Computational Economics
https://doi.org/10.1007/s10614-024-10689-z

Enhancing Option Pricing Accuracy in the Indian Market: A CNN-BiLSTM Approach

Akanksha Sharma[1] · **Chandan Kumar Verma**[1] · **Priya Singh**[1]

Accepted: 22 July 2024
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract
Due to overly optimistic economic and statistical assumptions, the classical option pricing model frequently falls short of ideal predictions. Rapid progress in artificial intelligence, the availability of massive datasets, and the rise in computational power in machines have all created an environment conducive to the development of complex methods for predicting financial derivatives prices. This study proposes a hybrid deep learning (DL) based predictive model for accurate and prompt prediction of option prices by fusing a one-dimensional convolutional neural network (CNN) and a bidirectional long short-term memory (BiLSTM). A set of 15 predictive factors is carefully built under the umbrella of fundamental market data and technical indicators. Our proposed model is compared with other DL-based models using six evaluation metrics-root mean square error (RMSE), mean absolute percentage error, mean percentage error, determination coefficient (R^2), maximum error and median absolute error. Further, statistical analysis of models is also done using one-way ANOVA and posthoc analysis using the Tukey HSD test to demonstrate that the CNN-BiLSTM model outperforms competing models in terms of fit and prediction accuracy.

Keywords CNN-BiLSTM · Technical indicators · Deep learning · Option pricing · Derivatives

Chandan Kumar Verma and Priya Singh have contributed equally to this work.

✉ Akanksha Sharma
akanksha199906@gmail.com

Chandan Kumar Verma
chandankverma@gmail.com

Priya Singh
priya.s.parihar94@gmail.com

[1] Department of Mathematics, Bioinformatics, and Computer Applications, Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh 462003, India

Published online: 01 August 2024

1 Introduction

Derivatives, integral to financial markets, facilitate the trading of specific financial risks tied to various instruments, indicators, or commodities. In India, the Securities Laws (Amendment) Ordinance 1995 marked the initiation of financial derivatives trading, primarily conducted on major stock exchanges like the National Stock Exchange (NSE) and the Bombay Stock Exchange (BSE) (Srivastava & Shastri, 2018). Following that, the derivatives market in India has experienced substantial expansion concerning both the value of trades and the quantity of contracts exchanged.

Derivative pricing is the process of determining optimal values based on market conditions and predictions of asset movements, influenced by factors such as time to maturity, underlying asset value, volatility, interest rates, dividend yields, and prevailing market circumstances. However, challenges arise in capturing market dynamics, handling interest rate fluctuations, and addressing liquidity concerns. Complexities further emerge in assessing counterparty risk, precise model calibration, and incorporating behavioral and regulatory aspects. Overcoming these challenges requires continual refinement and the incorporation of advanced methodologies, such as machine learning, while emphasizing robust risk management practices (Vashishtha & Kumar, 2010).

Over the past decades, options have become crucial financial derivatives, offering protection against share price fluctuations. Option pricing involves assessing fair values, considering factors like underlying asset movements, time to maturity, and current market conditions. The options market, particularly trading stock or index options, has garnered significant interest due to substantial profits in recent years (Park et al., 2014). Options are complicated financial contracts that confer the right, but not the duty, to purchase or sell an underlying asset at a defined price by a given date. In the market, there are numerous types of options, including European call (put) options and American call (put) options, exotic options, Bermudan options, Barrier options, Asian options, and lookback options (Liang et al., 2009).

Despite rapid growth, the Indian options market faces unique challenges in accurate pricing. Predicting the price of financial options in the Indian market is a challenging task due to the market's dynamic nature, intricate interest rate structure, historical data unavailability, market sentiment, limited option maturities and the complexity of underlying assets. Grasping these intricacies is key to unlocking the potential of the Indian options market for both traders and investors. The central topic of options research, the option pricing method, has been studied for decades, yet a precise match to the price process remains difficult to achieve. Many studies have been conducted by academics to refine option pricing's precision. Today, option pricing models are primarily classified as parametric or non-parametric.

Black and Scholes (1973) made history when they presented the famous Black-Scholes (B-S) pricing formula for European options and developed the classical parametric pricing model. Approaching its five-decade milestone since publication, the model remains a prevalent choice among practitioners globally. Despite its sustained popularity, certain assumptions inherent in the model, such as constant volatility and continuous underlying asset processes, diverge from the characteristics observed in the traded option data of the market. This misalignment underscores the necessity for more nuanced modeling approaches that better reflect the dynamics inherent in the traded financial options. Numerous financial engineering models have attempted to loosen the limitations of the Black-Scholes model and optimize empirical results, resulting in popular approaches such as statistical series expansion (Corrado & Su, 1996), local volatility models (Schroder, 1989), stochastic volatility models (Heston, 1993; Hull & White, 1987), models with jumps (Merton, 1976), stochastic interest model (Merton, 1974), and models with transactions costs (Davis et al., 1993; Barles & Soner, 1998). These models have shown to be more effective in terms of valuation, but they are computationally expensive due to the need to calibrate a large number of implicit factors. Furthermore, they continue to be constrained by certain economic and statistical hypotheses, such as no-arbitrage and market completeness principles. The assumption of constant volatility, a cornerstone of these models, does not align with empirical evidence, especially during turbulent economic periods. This oversimplification impedes their ability to accurately capture market fluctuations. Furthermore, these models rely on continuous underlying asset processes and constant interest rates, disregarding the discontinuities and changing rate environments observed in reality.

So, a new field of research has been created that uses data-driven models to rapidly and precisely price financial derivatives. As the demand for data intensifies, the constraints of inflexible and assumption-laden parametric models become more pronounced. These models struggle to grasp intricate relationships, manage extensive datasets, and adjust to evolving trends, limiting their ability to provide comprehensive insights. In response, data-driven models step up to the task, employing robust algorithms to uncover concealed patterns, navigate diverse data structures, and undergo continuous refinement. Machine Learning methods are extremely helpful if options are viewed as a functional relation between the contracted terms (inputs) and the premium (outputs). Furthermore, machine learning methods offer a resolution for pricing options without relying on analytical formulas, thereby circumventing the expenses associated with Monte Carlo simulation. There is a wealth of literature in this field such as option pricing using Support Vector Machine (Nikou et al., 2019), Decision Tree (Ivacu, 2021), Artificial Neural Network (Gradojevic et al., 2009).

Nevertheless, machine learning methods share a common drawback- the need for manual extraction of data features. This process is intricate and laborious, and certain methods exhibit insufficient nonlinear fitting ability, leading to the omission of implicit information in the extracted features. Financial option data, known for their high dimensional and nonlinear characteristics, further exacerbate the issue, resulting in suboptimal classification effects that ultimately impact the overall performance of the pricing model.

Hutchinson et al. (1994) advocated for a data-driven approach, emphasizing its adaptability to structural changes in the data-capabilities that parametric models lack. By avoiding reliance on restrictive parametric assumptions, these models exhibit robustness to specification errors that often constrain classical models. Moreover, the non-parametric approach proves highly flexible, suitable for valuing a diverse range of derivatives. The authors specifically demonstrated the efficacy of a neural network (NN) in approximating the option pricing function, showcasing its superior accuracy and computational efficiency compared to the Black-Scholes model.

Subsequent research from a variety of sources, including (Yang & Lee, 2011; Liu et al., 2019), and (Ruf & Wang, 2019), confirms that an artificial neural network is highly effective at options valuation. Goswami et al. (2021) applied artificial neural networks (ANN) to predict option prices on the Indian market indices NIFTY50 and BANKNIFTY. Vaswani et al. (2022) conducted their research utilizing a comprehensive dataset comprising more than 2 lakh observations of NSE Index options, specifically focusing on NIFTY 50 and BANKNIFTY. The dataset was sourced from the national stock exchange (NSE) of India, spanning the period from August 2020 to August 2022. The study strategically employed artificial neural networks (ANN) to achieve heightened accuracy in predicting option prices.

Notably, deep learning has garnered considerable interest in the field of financial data analysis because of its effective nonlinear fitting and feature capture capabilities. Deep learning can filter and learn data in depth, removing irrelevant factors and strengthening relevant ones while learning variability. Convolutional neural network performs particularly well in this respect by autonomously extracting features. CNN's low hyperparameter and computational requirements make it a popular choice for use in graphic and image processing (Li et al., 2020; Ikram & Liu, 2020). Hu (2018) made stock price predictions using CNN. CNN was found to be capable of time series prediction, and deep learning was found to be the optimal approach for dealing with time series problems. Wei et al. (2020) presented a novel system employing CNN to predict both implied volatility and option prices. The empirical experiments demonstrated the superior performance of the proposed framework, surpassing traditional methods. Liu and Wu (2023) introduced a novel methodology that involves the integration of 2D-CNN with market technical indicators (TIs). Through a meticulous selection process encompassing relevant market technical indicators and precise Black-Scholes (BS) model parameters, the CNN's predictive accuracy is significantly improved.

Hochreiter and Schmidhuber (1997) in 1997 first suggested the long short-term memory (LSTM) neural network. Recent years have seen a rise in the use of long short-term memory (LSTM) for predicting data that depends on the past, and LSTM has reasonably mature studies in fields like option pricing, volatility prediction in the option market (Xie & You, 2018), stock price prediction (Feng & Li, 2019), and so on. Sezer et al. (2020) suggested that LSTM is one of the most widely used DL models for predicting financial time series. However, LSTM occasionally failed to recognize when a pattern suddenly changed, which negatively impacted the reliability of the prediction model (Selvin et al., 2017). Liu and Zhang (2023) advanced an option pricing model based on Long Short-Term Memory (LSTM) architecture, integrating realized skewness to effectively account for the asymmetry observed in emerging market asset returns. The performance evaluation revealed the model's superiority, surpassing classical methods and other machine-learning approaches across various metrics. Zhang and Huang (2021) presented a modified LSTM model, derived from the approach proposed by Buehler et al. The research conducted an extensive comparative examination, showcasing the model's superior performance in the presence of transaction costs. The evaluation encompassed simulated market data generated through Geometric Brownian Motion (GBM) and the Heston model, as well as real market data. Liu and Wei (2022) introduced a refined LSTM network, incorporating technical indicators to enhance prediction accuracy. Zouaoui and Naas (2023) employed an ensemble of deep learning approaches, specifically LSTM and GRU, to predict option prices during the COVID-19 period, comparing the results against the Black-Scholes Model (BSM).

In contrast to LSTM, which can only use forward information, Bidirectional LSTM (BiLSTM) can use both forward and backward information which enhances the model performance. Siami-Namini et al. (2019) compared ARIMA and LSTM models to BiLSTM, and their findings indicate that BiLSTM results in more accurate predictions. As a result, it works wonderfully for making predictions about time sequences. Numerous studies have used BiLSTM for predicting time series (Lu et al., 2021). ZENG An (2019) utilized BiLSTM to make predictions about the S &P 500. The findings indicate that the accuracy of the predictions far exceeded that of the currently available prediction models.

For a variety of applications, including healthcare (Rai & Chatterjee, 2022), financial market prediction (Ishwarappa & Anuradha, 2021), portfolio optimization (Singh et al., 2023) etc., some researchers have tried to merge CNN and LSTM to build CNN-LSTM hybrid models. Numerous empirical findings demonstrate that the influence of trade discontinuities makes it challenging to reliably estimate option prices using a single neural network prediction model. To solve the option pricing prediction problem, Zhao et al. (2022) merged CNN and LSTM model with a standard stochastic volatility Heston model and a stochastic interests CIR model, and the results demonstrated the high accuracy of the dual hybrid model.

This research recommends combining the CNN and BiLSTM to appropriately price options, enhance prediction accuracy, and boost model performance by looking at the widespread popularity of DL-based hybrid models in several sectors for financial derivative price prediction. According to the research studies discussed above, most papers skim the surface of how we can use technical indicators in sequential deep networks. The suggested research utilizes a combination of selective features, derived from fundamental and technical data to construct the model.

This research's main takeaways are as follows: (a) To predict the price of NIFTY 50 index options, a novel hybrid DL-based model (CNN-BiLSTM) that incorporates fundamental and technical data is proposed.; (b) Experiments were carried out to compare our suggested approach to different DL models. The results demonstrate that our proposed model provides more precise predictions; (c) the model's validity and robustness are confirmed by doing a statistical and robustness testing experiment.

The remaining sections of the paper are arranged as follows. Section 2 describes the modeling approach discussed in this study. Dataset collection, feature selection procedure, input preparation, and evaluation criteria are discussed in Sect. 3. Experimental design, results, and prediction performances are explained in Sect. 4. It also describes robustness checks and statistical analysis of models. Finally, Sects. 5 and 6 present the discussion and conclusion with future work, followed by references.

2 Proposed Model

To improve the precision of option price predictions, we present a hybrid model CNN-BiLSTM comprising 1DCNN and BiLSTM. We synergize a Convolutional Neural Network (CNN) for feature extraction, enabling pattern detection and effective learning of internal data representations, with a bidirectional long short-term memory (BiLSTM) to predict prices based on the extracted features. CNN is utilized to extract local features of the data in a layer-by-layer manner. Extraction of highly expressive advanced features from the data can help overcome the subjectivity and constraints of manual feature extraction. In the realm of CNN architectures, it is conventional to process data using a defined number of convolutional layers, followed by a subsequent max-pooling layer. Essentially, the convolutional layers play a role in data filtration by eliminating noise and extracting features from the input, while the subsequent max-pooling layer serves as a mechanism to condense information, retaining only a succinct summary of the filtered features.

Recurrent neural networks (RNNs) constitute a neural network category achieved by integrating feedback connections into the feedforward architecture. Primarily employed for sequential data processing, including text and time series data, RNNs encompass a specific type known as Long Short-Term Memory (LSTM). LSTMs innovatively introduced gate mechanisms to enhance information flow control and facilitate the learning of long-term dependencies. A Bidirectional long short-term memory (BiLSTM) cell can be conceptualized as composed of two LSTM units: one processing the input sequence in a forward direction and the other in reverse, followed by concatenating both interpretations. This augmentation enhances the information accessible to the model by offering a more comprehensive context. While traditionally applied to natural language processing, its recent effectiveness has also been demonstrated in time series forecasting tasks (Jin et al., 2020). As BiLSTM can remember past contexts for a long period, it can accomplish time- and distance-dependent feature extraction. Moreover, BiLSTM may extract the long-term time series link between the index's influencing elements and its close price (Kavianpour et al., 2023). Hence, the CNN output data are fed into the BiLSTM to model the bidirectional temporal structure via the calculation of formulas(2–10). The pseudo-code of the proposed model is shown in Algorithm 1.

Algorithm 1 Pseudo-code of CNN-BiLSTM

```
Require: Historical & Technical Data
Step 1: Data Preprocessing
Cleaning and Scaling ⇐ Dataset
(X_train, Y_train, X_test, Y_test, X_val, Y_val) ⇐ Dataset Split
return Dataset
Step 2: Model training
# Initialize the parameters
epochs⇐100, loss⇐mse, optimizer⇐Nadam, units⇐256, padding⇐same,
batch size⇐128, learning rate⇐0.0001, dense_units⇐1
# Initialize the model
Create a sequential model
# CNN layers
Add TimeDistributed(Convo1D(filters,kernel_size,activation,padding),input_shape)
Add TimeDistributed(Convo1D(filters,kernel_size,activation,padding))
Add TimeDistributed(MaxPooling1D(pool_size))
Add TimeDistributed(Flatten())
Add Dropout(rate)
# BiLSTM layers
Add Bidirectional(LSTM(units,kernel_regularizer))
Dense(dense_units)
# Print summary and trainable parameters
model.summary()
Step 3: Model Fitting
model.compile(optimizer(learning_rate),loss)
for each i in epochs
    model.fit(X_train, Y_train,epochs)
end for
Results ⇐model.evaluate(X_val, Y_val,batch_size)
return Results
```

By maximizing the use of data information, our suggested model can automatically train and extract local features and long memory patterns in the time series, significantly lowering the model's complexity (Chen et al., 2021). Although the intricacy of the CNN-BiLSTM architecture requires careful adjustment of hyperparameters and significant computational resources, its efficacy in tackling crucial challenges and attaining superior accuracy establishes it as a powerful instrument in option price prediction research. Through the collaborative utilization of CNNs and BiLSTMs, the model not only enhances prediction accuracy but also furnishes valuable insights into the fundamental dynamics of option pricing. This contributes to the progression of this pivotal field, opening avenues for future advancements. Figure 1 is a simplified visual representation of the suggested study framework. The architecture of a proposed network is depicted in Fig. 2. These CNN and BiLSTM layers, which constitute the core of our proposed model, are briefly described in the following subsections.

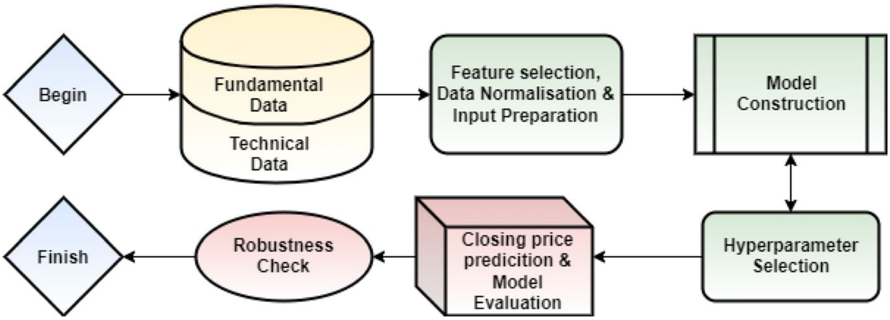


Fig. 1 Diagrammatic illustration of the proposed research plan

2.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is based on Hubel (1959) studies on how animals naturally perceive light and color. CNN architectural framework consists of input, convolutional layer with non-linear activation function, pooling layer, fully connected layer, and output. Numerous convolution kernels are included in each convolution layer (Lu et al., 2020). The formula for its computation is stated in Eq.{1}

$$Y_t = \phi(x_t * k_t + b_t) \tag{1}$$

where Y_t is the output after convolution, ϕ is the activation function, x_t is the input vector, k_t and b_t represents weight and bias of convolutional kernel respectively. During the convolution operation of the convolution layer, data features are extracted; however, the extracted feature dimensions are rather high; to address this issue and lower the cost of network training, a pooling layer has been added after the convolution layer. To learn the nonlinear combination of features extracted by the convolution layer to produce the final output, FC layers are used in the last layers of the CNN architecture (Hoseinzade & Haratizadeh, 2019).

2.2 Bidirectional Long Short-Term Memory Neural Network

LSTM network model, a version of RNN, was proposed by Hochreiter and Schmidhuber (1997). As such, it has found widespread application in both classification and regression problems not only for stock market prediction but additionally in a variety of areas such as text analysis, emotional analysis, speech recognition, rainfall-runoff modeling, anomaly detection, mobile traffic prediction, etc (Lindemann et al., 2021; Rahman et al., 2020; Li, 2022).

While traditional RNNs surpass classic networks in conserving information, it struggles to grasp long-term dependencies due to the vanishing gradient problem. LSTM employs memory cells to overcome this issue. BiLSTM is an enhanced form of LSTM that allows for accessibility to attribute values in both the forward and backward directions and was introduced by Graves and Schmidhuber (2005).

A cell state, an input layer, a hidden layer, and an output layer make the overall framework (Fig. 3). The essential element of the LSTM structure is the cell state, which passes through the chain with only linear interaction, preserving the information flow. The cell state information can be deleted or altered by the LSTM gate mechanism. A combination of a sigmoid layer, a hyperbolic tangent layer, and a pointwise multiplication operation is used to selectively transmit data (Md et al., 2023; Ke & Yang, 2016). LSTM calculating principle is as described below.

$$\alpha_t = \sigma(\Omega_\alpha \lambda_t + \Omega_{\psi\alpha}\psi_{t-1} + b_\alpha) \tag{2}$$

$$\beta_t = \sigma(\Omega_\beta \lambda_t + \Omega_{\psi\beta}\psi_{t-1} + b_\beta) \tag{3}$$

$$\gamma_t = \sigma(\Omega_\gamma \lambda_t + \Omega_{\psi\gamma}\psi_{t-1} + b_\gamma) \tag{4}$$

$$\hat{\tau}_t = tanh(\Omega_\tau \lambda_t + \Omega_{\psi\tau}\psi_{t-1} + b_\tau) \tag{5}$$

$$\tau_t = \beta_t \tau_{t-1} + \alpha_t \hat{\tau}_t \tag{6}$$

$$\psi_t = \gamma_t * tanh(\tau_t) \tag{7}$$

where, Ω and ψ are weight matrices α_t represents input gate, β_t is forget gate, $\hat{\tau}_t$ is current cell state, τ_t is candidate value, γ_t is output and, ψ_t is the hidden state of the lstm cell for timestep t (Liang & Cai, 2022). By introducing double-layer LSTM and configuring the forward and reverse layers, BiLSTM can be procured (Fig 4. BiLSTM's activation function is present at the output of the hidden layer for both forward and backward direction (Lee et al., 2022).

The mathematical equations of BiLSTM are expressed as follows:

$$\overrightarrow{\psi}_t = \phi(\Omega_{\lambda\overrightarrow{\psi}}\lambda_t + \Omega_{\overrightarrow{\psi}\overrightarrow{\psi}}\overrightarrow{\psi}_{t-1} + b_{\overrightarrow{\psi}}) \tag{8}$$

$$\overleftarrow{\psi}_t = \phi(\Omega_{\lambda\overleftarrow{\psi}}\lambda_t + \Omega_{\overleftarrow{\psi}\overleftarrow{\psi}}\overleftarrow{\psi}_{t-1} + b_{\overleftarrow{\psi}}) \tag{9}$$

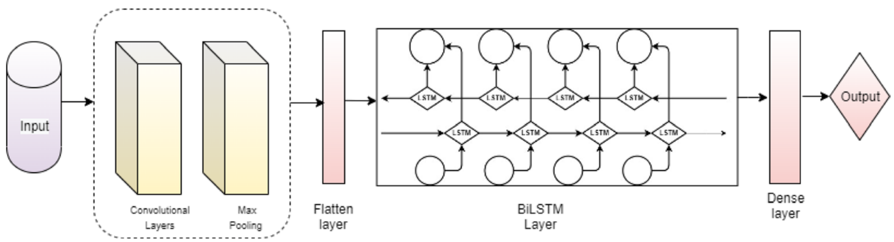


Fig. 2 CNN-BiLSTM

$$O_t = \Omega_{\lambda\overrightarrow{\psi}}\overrightarrow{\psi} + \Omega_{\chi\overrightarrow{\psi}}\overrightarrow{\psi} + b_\chi \tag{10}$$

where ϕ represents activation function of the model; Ω is the weight of matrix; $\Omega_{\lambda\psi}$ indicates the weight of input(λ) to hidden layer(ψ); O_t shows the hidden layer input; b_λ represents the bias of respective gates(λ).The output is achieved via updating in the forward direction $\overrightarrow{\psi}$ and structures that are created in backward $\overleftarrow{\psi}$.

3 Dataset Preparation

The Nifty index option stands out as the most heavily traded instrument on the National Stock Exchange (NSE) of India, constituting over 75% of the exchange's total trading volume. The prominence of Nifty index options contributes significantly to the NSE's standing, enabling it to secure a position among the top five exchanges globally. Trading of Nifty 50 index options commenced in India in June 2001, and in just under 12 years, it has ascended to become the foremost global index in terms of the volume of contracts traded. The research methodology employed in this study relies on the predictive analysis of the NIFTY 50 index options. In feature selection, the primary contributors of index value fluctuations are isolated. The data incorporated into the study range from January 2007 to January 2021. The selected period includes the effects of the 2008 global financial crisis on the Indian economy and the Great Recession around 2020, which coincides with the global COVID-19 pandemic. Using fundamental trading data and technical indicators of the underlying index, the close price is predicted. The features of the proposed approach are first briefly discussed.

3.1 Fundamental Data

The foundation of any successful stock trading strategy is a solid understanding of fundamentals, also known as historical data. The open price, close price, high price,

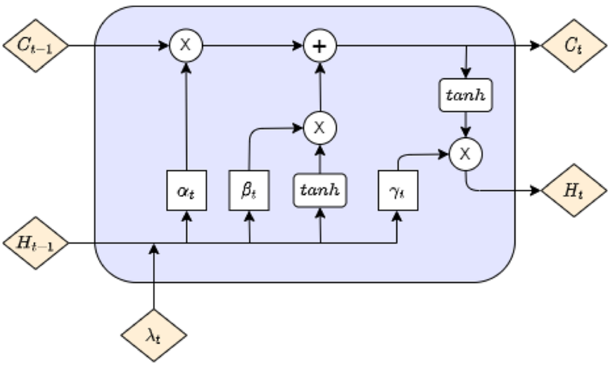


Fig. 3 Schematic diagram of LSTM

low price, and trading volume are all included. Indexes begin trading at the open price each day, while they end the day at the closing price. The high and low prices are, respectively, the daily high and low. The volume of a stock's trading activity is a measure of how actively investors are participating in the market on a given day. Interest is proportional to trading volume, therefore more trading indicates more interest. A ratio between the day's open and close prices is also used as a statistical feature.

3.2 Technical Indicators

Traders and analysts employ technical indicators-mathematical calculations based on factors like price, volume, and other technical indicators-to help them spot possible buy and sell signals. Because of their focus on short-term price changes, they are often used by active traders. This research delves into distinct technical indicator categories, each making unique contributions to the nuanced understanding of market behavior. Through a comprehensive exploration of overlay indicators, volatility metrics, oscillators, and volume indicators, we systematically unravel the intricate layers of market analysis (Table 1). This systematic investigation equips stakeholders with a robust toolkit, enhancing their capacity to navigate the intricacies inherent in the financial trading landscape.

- Overlay—Indicators that are plotted directly on the price chart and offer a visual depiction of average prices, trend directions, and levels of volatility. Examples include Simple Moving Average, Exponential Moving Average, Smoothed Moving Average, Triangular Moving Average, Bollinger Bands Width, etc.
- Volatility indicators—Concentrate on gauging the extent of price fluctuations assist traders in recognizing phases of heightened or diminished volatility. True range and average true range (ATR) are widely utilized for evaluating volatility.
- Oscillators—Indicators that fluctuate within designated values assist traders in pinpointing situations of overbought or oversold conditions, along with potential

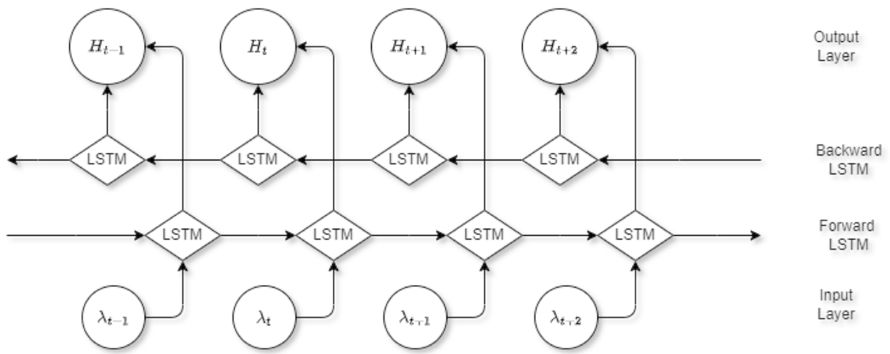


Fig. 4 BiLSTM

reversals in trends. Examples include Relative Strength Index, Rate of Change, Money Flow Index, Commodity Channel Index, Momentum, Chande Momentum Oscillator, Stochastic Oscillator, Awesome Oscillator, Mass Index, etc.
- Volume indicators—Centered on trading volumes aid traders in comprehending the robustness and durability of price movements. Examples such as On-Balance Volume and Volume Oscillator integrate volume data into their calculations.

where, C_p = Current price of the asset at period p; abs = absolute function; C_{p-n} = Current price of asset n periods ago; $\alpha = \frac{2}{1+n}$; Money flow ratio = $\frac{\text{14 Period positive Money flow}}{\text{14 Period negative Money flow}}$; TP (typical price) = (High+Low+Close)/3; MA (moving average) = $\sum_{i=1}^{n} \frac{TP}{n}$; MD (mean deviation)= $\sum_{i=1}^{n} \frac{abs(TP-MA)}{n}$;Up/Down= sum of positive/negative changes in price; $\beta = \frac{\text{(Close-low)-(High-Close)}}{\text{High-low}}$; L_p/H_p=Lowest/Highest price over period p; MA_S/MA_L= Short/Long term moving average of volume; EMA_9=Sum of 9 periods EMA of Range; EMA_9^{new}=Sum of 9-period EMA of 9-period EMA of Range.

3.3 Feature Selection Strategy

Deep learning is a method of processing massive amounts of data to draw relevant results. The effectiveness of deep learning algorithms, however, is typically hampered by the existence of irrelevant or duplicate input. To anticipate the final price, each of the input factors described above contributes to some extent (Tyralis & Papacharalampous, 2017). This section discusses the strategies for picking the essential features and deleting the extraneous ones from the original feature sets.

3.3.1 Pearson Correlation Coefficient

As a statistical measure of the degree to which two continuous variables are related to one another, Pearson's correlation coefficient(PCC) is a widely used test statistic (Huang et al., 2021). It illustrates not only the strength but also the direction of the link or correlation. Consider two-time series $A = \{a_1, a_2, ..., a_n\}$ and $B = \{b_1, b_2 ...., b_n\}$, the PCC between series A and B can be written as:

$$\phi_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B} \tag{11}$$

where Covariance (Cov) formula is $\frac{\sum_{j=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{\sqrt{\sum_{j=1}^{n}(a_i-\bar{A})^2}\sqrt{\sum_{j=1}^{n}(b_i-\bar{B})^2}}$, $\bar{A}$ is the mean of series A and $\bar{B}$ is the mean of series B.

Its value varies from -1 (absolute weak relation) to 1 (absolute strong correlation), with 0 denoting that there is no link.

The heatmap of the Pearson correlation coefficient is shown in Fig. 5. Closing prices may or may not have a high correlation with the other variables. Predictive

Table 1 List of technical indicators used in the study along with their formulas

Name	Abbreviation	Formula
Simple moving average	SMA	$\sum_{i=1}^{p} C_i/p$
True range	TR	$\max(High - Low, abs(High - PC), abs(Low - PC))$
Average true range	ATR	$\frac{Prev.ATR(n-1)+TR}{n}$
Exponential moving average	EMA	$(C_p - EMA_{i-1}) * \alpha + EMA_{i-1}$
Relative strength index	RSI	$100 - \frac{100}{1+\frac{Avg.Gain}{Avg.Loss}}$
Rate of change	RoC	$\frac{C-C_{p-n}}{C_{p-n}} * 100$
Money flow index	MFI	$100 - \frac{100}{1+MoneyFlowRatio}$
Commodity channel index	CCI	$\frac{TP-MA}{0.015*MD}$
Smoothed moving average	SMMA	$\frac{SMMA_{i-1}*(n-1)+C_p}{n}$
Momentum	MoM	$C_p - C_{p-n}$
Chande momentum oscillator	CMO	$\frac{Up-Down}{Up+Down} * 100$
Triangular moving average	TRIMA	$\frac{1}{n}\sum_{i=0}^{n} iC_i$
Efficiency ratio	ER	$\frac{C_{p-n}-C_p}{\sum C_p-C_{p-1}}$
Volume-weighted average price	VWAP	$\frac{\sum Price-Vol}{\sum Vol}$
Bollinger bands width	BBwidth	$\frac{UpperBB-LowerBB}{MA}$
Average directional index	ADX	$100 * \frac{MA_{+}-MA_{-}}{MA_{Total}}$
Volume price trend	VPT	$VPT_{i-1} + Vol * \frac{C_p-C_{p-1}}{C_{p-1}}$
Force volume energy	FVE	$Vol * (C_n - C_{n-1})$
On-balance volume	OBV	$OBV_{i-1} + Vol * (C_n - C_{n-1})$
Accumulation distribution line	ADL	$ADL_{i-1} + Vol * \beta$
Stochastic oscillator	STOCH	$\frac{C_p-L_p}{H_p-L_p} * 100$
Volume oscillator	VO	$\frac{MA_S-MA_L}{MA_L} * 100$
Awesome oscillator	AO	$SMA_5 - SMA_{34}$
Mass index	MI	$\frac{EMA_9}{EMA_9^{new}}$

strength may be poor if there is little to no association between the considered variables. Hence, a correlation coefficient of 0.70 is used as a cutoff for identifying and discarding features that provide irrelevant information.

During the modeling process, the filter method for selecting features is complemented by an embedding method that combines the best characteristics of both the filter and wrapper approaches, and for this L2(Ridge) regularization approach is used. The addition of the L2 regularizer has the effect of shrinking the coefficients that are less relevant to the target variable towards zero, encouraging them to take smaller values which helps to reduce the complexity of the model and prevent overfitting. Following the completion of the feature selection procedure, a total of 12 variables are utilized as inputs.

3.4 Input Preparation

Due to a significant number of immediate market fluctuations and trade noises, financial data contain a complicated structure of irregularities and roughness. Denoising time series data with the help of discrete wavelet transform (DWT) is a highly effective method (Alrumaih & Al-Fawzan, 2002). While analyzing time series data, Haar wavelets are widely used. Haar wavelet possesses a trait of compacted support, which provides the function with a remarkable acute drop-off performance. Further, its support length is 1, which expediently shortens the computing time and minimizes the data processing and the training duration. Additionally, it is also symmetric, so the true price can be recovered substantially after noise reduction (Liang et al., 2019). Using the scikit-image module from the Python library, we have de-noised the close price by employing the soft mode of the Haar wavelets (Meinl & Sun, 2015). We have applied the min-max normalization method for scaling features to boost our model's efficiency and reliability. The mathematical representation of the min-max normalization method is

$$\beta = \frac{\alpha - \alpha_{\text{min}}}{\alpha_{\text{max}} - \alpha_{\text{min}}} \tag{12}$$

where β, α are scaled and real input respectively and α_{max}, α_{min} are maximum and minimum values of input respectively.

4 Experiment and Findings

The objective is to accurately predict the closing price of the NIFTY 50 index, which has complicated, noisy, and volatile behavior. The original price data of the NIFTY50 index option is directly taken from the NSE website. Figure 6 illustrates the general trend of the predictor variables. The blue line shows the original closing price time series from September 2007 to January 2021. To further illustrate the long-term and short-term patterns of the closing price, the green and red curves reflect the 200-day and 50-day moving averages, respectively.

4.1 Environmental Setup and Dataset Processing

All the tests are executed in a Python environment(version 3.10.11), utilizing the TensorFlow and Keras APIs. Table 2 details the machine setup used for each experiment.

As an integral aspect of dataset preprocessing, the initial steps involve the replacement of missing values with zero in the dataframe to address data gaps, accompanied by the exclusion of trades with zero volume. Subsequently, the close

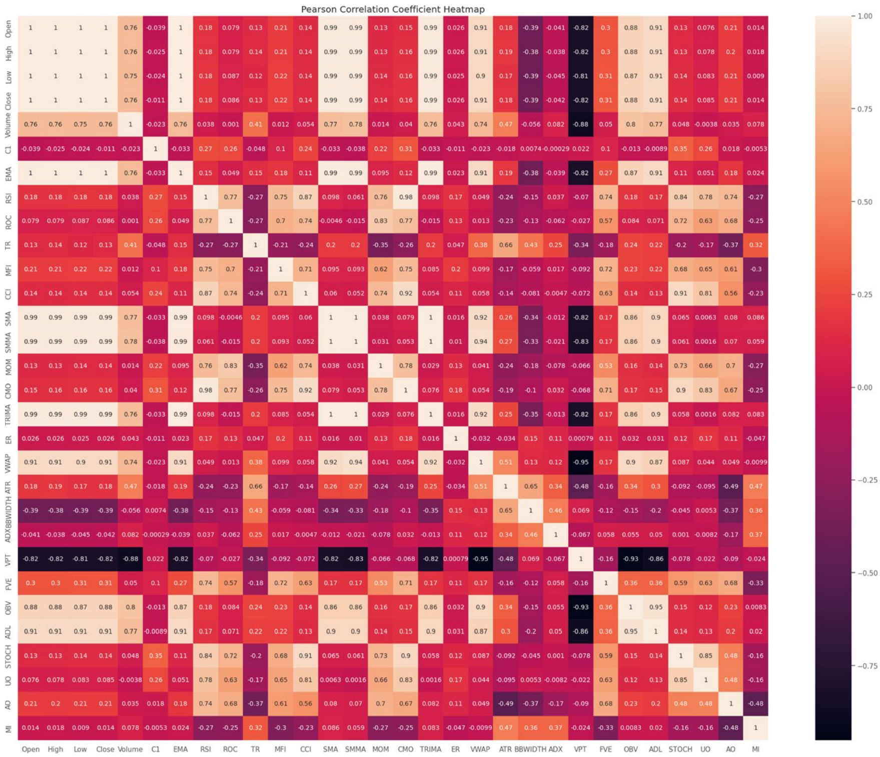


Fig. 5 Heatmap

price undergoes denoising through the application of the soft node of the Haar wavelet. Following this, relevant features are selectively chosen (see 3.3), and their values are normalized to a range of (0,1) using min-max scaler (see 3.4). The data is then transformed and segmented into two distinct parts. In this study, the initial 80% of the series is designated as the training set, while the remaining 20% serves as the testing set. Additionally, within the training set, a further 20% is set aside as the validation set during hyperparameter tuning. The transformed datasets are formatted as 2D arrays (number of observations, number of features). However, the model architecture necessitates 3D input data. To accommodate this requirement, the data is further transformed into three-dimensional arrays (number of observations, time step, and number of features) that incorporate the temporal dimension. Subsequently, the models are trained with optimal hyperparameters using the complete training dataset. The evaluation of model efficacy is conducted using the designated validation dataset.

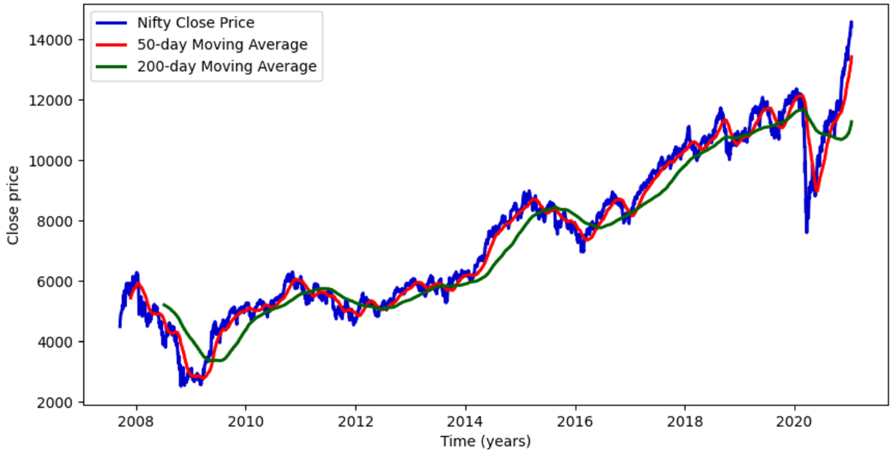


Fig. 6 Closing price and moving averages of NIFTY 50

Table 2 Computing setup

Machine configuration	Google colab with NIVIDA-SMI 525.85.12 GPU
Python version	3.10.11
Processor	Intel i5-4700H
Edition	Windows 8.1 with 64-bit OS

4.2 Hyperparameter Selection

The ideal value of the model's hyperparameters was determined after repeated experimental work. The following is an explanation of the impact that several hyperparameters have on the model, along with their significance:

- Batch size—It specifies the total number of input samples to be processed before affecting any internal model parameters. The degree of optimization and the training speed will be affected by its size. We experimented with 32, 64, and 128-sample batches to train our models.
- Epochs—A single epoch is one cycle through the entire training dataset. The neural network's weights are updated more frequently as the epoch count rises, leading to a shift from underfitting to overfitting as the curve progresses.
- Learning rate—This hyperparameter is used to fine-tune the convergence of the model to an accurate prediction. In our model, we tried different learning rates like 0.1,0.001..,0.00001.
- Optimizer—This is the optimization function that is applied in order to achieve the best possible outcomes. In our experiments, we tried the Adam, Nadam, and Adagrad optimizers. When compared to Adam, Nadam (which combines the

Adam optimizer with the Nesterov approach) has a somewhat shorter training duration.
- BiLSTM units—In a nutshell, the more neurons a system has, the more precise it is. Overfitting, however, is very common. Dropout should be implemented at this time.
- Dropout—Regularisation in DL models is achieved through the use of a dropout layer, which aids in reducing dependent learning between individual nodes. This layer's value, which can range from 0 to 1, serves to stop the model from being overfitting. Regularisation in DL models is achieved through the use of a dropout layer, which aids in reducing dependent learning between individual nodes. This layer's value, which can range from 0 to 1, serves to stop the model from being overfitting.
- Convolution Kernel and Filters—Their importance cannot be overstated in the process of gaining insight and useful knowledge from input data. 3 * 3 convolution kernel is adopted in this study. We have conducted experiments using 32 and 64-filter sizes.

Table 3 shows a description of the hyperparameters selected.

4.3 Prediction Performance

To validate the effectiveness, superiority, and generalizability of the suggested model, we compared its prediction performance to that of competing models CNN, LSTM, BiLSTM, and CNN-LSTM. Figure 7 shows the prediction results of all models. The red line represents the actual value, while the blue line represents the predicted value. This study uses the following six criteria to compute prediction error and accuracy in order to assess the effectiveness of the predictions made, and these are mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE), determination coefficient (R^2), maximum error (max error), and median absolute error (MedAE). Enhanced prediction performance is seen as the model's R^2 rises and the remaining metrics decline. The mathematical representation of these measures is as follows:

- Mean absolute error(MAE)—It is a statistical measure of how far the actual values in a dataset deviate from the predicted ones.

$$MAE = \frac{1}{N}\Sigma_{i=1}^{N} \mid Y_i - Y_i' \mid$$

- Root mean squared error(RMSE)—To put it another way, RMSE is the standard deviation of the absolute error between the anticipated value and the real values.

$$RMSE = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(Y_i - Y_i')^2}$$

- Mean absolute percentage error(MAPE)—In statistics, MAPE is used frequently as a more solid criterion for evaluating the success of a prediction.

$$MAPE = \frac{1}{n}\Sigma_{i=1}^{n}\left|\frac{Y_i - Y_i'}{Y_i'}\right|$$

- Determination coefficient(R^2)—It is a statistical indicator of the amount of variation in the target variable that can be accounted for by the model's independent variables. Its values are between zero and one.

$$R2 = 1 - \frac{\Sigma_{i=1}^{n}(Y_i - Y_i')^2}{\Sigma_{i=1}^{n}(Y_i - \bar{Y}_i')^2}$$

- Maximum error (Max Error)—It is a metric for determining the largest possible discrepancy between anticipated and observed values in a given dataset.
- Median absolute error(MedAE)—It's a statistic that takes the median of all the discrepancies between predictions and actuals.

$$MedAE = Median(\Sigma_{i=1}^{n} \mid Y_i - Y_i' \mid)$$

where Y_i = Actual value, $\bar{Y}_i'$ = Mean value & Y_i' = Predicted value
Table 4 displays the average results of the different model's prediction errors based on the six evaluation indicators after multiple runs.

Since the mean squared error most accurately represents the deviation between the resultant value generated by the output layer and the real value of the data, it is used as the loss function. Figure 8 represents the loss function graph for the above-mentioned models.

Table 3 List of optimal hyperparameter values

Hyper-parameter	Value	Hyper-parameter	Value
Epochs	100	Batch size	128
BiLSTM units	256	Activation Function	tanh
Conv1D layers	2	Filters	64
Kernel size	3	Padding	same
Pool size	1	Kernel regularizer(L2)	0.0001
Timesteps	5	Optimizer	Nadam
Loss function	Mean squared error	Dropout	0.1
Learning rate	0.0001		

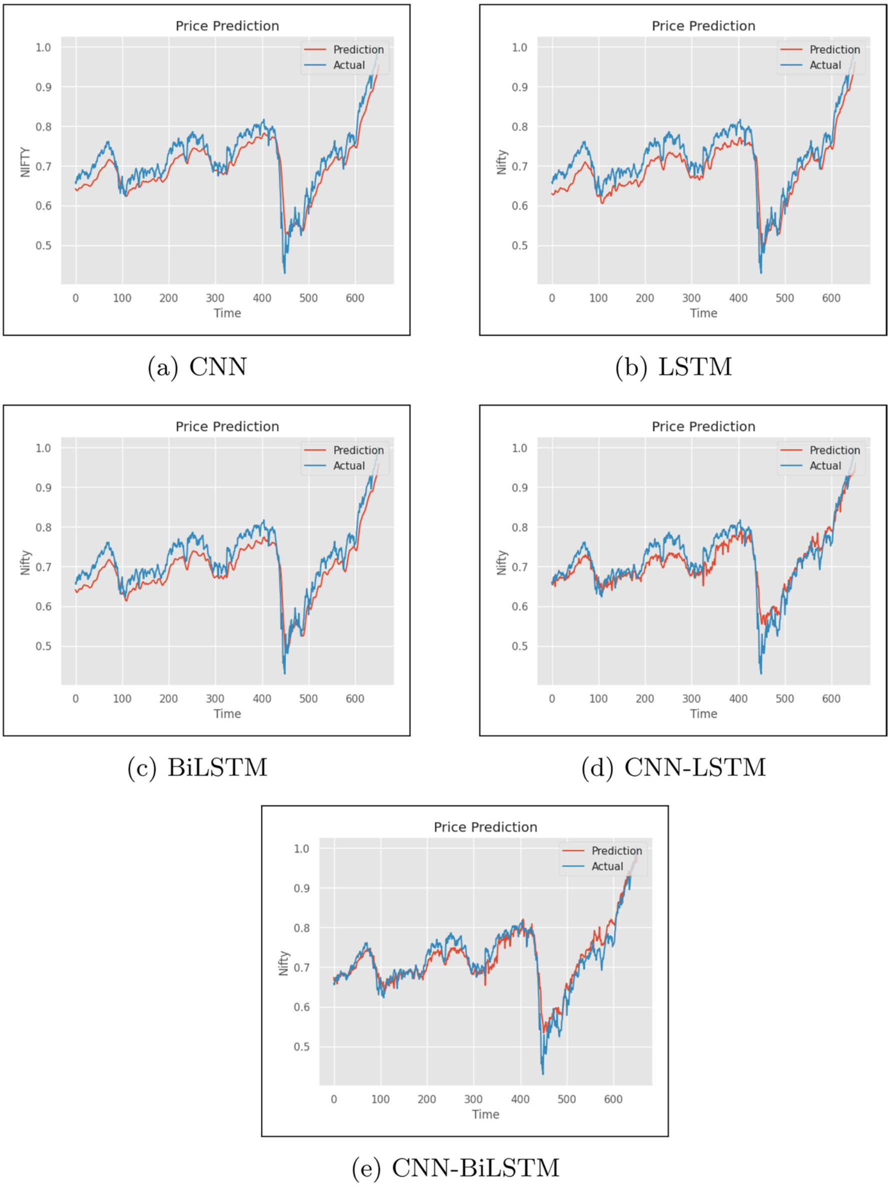


Fig. 7 Actual vs Predicted Graph

4.4 Statistical Evaluation

To demonstrate that the output of the model can be trusted, we carry out a statistical investigation to determine whether or not the results obtained by each of the five models are noticeably distinct from one another. As a first step, the

Shapiro-Wilk test is used to ensure that RMSEs follow a normal distribution. After ensuring that the RMSE distributions are normally distributed, a one-way analysis of variance(ANOVA test) to compare them is conducted; the resulting p-value was $0.00 < 0.05$ which indicates there is a significant difference between the performance of models. Following this primary statistical analysis, a post hoc analysis utilizing the Tukey-HSD test is performed to ascertain if there are statistically noteworthy distinctions between the groups. Figure 9 shows the p-values of each model with respect to the other. Comparing CNN-BiLSTM to CNN, LSTM, BiLSTM, CNN-LSTM, Lal and Timalsina (2022), and Madhu et al. (2021) its p-values are 0.0137, 0, 0.0295, 0.0192, 1, 0, and 0 correspondingly. As a result, there is sufficient data from both tests to assert that the CNN-BiLSTM model's performance is clearly above that of its competitors.

4.5 Robustness Check

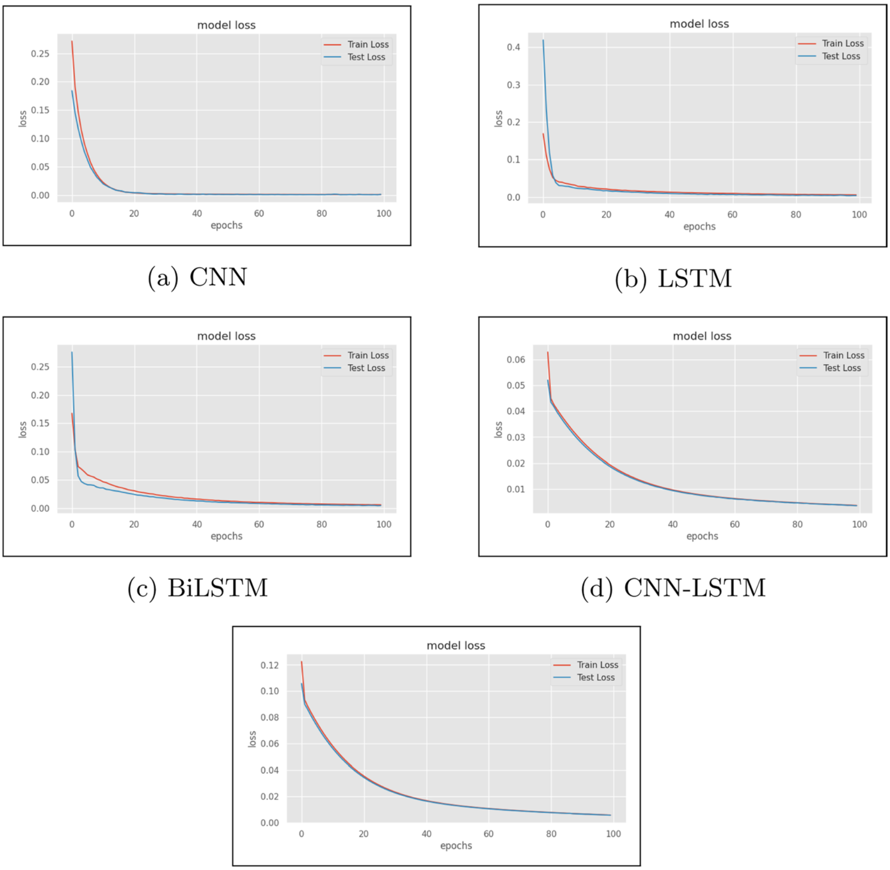
The experimental findings presented in Sect. 4.3 demonstrate that the CNN-BiLSTM model suggested in this paper significantly outperforms in terms of prediction accuracy. This research employs a pair of methodologies to examine the sturdiness of the proposed hybrid model with other benchmark models. First, we examine whether or not the model suggested in this study continues to produce top-tier outcomes under varying training and testing situations. We then investigate the prediction performance of the suggested model in the context of the current dispute between Russia and Ukraine.

4.5.1 Train-Test Ratio

We used a smaller sample size for the training set and a larger sample size for the test set, with 40% and 60% respectively. Figure 10a displays the comparative prediction results of the proposed model and its competing ones. The line chart of performance metrics is shown in Fig. 10b. The proposed model still has better accuracy than others.

Table 4 Average performance scores

Models Metrics	RMSE	MAE	MAPE	R2	MaxE	MedAE
CNN	0.0344	0.0291	0.0412	0.8393	0.1656	0.0286
LSTM	0.0377	0.0339	0.0474	0.8070	0.1486	0.0346
BiLSTM	0.0358	0.0323	0.0450	0.8260	0.1353	0.0330
CNN-LSTM	0.0307	0.0237	0.0345	0.8721	0.1423	0.0201
CNN-BiLSTM	0.0266	0.0191	0.0283	0.9036	0.1309	0.0140
Lal and Timalsina (2022)	0.0430	0.0373	0.0518	0.7494	0.1462	0.0350
Madhu et al. (2021)	0.03798	0.0278	0.0402	0.7855	0.2283	0.0213



(a) CNN (b) LSTM

(c) BiLSTM (d) CNN-LSTM

(e) CNN-BiLSTM

Fig. 8 Loss function graph

4.5.2 Validation of Recent Dataset

According to the results presented above, the model suggested in this work can effectively foresee the consequences of extreme events. With this in mind, we supplemented the model with data spanning January 2021 through March 2023 (including the effects of inflation and the current crisis in Ukraine) to ensure its efficacy. Figure 11a depicts the model's prediction comparison diagram for 0.4 training and 0.6 test sets. Again, the scores from the performance evaluations demonstrate the high quality of the prediction made by the model described in this study(Fig. 11b).

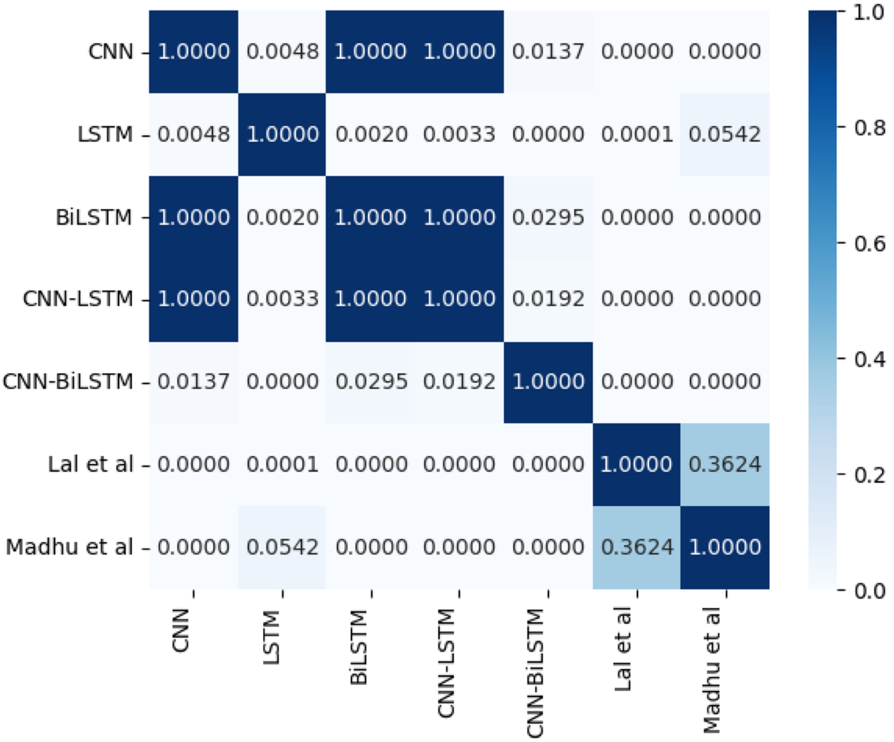
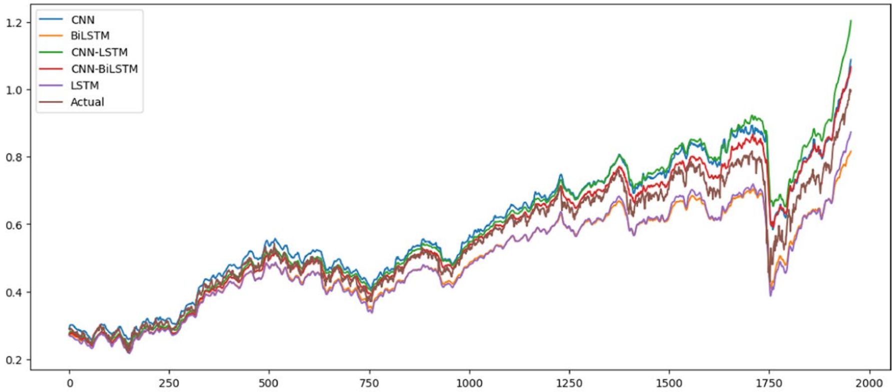


Fig. 9 P-Statistic Values

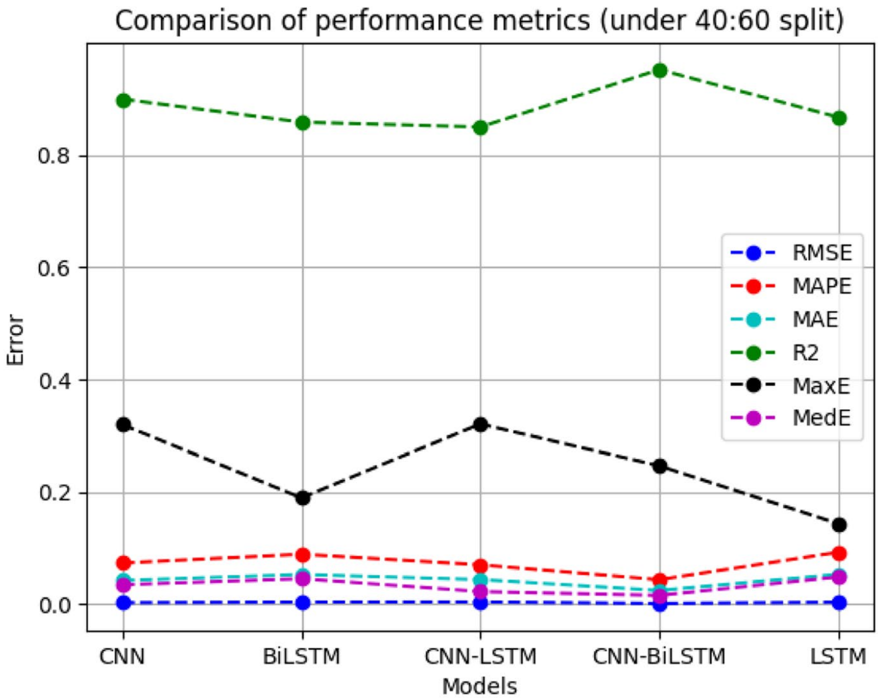
5 Discussion: Validity Threat

In this section, we address the internal and external validity concerns pertaining to potential threats in our research. An external validity concern arises from the prerequisite of a substantial volume of time series data for precise prediction outcomes in our model. This observation is substantiated by our findings, particularly evident when incorporating a recent dataset spanning an additional two years, resulting in diminished errors and heightened accuracy. It is crucial to acknowledge that this requirement presents an inherent challenge in the training of nearly all deep learning models.

Two internal validity threats have been identified in our study. The primary concern centers around the meticulous selection of the training dataset for the extraction of historical patterns, a critical element in our research framework. To address this, a comprehensive approach is employed, involving the exploration of various partition values, including 70–30, 80–20, 75–25, and 90–10. This method aims to systematically evaluate and ascertain the most effective dataset partition for enhancing the robustness of our model training process. Another internal validity concern involves the potential variability in the models' outputs based on the hyperparameter values employed during the training phase. The



(a)



(b)

Fig. 10 Prediction graph and performance scores of all models under 40:60 split absence of a universally acknowledged optimal selection approach for hyperparameters renders the process time-consuming and subjective. To address this challenge, we performed multiple trial runs, experimenting with diverse values for the model's hyperparameters to refine our proposed model. For instance, in the process of optimizer selection, our model underwent training using various optimizers such as Adam, Nadam, and Adagrad. Subsequent analysis revealed that Nadam yielded superior results compared to the other two alternatives. Consequently, Nadam was chosen as the optimizer for our experimental configuration. Following iterative adjustments, we ultimately determined the hyperparameter values for our model training, as detailed earlier in Sect. 4.2. These finalized values are comprehensively presented in Table 3. This is anticipated to provide the reader with a comprehensive understanding of the extensive real-world applicability of our prediction model.

6 Conclusion and Future Work

In this article, we suggest a hybrid deep learning model for predicting the prices of index options. Our suggested model, CNN-BiLSTM, combines the capabilities of both CNN and BiLSTM into a single computational framework. CNN is utilized for extracting features from raw data. The problem of long-term and prolonged dependencies can be addressed by the BiLSTM unit. The hyperparameters that were employed in the assessment of the model that provided the best prediction have also been covered in this article. This study constructs the predictive models by making use of thirty different predictors that are classified as either fundamental or technical data. We incorporated Pearson's correlation coefficient (filtration method) and L2 regularisation approach (embedding method) during model-building to extract a balanced set of input variables. Results from experiments demonstrate that, when compared to other DL Models, the CNN-BiLSTM achieves the highest prediction accuracy and the most remarkable overall performance. To ensure that RMSE values are normally distributed, the Shapiro test is utilized. As an added layer of verification, one-way ANOVA and post hoc analysis with the Tukey-HSD test show that the CNN-BiLSTM model is statistically superior to the rest. In conclusion, we demonstrate that the model developed in this research continues to exhibit superior performance and dependability by conducting a robustness check, varying the ratio of varying training and test sets, and conducting an additional evaluation of recent data sets.

Improved hybrid predictive models based on different neural network architectures will be developed shortly in an effort to improve accuracy. Combining features of both traditional and modern machine learning model architectures into a single predictive model is yet another viable option. More informative features from multivariate datasets like macroeconomic data, greeks, etc. can be used as well, leading to improved prediction accuracy. Hybrid optimization methods, which combine existing local optimizers with global optimizers like genetic algorithms, can be implemented to train the model parameters to increase the prediction accuracy even more.

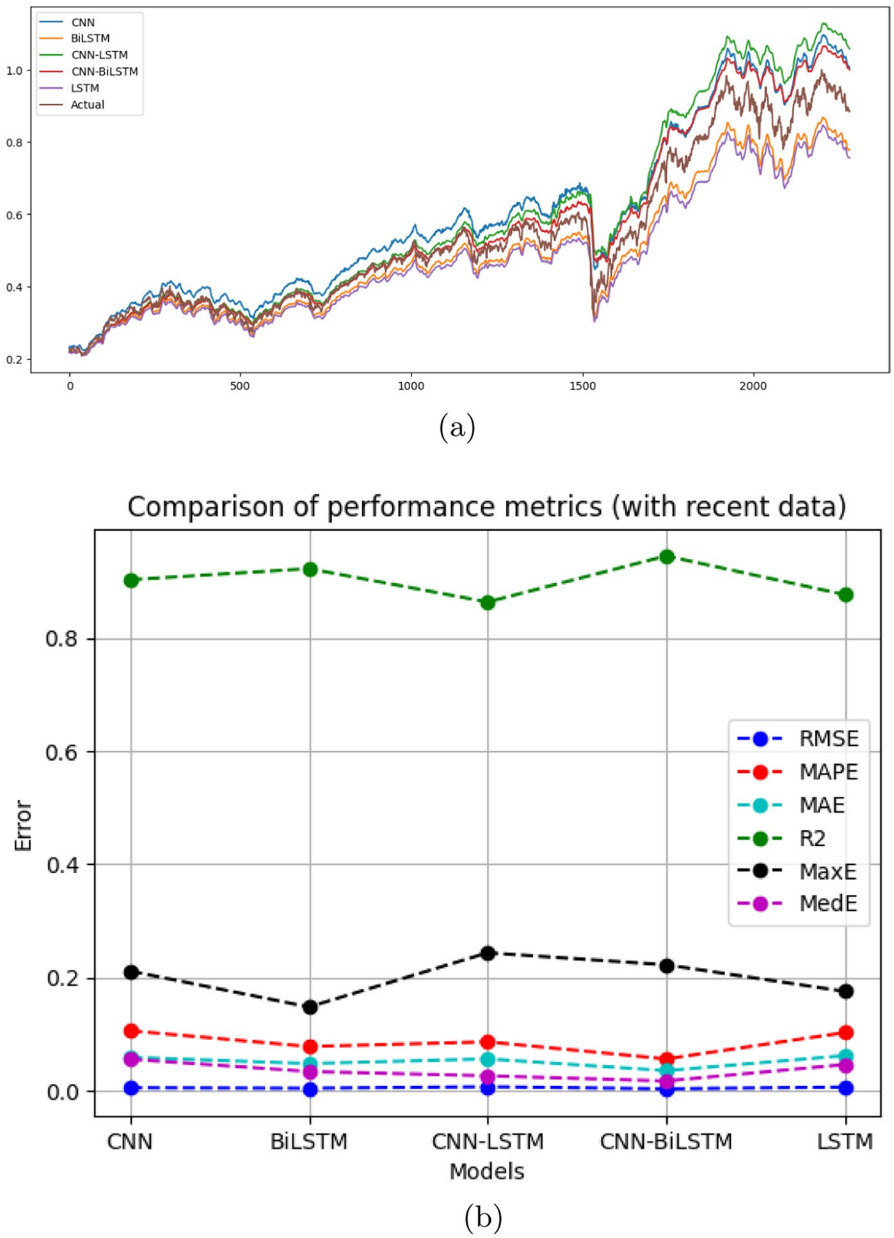


Fig. 11 Prediction graph and performance scores of all models under 40:60 split in addition to recent data

Funding This research did not receive any funding.

Declaration

Conflict of interest The authors state that they do not have any conflict of interest.

References

Alrumaih, R. M., & Al-Fawzan, M. A. (2002). Time series forecasting using wavelet denoising an application to Saudi stock index. *Journal of King Saud University-Engineering Sciences, 14*(2), 221–233.

Barles, G., & Soner, H. M. (1998). Option pricing with transaction costs and a nonlinear Black-Scholes equation. *Finance and Stochastics, 2*, 369–397.

Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy, 81*(3), 637–654.

Chen, Y., Fang, R., Liang, T., Sha, Z., Li, S., Yi, Y., Zhou, W., & Song, H. (2021). Stock price forecast based on CNN-BiLSTM-ECA model. *Scientific Programming, 2021*, 1–20.

Corrado, C. J., & Su, T. (1996). Skewness and kurtosis in s &p 500 index returns implied by option prices. *Journal of Financial Research, 19*(2), 175–192.

Davis, M. H., Panas, V. G., & Zariphopoulou, T. (1993). European option pricing with transaction costs. *SIAM Journal on Control and Optimization, 31*(2), 470–493.

Feng, Y., & Li, Y. (2019). A research on the CSI 300 index prediction model based on lSTM neural network. *Mathematics in Practice and Theory, 49*(7), 308–315.

Goswami, A., Rajani, S., & Tanksale, A. (2021). Data-driven option pricing using single and multi-asset supervised learning. *International Journal of Financial Engineering, 8*(02), 2141001.

Gradojevic, N., Gençay, R., & Kukolj, D. (2009). Option pricing with modular neural networks. *IEEE Transactions on Neural Networks, 20*(4), 626–637.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lSTM and other neural network architectures. *Neural Networks, 18*(5–6), 602–610.

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., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems with Applications, 129*, 273–285.

Hu, Y. (2018). Stock market timing model based on convolutional neural network-a case study of shanghai composite index. *Finance and Economy, 4*, 71–74.

Huang, H., Jia, R., Shi, X., Liang, J., & Dang, J. (2021). Feature selection and hyper parameters optimization for short-term wind power forecast. *Applied Intelligence*, *51*, 6752–6770.

Hubel, D. H. (1959). Single unit activity in striate cortex of unrestrained cats. *The Journal of Physiology, 147*(2), 226.

Hull, J., & White, A. (1987). The pricing of options on assets with stochastic volatilities. *The Journal of Finance, 42*(2), 281–300.

Hutchinson, J. M., Lo, A. W., & Poggio, T. (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks. *The Journal of Finance, 49*(3), 851–889.

Ikram, A., & Liu, Y. (2020). Skeleton based dynamic hand gesture recognition using LSTM and CNN. In *Proceedings of the 2020 2nd international conference on image processing and machine vision*, (pp. 63–68).

Ishwarappa, & Anuradha, J. (2021). Big data based stock trend prediction using deep CNN with reinforcement-lSTM model. *International Journal of System Assurance Engineering and Management*, 1–11. https://doi.org/10.1007/s13198-021-01074-2

Ivacu, C.-F. (2021). Option pricing using machine learning. *Expert Systems with Applications, 163*, 113799.

Jin, X., Yu, X., Wang, X., Bai, Y., Su, T., & Kong, J. (2020). Prediction for time series with CNN and LSTM. In *Proceedings of the 11th international conference on modelling, identification and control (ICMIC2019)* (pp. 631–641). Springer.

Kavianpour, P., Kavianpour, M., Jahani, E., & Ramezani, A. (2023). A CNN-BILSTM model with attention mechanism for earthquake prediction. *The Journal of Supercomputing*, *79*, 19194–19226.

Ke, A., & Yang, A. (2019). Option pricing with deep learning. In *Department of Computer Science, Standford University, in CS230: Deep learning*, (vol. 8, pp. 1–8).

Lal, J. K., & Timalsina, A. K. (2022). A CNN-BGRU method for stock price prediction.

Lee, M.-C., Chang, J.-W., Yeh, S.-C., Chia, T.-L., Liao, J.-S., & Chen, X.-M. (2022). Applying attention-based BILSTM and technical indicators in the design and performance analysis of stock trading strategies. *Neural Computing and Applications, 34*(16), 13267–13279.

Li, J. (2022). A comparative study of LSTM variants in prediction for tesla's stock price. *BCP Business and Management, 34*, 30–38.

Liang, L., & Cai, X. (2022). Time-sequencing European options and pricing with deep learning-analyzing based on interpretable ale method. *Expert Systems with Applications, 187*, 115951.

Liang, X., Ge, Z., Sun, L., He, M., & Chen, H. (2019). LSTM with wavelet transform based data preprocessing for stock price prediction. *Mathematical Problems in Engineering, 2019*(2019), 1340174.

Liang, X., Zhang, H., Xiao, J., & Chen, Y. (2009). Improving option price forecasts with neural networks and support vector regressions. *Neurocomputing, 72*(13–15), 3055–3065.

Lindemann, B., Maschler, B., Sahlab, N., & Weyrich, M. (2021). A survey on anomaly detection for technical systems using LSTM networks. *Computers in Industry, 131*, 103498.

Li, J., Sun, Y., & Zhang, B. (2020). Interactive behavior recognition based on sparse coding feature fusion. *Laser and Optoelectronics Progress, 57*(11), 181006.

Liu, D. & Wu, Y. (2023). Option pricing using deep convolutional neural networks enhanced by technical indicators. In *2023 IEEE 9th international conference on cloud computing and intelligent systems (CCIS)* (pp. 143–147). IEEE.

Liu, S., Oosterlee, C. W., & Bohte, S. M. (2019). Pricing options and computing implied volatilities using neural networks. *Risks, 7*(1), 16.

Liu, D., & Wei, A. (2022). Regulated LSTM artificial neural networks for option risks. *FinTech, 1*(2), 180–190.

Liu, Y., & Zhang, X. (2023). Option pricing using LSTM: A perspective of realized skewness. *Mathematics, 11*(2), 314.

Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-based model to forecast stock prices. *Complexity, 2020*, 1–10.

Lu, W., Li, J., Wang, J., & Qin, L. (2021). A CNN-BiLSTM-AM method for stock price prediction. *Neural Computing and Applications, 33*, 4741–4753.

Madhu, B., Rahman, M. A., Mukherjee, A., Islam, M. Z., Roy, R., & Ali, L. E. (2021). A comparative study of support vector machine and artificial neural network for option price prediction. *Journal of Computer and Communications, 9*(05), 78–91.

Md, A. Q., Kapoor, S., AV, C. J., Sivaraman, A. K., Tee, K. F., Sabireen, H., & Janakiraman, N. (2023). Novel optimization approach for stock price forecasting using multi-layered sequential ISTM. *Applied Soft Computing, 134*, 109830.

Meinl, T., & Sun, E. W. (2015). Methods of denoising financial data. In *Handbook of Financial Econometrics and Statistics* (pp. 519–538). Springer.

Merton, R. C. (1974). On the pricing of corporate debt: The risk structure of interest rates. *The Journal of Finance, 29*(2), 449–470.

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics, 3*(1–2), 125–144.

Nikou, M., Mansourfar, G., & Bagherzadeh, J. (2019). Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms. *Intelligent Systems in Accounting, Finance and Management, 26*(4), 164–174.

Park, H., Kim, N., & Lee, J. (2014). Parametric models and non-parametric machine learning models for predicting option prices: Empirical comparison study over Kospi 200 index options. *Expert Systems with Applications, 41*(11), 5227–5237.

Rahman, M. M., Usman, O. L., Muniyandi, R. C., Sahran, S., Mohamed, S., & Razak, R. A. (2020). A review of machine learning methods of feature selection and classification for autism spectrum disorder. *Brain Sciences, 10*(12), 949.

Rai, H. M., & Chatterjee, K. (2022). Hybrid CNN-LSTM deep learning model and ensemble technique for automatic detection of myocardial infarction using big ECG data. *Applied Intelligence, 52*(5), 5366–5384.

Ruf, J. & Wang, W. (2019). Neural networks for option pricing and hedging: A literature review. arXiv preprint arXiv:1911.05620.
Schroder, M. (1989). Computing the constant elasticity of variance option pricing formula. *The Journal of Finance, 44*(1), 211–219.
Selvin, S., Vinayakumar, R., Gopalakrishnan, E., Menon, V. K., & Soman, K. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017 international conference on advances in computing, communications and informatics (ICACCI)* (pp. 1643–1647). IEEE.
Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing, 90*, 106181.
Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019). The performance of lstm and bilstm in forecasting time series. In: *2019 IEEE international conference on big data (big data)*, (pp. 3285–3292). IEEE.
Singh, P., Jha, M., Sharaf, M., Elmeligy, M. A., & Gadekallu, T. R. (2023). Harnessing a hybrid CNN-LSTM model for portfolio performance: A case study on stock selection and optimization. IEEE Access.
Srivastava, A., & Shastri, M. (2018). A study of relevance of Black-Scholes model on option prices of Indian stock market. *International Journal of Governance and Financial Intermediation, 1*(1), 82–104.
Tyralis, H., & Papacharalampous, G. (2017). Variable selection in time series forecasting using random forests. *Algorithms, 10*(4), 114.
Vashishtha, A., & Kumar, S. (2010). Development of financial derivatives market in India-a case study. *International Research Journal of Finance and Economics, 37*(37), 15–29.
Vaswani, P., Mundakkad, P., & Jayaprakasam, K. (2022). Financial option pricing using random forest and artificial neural network: A novel approach. In *International joint conference on advances in computational intelligence* (pp. 419–433). Springer.
Wei, X., Xie, Z., Cheng, R., & Li, Q. (2020). A CNN based system for predicting the implied volatility and option prices.
Xie, H., & You, T. (2018). Research on European stock index options pricing based on deep learning algorithm: Evidence from 50ETF options markets. *Statistics and Information Forum, 33*, 99–106.
Yang, S.-H., & Lee, J. (2011). Predicting a distribution of implied volatilities for option pricing. *Expert Systems with Applications, 38*(3), 1702–1708.
Zeng, A., & Nie, W. J. (2019). Stock recommendation system based on deep bidirectional LSTM. *Computer Science, 46*(10), 84.
Zhang, J., & Huang, W. (2021). Option hedging using LSTM-RNN: An empirical analysis. *Quantitative Finance, 21*(10), 1753–1772. https://doi.org/10.1080/14697688.2021.1905171
Zhao, K., Zhang, J., & Liu, Q. (2022). Dual-hybrid modeling for option pricing of CSI 300ETF. *Information, 13*(1), 36.
Zouaoui, H., & Naas, M.-N. (2023). Option pricing using deep learning approach based on LSTM-GRU neural networks: Case of London stock exchange. *Data Science in Finance and Economics, 3*(3), 267–284.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.