Value of the long positions minus the short.
‡ Sum of cash and net value of options portfolio.

Funding

This research was partially funded by The Research Council of Norway throughout the project COMPAMA (https://www.ntnu.edu/compama/), with grant number 314609.

CRediT statement

Vinje: Conceptualization, data curation, methodology, formal analysis, implementation, interpretation, writing–original draft, editing, and reviewing.

Rygg and Wu: Conceptualization, data curation, methodology, formal analysis, implementation, interpretation.

Risstad: Supervision, conceptualization, methodology, validation, interpretation, writing–improving original draft, editing, and reviewing.

Westgaard: Supervision, conceptualization, validation.

Pimentel: Funding, interpretation, writing – improving original draft, editing, and reviewing.

Ewald: Conceptualization, methodology, writing – improving original draft and editing.

References

Amilon, H., A neural network versus Black-Scholes: A comparison of pricing and hedging performances. *J. Forecast.*, 2003, **22**, 317–335.

Andersen, T.G., Fusari, N. and Todorov, V., Parametric inference and dynamic state recovery from option panels. *Econometrica*, 2015, **83**, 1081–1145.

Andreou, P.C., Charalambous, C. and Martzoukos, S.H., Pricing and trading European options by combining artificial neural networks and parametric models with implied parameters. *Eur. J. Oper. Res.*, 2008, **185**, 1415–1433.

Apley, D.W. and Zhu, J., Visualizing the effects of predictor variables in black box supervised learning models. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 2020, **82**, 1059–1086.

Bali, T.G., Beckmeyer, H., Mörke, M. and Weigert, F., Option return predictability with machine learning and big data. *Rev. Financ. Stud.*, 2023, **36**(9), 3548–3602.

Bayer, C., Friz, P. and Gatheral, J., Pricing under rough volatility. *Quant. Finance*, 2016, **16**, 887–904.

Black, F. and Scholes, M., The valuation of option contracts and a test of market efficiency. *J. Finance*, 1972, **27**, 399.

Black, F. and Scholes, M., The pricing of options and corporate liabilities. *J. Pol. Econ.*, 1973, **81**, 637–654.

Bollerslev, T., Generalized autoregressive conditional heteroskedasticity. *J. Econom.*, 1986, **31**, 307–327.

Cao, Y., Liu, X. and Zhai, J., Option valuation under no-arbitrage constraints with neural networks. *Eur. J. Oper. Res.*, 2021, **293**, 361–374.

Cont, R. and Tankov, P., *Financial Modelling with Jump Processes*, 2003 (Chapman and Hall/CRC: Boca Raton, FL).

Culkin, R. and Das, R., Machine learning in finance: The case of deep learning for option pricing. *J. Invest. Manag.*, 2017, **15**, 1–9.

Diebold, F.X. and Mariano, R.S., Comparing predictive accuracy. *J. Bus. Econ. Stat.*, 1995, **13**(3), 253–263.

Ewald, C.O., Haugom, E., Lien, G., Størdal, S. and Wu, Y., Trading time seasonality in commodity futures: An opportunity for arbitrage in the natural gas and crude oil markets? *Energy Econ.*, 2022, **115**, 106324.

Gençay, R. and Gibson, R., Model risk for European-style stock index options. *IEEE Trans. Neural Networks*, 2007, **18**, 193–202.

Gradojevic, N. and Kukolj, D., Unlocking the black box: Non-parametric option pricing before and during COVID-19. *Ann. Oper. Res.*, 2022, **334**(1), 59–82.

He, C., Kennedy, J.S., Coleman, T.F., Forsyth, P.A., Li, Y. and Vetzal, K.R., Calibration and hedging under jump diffusion. *Rev. Deriv. Res.*, 2006, **9**, 1–35.

Heston, S., A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Financ. Stud.*, 1993, **6**, 327–343.

Hornik, K., Stinchcombe, M. and White, H., Multilayer feedforward networks are universal approximators. *Neural Netw.*, 1989, **2**, 359–366.

Hsu, P.Y., Chou, C., Huang, S.H. and Chen, A.P., A market-making quotation strategy based on dual deep learning agents for option pricing and bid-ask spread estimation. In *Proceedings - 2018 IEEE International Conference on Agents, ICA 2018*, pp. 99–104, 2018 (Atlantis Press).

Hutchinson, J.M., Lo, A.W. and Poggio, T., A nonparametric approach to pricing and hedging derivative securities via learning networks. *J. Finance*, 1994, **49**, 851–889.

Iltüzer, Z., Option pricing with neural networks vs. Black-Scholes under different volatility forecasting approaches for BIST 30 index options. *Borsa Istanb. Rev.*, 2022, **22**, 725–742.

Ioffe, S. and Szegedy, C., Batch normalization: Accelerating deep network training by reducing internal covariate shift, In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448–456, 2015.

Ivaşcu, C., Option pricing using machine learning. *Expert Syst. Appl.*, 2021, **163**, 113799.

Kingma, D. and Ba, J., Adam: A method for stochastic optimization, 2014. arXiv preprint arXiv:1412.6980.

Leland, H.E., Beyond mean-variance: Performance measurement in a nonsymmetrical world. *Financ. Anal. J.*, 1999, **55**, 27–36.

Liang, L. and Cai, X., Time-sequencing European options and pricing with deep learning – analyzing based on interpretable ALE method. *Expert Syst. Appl.*, 2022, **187**, 115951.

Liu, D. and Wei, A., Regulated LSTM artificial neural networks for option risks. *FinTech*, 2022, **1**, 180–190.

Liu, S., Oosterlee, C. and Bohte, S., Pricing options and computing implied volatilities using neural networks. *Risks*, 2019, **7**, 16.

Lo, A.W., Statistical tests of contingent claims asset pricing models. *J. Financ. Econ.*, 1986, **17**, 143–173.

Maas, A.L., Hannun, A.Y. and Ng, A.Y., Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning*, Vol. 30, 2013. JMLR Workshop and Conference Proceedings.

Merton, R., Theory of rational option pricing. *Bell J. Econ. Manage. Sci.*, 1973, **4**, 141–183.

Muravyev, D. and Ni, X., Why do option returns change sign from day to night? *J. Financ. Econ.*, 2020, **136**, 219–238.

Ruf, J. and Wang, W., Neural networks for option pricing and hedging: A literature review. *J. Comput. Finance*, 2020, **24**, 1–46.

Sharpe, W.F., Mutual fund performance. *J. Bus.*, 1966, **39**, 119–138.

Sortino, F.A., Performance measurement in a downside risk framework. *J. Invest.*, 1994, **3**, 59–64.

Svensson, L.E., Estimating and interpreting forward interest rates: Sweden 1992–1994, 1994. IMF Working Paper No. 94/114, 76 pp. International Monetary Fund.

Taylor, S.J., *Modelling Financial Time Series*, 1986 (Wiley: Chichester).

Wang, C.P., Lin, S.H., Huang, H.H. and Wu, P.C., Using neural network for forecasting TXO price under different volatility models. *Expert Syst. Appl.*, 2012, **39**, 5025–5032.

Wang, M., Zhang, Y., Qin, C., Liu, P. and Zhang, Q., Option pricing model combining ensemble learning methods and network learning structure. *Math. Probl. Eng.*, 2022, **2022**, Article ID 5843491, 1–20.

Yang, Y., Zheng, Y. and Hospedales, T., Gated neural networks for option pricing: Rationality by design. In *31st AAAI Conference*

on *Artificial Intelligence*, Vol. 31, No. 1, pp. 52–58, 2017 (AAAI Press).
Yao, J., Li, Y. and Tan, C., Option price forecasting using neural networks. *Omega*, 2000, **28**, 455–466.

Appendix 1. Hyperparameter configuration

The choice of hyperparameters in machine learning models is crucial because it can have a meaningful impact on performance. Therefore, to ensure a systematic and efficient approach, we conduct a hyperparameter search using Wandb,† a machine learning development tool for experiment tracking. Wandb is helpful in keeping a detailed record of model runs and facilitates parallel exploration of multiple hyperparameter configurations. This is highly beneficial for the merged LSTM-MLP architecture due to the significant number of hyperparameters that need to be specified.

The hyperparameter search for the merged LSTM-MLP is conducted using a window ending in December 2014. The validation period is purposefully set before the first out-of-sample test period in January 2015 to avoid data leakage between validation and out-of-sample testing. To match the total duration of the one-month validation and one-month test period in the 96 sliding windows used for out-of-sample testing, a two-month validation period is used for the hyperparameter search. With this setup, the hyperparameter search uses training data for the three years from November 2011 to October 2014, and validation data for the two months from November to December 2014.

The ranges of hyperparameter values to be searched are determined based on values that have been used in previous studies, including Liu *et al.* (2019) and Liang and Cai (2022). Further adjustments to these ranges are made based on observed results. Due to the extensive number of hyperparameters, it is not computationally feasible to conduct a grid search testing all possible combinations. Exploration of the hyperparameter space is therefore carried out using a random search algorithm where 150 unique combinations of hyperparameters are generated and tested. Figure A1 shows the model run for each unique combination of hyperparameters and the resulting validation loss.

The hyperparameter search shows a clear preference for a higher number of LSTM timesteps, meaning more past S&P 500 returns. It seems intuitive that providing more historical data can improve model performance, assuming that the additional information does not introduce too much unnecessary noise or lead to excessive overfitting. This is where the strength of the LSTM network comes in, with its ability to forget irrelevant historical data while retaining valuable information selectively. The clear preference for longer periods of historical data supports our hypothesis that the LSTM is capable of extracting useful information from long-term patterns in historical data.

The final hyperparameters are chosen based on inspection of Figure A1 and analysis of the highest performing runs during the hyperparameter search. The hyperparameter and configuration choices made based on the hyperparameter search and discussion are summarized in Table A1.

† https://wandb.ai/

Appendix 2. Data set

Table A2 presents detailed descriptive statistics per year.

Table A1. LSTM-MLP model configuration grouped by model component. The parameters layers, units, interface units, timesteps, BN momentum, alpha constant, learning rate and learning rate decay were found during the hyperparameter search using Wandb while the others were determined manually beforehand to reduce the hyperparameter space.

	Parameter	Value
LSTM	Layers	3
	Units per layer	8
	Units in the last layer (Interface units)	8
	Lagged returns (Timesteps)	140
	Activation function	Tanh
	Recurrent activation function	Sigmoid
MLP	Layers	4
	Units per layer	200
	Batch normalization momentum	0.10
	Activation function	Leaky ReLu
	Alpha constant for Leaky ReLu	0.05
Training	Training set samples	1 000 000
	Learning rate	5.7×10^{-2}
	Learning rate decay	0.91
	Minibatch size	4096
	Epochs	Early stopping
	Optimizer	Adam
Sliding windows setup	Training length	3 years
	Validation length	1 month
	Test length	1 month
	Number of windows	96
Computations	Hardware	Nvidia Tesla T4 GPU
	Hyperparameter search run time	26 hours
	Final model average run time per window	13 min
	Total run time all windows	21 hours

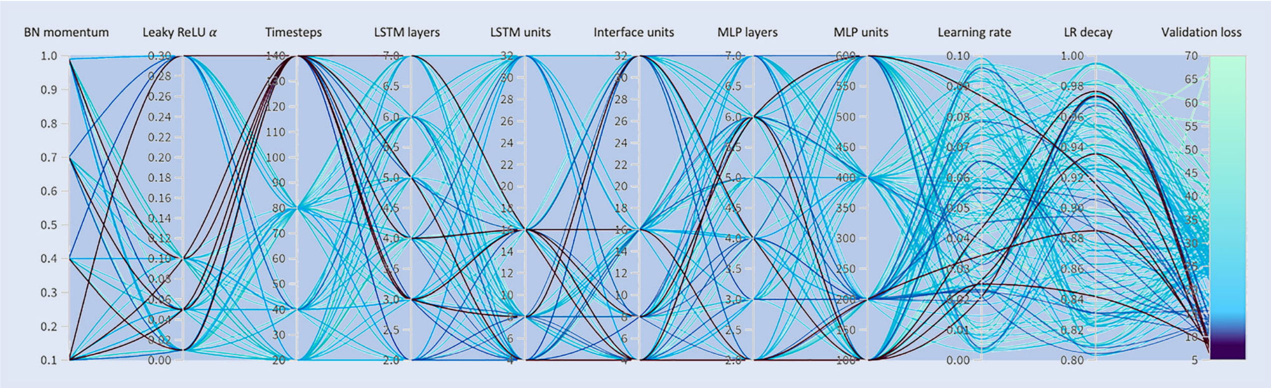

Figure A1. Hyperparameter search for the LSTM-MLP model in terms of hyperparameter combinations, hyperparameter value ranges and MSE validation loss. Results are shown for model runs testing 150 distinct combinations of hyperparameter values. The data set used consists of training data for the three years from November 2011 to October 2014, and validation data for the two months from November to December 2014.

Table A2. Descriptive statistics of all options data used for hyperparameter search, training, validation, and testing.

Year	Measure	Option price	S&P 500	Strike	TtM (years)	Rf rate (%)	Moneyness
2011	count	33 080	33 080	33 080	33 080	33 080	33 080
	mean	192.32	1236.11	1098.89	0.32	0.03	1.18
	std	173.97	27.61	228.79	0.39	0.05	0.28
2012	count	259 343	259 343	259 343	259 343	259 343	259 343
	mean	191.6	1383.49	1239.55	0.34	0.10	1.17
	std	187.5	45.79	245.32	0.42	0.05	0.26
2013	count	345 503	345 503	345 503	345 503	345 503	345 503
	mean	227.38	1653.16	1462.91	0.32	0.07	1.17
	std	218.48	98.33	276.85	0.4	0.06	0.24
2014	count	554 101	554 101	554 101	554 101	554 101	554 101
	mean	277.43	1942.64	1700.84	0.28	0.05	1.18
	std	250.94	80.19	302.33	0.38	0.07	0.24
2015	count	869 602	869 602	869 602	869 602	869 602	869 602
	mean	294.05	2059.37	1812.5	0.25	0.08	1.18
	std	268.59	56.83	326.75	0.32	0.13	0.24
2016	count	811 805	811 805	811 805	811 805	811 805	811 805
	mean	275.46	2097.31	1870.22	0.24	0.33	1.16
	std	272.01	102.25	338.59	0.34	0.14	0.24
2017	count	919 659	919 659	919 659	919 659	919 659	919 659
	mean	280.88	2451.22	2211.72	0.21	0.92	1.14
	std	299.24	111.24	358.84	0.33	0.26	0.22
2018	count	1 249 550	1 249 550	1 249 550	1 249 550	1 249 550	1 249 550
	mean	299.29	2745.12	2525.13	0.23	1.97	1.12
	std	330.92	101.72	404.58	0.32	0.34	0.22
2019	count	1 317 850	1 317 850	1 317 850	1 317 850	1 317 850	1 317 850
	mean	343.64	2917.99	2644.67	0.23	2.09	1.14
	std	358.96	155.2	445.94	0.32	0.34	0.22
2020	count	1 454 860	1 454 860	1 454 860	1 454 860	1 454 860	1 454 860
	mean	408.38	3219.94	2928.58	0.24	0.38	1.14
	std	404.78	314.85	553.86	0.32	0.55	0.23
2021	count	1 762 540	1 762 540	1 762 540	1 762 540	1 762 540	1 762 540
	mean	526.87	4295.38	3887.66	0.28	0.05	1.14
	std	525.04	286.16	695.49	0.34	0.05	0.22
2022	count	1 750 820	1 750 820	1 750 820	1 750 820	1 750 820	1 750 820
	mean	397.96	4108.23	3916.73	0.27	2.00	1.08
	std	456.77	295.95	639.11	0.34	1.52	0.20