



Leveraging BiLSTM-GAT for enhanced stock market prediction: a dual-graph approach to portfolio optimization

Xiaobin Lu¹ · Josiah Poon¹ · Matloob Khushi^{1,2} 

Accepted: 14 March 2025
© The Author(s) 2025

Abstract

Stock price prediction remains a critical challenge in financial research due to its potential to inform strategic decision-making. Existing approaches predominantly focus on two key tasks: (1) regression, which forecasts future stock prices, and (2) classification, which identifies trading signals such as buy, sell, or hold. However, the inherent limitations of financial data hinder effective model training, often leading to suboptimal performance. To mitigate this issue, prior studies have expanded datasets by aggregating historical data from multiple companies. This strategy, however, fails to account for the unique characteristics and interdependencies among individual stocks, thereby reducing predictive accuracy. To address these limitations, we propose a novel BiLSTM-GAT-AM model that integrates bidirectional long short-term memory (BiLSTM) networks with graph attention networks (GAT) and an attention mechanism (AM). Unlike conventional graph-based models that define edges based solely on technical or fundamental relationships, our approach employs a dual-graph structure: one graph captures technical similarities, while the other encodes fundamental industry relationships. These two representations are aligned through an attention mechanism, enabling the model to exploit both technical and fundamental insights for enhanced stock market predictions. We conduct extensive experiments, including ablation studies and comparative evaluations against baseline models. The results demonstrate that our model achieves superior predictive performance. Furthermore, leveraging the model's forecasts, we construct an optimized portfolio and conduct backtesting on the test dataset. Empirical results indicate that our portfolio consistently outperforms both baseline models and the S&P 500 index, highlighting the effectiveness of our approach in stock market prediction and portfolio optimization.

Keywords Attention mechanism · BiLSTM · Graph attention network · Portfolio management · Stock selection · Technical indicators

1 Introduction

Stock market prediction has long been a critical area of research due to its potential to offer significant financial benefits [1]. Accurate forecasting of stock prices is highly sought

after in the financial industry [2], as it can enable investors and traders to make informed decisions and optimize their portfolios for maximum returns [3]. However, the complex and volatile nature of financial markets makes stock market prediction an inherently challenging task [4]. While deep learning models have made strides in improving predictive accuracy [5], there remain significant limitations that hinder their effectiveness in real-world trading scenarios.

Traditional deep learning approaches to stock market prediction are often constrained by two primary factors. First, these models typically aggregate data from multiple companies into a single dataset without considering the heterogeneity inherent in the stock market [6, 7]. By treating companies as homogenous entities, these models overlook the distinct characteristics of individual companies and the intricate relationships between them [8, 9]. Factors such as industry sector, market capitalization, and financial health

✉ Matloob Khushi
matloob.khushi@brunel.ac.uk

Xiaobin Lu
xilu5047@uni.sydney.edu.au

Josiah Poon
josiah.poon@sydney.edu.au

¹ School of Computer Science, University of Sydney, City Road, Sydney 2006, NSW, Australia

² Department of Computer Science, Brunel University of London, Kingston Lane, Uxbridge UB8 3PH, Middlesex, United Kingdom

can all influence stock performance, and ignoring these nuances can result in oversimplified models that do not reflect the true dynamics of the market [10].

Second, the narrow focus on predicting stock prices alone provides little operational value for actual trading [11]. While price predictions may achieve high accuracy, they often fail to offer actionable insights that can inform profitable trading decisions [12]. In practice, financial markets require more than accurate price forecasts; they demand a comprehensive understanding of market structure, inter-company correlations, and risk-adjusted returns [13, 14]. Thus, moving from pure price prediction to a framework that considers both predictive performance and practical decision-making is crucial [15].

To address these limitations, recent advances in Graph Neural Networks (GNNs) have shown significant potential in capturing the complex relationships between stocks [16, 17]. Unlike traditional models, GNNs can treat each stock as a node and define the connections between them using various criteria [17]. This is particularly useful in financial markets where inter-company relationships can significantly impact stock performance [18, 19]. For example, companies within the same industry often exhibit similar price behaviors due to shared economic factors, while companies with different business models may demonstrate weaker correlations [20]. Graph Attention Networks (GATs), an extension of GNNs, are particularly powerful as they assign attention weights to different edges, enabling the model to focus more on relevant relationships between companies and discount less important ones. However, the success of graph-based models is highly sensitive to the design of the adjacency matrix, which determines how connections between companies are defined [21, 22], and [23].

In financial markets, these connections can be understood through two major analytical frameworks: technical analysis [24] and fundamental analysis [25]. Technical analysis focuses on price movements and trends, while fundamental analysis examines a company's underlying attributes, such as industry classification, financial performance, and market position [26]. Most existing models that use GNNs for stock prediction primarily focus on one of these aspects, typically using price data to model relationships [27, 28]. However, to fully capture the complexity of stock markets, it is necessary to consider both technical and fundamental factors.

We propose a novel stock market prediction and portfolio optimization model that integrates Bi-directional Long Short-Term Memory (BiLSTM) networks, Graph Attention Networks (GATs), and Attention Mechanisms (AMs). Our approach addresses key limitations of traditional models by treating each company as a distinct node and constructing two separate adjacency matrices: one capturing technical similarities based on price movements and the other reflecting fundamental similarities based on industry sectors. This

dual-graph representation enables a more comprehensive understanding of inter-company relationships.

Beyond price prediction, our model ranks stocks by predicted return rates, constructing a portfolio optimized for risk-adjusted returns. Performance evaluation against benchmark indices (e.g., S&P 500) and baseline models demonstrates superior results across key financial metrics, including internal rate of return (IRR), Sharpe ratio, and final balance. These findings highlight the model's effectiveness in generating higher returns while managing risk.

This research advances stock market prediction by integrating technical and fundamental analysis into a unified framework. Furthermore, it shifts focus from pure forecasting to actionable trading strategies, bridging the gap between financial prediction and decision-making.

2 Related work

Predicting stock prices has long been a complex and enduring challenge within financial markets, captivating both researchers and industry professionals. The dynamic and volatile nature of financial markets has led to the development and continual refinement of a wide range of predictive methodologies. Early efforts were grounded in traditional statistical models, which sought to capture patterns in historical price data. However, with the rapid advancements in computational power and data availability, machine learning models have emerged as powerful alternatives, offering enhanced predictive capabilities. These advanced approaches incorporate not only historical price data but also a broad array of market factors, including technical indicators, macroeconomic variables, and even sentiment analysis. This evolution from traditional to machine learning-based techniques reflects the increasing sophistication in tackling the inherent complexities of financial markets [29–31].

2.1 Technical indicators

Technical indicators are mathematical constructs derived from historical price data, including previous open, high, low, and close prices, as well as trading volumes. These tools are instrumental in identifying trends, momentum, volatility, and other critical aspects of market behavior. According to [32], technical indicators are pivotal in analyzing price movements by providing a deeper understanding of market dynamics, making them indispensable for traders and analysts. Commonly integrated into deep learning models for stock price prediction, widely recognized indicators such as Moving Averages (MA), Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD) serve as important features that enable these models to discern patterns and relationships within historical data [33].

Numerous studies, including those by [34, 35], and [36], have demonstrated that the incorporation of technical indicators into predictive models can improve their performance, allowing models to detect subtle market signals and enhance forecasting precision. Despite their utility, however, the contribution of technical indicators alone to the predictive accuracy of models remains modest, as highlighted by [37]. These findings suggest that while technical indicators provide valuable information, they may not fully capture the complexities of financial markets when used in isolation.

In response to these limitations, recent research has shifted towards integrating technical indicators with a broader range of data sources and more advanced modeling techniques. This approach aims to capture the intricate, multi-dimensional nature of financial markets, leading to more robust and accurate predictions. For instance, models proposed by [38, 39], and [40] combine technical indicators with additional inputs such as sentiment data, macroeconomic variables, and alternative data sources, as well as employing sophisticated methodologies like convolutional neural networks and hybrid models. These advancements have resulted in significant improvements in prediction accuracy, underscoring the importance of comprehensive, multi-faceted approaches to stock price forecasting.

2.2 Long short-term memory networks

Long Short-Term Memory (LSTM) networks, a specialized form of recurrent neural network (RNN), are particularly adept at handling sequential data, making them an ideal choice for time series forecasting tasks such as stock price prediction. Their architecture is uniquely designed to capture long-term dependencies in data by selectively retaining information over extended time periods, a critical feature when dealing with the complex and often nonlinear dynamics present in financial markets [41]. This ability to model temporal dependencies has been consistently shown to outperform traditional machine learning approaches, particularly in financial contexts where time-based patterns play a crucial role.

Numerous studies, including [42, 43], and [44], have demonstrated the superiority of LSTM models over conventional methods in stock price forecasting. These studies emphasize LSTM's capacity to effectively capture nonlinear relationships and intricate temporal patterns that are otherwise difficult to model using standard approaches. By leveraging memory cells that control the flow of information through gates, LSTM models can maintain a balance between short-term and long-term trends in stock prices, thus offering enhanced predictive accuracy in dynamic market environments.

Moreover, recent advancements in stock price prediction have explored the integration of LSTM units within hybrid

models to further boost predictive performance. Research conducted by [45, 46], and [47] has shown that combining LSTM with other methods, such as Convolutional Neural Networks (CNNs) or attention mechanisms, can yield significant improvements in both predictive accuracy and model robustness. These hybrid models capitalize on the temporal modeling capabilities of LSTM while incorporating the feature extraction strengths of CNNs or the ability of attention mechanisms to focus on relevant data points within the sequence.

The incorporation of hybrid LSTM models has proven particularly effective in addressing the volatile and non-stationary nature of stock prices. For instance, these models can better generalize across different market conditions by enhancing their adaptability and reducing overfitting to specific patterns. As demonstrated in [48], the combination of LSTM with other deep learning techniques provides a more holistic approach to stock price forecasting, improving not only the model's accuracy but also its ability to handle the uncertainty and complexity inherent in financial data. This ongoing research highlights the continued evolution of LSTM-based models as a cornerstone in the development of more sophisticated stock prediction systems.

2.3 Graph attention networks

Graph Neural Networks (GNNs) have emerged as a powerful tool for analyzing data structured in the form of graphs, where entities (nodes) are interconnected through relationships (edges) [49, 50]. Unlike traditional machine learning models, which primarily focus on independent and identically distributed (IID) data, GNNs excel at capturing the complex interactions between entities within graph structures [51]. This makes GNNs particularly suitable for tasks such as social network analysis [52], molecular property prediction [53], and, more recently, financial market modeling [54]. In stock prediction, the relationships between different stocks, sectors, or even external factors such as news sentiment can be represented as a graph, allowing GNNs to uncover hidden dependencies and trends that are otherwise difficult to model with traditional approaches [55, 56].

Building upon the GNN framework, Graph Attention Networks (GATs) introduce attention mechanisms to further enhance the learning process [57–62]. While GNNs aggregate information from neighboring nodes, they typically treat all neighbors equally [63]. GATs, on the other hand, assign varying levels of importance to different neighbors based on their relevance to the task at hand. This is achieved through the attention mechanism, which allows the model to weigh the influence of each neighboring node dynamically, ensuring that more critical connections are given greater focus [64, 65]. As a result, GATs can capture more nuanced and contextually important relationships within the graph, making

them highly effective in tasks where the importance of relationships varies significantly across the graph [66, 67].

In the context of stock price prediction, GATs have been increasingly applied to model the intricate and dynamic relationships between stocks [68]. Financial markets are highly interconnected, with stocks often influencing one another due to factors such as industry sector, supply chain dependencies, or macroeconomic conditions [69]. GATs allow for the modeling of these dependencies by constructing a graph where each node represents a stock, and edges represent the relationships between them, whether they be based on historical price correlations, shared market sectors, or external influences such as news sentiment [70, 71], and [72].

Several studies have demonstrated the effectiveness of GATs in improving the accuracy of stock price prediction models. By leveraging the attention mechanism, GATs can identify and focus on the most relevant relationships between stocks, thereby enhancing the model's ability to capture market dynamics [73]. For example, recent work from [74, 75], and [76] has shown that GAT-based models outperform traditional time series models by effectively incorporating relational information into the forecasting process. This has led to improved prediction accuracy, particularly in capturing inter-stock dependencies and broader market trends that are essential for making informed financial decisions. The adaptability and precision of GATs in financial modeling underscore their growing importance in stock market prediction research [77].

2.4 Attention mechanism

The attention mechanism has become a foundational element in modern deep learning models [78], particularly for tasks involving complex data with multiple sources or channels [79]. At its core, the attention mechanism dynamically weighs the importance of different parts of the input, allowing models to focus on the most relevant information [80, 81]. This is particularly useful in scenarios where the model needs to integrate diverse types of input or outputs from different channels, such as in natural language processing [82], computer vision [83], or financial forecasting [84].

In broader applications, including stock price prediction, attention mechanisms are equally valuable. For instance, when multiple channels of information—such as historical price data, technical indicators, and external market factors—are available, the attention mechanism can combine these diverse outputs by assigning appropriate importance to each channel [85, 86]. This ability to focus on the most informative data points helps improve predictive performance, particularly in volatile and interconnected environments like financial markets [87].

The flexibility of attention mechanisms lies in their capacity to integrate and weigh information from different

channels in a data-driven manner [88]. By doing so, attention-based models can better capture complex dependencies, making them highly effective in tasks that require the synthesis of information from multiple sources [89, 90]. In financial prediction models, this mechanism allows for more refined and contextually aware forecasts by dynamically prioritizing the most relevant inputs, leading to more accurate and robust predictions [91].

3 Methodology

3.1 Technique indicators

This study leverages a set of basic and derived technical indicators to analyze stock price movements and inform trading decisions. The fundamental indicators used are the Open, High, Low, and Close (OHLC) prices, as well as tick volume and spread. These indicators form the basis for generating an additional 44 features, categorized Table 1.

3.1.1 Simple moving averages (SMA)

The Simple Moving Average (SMA) is a widely used technical indicator that smooths out price data by creating a constantly updated average price. It helps in identifying the direction of the trend over a specified period [92].

The SMA is calculated by taking the arithmetic mean of a given set of prices over a specific number of periods.

$$SMA_n(t) = \frac{1}{n} \sum_{i=0}^{n-1} P(t-i) \quad (1)$$

where:

- $SMA_n(t)$ is the Simple Moving Average at time t over n periods.
- $P(t-i)$ is the close price at time $t-i$.
- n is the number of periods over which the average is calculated.

In this research the 'mv100', 'mv50', 'mv9' are moving averages over 100, 50, and 9 periods, respectively. SMAs help in smoothing out price data to identify trends over different time frames.

3.1.2 Bollinger bands

Bollinger Bands consist of a set of lines plotted two standard deviations (positively and negatively) away from a simple moving average (SMA) of the price which provide a relative

Table 1 List of indicators, labels, and number of indicators used

Indicators	Labels	Count
Simple Moving Averages	'mv100', 'mv50', 'mv9'	3
Bollinger Bands	'bb_bbm', 'bb_bbh', 'bb_bbl'	3
Relative Strength Index	'rsi15', 'rsi9', 'rsi50'	3
Percentage Change Features	'f1' to 'f10'	10
Moving Average Comparisons	'f11' to 'f16'	6
RSI Comparisons	'f17' to 'f18'	2
Bollinger Band Comparisons	'f19' to 'f22'	4
Rolling Maximum and Minimum	'f23' to 'f28'	6
Close Price Shifts	'f29' to 'f33'	5
Trading time	'h1', 'w1'	2
Total		44

definition of high and low prices of a financial instrument [93].

Middle Band (MB) The middle band is the simple moving average (SMA) of the close price, typically over 20 periods.

$$MB(t) = SMA_{20}(t) = \frac{1}{20} \sum_{i=0}^{19} P(t-i) \quad (2)$$

Upper Band (UB) The upper band is calculated by adding two standard deviations to the middle band.

$$UB(t) = MB(t) + 2 \times \sigma_{20}(t) \quad (3)$$

where $\sigma_{20}(t)$ is the standard deviation of the close price over 20 periods.

Lower Band (LB) The lower band is calculated by subtracting two standard deviations from the middle band.

$$LB(t) = MB(t) - 2 \times \sigma_{20}(t) \quad (4)$$

The 'bb_bbm', 'bb_bbh', 'bb_bbl' represent the middle band (moving average), upper band, and lower band respectively.

3.1.3 Relative strength index (RSI)

The Relative Strength Index (RSI) [94] is a momentum oscillator that measures the speed and change of close price movements. It is used to identify overbought or oversold conditions in a market. The RSI oscillates between 0 and 100 and is typically used with a 14-period setting.

1. Calculate the average gains and losses over the specified period (e.g., 14 or 50 periods).

2. Calculate the Relative Strength (RS):

$$RS = \frac{\text{Average Gain}}{\text{Average Loss}} \quad (5)$$

3. Calculate the RSI:

$$RSI = 100 - \left(\frac{100}{1 + RS} \right) \quad (6)$$

'rsi14', 'rsi50' are RSI over 14 and 50 periods, respectively, measures the speed and change of close price movements to identify overbought or oversold conditions. 'rsimv9' is a 9-period moving average of the 14-period RSI.

3.1.4 Price percentage change features

'f1' to 'f10' calculate the percentage change between different prices (open, close, high, low) and their shifts over different periods.

$$f1 = \left(\frac{\text{Close} - \text{Open}}{\text{Open}} \right) \times 100 \quad (7)$$

$$f2 = \left(\frac{\text{High} - \text{Low}}{\text{Low}} \right) \times 100 \quad (8)$$

$$f3 = \left(\frac{\text{High}_{t-1} - \text{Low}_{t-1}}{\text{Low}_{t-1}} \right) \times 100 \quad (9)$$

$$f4 = \left(\frac{\text{High}_{t-2} - \text{Low}_{t-2}}{\text{Low}_{t-2}} \right) \times 100 \quad (10)$$

$$f5 = \left(\frac{\text{High}_{t-3} - \text{Low}_{t-3}}{\text{Low}_{t-3}} \right) \times 100 \quad (11)$$

$$f6 = \left(\frac{\text{High}_{t-4} - \text{Low}_{t-4}}{\text{Low}_{t-4}} \right) \times 100 \quad (12)$$

$$f7 = \left(\frac{\text{High} - \text{Open}}{\text{Open}} \right) \times 100 \quad (13)$$

$$f8 = \left(\frac{\text{High} - \text{Close}}{\text{Close}} \right) \times 100 \quad (14)$$

$$f9 = \left(\frac{\text{Open} - \text{Low}}{\text{Low}} \right) \times 100 \quad (15)$$

$$f10 = \left(\frac{\text{Close} - \text{Low}}{\text{Low}} \right) \times 100 \quad (16)$$

3.1.5 Moving average percentage change features

'f11' to 'f13' calculate the percentage change between the closing price and the moving averages (50-period, 9-period, and 100-period, respectively). Features 'f14' to 'f16' compute the percentage changes between different moving averages themselves.

$$f11 = \left(\frac{\text{Close} - \text{MV}_{50}}{\text{MV}_{50}} \right) \times 100 \quad (17)$$

$$f12 = \left(\frac{\text{Close} - \text{MV}_9}{\text{MV}_9} \right) \times 100 \quad (18)$$

$$f13 = \left(\frac{\text{Close} - \text{MV}_{100}}{\text{MV}_{100}} \right) \times 100 \quad (19)$$

$$f14 = \left(\frac{\text{MV}_9 - \text{MV}_{50}}{\text{MV}_{50}} \right) \times 100 \quad (20)$$

$$f15 = \left(\frac{\text{MV}_9 - \text{MV}_{100}}{\text{MV}_{100}} \right) \times 100 \quad (21)$$

$$f16 = \left(\frac{\text{MV}_{50} - \text{MV}_{100}}{\text{MV}_{100}} \right) \times 100 \quad (22)$$

3.1.6 RSI percentage change features

'f17', 'f18' calculate the percentage difference between different RSI values (rsi14, rsi50, rsmv9). f17 computes the percentage change between the 14-period RSI and the 50-period RSI, while f18 calculates the percentage change between the 50-period RSI and a 9-period simple moving average of the 14-period RSI.

$$f17 = \left(\frac{\text{RSI}_{14} - \text{RSI}_{50}}{\text{RSI}_{50}} \right) \times 100 \quad (23)$$

$$f18 = \left(\frac{\text{RSI}_{50} - \text{RSI}_{\text{mv9}}}{\text{RSI}_{\text{mv9}}} \right) \times 100 \quad (24)$$

3.1.7 Bollinger band percentage change features

'f19' to 'f23' calculate the percentage difference between the close price and Bollinger Bands (bb_bbm, bb_bbh, bb_bbl), and between the bands themselves. Specifically, f19 computes the percentage change between the closing price and the middle Bollinger Band (20-period simple moving average), f20 calculates the percentage change between the closing price and the upper Bollinger Band, and f21 calculates the percentage change between the closing price and the lower Bollinger Band. Additionally, f22 computes the percentage change between the lower and upper Bollinger Bands.

$$f19 = \left(\frac{\text{Close} - \text{BB}_{\text{Middle}}}{\text{BB}_{\text{Middle}}} \right) \times 100 \quad (25)$$

$$f20 = \left(\frac{\text{Close} - \text{BB}_{\text{Upper}}}{\text{BB}_{\text{Upper}}} \right) \times 100 \quad (26)$$

$$f21 = \left(\frac{\text{Close} - \text{BB}_{\text{Lower}}}{\text{BB}_{\text{Lower}}} \right) \times 100 \quad (27)$$

$$f22 = \left(\frac{\text{BB}_{\text{Lower}} - \text{BB}_{\text{Upper}}}{\text{BB}_{\text{Upper}}} \right) \times 100 \quad (28)$$

3.1.8 Rolling maximum and minimum

'f23' to 'f28' calculate the percentage difference between the close price and its rolling maximum or minimum over different periods (20, 50, 100). Specifically, 'f23' to 'f25' compute the percentage change between the rolling maximum closing prices over 20, 50, and 100 periods, respectively, and the current closing price. Conversely, 'f26' to 'f28' calculate the percentage change between the rolling minimum closing prices over the same periods and the current closing price.

$$f23 = \left(\frac{\max(\text{Close}_{t-20:t}) - \text{Close}}{\text{Close}} \right) \times 100 \quad (29)$$

$$f24 = \left(\frac{\max(\text{Close}_{t-50:t}) - \text{Close}}{\text{Close}} \right) \times 100 \quad (30)$$

$$f25 = \left(\frac{\max(\text{Close}_{t-100:t}) - \text{Close}}{\text{Close}} \right) \times 100 \quad (31)$$

$$f26 = \left(\frac{\min(\text{Close}_{t-20:t}) - \text{Close}}{\text{Close}} \right) \times 100 \quad (32)$$

$$f27 = \left(\frac{\min(\text{Close}_{t-50:t}) - \text{Close}}{\text{Close}} \right) \times 100 \quad (33)$$

$$f28 = \left(\frac{\min(\text{Close}_{t-100:t}) - \text{Close}}{\text{Close}} \right) \times 100 \quad (34)$$

3.1.9 Close price shifts

‘f29’ to ‘f33’ calculate the percentage change of the close price compared to its previous values over different periods (1 to 5). ‘f29’ computes the percentage change from the closing price of the previous day to the current closing price. Features ‘f30’ to ‘f33’ extend this calculation to the closing prices from 2 to 5 days prior, respectively.

$$f29 = \left(\frac{\text{Close}_{t-1} - \text{Close}}{\text{Close}} \right) \times 100 \quad (35)$$

$$f30 = \left(\frac{\text{Close}_{t-2} - \text{Close}}{\text{Close}} \right) \times 100 \quad (36)$$

$$f31 = \left(\frac{\text{Close}_{t-3} - \text{Close}}{\text{Close}} \right) \times 100 \quad (37)$$

$$f32 = \left(\frac{\text{Close}_{t-4} - \text{Close}}{\text{Close}} \right) \times 100 \quad (38)$$

$$f33 = \left(\frac{\text{Close}_{t-5} - \text{Close}}{\text{Close}} \right) \times 100 \quad (39)$$

3.1.10 Trading time

‘h1’ captures the hour of the day from the datetime values. The second line creates a new column wd that captures the day of the week (with Monday as 0 and Sunday as 6) from the datetime values, which could be useful for identifying patterns related to different weekdays.

$$h1 = \text{Hour}(\text{datetime}) \quad (40)$$

$$wd = \text{Weekday}(\text{datetime}) \quad (41)$$

3.2 Long short-term memory (LSTM) network

In this study, we employ Long Short-Term Memory (LSTM) networks to model sequential data and capture both short-term and long-term dependencies. LSTM networks, a variant of Recurrent Neural Networks (RNNs), are particularly well-suited for tasks involving temporal sequences due to their ability to mitigate the vanishing gradient problem inherent in traditional RNNs. This is achieved through the inclusion of a memory cell that maintains information over extended time steps.

The structure of an LSTM cell is defined by three key gates: the input gate, the forget gate, and the output gate. These gates regulate the flow of information into and out of the memory cell, enabling the network to selectively retain or discard information at each time step. The equations governing the behavior of the LSTM cell are as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (42)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (43)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (44)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (45)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (46)$$

$$h_t = o_t * \tanh(C_t) \quad (47)$$

In these equations, f_t represents the forget gate, which determines the extent to which the previous cell state C_{t-1} should be forgotten. i_t is the input gate, controlling what new information is stored in the current cell state. The candidate cell state, $\tilde{C}_t$, is computed based on the previous hidden state h_{t-1} and current input x_t . The updated cell state C_t is a combination of the previous cell state and the candidate cell state, modulated by the forget and input gates. Finally, o_t is the output gate, which controls the output of the LSTM cell, and h_t represents the hidden state, which is passed to the next time step.

LSTMs are particularly effective in time series forecasting tasks, such as stock price prediction, as they can capture both short-term fluctuations and long-term trends in financial data. By leveraging historical stock prices and other relevant indicators, our LSTM model provides improved forecasting accuracy compared to traditional models.

The weight matrices W_f , W_i , W_C , W_o and bias vectors b_f , b_i , b_C , b_o are learned during the training process, ensuring the model adapts to the dynamics of the data.

3.3 Graph construction

In our approach, each company is represented as a unique node within a graph, where edges are established based on similarity metrics. Companies exhibiting higher similarity are interconnected, effectively capturing interdependencies within the financial market. This graph-based framework enables the incorporation of relational information among stocks, allowing for the simultaneous prediction of closing prices across all companies.

3.3.1 Scaling

Before constructing the weighted edge graph, we first normalize the closing prices of all companies, scaling them to the range [0, 1]. Let $X_i = [x_1, x_2, \dots, x_T]$ represent the normalized time series of closing prices for company i . The normalization is done as follows:

$$X_i = \frac{X_i - \min(X_i)}{\max(X_i) - \min(X_i)} \quad (48)$$

3.3.2 Graph for technical similarity

After normalizing the closing prices, we compute the Dynamic Time Warping (DTW) distance between every pair of companies using the fastdtw algorithm, which provides an efficient approximation of the DTW distance. Given two normalized time series X_i and X_j for companies i and j , the DTW distance $DTW(X_i, X_j)$ is calculated.

Next, we introduce a threshold τ to determine which company pairs should be connected by edges. For any two companies i and j , if their DTW distance $DTW(X_i, X_j)$ is less than the threshold τ , an edge is established between them. The weight of the edge w_{ij} is computed as:

$$w_{ij} = \tau - DTW(X_i, X_j) \quad (49)$$

This formula ensures that the closer the DTW distance is to 0 (i.e., the more similar the two companies' stock prices are), the larger the edge weight will be. Conversely, as the DTW distance approaches the threshold, the edge weight decreases.

The result is a weighted graph, where each node represents a company, and edges are established only between companies whose DTW distance is below the threshold τ . The edge weight reflects the similarity of the companies' stock price movements, with smaller DTW distances leading to stronger connections. This graph provides a structural representation of the relationships between companies based on their normalized closing prices.

This weighted graph as is shown in Fig. 5 is then used in subsequent Graph Neural Network (GNN) and Graph Attention Network (GAT) models to predict future trends by incorporating both individual company data and the relationships captured in the graph.

3.3.3 Graph for fundamental similarity

The construction of the second graph is predicated on the industry sectors of the selected companies. We introduce this graph under the hypothesis that companies within the same industry sector exhibit similar patterns in price performance. To operationalize this concept, an edge is established between companies belonging to the same industry sector, with each edge assigned a uniform weight. This approach assumes that the shared sectorial characteristics will reflect in comparable price change behaviors, providing a structured framework to analyze the interconnectedness of market performance across similar industries.

3.4 Graph neural networks (GNN) and graph attention networks (GAT)

In this study, we also incorporate Graph Neural Networks (GNNs) to model complex relationships between entities

represented as graphs. GNNs are particularly suited for structured data, where the relationships between nodes can provide crucial insights that are otherwise missed by traditional neural networks. Each node in a graph represents an entity, and edges define relationships or interactions between these entities. The objective of a GNN is to learn a node representation by aggregating features from neighboring nodes through message passing mechanisms.

The key operation in a GNN is the neighborhood aggregation or message passing, where the representation of each node is updated based on its neighbors' features. Formally, the node representation $h_v^{(k)}$ at the k -th layer is computed as:

$$h_v^{(k)} = \text{aggregate} \left(h_u^{(k-1)} : u \in \mathcal{N}(v) \right) \quad (50)$$

where $\mathcal{N}(v)$ represents the set of neighbors of node v , and $h_u^{(k-1)}$ is the representation of node u from the previous layer. The aggregation function can vary based on the GNN variant (e.g., summation, averaging, or more complex functions).

3.4.1 Graph attention networks (GAT)

To further enhance the expressiveness of GNNs, we utilize Graph Attention Networks (GATs), which introduce attention mechanisms to assign different importance weights to neighboring nodes during the aggregation process. In a GAT, rather than treating all neighbors equally, attention scores are learned, allowing the model to focus on the most relevant nodes. This attention mechanism is particularly useful in scenarios where not all neighbors contribute equally to a node's final representation.

For each node v , the attention coefficient α_{vu} between node v and its neighbor u is computed as:

$$\alpha_{vu} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T [Wh_v \| Wh_u]))}{\sum_{k \in \mathcal{N}(v)} \exp(\text{LeakyReLU}(\mathbf{a}^T [Wh_v \| Wh_k]))} \quad (51)$$

where W is a weight matrix applied to the node features, $\mathbf{a}$ is a learnable attention vector, and $\|$ denotes concatenation. The attention coefficients α_{vu} are then used to compute the weighted sum of the neighboring features to update the node representation:

$$h_v^{\text{new}} = \sigma \left(\sum_{u \in \mathcal{N}(v)} \alpha_{vu} Wh_u \right) \quad (52)$$

Here, σ is a non-linear activation function, such as ReLU. This mechanism allows GATs to adaptively attend to the most informative neighbors, improving the model's performance on tasks where the relevance of neighbors varies (Fig. 1).

By leveraging GNNs and GATs, our methodology captures both local and global dependencies in graph-structured

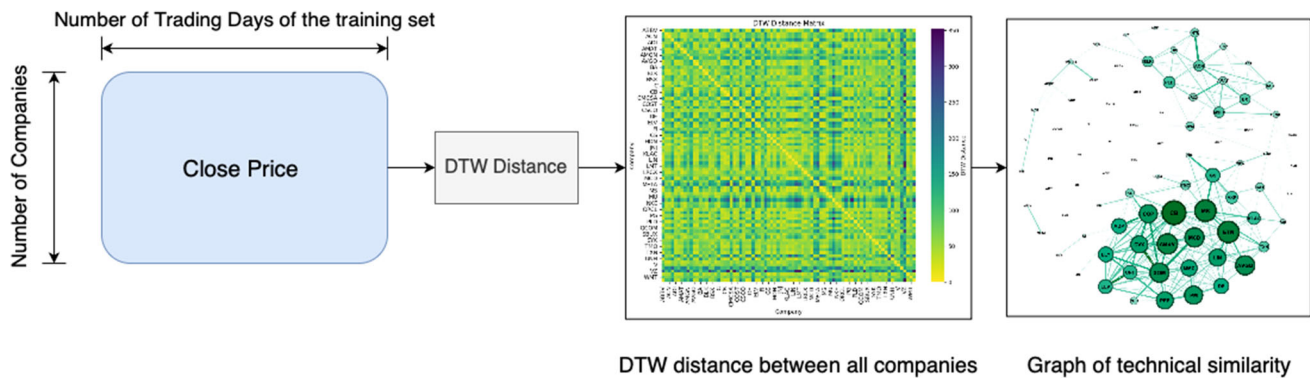


Fig. 1 Illustration of the step-by-step process for constructing the first graph, which quantifies technical similarity using DTW

data, offering a robust framework for learning node representations. The graph attention mechanism further enhances the model's ability to focus on critical nodes in the graph, leading to more accurate predictions, particularly in tasks involving structured data such as social networks, molecular graphs, or citation networks.

3.5 Attention mechanism for multi-channel graph networks

In our approach, we utilize an attention mechanism to combine the outputs of two separate graph networks, each representing a different channel of information for every node in the graph. By leveraging this mechanism, we are able to aggregate information from both graph channels and make more informed predictions based on the features of each node.

Given two graph networks, G_1 and G_2 , the output of each network for a node v is represented by the feature vectors $h_v^{G_1}$ and $h_v^{G_2}$, respectively. Our goal is to combine these two outputs using an attention mechanism that determines the contribution of each graph channel to the final representation of node v .

The attention mechanism assigns a weight to each graph channel based on the relevance of the information provided by each channel for the specific node. For each node v , the attention coefficients $\beta_v^{G_1}$ and $\beta_v^{G_2}$ are computed as follows:

$$\beta_v^{G_i} = \frac{\exp(\mathbf{a}^T \cdot \text{concat}(h_v^{G_1}, h_v^{G_2}))}{\sum_{j \in \{1,2\}} \exp(\mathbf{a}^T \cdot \text{concat}(h_v^{G_1}, h_v^{G_2}))} \quad (53)$$

Here, $\mathbf{a}$ is a learnable attention vector, and $\text{concat}(h_v^{G_1}, h_v^{G_2})$ represents the concatenation of the feature vectors from both graph channels for node v . The softmax function ensures that the attention coefficients $\beta_v^{G_1}$ and $\beta_v^{G_2}$ sum to 1, allowing the model to weigh the importance of each channel dynamically.

The final representation h_v^{final} for node v is then computed as the weighted sum of the two graph channels:

$$h_v^{\text{final}} = \beta_v^{G_1} h_v^{G_1} + \beta_v^{G_2} h_v^{G_2} \quad (54)$$

This mechanism enables the model to focus on the most relevant graph channel for each node, depending on the specific task and data characteristics.

3.5.1 Node-level prediction

Once the final node representation h_v^{final} is obtained by combining the two graph outputs, we use it to make predictions at the node level. Specifically, for each node v , we apply a fully connected layer followed by a softmax function to output the predicted class or value for that node. The prediction $\hat{y}_v$ for node v is computed as:

$$\hat{y}_v = \text{softmax}(W \cdot h_v^{\text{final}} + b) \quad (55)$$

where W is a weight matrix, and b is a bias term learned during the training process. The softmax function converts the raw scores into probabilities for classification tasks, while other activation functions may be used for regression tasks.

By incorporating the attention mechanism to fuse the two graph channels and making node-specific predictions, our model is able to leverage the full potential of multi-channel graph information, leading to more accurate and context-aware predictions for each node.

3.6 Proposed LSTM-GAT attention model

In this study, we propose a novel model that combines Long Short-Term Memory (LSTM) networks, Graph Attention Networks (GATs), and an attention mechanism to capture both temporal and relational dependencies in time series data. A single input at time t consists of a 2-dimensional vector representing 82 companies, with each company characterized

by 51 features, including the technical indicators mentioned previously.

3.6.1 Model architecture

Our proposed model consists of three main components: the LSTM layers, the GAT layers for two graph channels, and an attention mechanism to fuse the outputs from these channels. The final output is generated through a series of fully connected layers. Below is a detailed description of each component:

LSTM Layers The LSTM layers are used to capture the temporal dependencies in the time series data. The input time series data for each node is passed through a two-layer bidirectional LSTM, which processes the sequential information in both forward and backward directions. The LSTM output at the final time step is used as the node's representation of the sequence. This results in a feature vector for each node that summarizes the historical information.

GAT Layers The model employs two sets of GAT layers to process graph-structured data from two distinct adjacency matrices (representing two different graph channels). Each graph is processed through three consecutive GAT layers. Each GAT layer updates the node features by attending to the features of neighboring nodes, where the attention mechanism learns to assign different importance weights to each neighbor.

For both graph channels, the node features from the LSTM layers are first passed through the GAT layers to propagate information from neighboring nodes, resulting in updated node representations for each channel.

Attention Mechanism After processing the node features through the two sets of GAT layers, the model employs an attention mechanism to fuse the outputs from the two graph channels. The GAT outputs from the two channels are stacked and passed through a linear layer to compute attention scores for each channel. These scores are then normalized using a softmax function to obtain attention weights, which are

applied to the GAT outputs. The final node representation is computed as the weighted sum of the two GAT outputs.

Fully Connected Layers

The combined node features from the attention mechanism are passed through a series of fully connected layers. The first two layers use ReLU activations, while the final layer outputs the prediction for each node. This part of the model captures complex non-linear relationships in the learned node representations as is shown in Fig. 2.

4 Evaluation metrics

In this study, four primary evaluation metrics are employed to assess the performance of our proposed model: the Mean Squared Error (MSE) for prediction accuracy, the Sharpe Ratio for portfolio performance, the Mean Absolute Error (MAE) for absolute prediction deviations, and the Annual Return and Maximum Drawdown for overall financial performance. These metrics provide a comprehensive analysis of the predictive capability, risk-adjusted return, and financial resilience of the model.

4.1 Mean squared error (MSE)

The Mean Squared Error (MSE) is used to measure the accuracy of the model's predictions of the next day's stock prices. It evaluates how close the predicted values are to the actual values, with a lower MSE indicating a more accurate model. The MSE is calculated as follows:

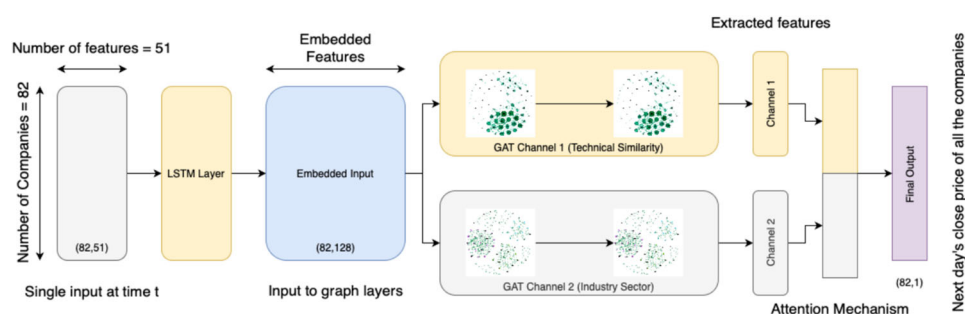
$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (56)$$

where y_i is the actual stock price, $\hat{y}_i$ is the predicted stock price, and n represents the total number of data points.

4.2 Sharpe ratio

The Sharpe Ratio is utilized to evaluate the risk-adjusted return of the portfolio generated from the model's predictions.

Fig. 2 Visualisation of proposed LSTM-GAT-AM model



It compares the portfolio's excess return to its volatility, with a higher Sharpe Ratio indicating a more favorable risk-adjusted return. The Sharpe Ratio is computed as follows:

$$\text{Sharpe Ratio} = \frac{\mathbb{E}[R_p - R_f]}{\sigma_p} \quad (57)$$

where R_p is the portfolio return, R_f is the risk-free rate, and σ_p is the standard deviation of the portfolio's excess return.

4.3 Mean absolute error (MAE)

The Mean Absolute Error (MAE) measures the average magnitude of the errors in a set of predictions, without considering their direction. It is less sensitive to outliers than the MSE and provides a clearer measure of actual prediction errors. The MAE is calculated as follows:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (58)$$

where y_i and $\hat{y}_i$ are defined as in the MSE section.

4.4 Annual return and maximum drawdown

The Annual Return measures the percentage change in the portfolio value over a year, reflecting the overall profitability of the investment strategy. The Maximum Drawdown assesses the largest single drop from peak to trough in the portfolio during the investment period, providing insight into the potential risk of losses. These metrics together offer a complete picture of the financial performance and risk resilience of the portfolio.

- **Annual Return:** Calculated based on the cumulative returns at the end of the year compared to the initial portfolio value.
- **Maximum Drawdown:** Defined as the maximum observed loss from a peak to a trough of the portfolio, before a new peak is attained.

5 Experimental setup

5.1 Data source

The data for this study were obtained from Yahoo Finance, comprising the largest 100 companies by market share in the S&P 500. The sampling period spanned from January 1, 2020, to December 31, 2023 including the Open, High, Low, Close, and Volume. Then we generate the technical indicators respectively as is shown in Fig. 3.

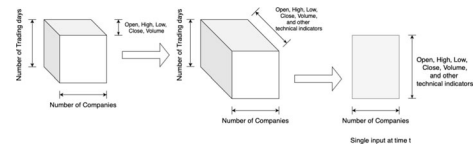


Fig. 3 Visualisation of the process of data pre-processing

To ensure data consistency, we specifically excluded companies that underwent stock splits during the sample period. Stock splits introduce abrupt, non-fundamental shifts in stock prices, which could obscure the true patterns the model is designed to detect. By omitting such companies, we minimize distortions and focus on capturing the underlying relationships between company characteristics and stock price movements. This approach enhances the model's predictive accuracy and robustness.

5.2 Data pre-processing

5.2.1 Scaling

To ensure that all features contribute equally to the prediction model and to improve the performance of the regression algorithms, data scaling will be applied during the pre-processing step. We will employ *StandardScaler()* to standardize the features and stock prices by removing the mean and scaling to unit variance. This transformation is expressed as:

$$z_i = \frac{x_i - \mu}{\sigma} \quad (59)$$

where x_i represents the original feature value, μ is the mean, and σ is the standard deviation of the feature values. This standardization ensures that the data has a mean of 0 and a standard deviation of 1, aligning all features on the same scale. This step is essential for models that are sensitive to feature scales or assume normally distributed input data.

5.2.2 Train test split

To prevent the model from learning from future data, which could lead to overfitting, the dataset is split into training, validation, and test sets based on chronological order. The training data spans from 2020-01-01 to 2023-04-01, the validation set covers the period from 2023-04-01 to 2023-08-01, and the test set includes data from 2023-08-01 to 2023-12-31 as is shown in Fig. 4. It is important to note that the scaling of features is performed using the statistics (mean and standard deviation) derived only from the training set to ensure that no future information is leaked into the model during the scaling process. This approach helps maintain the integrity of the model evaluation.



Fig. 4 The split interval of train, validation, and test set

5.3 Graph preparing

5.3.1 Graph with technical analysis

As is shown in Fig. 5 the first graph is a weighted and undirected graph based on the Dynamic Time Warping (DTW) distance between the companies. Companies with smaller distances will have a higher weight on the edges that link them.

5.3.2 Graph with fundamental analysis

As is shown in Fig. 6 the second graph is an undirected and unweighted graph based on the industry sector the company belongs to. Companies within the same industry sector will be linked.

5.4 Software and hardware setup

The experiments in this study were conducted using the following software environment: PyTorch 2.4.0, TensorFlow 2.13.0, Keras 2.13.1, Pandas 2.0.3, and Numpy 1.24.3. The hardware configuration consisted of an Apple Silicon processor (ARM architecture) with 12 CPU cores (12 physical, 12 logical) and 32.0 GB of RAM, running on macOS (Darwin 23.6.0). GPU acceleration was leveraged using the MPS Backend (Metal Performance Shaders), which was enabled and available for PyTorch, with the MPS device specified as mps.

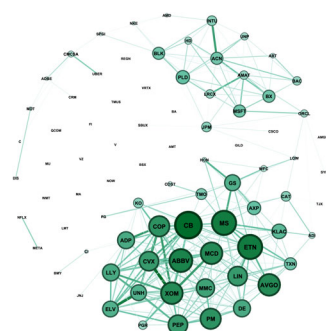


Fig. 5 Visualization of the adjacency matrix using the Fruchterman-Reingold algorithm. Each company is represented as a unique dot. Companies are connected with edges where the weight is inversely related to their DTW distance, with closer companies having heavier connections. The size of each dot is scaled based on the number of connections to enhance visual clarity

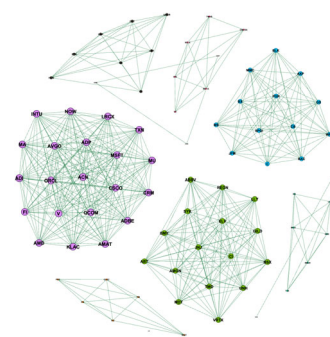


Fig. 6 Visualization of the adjacency matrix using the Fruchterman-Reingold algorithm. Each company is represented as a unique dot. Companies are connected within the same industry sector. The size of each dot is scaled based on the number of connections to enhance visual clarity

6 Experimental results and comparative analysis

In this section, we present the results in a two-fold manner. Initially, we evaluate our model's ability to predict the next day's closing price, comparing its performance against other models with single graph generated by DTW distance or Industry sector using the Mean Squared Error (MSE) as the evaluation metric.

Subsequently, utilizing the predicted closing prices, we compute the daily return rate, ranking companies based on this metric. A portfolio is then formulated by selecting the top-performing companies. This portfolio is backtested with actual market data to assess its practical viability.

The process involves recalculating the true average return rate of the selected companies daily and adjusting the portfolio composition accordingly. This iterative procedure is replicated across each trading day to emulate the portfolio's temporal performance.

We conclude by analyzing and contrasting the final balance and Sharpe ratio achieved by our model against those of other models during the testing period.

6.1 Results on prediction

In this section, we conduct a detailed comparison of our proposed model against several baseline models including ablation studies, employing identical experimental setups and performance metrics to ensure a fair and rigorous evaluation. The results of this comparison are presented in Table 2.

The baseline BiLSTM-GAT model retains its original structure, utilizing a single graph constructed from Dynamic Time Warping (DTW) distance or the industry sector graph. In contrast, the BiLSTM-GNN model replaces the Graph Attention Network (GAT) framework with a more general Graph Neural Network (GNN) architecture to highlight the differences in performance attributable to the graph modeling

Table 2 Comparison of proposed model and baseline model in MSE/MAE for prediction of next day's close price

Model	MSE (Mean Squared Error)	MAE (Mean Absolute Error)
BiLSTM-GAT-AM (DTW & Sector graph)	0.00440	0.04680
BiLSTM-GAT (DTW graph)	0.00454	0.04730
BiLSTM-GAT (Sector graph)	0.00543	0.05408
BiLSTM-GNN (DTW graph)	0.01518	0.08032
BiLSTM-GNN (Sector graph)	0.00561	0.05484

Bold items are best score

approach. Notably, our proposed BiLSTM-GAT-AM model introduces an enhanced architecture by incorporating two distinct graphs, enabling it to capture more complex relational structures within the data.

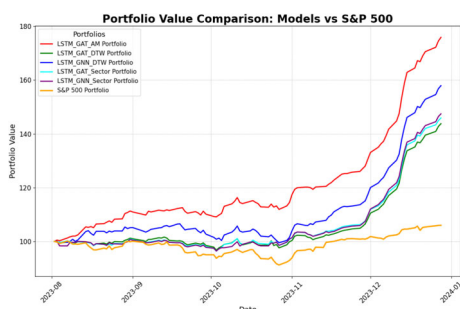
By comparing these models side by side, we aim to illustrate the improvements brought about by the additional graph in the BiLSTM-GAT-AM model, which allows for richer feature extraction and, ultimately, superior predictive performance.

6.2 Results on portfolio return

In this section, we compare the portfolio returns of the three models with the performance of the S&P 500 index during the testing period. The results, as illustrated in Fig. 7, demonstrate that our proposed model achieves the highest final portfolio balance, outperforming both the baseline models and the S&P 500 index. Furthermore, the proposed model exhibits the largest Sharpe ratio, smaller drawdown, and higher annual return rate as is shown in Table 3, indicating a superior risk-adjusted return compared to the other models. This highlights the model's ability to generate consistent returns while effectively managing risk throughout the evaluation period.

7 Discussion of results and ablation studies

Our study highlights the critical relationship between prediction accuracy and decision-making in stock market models.

**Fig. 7** Comparison of the final balance of the proposed model and baseline models on testing period

While prediction accuracy, as measured by Mean Squared Error (MSE), is a key factor in evaluating a model's performance, it is not the sole indicator of its practical utility in real-world trading. The ultimate goal in financial modeling is not just to accurately predict stock prices but to make profitable and risk-adjusted portfolio decisions. This is where the strength of the proposed BiLSTM-GAT-AM model truly stands out.

From the MSE results, we observe that the proposed model slightly outperforms the BiLSTM-GAT model that uses only the DTW graph. This shows that our dual-graph approach, combining both technical and fundamental insights, marginally improves predictive accuracy. However, it is in the backtesting phase-where the model's predictions are translated into actionable decisions-that the true novelty and superiority of our approach become apparent.

Backtesting results reveal that the proposed BiLSTM-GAT-AM model consistently generates higher portfolio returns compared to all baseline models, including those using single-graph approaches. The final portfolio balance achieved by the proposed model is the highest, as is the Sharpe ratio, which indicates that the model is not only generating higher returns but is also managing risk more effectively. In contrast, the baseline models, particularly those using only the sector graph, fail to capture the full complexity of the stock market and deliver inferior performance in both return and risk metrics.

The DTW-based graph, which captures technical relationships between stocks, yields better results than the sector-based graph alone, but it still does not fully account for the broader, fundamental relationships that can influence long-term portfolio performance. The combination of the two graphs in the BiLSTM-GAT-AM model enables it to capture both short-term price movements and long-term industry-based relationships, leading to more informed portfolio decisions. This dual-graph structure allows the model to generalize across different market conditions, thus providing superior results at the end of the testing period, even when the market is favorable for all models.

In addition, while predicting stock prices accurately is important, the decision-making process that follows from these predictions is crucial. Our model's slight edge in

Table 3 Comparison of proposed model and baseline model on Final balance, Sharpe ratio, Max Draw-down, and Annual return

Model	Final Balance	Sharpe ratio	maximum DD	Annual RR
BiLSTM-GAT-AM (DTW & Sector graph)	176.56	0.5554	-3.79%	302.47%
BiLSTM-GAT (DTW graph)	166.63	0.5070	-5.10%	252.88%
BiLSTM-GAT (Sector graph)	145.89	0.4070	-4.51%	165.57%
BiLSTM-GNN (DTW graph)	157.90	0.4206	-5.59%	222.47%
BiLSTM-GNN (Sector graph)	147.43	0.4089	-4.29%	160.51%
S&P500	105.98	0.0777	-8.88%	14.50%

Bold items are best score

MSE over the BiLSTM-GAT with only the DTW graph underscores its better predictive capacity. However, it is the backtesting phase—where portfolio strategies based on these predictions are evaluated—that demonstrates the full potential of the BiLSTM-GAT-AM model. By consistently outperforming baseline models in portfolio management, the proposed model showcases its ability to translate prediction into more effective decision-making, delivering superior financial returns.

This underscores the importance of not only developing models that predict stock movements accurately but also focusing on their capacity to make profitable trading decisions. The dual-graph structure of the BiLSTM-GAT-AM model, supported by the attention mechanism that dynamically weights relevant relationships, provides a powerful framework for making these decisions, ultimately bridging the gap between prediction and actionable outcomes in stock market trading.

8 Conclusion and future work

8.1 Conclusion

In this study, the use of a dual-graph approach — incorporating both Dynamic Time Warping (DTW) and industry sector-based graphs — has proven to be an effective strategy in improving stock prediction and portfolio optimization. Notably, the DTW-based graph, which captures technical similarities by measuring the temporal alignment of stock price movements, has delivered superior results compared to the industry sector graph when considered individually. This observation highlights the value of technical analysis in detecting short-term correlations and subtle relationships between stocks based purely on their historical price behavior. The DTW graph's ability to connect stocks with highly similar price trends allows the model to leverage more precise and targeted insights, leading to better predictive accuracy and portfolio performance.

However, the industry sector graph, while not outperforming the DTW graph in isolation, should not be viewed as a detriment to the overall model performance. On the contrary, its contribution to the hybrid graph framework provides significant complementary benefits. The sector graph establishes connections between companies within the same industry, ensuring that every node (company) is linked to at least some others. This is in contrast to the DTW graph, where certain nodes may remain disconnected due to the lack of strong price movement similarity. The inclusion of the industry sector graph helps bridge these gaps by creating a more complete and interconnected structure, thereby ensuring that no company is entirely isolated from the graph.

This hybridization has a positive impact on the overall performance of the model. By combining both technical and fundamental perspectives, the hybrid graph capitalizes on the strengths of each approach. The DTW graph excels at capturing nuanced and short-term relationships, while the sector-based graph ensures that longer-term, industry-level connections are accounted for. The result is a more robust representation of the stock market's structure, which leads to improved predictions and portfolio outcomes. The sector-based graph's ability to link nodes that are otherwise disconnected in the DTW graph enhances the information flow across the network, allowing the model to make more informed and comprehensive decisions.

Moreover, the combination of these two graphs allows the Graph Attention Networks (GATs) to assign more contextually aware attention weights. In scenarios where technical similarities alone might not provide sufficient insight due to sparse connections in the DTW graph, the sector graph ensures that the model still has access to relevant information through the industry-based relationships. This integrated approach mitigates the risk of missing out on critical inter-stock relationships and creates a more reliable decision-making framework.

In summary, while the DTW graph has demonstrated better standalone performance, the industry sector graph plays a crucial role in enhancing the hybrid model. Its contribution to creating a fully connected network, especially in cases

where the DTW graph leaves certain nodes disconnected, ensures that the model can access both short-term price movements and long-term industry insights. This complementary relationship between the two graphs is key to the superior performance of the hybrid BiLSTM-GAT-AM model, underscoring the importance of leveraging both technical and fundamental analyses in stock market prediction and portfolio optimization.

8.2 Future work

While this research has laid a strong foundation through backtesting, future work should focus on conducting real-world market tests to evaluate the model's practical performance in live trading environments. Market conditions are often unpredictable, and backtesting results may not fully capture the complexities encountered in real-time trading, such as liquidity constraints, transaction costs, or market impact. To bridge this gap, it is crucial to deploy the model in a live trading setting and assess its performance over extended periods and diverse market conditions.

Additionally, future studies could explore incorporating more diverse forms of data, such as news sentiment, macroeconomic indicators, or even social media analytics, to enhance the model's understanding of external factors that influence stock prices. Further improvements might also involve testing different graph construction methods, including more sophisticated inter-company relationships or dynamic graphs that evolve based on real-time data. Expanding the graph representation could uncover deeper insights into stock behaviors and improve both the predictive accuracy and portfolio optimization strategy.

Finally, including more visual representations of these relationships in the form of detailed graphs and heatmaps could provide traders with more intuitive insights into how individual stocks interact, aiding in more informed trading decisions.

Acknowledgements No other individuals contributed to the research, and therefore no additional acknowledgements are necessary.

Author Contributions The first author was responsible for all aspects of the research, including literature review, coding, and manuscript preparation, while the other two authors supervised the work and provided guidance, mentoring and suggestions.

Funding This research did not receive any funding.

Data Availability The data used in this research were obtained from a publicly available dataset from Yahoo Finance.

Declarations

The authors declare that ethics approval and consent to participate were not applicable to this study, and consent for publication is also not applicable.

Competing Interests The authors report no financial or non-financial competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Gandhmal DP, Kumar K (2019) Systematic analysis and review of stock market prediction techniques. *Comput Sci Rev* 34:100190
2. Pang X, Zhou Y, Wang P, Lin W, Chang V (2020) An innovative neural network approach for stock market prediction. *J Supercomput* 76:2098–2118
3. Kumar D, Sarangi PK, Verma R (2022) A systematic review of stock market prediction using machine learning and statistical techniques. *Materials Today: Proceedings* 49:3187–3191
4. Thakkar A, Chaudhari K (2021) A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. *Expert Syst Appl* 177:114800
5. Bustos O, Pomares-Quimbaya A (2020) Stock market movement forecast: A systematic review. *Expert Syst Appl* 156:113464
6. Jiang W (2021) Applications of deep learning in stock market prediction: recent progress. *Expert Syst Appl* 184:115537
7. Ji X, Wang J, Yan Z (2021) A stock price prediction method based on deep learning technology. *Inter J Crowd Sci* 5(1):55–72
8. Rouf N, Malik MB, Arif T, Sharma S, Singh S, Aich S, Kim H-C (2021) Stock market prediction using machine learning techniques: a decade survey on methodologies, recent developments, and future directions. *Electronics* 10(21):2717
9. Roe MJ (2020) Missing the Target: Why Stock-market Short-termism Is Not the Problem. Oxford University Press, ???
10. Htun HH, Biehl M, Petkov N (2023) Survey of feature selection and extraction techniques for stock market prediction. *Finan Innov* 9(1):26
11. Liu H, Zhao T, Wang S, Li X (2023) A stock rank prediction method combining industry attributes and price data of stocks. *Inf Process Manag* 60(4):103358
12. Ma Y, Mao R, Lin Q, Wu P, Cambria E (2023) Multi-source aggregated classification for stock price movement prediction. *Inf Fusion* 91:515–528

13. Cagliero L, Fior J, Garza P (2023) Shortlisting machine learning-based stock trading recommendations using candlestick pattern recognition. *Expert Syst Appl* 216:119493
14. Cao S, Jiang W, Wang J, Yang B (2024) From man vs. machine to man+ machine: The art and ai of stock analyses. *J Finan Econ* 160:103910
15. Ma Y, Mao R, Lin Q, Wu P, Cambria E (2024) Quantitative stock portfolio optimization by multi-task learning risk and return. *Inf Fusion* 104:102165
16. Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY (2020) A comprehensive survey on graph neural networks. *IEEE Trans Neural Netw Learn Syst* 32(1):4–24
17. Cheng D, Yang F, Xiang S, Liu J (2022) Financial time series forecasting with multi-modality graph neural network. *Pattern Recogn* 121:108218
18. Xu B, Shen H, Sun B, An R, Cao Q, Cheng X (2021) Towards consumer loan fraud detection: Graph neural networks with role-constrained conditional random field. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 4537–4545
19. Hsu Y-L, Tsai Y-C, Li C-T (2021) Fingat: Financial graph attention networks for recommending top- k profitable stocks. *IEEE Trans Knowl Data Eng* 35(1):469–481
20. Han S, Huang H, Huang X, Li Y, Yu R, Zhang J (2024) Core patent forecasting based on graph neural networks with an application in stock markets. *Technol Anal Strat Manag* 36(8):1680–1694
21. Cheng R, Li Q (2021) Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol 35, pp 55–62
22. Liu C, Sun L, Ao X, Feng J, He Q, Yang H (2021) Intention-aware heterogeneous graph attention networks for fraud transactions detection. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp 3280–3288
23. Wang Y, Jing C, Xu S, Guo T (2022) Attention based spatiotemporal graph attention networks for traffic flow forecasting. *Inf Sci* 607:869–883
24. Nti IK, Adekoya AF, Weyori BA (2020) A systematic review of fundamental and technical analysis of stock market predictions. *Artif Intell Rev* 53(4):3007–3057
25. Rahman SM, Fatema K, Miah N (2022) Fundamental analysis of dhaka stock exchange (dse) listed top five nbfis: A study on bangladesh. *J Intern Bus Manag* 5(6):01–20
26. Singh A, Bhardwaj G, Srivastava AP, Bindra A, Chaudhary P et al (2022) Application of neural network to technical analysis of stock market prediction. In: *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*, pp 302–306. IEEE
27. Song G, Zhao T, Wang S, Wang H, Li X (2023) Stock ranking prediction using a graph aggregation network based on stock price and stock relationship information. *Inf Sci* 643:119236
28. Xu Y, Zhang Y (2023) Enhancement economic system based-graph neural network in stock classification. *IEEE Access* 11:17956–17967
29. Strader TJ, Rozycki JJ, Root TH, Huang Y-HJ (2020) Machine learning stock market prediction studies: review and research directions. *J Int Technol Inf Manag* 28(4):63–83
30. Lu X, Poon J, Khushi M (2024) Bridging the gap between machine and human in stock prediction: Addressing heterogeneity in stock market. *IEEE Access*
31. Hu Y, Ma X, Sui J, Pun M-O (2024) Ppmamba: A pyramid pooling local auxiliary ssm-based model for remote sensing image semantic segmentation. *arXiv preprint arXiv:2409.06309*
32. Neely CJ, Rapach DE, Tu J, Zhou G (2014) Forecasting the equity risk premium: the role of technical indicators. *Manage Sci* 60(7):1772–1791
33. Agrawal M, Khan AU, Shukla PK (2019) Stock price prediction using technical indicators: a predictive model using optimal deep learning. *Learning* 6(2):7
34. Chandar SK (2022) Convolutional neural network for stock trading using technical indicators. *Autom Softw Eng* 29(1):16
35. Salkar T, Shinde A, Tamhankar N, Bhagat N (2021) Algorithmic trading using technical indicators. In: *2021 International Conference on Communication Information and Computing Technology (ICCICT)*, pp 1–6. IEEE
36. Gong X, Ye X, Zhang W, Zhang Y (2023) Predicting energy futures high-frequency volatility using technical indicators: The role of interaction. *Energy Econ* 119:106533
37. Shah J, Vaidya D, Shah M (2022) A comprehensive review on multiple hybrid deep learning approaches for stock prediction. *Intell Syst Appl* 16:200111
38. Li X, Wu P, Wang W (2020) Incorporating stock prices and news sentiments for stock market prediction: A case of hong kong. *Inf Process Manag* 57(5):102212
39. Yildirim DC, Toroslu IH, Fiore U (2021) Forecasting directional movement of forex data using lstm with technical and macroeconomic indicators. *Finan Innov* 7:1–36
40. Alonso-Monsalve S, Suárez-Cetrulo AL, Cervantes A, Quintana D (2020) Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators. *Expert Syst Appl* 149:113250
41. Yu Y, Si X, Hu C, Zhang J (2019) A review of recurrent neural networks: Lstm cells and network architectures. *Neural Comput* 31(7):1235–1270
42. Ding G, Qin L (2020) Study on the prediction of stock price based on the associated network model of lstm. *Int J Mach Learn Cybern* 11(6):1307–1317
43. Bhandari HN, Rimal B, Pokhrel NR, Rimal R, Dahal KR, Khatri RK (2022) Predicting stock market index using lstm. *Mach Learn Appl* 9:100320
44. Mehtab S, Sen J, Dutta A (2021) Stock price prediction using machine learning and lstm-based deep learning models. In: *Machine Learning and Metaheuristics Algorithms, and Applications: Second Symposium, SoMMA 2020, Chennai, India, October 14–17, 2020, Revised Selected Papers 2*, pp 88–106. Springer
45. Jing N, Wu Z, Wang H (2021) A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction. *Expert Syst Appl* 178:115019
46. Shahvaroughi Farahani M, Razavi Hajiagha SH (2021) Forecasting stock price using integrated artificial neural network and metaheuristic algorithms compared to time series models. *Soft Comput* 25(13):8483–8513
47. 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. *Sci Program* 2021(1):2446543
48. Zhang C, Sjarif NNA, Ibrahim R (2024) Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022. *Wiley Interdiscip Rev: Data Min Knowl Discov* 14(1):1519
49. Wu L, Cui P, Pei J, Zhao L, Guo X (2022) Graph neural networks: foundation, frontiers and applications. In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp 4840–4841
50. Ke Q, Jing X, Woźniak M, Xu S, Liang Y, Zheng J (2024) Apgvae: Adaptive disentangled representation learning with the graph-based structure information. *Inf Sci* 657:119903
51. Liu M, Gao H, Ji S (2020) Towards deeper graph neural networks. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp 338–348
52. Gao C, Wang X, He X, Li Y (2022) Graph neural networks for recommender system. In: *Proceedings of the Fifteenth ACM*

- International Conference on Web Search and Data Mining, pp 1623–1625
53. Walters WP, Barzilay R (2020) Applications of deep learning in molecule generation and molecular property prediction. *Acc Chem Res* 54(2):263–270
 54. Li W, Bao R, Harimoto K, Chen D, Xu J, Su Q (2021) Modeling the stock relation with graph network for overnight stock movement prediction. In: *Proceedings of the Twenty-ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp 4541–4547
 55. Wu JM-T, Li Z, Herencsar N, Vo B, Lin JC-W (2023) A graph-based cnn-lstm stock price prediction algorithm with leading indicators. *Multimedia Syst* 29(3):1751–1770
 56. Yang S, Zhang Z, Zhou J, Wang Y, Sun W, Zhong X, Fang Y, Yu Q, Qi Y (2021) Financial risk analysis for smes with graph-based supply chain mining. In: *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp 4661–4667
 57. Li Q, Lin W, Liu Z, Prorok A (2021) Message-aware graph attention networks for large-scale multi-robot path planning. *IEEE Robot Autom Lett* 6(3):5533–5540
 58. Zhou W, Yue Y, Fang M, Qian X, Yang R, Yu L (2023) Bcnet: Bilateral cross-modal interaction network for indoor scene understanding in rgb-d images. *Inf Fusion* 94:32–42
 59. Zhou W, Dong S, Lei J, Yu L (2022) Mtanet: Multitask-aware network with hierarchical multimodal fusion for rgb-t urban scene understanding. *IEEE Trans Intel Vehicles* 8(1):48–58
 60. Zhou W, Lin X, Lei J, Yu L, Hwang J-N (2021) Mffnet: Multiscale feature fusion and enhancement network for rgb-thermal urban road scene parsing. *IEEE Trans Multimedia* 24:2526–2538
 61. Zhou W, Yang E, Lei J, Yu L (2022) Frnet: Feature reconstruction network for rgb-d indoor scene parsing. *IEEE J Sel Topics Signal Process* 16(4):677–687
 62. Gómez E, Rodríguez-Marciel C (2012) Pgdnet: a new problem-solving virtual learning environment. *Br J Edu Technol* 43(4):576–591
 63. Jung J-w, Heo H-S, Tak H, Shim H-j, Chung JS, Lee B-J, Yu H-J, Evans N (2022) Aassist: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp 6367–6371. IEEE
 64. Dong Y, Liu Q, Du B, Zhang L (2022) Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification. *IEEE Trans Image Process* 31:1559–1572
 65. Gao H, Xiao J, Yin Y, Liu T, Shi J (2022) A mutually supervised graph attention network for few-shot segmentation: The perspective of fully utilizing limited samples. *IEEE Trans Neural Netw Learn Syst* 35(4):4826–4838
 66. Ishiwatari T, Yasuda Y, Miyazaki T, Goto J (2020) Relation-aware graph attention networks with relational position encodings for emotion recognition in conversations. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp 7360–7370
 67. Wang Y, Wang H, Lu W, Yan Y (2023) Hygge: hyperbolic graph attention network for reasoning over knowledge graphs. *Inf Sci* 630:190–205
 68. Zhang Q, Zhang Y, Yao X, Li S, Zhang C, Liu P (2023) A dynamic attributes-driven graph attention network modeling on behavioral finance for stock prediction. *ACM Trans Knowl Discov Data* 18(1):1–29
 69. Zhang Q, Qin C, Zhang Y, Bao F, Zhang C, Liu P (2022) Transformer-based attention network for stock movement prediction. *Expert Syst Appl* 202:117239
 70. Bai X, Liu P, Zhang Y (2020) Investigating typed syntactic dependencies for targeted sentiment classification using graph attention neural network. *IEEE/ACM Trans Audio, Speech, Lang Process* 29:503–514
 71. Zhong Q, Liu Y, Ao X, Hu B, Feng J, Tang J, He Q (2020) Financial defaulter detection on online credit payment via multi-view attributed heterogeneous information network. In: *Proceedings of the Web Conference 2020*, pp 785–795
 72. Wang J, Liu G, Xu X, Xing X (2024) Credit risk prediction for small and medium enterprises utilizing adjacent enterprise data and a relational graph attention network. *J Manag Sci Eng* 9(2):177–192
 73. Lei Z, Zhang C, Xu Y, Li X (2024) Dr-gat: Dynamic routing graph attention network for stock recommendation. *Inf Sci* 654:119833
 74. Huang K, Li X, Xiong N, Yang Y (2024) Mf-dat: a stock trend prediction of the double-graph attention network based on multisource information fusion. *Multimed Syst* 30(3):136
 75. Liu H, Zhou Y, Zhou Y, Hu B (2024) Heterogeneous dual-dynamic attention network for modeling mutual interplay of stocks. *IEEE Trans Artif Intell*
 76. Xia H, Ao H, Li L, Liu Y, Liu S, Ye G, Chai H (2024) Ci-sthan: Pre-trained attention network for stock selection with channel-independent spatio-temporal hypergraph. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol 38, pp 9187–9195
 77. Yan Y, Zhang C, Li X, Zhang B (2024) A framework for stock selection via concept-oriented attention representation in hypergraph neural network. *Knowl-Based Syst* 284:111326
 78. Niu Z, Zhong G, Yu H (2021) A review on the attention mechanism of deep learning. *Neurocomputing* 452:48–62
 79. Guo M-H, Xu T-X, Liu J-J, Liu Z-N, Jiang P-T, Mu T-J, Zhang S-H, Martin RR, Cheng M-M, Hu S-M (2022) Attention mechanisms in computer vision: A survey. *Comput Vis Media* 8(3):331–368
 80. Soydaner D (2022) Attention mechanism in neural networks: where it comes and where it goes. *Neural Comput Appl* 34(16):13371–13385
 81. Fukui H, Hirakawa T, Yamashita T, Fujiyoshi H (2019) Attention branch network: Learning of attention mechanism for visual explanation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp 10705–10714
 82. Choi H, Cho K, Bengio Y (2018) Fine-grained attention mechanism for neural machine translation. *Neurocomputing* 284:171–176
 83. Yang, X (2020) An overview of the attention mechanisms in computer vision. In: *Journal of Physics: Conference Series*, vol 1693, p 012173. IOP Publishing
 84. Chen S, Ge L (2019) Exploring the attention mechanism in lstm-based hong kong stock price movement prediction. *Quant Fin* 19(9):1507–1515
 85. Qiu J, Wang B, Zhou C (2020) Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PLoS ONE* 15(1):0227222
 86. Li X, Wang J, Jia H, Xiao L (2022) Stock market volatility prediction method based on graph neural network with multi-attention mechanism. *J Comp Appl* 42(7):2265
 87. Li J, Xu C, Feng B, Zhao H (2023) Credit risk prediction model for listed companies based on cnn-lstm and attention mechanism. *Electronics* 12(7):1643
 88. Li C, Zhang X, Qaasar M, Ahmed S, Alam KMR, Morimoto Y (2019) Multi-factor based stock price prediction using hybrid neural networks with attention mechanism. In: *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech)*, pp 961–966. IEEE
 89. Cheng W, Chen S (2021) Sentiment analysis of financial texts based on attention mechanism of finbert and bilstm. In: *2021 International Conference on Computer Engineering and Application (ICCEA)*, pp 73–78. IEEE

90. Wang C, Han D, Liu Q, Luo S (2018) A deep learning approach for credit scoring of peer-to-peer lending using attention mechanism lstm. *Ieee Access* 7:2161–2168
91. Sun L, Xu W, Liu J (2021) Two-channel attention mechanism fusion model of stock price prediction based on cnn-lstm. *Trans Asian Low-Res Lang Inf Process* 20(5):1–12
92. Johnston F, Boyland JE, Meadows M, Shale E (1999) Some properties of a simple moving average when applied to forecasting a time series. *J Oper Res Soc* 50(12):1267–1271
93. Bollinger J (2002) *Bollinger on Bollinger Bands*. McGraw-Hill New York, ???
94. Gumparthy S et al (2017) Relative strength index for developing effective trading strategies in constructing optimal portfolio. *Int J Appl Eng Res* 12(19):8926–8936

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