IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 October 2024, accepted 27 November 2024, date of publication 5 December 2024, date of current version 18 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3511613

RESEARCH ARTICLE

Bridging the Gap Between Machine and Human in Stock Prediction: Addressing Heterogeneity in Stock Market

XIAOBIN LU [1], JOSIAH POON [1], AND MATLOOB KHUSHI [1,2]
[1]School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia
[2]Department of Computer Science, Brunel University of London, UB8 3PH Uxbridge, U.K.

Corresponding author: Matloob Khushi (matloob.khushi@brunel.ac.uk)

The work of Matloob Khushi was supported by UKRI NERC under Grant NE/X000192/12.

ABSTRACT Accurately predicting stock prices remains a formidable challenge in financial markets. Traditional predictive models often aggregate data from multiple companies, failing to account for the unique characteristics of each firm, which can hinder the model's ability to identify company-specific patterns. Moreover, existing research on stock price prediction frequently trains and tests models within the same group of companies, neglecting to assess their generalizability on 'Out-of-Sample' companies. This study addresses these limitations by employing BERT to encode business descriptions into vectors, capturing the distinctive attributes of each company. We further enhance the predictive modeling framework by developing features that describe the percentage change of existing indicators, adding significant novelty to the existing research. Additionally, we apply a Restricted Boltzmann Machine (RBM) for dimensionality reduction after the BERT encoding process. In our approach, both the technical indicators and the vectorized descriptions are treated as distinct elements within the transformer encoder. By integrating these representations, our model is better equipped to differentiate between firms and recognize their individual patterns. The proposed model demonstrates superior performance over baseline models, particularly when tested on 'Out-of-Sample' companies, highlighting its ability to learn, understand, and analyze company-specific descriptions for more accurate predictions. This research offers novel insights into addressing the heterogeneity in stock price prediction.

INDEX TERMS Bert, BiLSTM, financial markets, heterogeneity analysis, predictive modeling, restricted Boltzmann machine (RBM), stock prediction, technical indicators, textual data, transfer learning, transformer.

I. INTRODUCTION

The accurate prediction of stock prices has long been a critical focus in financial markets due to its significant implications for investment strategies and economic forecasting [1]. This area of research has seen substantial development, with numerous models and approaches being proposed over the years to enhance prediction accuracy [2]. Traditional statistical methods, such as time series analysis and econometric models, have been widely used in this domain [3]. However, the inherent complexity and volatility of financial markets often limit the predictive power of these conventional approaches.

In recent years, the advent of machine learning and deep learning techniques [4] has sparked renewed interest in stock price prediction [5]. These advanced methodologies offer the ability to learn from large volumes of data, uncovering intricate patterns and relationships that are not easily detectable by traditional models [6]. Among these approaches, the application of Natural Language Processing (NLP) and transformer models has gained considerable attention [7]. These models excel at understanding and

The associate editor coordinating the review of this manuscript and approving it for publication was Olarik Surinta.

processing textual data, enabling more nuanced analysis of market sentiment, news articles, and other relevant text-based inputs that influence stock prices [8].

Despite the progress made, much of the existing work in stock price prediction continues to aggregate data from multiple companies without differentiating between the unique characteristics of each firm [9]. This approach, while increasing the dataset size and potentially improving general model performance, tends to overlook the distinct patterns that are specific to individual companies. Although some research has attempted to address this issue by grouping similar companies through clustering techniques [10], these methods still fall short of fully capturing the unique business models, market positions, and defining attributes of individual companies.

To address this gap, we propose a novel approach that enhances stock price prediction by explicitly distinguishing between companies using vectorized business descriptions. Our method introduces a new way of integrating textual descriptions as static, descriptive vectors that encapsulate the specific characteristics of each firm, such as industry sector, geographic location, and market scale. These vectors are embedded into predictive modeling frameworks, enabling the model to differentiate between companies more effectively and learn from the unique patterns associated with each firm.

The core contributions of this study are twofold. First, we introduce a novel approach by incorporating textual business descriptions into the model, providing a richer and more detailed representation of each company. This allows the model to capture the inherent differences and similarities between firms, leading to more accurate predictions. Second, we rigorously test and compare the performance of the model on unseen companies with new descriptions, highlighting the model's ability to generalize and apply learned patterns to new, out-of-sample data. This underscores the potential of transfer learning in stock price prediction, demonstrating how the model can leverage learned knowledge to perform well on new companies with different characteristics.

This paper is organized as follows: Section II provides a comprehensive review of the relevant literature and previous research that form the foundation of our study. Section III offers an in-depth overview of the technical indicators, deep learning models, and Word2Vec techniques employed in our experiments. Section IV dives into the architecture of both the proposed model and the baseline model used in this study. Section V presents the experimental results and compares them with findings from existing literature. Section VI discusses the conclusions drawn from our research and suggests potential directions for future research on the heterogeneity of the stock market.

II. RELATED WORK

The prediction of stock prices has been a longstanding challenge in financial markets, drawing considerable attention from researchers and practitioners alike [11]. Over the years, various methodologies have been proposed and refined, ranging from traditional statistical techniques to advanced machine learning models [12].

A. TECHNICAL INDICATORS

Technical indicators, mathematical tools derived from historical price data such as previous open, high, low, close prices, and trading volumes, are essential for capturing trends, momentum, and other key aspects of market behavior. As summarized by [13], these indicators play a crucial role in price movement analysis by offering insights into various market dynamics. Widely utilized in deep learning models for stock price prediction, indicators like Moving Averages, Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD) provide valuable features that help these models identify relevant patterns in historical data [14]. Studies, including those by [15], [16], and [17], have shown that incorporating technical indicators into predictive models can enhance their accuracy. However, as noted by [5], the improvements achieved through these indicators alone are often marginal. Consequently, recent research has focused on developing more sophisticated models, such as those proposed by [18], [19], and [20], which integrate technical indicators with additional data sources and advanced methodologies to better capture the complex dynamics of financial markets, thereby achieving more substantial gains in prediction accuracy.

B. LONG SHORT-TERM MEMORY

Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN), are particularly well-suited for time series forecasting, including stock price prediction, due to their ability to learn and retain long-term dependencies in sequential data [21]. Research consistently demonstrates that LSTM models outperform traditional machine learning methods in capturing the complex temporal dynamics of stock prices. For instance, works such as [22], [23], and [24] indicate that LSTM models achieve superior predictive accuracy by effectively modeling the nonlinear relationships inherent in financial time series data. Furthermore, studies like [25], [26], and [27] highlight the advantages of hybrid models that incorporate LSTM units, demonstrating that these models yield outstanding results by combining the strengths of LSTM with other methods, such as convolutional neural networks (CNNs) or attention mechanisms. These hybrid approaches not only improve predictive performance but also enhance the model's robustness and generalization capabilities, making them highly effective in the volatile and complex domain of stock price prediction [28].

C. TEXT ENCODING

Text encoding is the process of converting textual data into a numerical format that can be understood and processed by machine learning models [7]. This transformation is crucial because models cannot directly interpret raw text; they require the text to be encoded into vectors or other numerical representations [29]. One of the most advanced methods for text encoding in recent years is the BERT (Bidirectional Encoder Representations from Transformers) model [30]. BERT, which utilizes the transformer architecture, has dramatically improved the understanding of context and semantics in text data [31]. It encodes text by considering the bidirectional context, meaning it takes into account both the preceding and following words in a sentence, allowing for a deeper and more nuanced understanding of language [32]. Studies such as [33], [34], and [35] have successfully employed BERT for stock price prediction, demonstrating that the model's ability to extract features from textual sources like news articles and financial reports significantly enhances predictive accuracy. These works have shown that using text encoding with BERT leads to more informed and accurate predictions in the volatile domain of financial markets.

D. FEATURE EXTRACTION

Feature extraction is a critical process in machine learning, particularly in the context of unsupervised learning, where models identify and learn patterns from data without labeled outputs [36]. Unsupervised learning techniques enable the discovery of hidden structures within the data, which can then be used to improve the performance of predictive models as presented by [37], [38], and [39]. One powerful approach to feature extraction in this domain is the use of Restricted Boltzmann Machines (RBMs) [40]. RBMs are stochastic neural networks designed to learn a probability distribution over input data, making them highly effective for unsupervised feature learning [41]. They are capable of uncovering latent patterns in complex datasets, which can then be utilized as informative inputs for downstream tasks [42]. Research has shown that unsupervised learning techniques, such as clustering, have been effectively applied in studies [43], [44], and [45], where they have been used to categorize financial data into meaningful groups. Additionally, works like [46], [47], and [48] have employed RBMs for feature extraction in financial data, demonstrating that these models can significantly enhance the accuracy and performance of stock price prediction models by capturing underlying patterns that other methods might miss.

E. TRANSFORMER

The Transformer model, introduced by [49], is a deep learning architecture that has revolutionized the field of Natural Language Processing (NLP) by enabling efficient handling of sequential data [50]. Unlike traditional models that process data sequentially, the Transformer uses self-attention mechanisms to capture the relationships between words in a sentence, regardless of their position [51]. This allows the model to process sequences in parallel, significantly improving computational efficiency and performance [52]. In the context of stock prediction, the Transformer model offers several advantages, particularly in its ability to capture complex dependencies within financial time series data [53]. By leveraging its self-attention mechanism, the Transformer can effectively model both short-term and long-term relationships in stock prices, leading to more accurate predictions.

Studies such as [54], [55], and [56] have successfully applied Transformer models to stock price prediction, demonstrating that these models outperform traditional methods by better capturing the intricate patterns and trends in financial data. These works highlight the Transformer's ability to enhance predictive accuracy and robustness, making it a powerful tool in the highly volatile and dynamic field of financial forecasting.

F. TIME-SERIES PREDICTION WITH DEEP LEARNING

Recent advancements in time series prediction and contrastive learning have led to the development of novel methods that improve both accuracy and interpretability. [57] introduced a dual-stage attention-based recurrent neural network (DA-RNN), which addresses the challenges of selecting relevant input features and capturing long-term dependencies by employing an input attention mechanism and a temporal attention mechanism. This model has shown superior prediction performance using datasets like SML 2010 and NASDAQ 100. Building on the theme of improving time series modeling, [58] presented AutoTCL, a parametric augmentation framework for contrastive learning that factorizes time series into informative and task-irrelevant parts, applying adaptive transformations to enhance both univariate and multivariate tasks. Similarly, [59] proposed InfoTS, a method that utilizes information-aware augmentations through a meta-learner network to select optimal augmentations, leading to improved performance in both forecasting and classification. Together, these approaches demonstrate significant advancements in time series forecasting and classification, improving accuracy and generalization while addressing the complexities inherent in temporal data.

Together, these advancements in technical indicators, deep learning models, and feature extraction techniques have pushed the boundaries of what is possible in stock price prediction. The integration of these methods offers a multifaceted approach to understanding and forecasting market movements, with each component contributing unique strengths to the predictive process.

III. METHODOLOGY

A. 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 as follows in Table 1.

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

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 [60].

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

$$\mathrm{SMA}_n(t) = \frac{1}{n}\sum_{i=0}^{n-1} P(t-i) \tag{1}$$

where:

- $\mathrm{SMA}_n(t)$ 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.

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 definition of high and low prices of a financial instrument [61].

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

$$\mathrm{MB}(t) = \mathrm{SMA}_{20}(t) = \frac{1}{20}\sum_{i=0}^{19} P(t-i) \tag{2}$$

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

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

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.

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

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

3) RELATIVE STRENGTH INDEX (RSI)

The Relative Strength Index (RSI) [62]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):

$$\mathrm{RS} = \frac{\text{Average Gain}}{\text{Average Loss}} \tag{5}$$

3) Calculate the RSI:

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

'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.

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.

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

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

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

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

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

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

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

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

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

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

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.

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

$$f12 = \left(\frac{\text{Close} - MV_{9}}{MV_{9}} \right) \times 100 \quad (18)$$

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

$$f14 = \left(\frac{MV_{9} - MV_{50}}{MV_{50}} \right) \times 100 \quad (20)$$

$$f15 = \left(\frac{MV_{9} - MV_{100}}{MV_{100}} \right) \times 100 \quad (21)$$

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

6) RSI PERCENTAGE CHANGE FEATURES

f17, f18' calculate the percentage difference between different RSI values (rsi14, rsi50, rsimv9). 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.

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

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

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.

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

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

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

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

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.

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

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

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

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

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

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

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.

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

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

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

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

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

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.

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

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

B. BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (BERT)

BERT is a pre-trained model that leverages the encoder component of the Transformer architecture, distinguishing itself from convolutional and recurrent neural networks. The core strength of BERT lies in the powerful Transformer encoder, which allows the model to be extended to considerable depths, thereby fully exploiting the properties of deep neural networks and enhancing model accuracy. The BERT model employs a multi-headed attention mechanism, where the input vector $X_a \in \mathbb{R}^k$ undergoes multiple linear transformations to generate different linear values, which are then input into the attention block to compute attention weights. This is mathematically represented as:

$$\text{Att}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{42}$$

where h denotes the number of heads in the attention mechanism. The final output of the multi-headed self-attention mechanism is derived by concatenating the outputs from each head and performing another linear transformation, as expressed by:

$$V_l = \text{Linear}\left(W_l \cdot \text{concat}(\text{Att}_1, \text{Att}_2, \ldots, \text{Att}_h) + b_l\right) \tag{43}$$

Here, Q, K, and V are the word-embedding representations of the input text, where $X_a = Q = K = V$. The similarity between words is calculated using dot product, followed by scaling by a factor $\frac{1}{\sqrt{d_k}}$ to prevent excessively large values that could adversely affect gradient backpropagation. This is then followed by the application of the softmax function to compute the attention weights, which are subsequently multiplied by V to yield the attentional output V_l.

After obtaining V_l through the multi-headed attention mechanism, a new vector $V_a = V_l + X_a$ is formed via a residual connection. This vector V_a is then normalized and passed through a feedforward network, and the final output of the Transformer is computed by applying another residual connection, as shown below:

$$V_t = \text{Feed}(W_f V_a + b_f) + V_a \tag{44}$$

In this equation, "Feed" denotes a linear function, and the final output of the Transformer is represented by V_t. The entire computation within the Transformer for any input vector X_a is encapsulated by the expression "Trans." The BERT model itself is constructed by stacking multiple Transformer layers, allowing it to model complex dependencies and capture deep contextual information within the text. This deep architecture enables BERT to excel in a wide range of natural language processing tasks.

C. RESTRICTED BOLTZMANN MACHINE (RBM)

Restricted Boltzmann Machines (RBMs) are stochastic neural networks that are particularly effective for unsupervised learning, allowing them to learn a probability distribution over a set of inputs. An RBM consists of two layers: a visible layer (representing the input data) and a hidden layer (capturing the underlying features). The connection between the visible and hidden layers is undirected, and there are no connections within a layer, making the architecture bipartite.

The energy function, which the RBM minimizes, is defined as:

$$E(v, h) = -\sum_i v_i a_i - \sum_j h_j b_j - \sum_{i,j} v_i W_{ij} h_j \tag{45}$$

where: v_i and h_j represent the binary states of the visible and hidden units, respectively, a_i and b_j are the biases associated with the visible and hidden units, W_{ij} represents the weight between visible unit i and hidden unit j.

The probability of a particular visible vector v is defined as:

$$P(v) = \frac{1}{Z} \sum_h e^{-E(v,h)} \tag{46}$$

where Z is the partition function, calculated as:

$$Z = \sum_{v,h} e^{-E(v,h)} \tag{47}$$

RBMs are typically trained using contrastive divergence, an efficient approximation to maximum likelihood learning. During training, the model updates the weights and biases to minimize the difference between the data distribution and the model distribution. The updates are computed as follows:

$$\Delta W_{ij} = \epsilon \left(\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}\right) \tag{48}$$

$$\Delta a_i = \epsilon \left(\langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{model}}\right) \tag{49}$$

$$\Delta b_j = \epsilon \left(\langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{model}}\right) \tag{50}$$

where ϵ is the learning rate, and $\langle \cdot \rangle_{\text{data}}$ and $\langle \cdot \rangle_{\text{model}}$ denote expectations under the data and model distributions, respectively.

Once trained, the RBM can be used to extract features by computing the hidden layer activations given the visible layer inputs. The hidden unit activations are calculated as:

$$P(h_j = 1 \mid v) = \text{sigmoid}\left(b_j + \sum_i v_i W_{ij}\right) \tag{51}$$

These learned features can then be used as inputs to downstream tasks, such as classification or regression models, significantly enhancing the model's ability to capture complex patterns in the data. RBMs have been successfully applied in various domains, including financial data analysis, where they have proven to be effective in feature extraction and improving prediction accuracy.

D. LONG SHORT-TERM MEMORY (LSTM)

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) that are particularly well-suited for modeling sequential data, such as time series or natural language. Unlike standard RNNs, which suffer from the vanishing gradient problem, LSTMs are designed to capture long-term dependencies by incorporating a memory cell that can maintain information across long sequences.

An LSTM cell consists of three gates: the input gate, the forget gate, and the output gate. These gates control the flow of information into and out of the memory cell. The equations governing the operation of an LSTM cell are as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (52)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (53)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (54)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (55)$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (56)$$
$$h_t = o_t * \tanh(C_t) \quad (57)$$

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

Here, σ represents the sigmoid function, and tanh represents the hyperbolic tangent function. The weight matrices W_f, W_i, W_C, W_o and the bias vectors b_f, b_i, b_C, b_o are learned during the training process.

LSTMs have been extensively used in various applications, including stock price prediction, where their ability to capture both short-term and long-term dependencies in financial time series data leads to more accurate predictions. For example, by maintaining a memory of past stock prices and other relevant financial indicators, LSTM networks can better forecast future trends compared to traditional models.

E. TRANSFORMER ENCODER

The Transformer encoder is a key component of the Transformer architecture, which has revolutionized natural language processing by allowing for efficient handling of sequential data. Unlike traditional models such as recurrent neural networks (RNNs), the Transformer encoder processes the entire sequence of data in parallel, enabling faster training and better capture of long-range dependencies.

The Transformer encoder consists of multiple layers, each composed of two main components: a multi-headed self-attention mechanism and a position-wise fully connected feedforward network. The self-attention mechanism allows the model to focus on different parts of the input sequence when encoding each element, while the feedforward network further transforms these representations.

1) MULTI-HEADED SELF-ATTENTION

The self-attention mechanism computes a weighted sum of input vectors, where the weights are determined by the similarity between different elements of the sequence. The equations for self-attention are as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \quad (58)$$

Here: Q (queries), K (keys), and V (values) are the input matrices obtained by linearly projecting the input sequence. d_k is the dimension of the key vectors. The dot product QK^T computes the similarity between the queries and keys. The Softmax function normalizes the result to obtain the attention weights.

In a multi-headed attention mechanism, this process is repeated multiple times (with different linear projections), allowing the model to focus on different parts of the sequence simultaneously:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \quad (59)$$

where each head head_i is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, K\ W_i^K, V\ W_i^V) \quad (60)$$

W_i^Q, W_i^K, W_i^V, and W^O are learned weight matrices.

2) POSITION-WISE FEEDFORWARD NETWORK

After the multi-headed self-attention mechanism, the output is passed through a fully connected feedforward network, which is applied identically to each position in the sequence:

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2 \quad (61)$$

Here, W_1 and W_2 are learned weight matrices, and b_1 and b_2 are bias vectors. The ReLU function introduces non-linearity, allowing the network to capture complex patterns.

3) LAYER NORMALIZATION AND RESIDUAL CONNECTIONS

Each sub-layer (self-attention and feedforward network) is followed by layer normalization and a residual connection, which helps stabilize training and allows the model to learn more effectively:

$$\text{Output} = \text{LayerNorm}(x + \text{SubLayer}(x)) \quad (62)$$

The output of the Transformer encoder is a set of encoded vectors, one for each input element, which can be used for various downstream tasks such as classification, translation, or sequence modeling. The ability of the Transformer encoder to process sequences in parallel and capture long-range dependencies has made it highly effective in tasks such as stock price prediction, where it can model complex temporal relationships in financial data.

F. PROPOSED MODEL

In this research, we propose a novel predictive model designed to forecast the next day's closing price of stocks, with a focus on differentiating between specific companies. The model architecture consists of two primary channels. The first channel processes the Open, High, Low, and Close (OHLC) data along with various generated technical

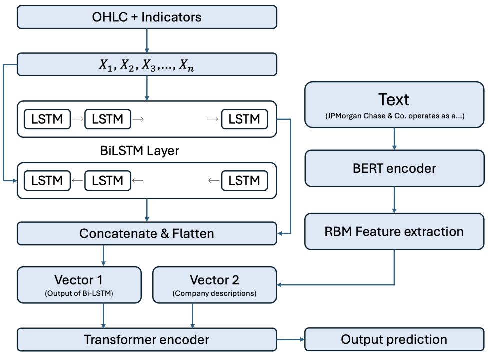

FIGURE 1. Proposed model structure.

indicators. These inputs are fed into a Bidirectional Long Short-Term Memory (BiLSTM) network, which is capable of capturing temporal dependencies and patterns within the time-series data. The second channel handles the vectorized long business descriptions of the companies, utilizing a Restricted Boltzmann Machine (RBM) for feature extraction.

The outputs from both channels are then analyzed using a Transformer model, which integrates the results to identify patterns specific to individual companies. This approach III-F allows the model to differentiate between companies and improve the accuracy of the stock price predictions.

G. EVALUATION METRICS

In this study, we aim to predict the next day's closing price of selected companies, framing the problem as a regression task. We will employ various models to perform the prediction, and the effectiveness of these models will be evaluated using the two key metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE). The formulas for these metrics are as follows:

- **Mean Squared Error (MSE):**

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{63}$$

- **Mean Absolute Error (MAE):**

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{64}$$

Here, y_i represents the actual value, $\hat{y}_i$ represents the predicted value, and n is the total number of data points. These metrics will allow us to comprehensively assess the performance of the models and their suitability for predicting financial time series data.

IV. EXPERIMENTAL SETUP

A. DATA SOURCE

The data utilized in this study were sourced from Yahoo Finance. For the first dataset, the companies selected were the 100 largest by market capitalization, spanning from Apple Inc. (AAPL) to European Metal Recycling Limited (EMR), as shown in Table 2. An additional dataset was compiled for companies ranked 101st to 170th by market capitalization, covering firms from O'Reilly Automotive (ORLY) to WEC Energy Group (WEC), as shown in Table 3. Both datasets consist of daily prices from January 1, 2020, to December 31, 2023.

TABLE 2. Dataset 1: Companies with no stock split among the Top 100 largest companies in US.

Industry	Companies	Count
Healthcare	UNH, LLY, JNJ, MRK, PFE, ABBV, AMGN, ABT, TMO, ISRG, SYK, GILD, BMY, ZTS, HCA, ELV, VRTX	17
Technology	AAPL, MSFT, AMZN, NVDA, GOOGL, GOOG, META, ORCL, ADBE, AMD, TXN, QCOM, CSCO, MU, CDNS, PLD	16
Consumer Staples	PG, KO, PEP, MDLZ, PM, CL, STZ, GIS, KMB, ADM	10
Finance	V, MA, JPM, MS, SPGI, AXP, BLK, SCHW, C	9
Consumer Discretionary	TSLA, HD, MCD, NKE, COST, TGT, GM, F, DG	9
Industrials	UPS, HON, RTX, CAT, LMT, UNP, DE, EMR, ITW	9
Energy	XOM, CVX, SLB, NEE, D, SO	6
Meterials	LIN, APD, ADM, WM	4
Utilities	NEE, D, SO	3
Total		83

In this study, we did not select companies that had undergone stock splits within the sample period to ensure the consistency and integrity of the dataset. Stock splits can introduce sudden, non-fundamental changes in stock prices, which could distort the true underlying patterns that the model aims to learn. By excluding companies with stock splits, we eliminate this potential source of noise, allowing the model to focus on capturing the genuine relationships between the company-specific characteristics and stock price movements. This approach helps to enhance the accuracy and reliability of the predictive model.

For the unseen companies, we selected companies ranked 101th to 170th in market share. After removing companies that underwent stock splits during the period, the remaining companies are as follows:

B. DATA PRE-PROCESSING

1) VECTORISED COMPANY DESCRIPTION

The long business descriptions of the companies were retrieved from Yahoo Finance, specifically from the 'Long Business Description' section under the category of 'asset_profile'. These descriptions were then vectorized using the BERT (Bidirectional Encoder Representations from Transformers) model. Subsequently, an independent Restricted Boltzmann Machine (RBM) was trained for

TABLE 3. Dataset 2: Companies with no stock split among the 101-170 largest companies in US.

Industry	Companies	Count
Finance	USB, TFC, AON, MET, STT, ICE, CME, PNC, EWBC, AFL, AIG	11
Healthcare	REGN, BDX, BSX, MRNA, DXCM, IDXX, MCK, CNC, BAX	9
Industrials	NSC, FDX, CTAS, ROP, TRV, TT, ROK	7
Consumer Discretionary	ORLY, APTV, SBUX, FDX, CHTR, LUV, DHIT	7
Energy	EOG, OXY, PSX, MPC, VLO, HAL, WMB	7
Materials	SHW, CTVA, DOW, PPG, LYB, GLW	6
Technology	ADI, PANW, IT, KLAC, HPQ, FTNT	6
Consumer staples	MO, KHC, WBA, KR, HSY	5
Real Estate	EQIX, PSA, SPG, WELL	4
Utilities	AEP, SRE, WEC	3
Insurance	PGR, ALL	2
Communication Services	WMG	1
Total		68

feature extraction, reducing the dimensionality of each vector to a length of 100.

2) SCALING

To ensure that all features contribute equally to the prediction model and to enhance the performance of the regression algorithms, data scaling will be applied as part of the preprocessing step. Specifically, we will use the *StandardScaler()* to standardize the features and the stock prices by removing the mean and scaling to unit variance. The transformation can be expressed as follows:

$$z_i = \frac{x_i - \mu}{\sigma} \tag{65}$$

where x_i is the original feature value, μ is the mean of the feature values, and σ is the standard deviation. This standardization process transforms the data to have a mean of 0 and a standard deviation of 1, ensuring that each feature is on the same scale. This step is crucial for models that rely on the assumption that the input data is normally distributed or models sensitive to the scale of the input features.

3) 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 in Figure 2. 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

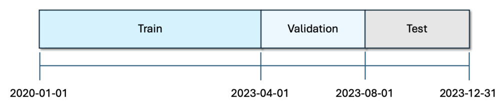

FIGURE 2. The split interval of train, validation, and test set.

during the scaling process. This approach helps maintain the integrity of the model evaluation.

C. 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.

V. EXPERIMENTAL RESULTS AND COMPARATIVE ANALYSIS

In this section, we will present the results in a comparative manner. First, we demonstrate that our proposed model, which incorporates additional company-specific features such as detailed business descriptions, outperforms the baseline model that does not include these features. We will then compare the performance of our model against several recent models from the existing literature.

Next, we assess the generalization capability of our model by testing a pre-trained version (trained on data from Companies 1-100) on a new set of companies (Companies 101-170) using the same data preprocessing steps. We will also train the baseline model on this new dataset (Companies 101-170) for comparison. Our proposed model continues to outperform the baseline, even when evaluated on unseen business descriptions from new companies. This indicates that our model not only differentiates between companies based on their descriptions but also generalizes the knowledge gained from previous descriptions, effectively transferring this understanding to new, unseen companies.

To ensure the robustness of our results, we conducted the experiments 20 times independently and calculated the average of the evaluation metrics for each model.

A. RESULTS ON 1-100 COMPANIES AND COMPARISON WITH THE SOTA MODELS

In this section, we explore the advantages of the proposed model by comparing its performance against six other methods, including a baseline model trained exclusively on the generated features. The baseline model is trained and validated on the same dataset, and the comparison is conducted using evaluation metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE).

The other four models represent state-of-the-art techniques in the domain of stock price prediction, including transformer and LSTM architectures. These models were chosen as baselines for several reasons. First, they represent the current state-of-the-art (SOTA) in predictive modeling, making them suitable benchmarks to evaluate the performance of the proposed model. Second, their underlying architectures share similarities with the proposed model, such as the use of deep learning components like LSTMs, GRUs, transformers, and hybrid structures. This allows for a fair comparison in terms of model structure and methodology. Additionally, these baseline models are all specifically designed for stock price prediction, making them highly relevant for assessing the effectiveness of the proposed approach in this specific application domain.

For instance, [51] integrates the Empirical Mode Decomposition (EMD) algorithm with LSTM, GRU, and transformer units to enhance feature extraction and capture complex patterns in stock data. Similarly, [63] employs a CNN-LSTM hybrid model, leveraging the strengths of convolutional layers for spatial feature extraction and LSTMs for sequential data processing. Reference [64] utilizes a deep attention network, which aligns closely with transformer-based models in emphasizing important temporal patterns in stock sequences. Lastly, [52] implements a hybrid LSTM model that combines multiple techniques to improve prediction accuracy. By selecting these models, the comparison not only demonstrates the novelty and effectiveness of the proposed model but also highlights its advantages when benchmarked against well-established and competitive approaches in the field detailed is shown in Table 4.

TABLE 4. Comparison of results including baseline model and existing literature on 1-100 (seen) companies.

Models	MSE	MAE
Proposed Model	0.0019	0.0272
BiLSTM with indicators only	0.0036	0.0430
DeepLOB [63]	0.0302	0.0945
DeepAtt [64]	0.0317	0.0932
FDG-Trans [51]	0.0291	0.0917
BiLSTM-MTRAN-TCN [52]	0.0143	0.0874

Figure 3 illustrates the prediction results from our proposed model for the seen companies (1)-100) on the first 6 trading days of the testing set, dated from 2023-08-01.

Figure 4 provided demonstrates the training process, showcasing the Mean Squared Error (MSE) and Mean Absolute Error (MAE) metrics for the training, and validation sets. The close alignment of these error metrics across all three sets suggests that the model is not overfitting. The absence of significant divergence between the training and validation errors indicates that the model is generalizing well to unseen data, as the errors remain consistent across different datasets. Thus, this plot provides strong evidence that the model maintains a balanced performance and does not exhibit signs of overfitting.

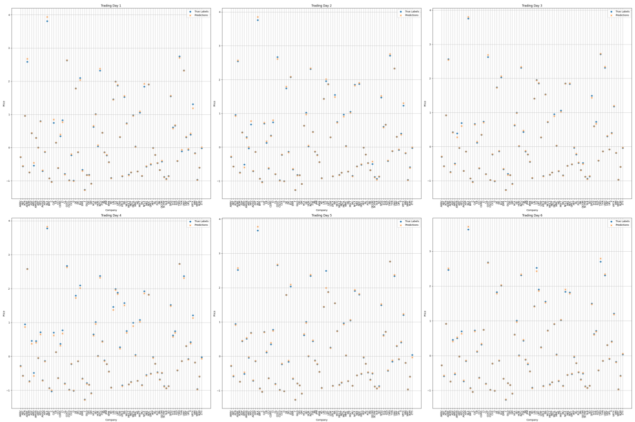

FIGURE 3. Results for the 1-100 (seen) companies on the first 6 trading days of the testing set, dated from 2023-08-01.

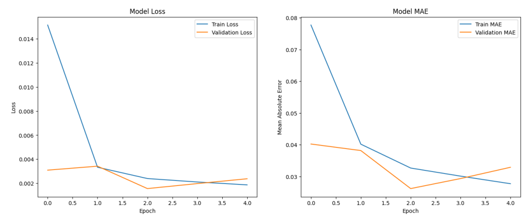

FIGURE 4. Results for the 1-100 (seen) companies on the first 6 trading days of the testing set, dated from 2023-08-01.

The MSE and MAE metrics for the training, validation, and test sets are presented in Table 5. With the implementation of early stopping during the training process, we believe that significant overfitting has been effectively mitigated.

TABLE 5. Comparison of training, validation, and test set of proposed model on 1-100 (seen) companies.

Daraset	MSE	MAE
Training	0.0014	0.0233
Validation	0.0032	0.0229
Test	0.0019	0.0272

Based on the sector-specific analysis, as is shown in Figure 5, we observe varying performance across different sectors as measured by the MSE (Mean Square Error). The analysis reveals that certain sectors, such as Consumer Staples and Utilities, tend to have lower MSE values, indicating that the model performs better in these industries where market dynamics might be more stable or predictable. On the other hand, sectors such as Consumer Discretionary and Healthcare exhibit relatively higher RMSE, which could be attributed to the more volatile nature of these sectors. This variability in performance suggests that while the model provides accurate predictions in some industries, its applicability may be less reliable in sectors characterized by rapid changes in external factors, such as geopolitical influences or fluctuating commodity prices. Hence, understanding these

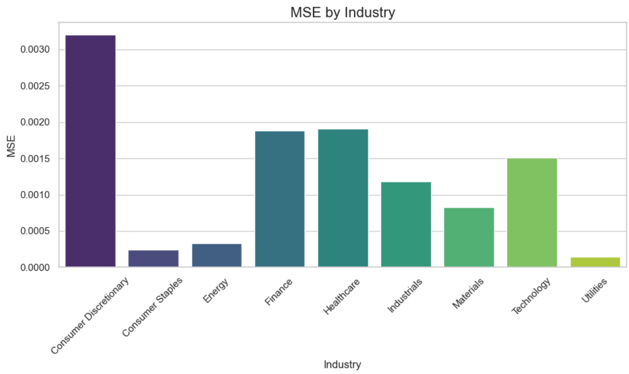

FIGURE 5. Results for the 1-100 (seen) companies group by industry sectors of the testing set, dated from 2023-08-01.

sectoral differences is crucial for refining the model and ensuring it is tailored to specific market environments.

B. RESULTS ON 101-170 (UNSEEN) COMPANIES

In this section, the proposed model was tested on a set of 70 additional companies (101-170) that were not seen during its training phase and compared with a baseline BiLSTM model, which was both trained and tested on the same set of companies (101-170). The results in Table 6 demonstrate that by incorporating vectorized features, our model can not only differentiate between different companies but also effectively learn from the provided descriptions and generalize this knowledge to unseen companies with new descriptions.

TABLE 6. Comparison of results with baseline model on 101-170 (unseen) companies.

Models	MSE	MAE	Time
Proposed Model	0.0014	0.0233	36s
BiLSTM with no additional features	0.0028	0.0351	45s

The proposed model utilizes an additional channel for vectorized descriptions and introduces a transformer encoder to integrate the outputs of both channels into a fully connected layer. While this architecture increases the model's size, it enhances the learning efficiency, allowing the model to converge faster and requiring fewer epochs for complete training. In contrast, the baseline model, although taking less time per epoch, requires significantly more epochs to achieve full training. As a result, the overall training time of the proposed model is shorter compared to the baseline model.

Figure 6 illustrates the prediction results from proposed model for the unseen companies on the first trading 6 days of the testing set, dated from 2023-08-01.

On the sector-specific analysis as is shown in Figure 7, we observe varying performance across different sectors as measured by the MSE (Mean Squared Error). The analysis reveals that certain sectors, such as Communication Services and Utilities, tend to have lower MSE values, indicating that the model performs better in these industries where market dynamics are relatively stable and predictable. Conversely,

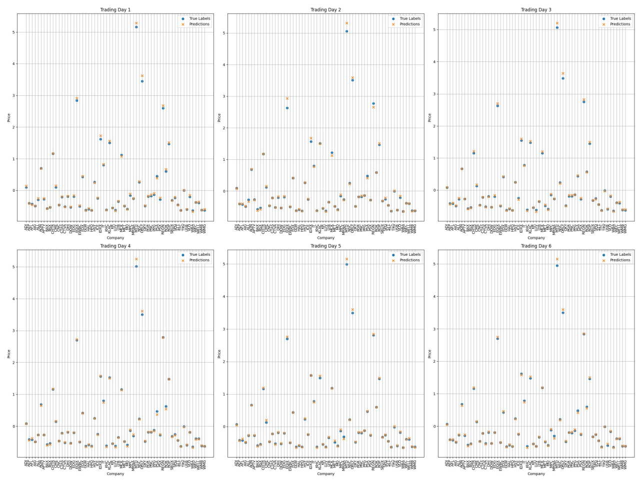

FIGURE 6. Results for the 101-170 (unseen) companies on the first 6 trading days of the testing set, dated from 2023-08-01.

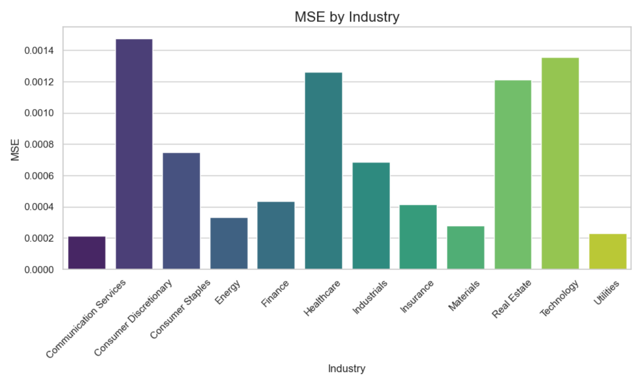

FIGURE 7. Results for the 101-170 companies group by industry sectors of the testing set, dated from 2023-08-01.

sectors like Consumer Discretionary and Technology exhibit higher MSE, which could be attributed to the more volatile and fast-changing nature of these markets. This variability in performance suggests that while the model demonstrates strong predictive accuracy in some industries, it may be less reliable in sectors influenced by external factors such as rapid technological advancements or fluctuating consumer preferences. Therefore, understanding these sector-specific differences is essential for further refinement of the model and its application across varying market environments.

VI. DISCUSSION OF RESULTS

In this section, we provide a detailed analysis of the results obtained from our proposed model, specifically focusing on its comparative performance with state-of-the-art (SOTA) stock prediction models and its efficacy in predicting stock prices for companies outside the training set.

A. COMPARATIVE PERFORMANCE WITH SOTA MODELS

The results of our experiments demonstrate that our proposed model, which incorporates vectorized company descriptions as static features, significantly outperforms existing SOTA models in stock price prediction tasks. Traditional models typically aggregate indicators from multiple companies into a unified training set without distinguishing the unique characteristics inherent to each company. This approach, while it increases the dataset's size, inherently overlooks the heterogeneity among companies, which can lead to sub-optimal prediction accuracy. Each company operates under unique market conditions and possesses distinct financial and operational characteristics, which influence its stock price movements. Aggregating all companies into a single dataset assumes a homogeneity that does not exist, resulting in a model that may fail to capture the nuanced patterns that are specific to individual companies.

Our model addresses this critical limitation by integrating company-specific descriptions into the predictive modeling framework. These descriptions, encoded into vectors using advanced natural language processing techniques, allow the model to recognize and differentiate between companies, much like how an experienced human trader would consider both quantitative data (e.g., price history, trading volumes) and qualitative information (e.g., company fundamentals, market position) before making trading decisions. The incorporation of these descriptive vectors enables the model to identify distinct patterns within each company's data, leading to more accurate and reliable stock price predictions. This is particularly crucial in financial markets where the same market event can have varying effects on different companies, depending on their individual characteristics.

B. GENERALIZATION TO UNSEEN COMPANIES

To further validate the robustness and generalizability of our model, we conducted experiments on a dataset comprising companies ranked 101st to 170th by market capitalization, which were not included in the training set. This experiment aimed to assess the model's ability to apply learned pattern from seen companies to predict the stock prices of unseen companies based on their descriptive vectors and technical indicators.

The results were promising and indicated that the model successfully generalized the patterns and descriptions learned from the training companies to unseen companies. This was evidenced by the model's ability to analysis unseen companies based on their business descriptions together with technical indicators and predict their price movements with a higher degree of accuracy. The model achieved this by leveraging the underlying similarities between the vectorized descriptions of unseen companies and those of companies in the training set. For instance, companies within the same sector often share common market dynamics and risk factors, which influence their stock price movements in similar ways. Our model was able to capture these specific patterns and apply them effectively to unseen and similar companies, thereby supporting the hypothesis that companies with similar characteristics tend to exhibit similar price behaviors.

Moreover, the success of our model in predicting the stock prices of unseen companies highlights its potential for practical applications in real-world trading scenarios. The ability to generalize from a known set of companies to new companies without requiring retraining on the entire dataset presents a significant advantage in dynamic financial markets where new companies frequently emerge and existing companies undergo changes.

C. TIME COMPLEXITY ANALYSIS AND EXPERIMENTAL STUDIES

In this section, we analyze the time complexity and discuss the experimental results comparing the proposed model with the baseline model in terms of training efficiency. The proposed model introduces an additional channel for vectorized descriptions and employs a transformer encoder to fuse the outputs from the two channels into a fully connected layer. While this architectural enhancement increases the size and complexity of the model, it leads to a faster learning process.

From a time complexity perspective, the use of a transformer encoder, despite adding computational overhead, significantly improves the model's ability to capture complex patterns in the data. This results in faster convergence, allowing the proposed model to be fully trained in fewer epochs. The transformer encoder's self-attention mechanism efficiently captures relationships across input sequences, leading to more robust feature representations.

In contrast, the baseline model, while simpler and requiring less time per epoch, lacks the enhanced learning capabilities provided by the transformer. Consequently, it requires a greater number of epochs to achieve comparable performance. Although the baseline model benefits from lower per-epoch computational costs, the overall training time is extended due to the larger number of epochs required for convergence.

Our experimental studies corroborate this analysis. The proposed model, despite the higher per-epoch complexity, exhibited faster convergence, resulting in a shorter total training time compared to the baseline model. This demonstrates the trade-off between per-epoch time complexity and the total number of epochs required for training. Ultimately, the proposed model offers a more efficient training process, achieving superior performance with fewer epochs, thus reducing the overall computational cost in practice.

D. IMPLICATIONS FOR FINANCIAL MODELING

The implications of our findings are multifaceted. Firstly, the integration of company-specific descriptive vectors represents a significant advancement in stock price prediction models, particularly in addressing the issue of heterogeneity among companies. By moving away from a one-size-fits-all approach and towards a more nuanced model that accounts for individual company characteristics, our approach offers a more precise tool for financial forecasting. Secondly, the model's ability to generalize from known to unknown entities opens up new possibilities for the application of transfer learning in financial markets. This approach not only enhances prediction accuracy but also reduces the need for frequent retraining, thus offering a more efficient and scalable solution for stock price prediction.

E. ADDRESSING THE LIMITATION OF COMPARATIVE ANALYSIS

A key limitation of this study is the comparative analysis between our proposed model and other state-of-the-art (SOTA) stock prediction models. Although our research highlights the benefits of integrating vectorized company descriptions into the predictive framework, the comparison with existing models was not as comprehensive as it could have been. This limitation might raise questions regarding the relative superiority of our model.

There are several factors contributing to this challenge. Firstly, each SOTA model in the field of stock prediction typically employs its own set of technical indicators and methodologies. These unique characteristics make it difficult to establish a direct and fair comparison across models. The diversity in the indicators used means that models are often optimized for specific tasks or datasets, which complicates attempts to compare their performance on a more general level.

Secondly, SOTA models are frequently trained and tested on different datasets, drawn from various markets, time periods, and companies. The lack of a standardized dataset for stock price prediction research means that models are often evaluated under different conditions. This variation is inherent to financial prediction, where market dynamics can differ significantly depending on the time period, geographic region, and sector of the companies involved. Consequently, it is challenging to compare models directly, as they may have been optimized for distinct market environments or temporal contexts.

In practice, researchers typically present their results using metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE) to evaluate and compare the performance of their models. However, these metrics alone do not fully account for the variations in market conditions and datasets, making it difficult to draw definitive conclusions about one model's superiority over another.

Despite these challenges, our research has demonstrated the potential of our proposed model to outperform a selected set of baseline models, particularly in its ability to generalize to unseen companies. However, we acknowledge that a more extensive and standardized comparison with a broader range of SOTA models could have provided a stronger validation of our model's performance. Future research should aim to address this limitation by utilizing more standardized datasets or developing methodologies that allow for fairer comparisons across different models and market conditions.

While our study makes a valuable contribution to the field of stock price prediction, the inherent challenges of comparing models with different technical indicators and datasets must be recognized. Addressing these challenges in future work will be essential to provide a more robust and convincing case for the adoption of our model in financial prediction tasks.

VII. CONCLUSION AND FUTURE WORK

A. CONCLUSION

This research presents a significant advancement in the field of stock price prediction by introducing a model that integrates vectorized company descriptions as static features. Unlike traditional state-of-the-art (SOTA) models, which often rely on aggregated data from multiple companies, our approach acknowledges the unique characteristics and heterogeneity inherent in different firms. This recognition of company-specific attributes represents a substantial departure from existing methods, which tend to treat all companies as if they operate under similar conditions. By incorporating descriptive vectors into the predictive framework, our model is able to discern and leverage the distinct patterns that are specific to each company, leading to more accurate and reliable stock price predictions.

The novelty of our approach lies in its ability to simulate the decision-making process of human traders, who consider both quantitative data, such as historical prices, and qualitative information, such as a company's market position and business model. This holistic view allows the model to capture a broader range of factors that influence stock prices, offering a more comprehensive and nuanced prediction tool. The success of our model in outperforming existing SOTA models highlights the importance of addressing the heterogeneity in stock markets, a factor that has been largely overlooked in previous research.

In summary, our research contributes a novel and effective method to the existing body of work on stock price prediction, addressing critical gaps in current SOTA models by incorporating company-specific features. This advancement not only improves prediction accuracy but also opens new avenues for integrating qualitative data into predictive models, paving the way for future research and practical applications in finance.

B. FUTURE WORK

Building on the promising results of this study, several avenues for future research are proposed to further enhance the model's robustness, generalizability, and applicability across diverse financial contexts. One significant direction for future work involves expanding the scope of the model to include a more varied set of companies. By testing the model on companies with diverse descriptions from a wide array of sectors, researchers can evaluate its performance across different industries and market environments. This would help ascertain whether the model's ability to capture and leverage company-specific patterns holds consistently across various sectors, including those that may have different market dynamics or are subject to different regulatory environments.

Another potential area for future exploration is the application of the model to different geographic markets. Financial markets in different regions are often influenced by distinct economic conditions, cultural factors, and regulatory frameworks. Testing the model in diverse markets, such as emerging markets or markets with different levels of liquidity and volatility, could provide valuable insights into its adaptability and effectiveness in predicting stock prices under various economic conditions. This would not only contribute to the model's generalizability but also enhance its utility for global financial forecasting.

Additionally, future research could investigate the integration of other qualitative features into the model, such as news sentiment, social media trends, and macroeconomic indicators. These factors can have significant impacts on stock prices, and their inclusion could further improve the model's predictive accuracy. For instance, incorporating real-time sentiment analysis from financial news or social media platforms could enable the model to react more quickly to market-moving events, providing more timely and accurate predictions.

In conclusion, while this study makes a significant contribution to the field of stock price prediction, the potential for further advancements is vast. By exploring these proposed avenues of future research, the model's capabilities can be extended and refined, leading to even more accurate, adaptable, and generalizable tools for financial forecasting.

DATA ACCESS

The data for this study has been sourced from the public repository Yahoo Finance. Related code can be found at: https://github.com/xilu5047/Stock_prediction

REFERENCES

[1] Z. Hu, Y. Zhao, and M. Khushi, "A survey of forex and stock price prediction using deep learning," *Appl. Syst. Innov.*, vol. 4, no. 1, p. 9, Feb. 2021.

[2] W. Jiang, "Applications of deep learning in stock market prediction: Recent progress," *Expert Syst. Appl.*, vol. 184, Dec. 2021, Art. no. 115537.

[3] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, and S. Shahab, "Deep learning for stock market prediction," *Entropy*, vol. 22, no. 8, p. 840, Jul. 2020.

[4] Y. Hu, X. Ma, J. Sui, and M.-O. Pun, "PPMamba: A pyramid pooling local auxiliary SSM-based model for remote sensing image semantic segmentation," 2024, *arXiv:2409.06309*.

[5] J. Shah, D. Vaidya, and M. Shah, "A comprehensive review on multiple hybrid deep learning approaches for stock prediction," *Intell. Syst. Appl.*, vol. 16, Nov. 2022, Art. no. 200111.

[6] M. Nikou, G. Mansourfar, and J. Bagherzadeh, "Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms," *Intell. Syst. Accounting, Finance Manage.*, vol. 26, no. 4, pp. 164–174, Oct. 2019.

[7] Y. Kang, Z. Cai, C.-W. Tan, Q. Huang, and H. Liu, "Natural language processing (NLP) in management research: A literature review," *J. Manage. Anal.*, vol. 7, no. 2, pp. 139–172, Apr. 2020.

[8] M. Li, W. Li, F. Wang, X. Jia, and G. Rui, "Applying BERT to analyze investor sentiment in stock market," *Neural Comput. Appl.*, vol. 33, no. 10, pp. 4663–4676, May 2021.

[9] P. Soni, Y. Tewari, and D. Krishnan, "Machine learning approaches in stock price prediction: A systematic review," *J. Phys., Conf. Ser.*, vol. 2161, no. 1, Jan. 2022, Art. no. 012065.

[10] G. Marti, F. Nielsen, M. Bińkowski, and P. Donnat, "A review of two decades of correlations, hierarchies, networks and clustering in financial markets," in *Progress in Information Geometry*, Jan. 2021, pp. 245–274.

[11] T. J. Strader, J. J. Rozycki, T. H. Root, and Y.-H.-J. Huang, "Machine learning stock market prediction studies: Review and research directions," *J. Int. Technol. Inf. Manage.*, vol. 28, no. 4, pp. 63–83, Jan. 2020.

[12] I. K. Nti, A. F. Adekoya, and B. A. Weyori, "A systematic review of fundamental and technical analysis of stock market predictions," *Artif. Intell. Rev.*, vol. 53, no. 4, pp. 3007–3057, Apr. 2020.

[13] C. J. Neely, D. E. Rapach, J. Tu, and G. Zhou, "Forecasting the equity risk premium: The role of technical indicators," *Manage. Sci.*, vol. 60, no. 7, pp. 1772–1791, Jul. 2014.

[14] M. Agrawal, A. U. Khan, and P. K. Shukla, "Stock price prediction using technical indicators: A predictive model using optimal deep learning," *Learning*, vol. 6, no. 2, p. 7, 2019.

[15] S. K. Chandar, "Convolutional neural network for stock trading using technical indicators," *Automated Softw. Eng.*, vol. 29, no. 1, p. 16, May 2022.

[16] T. Salkar, A. Shinde, N. Tamhankar, and N. Bhagat, "Algorithmic trading using technical indicators," in *Proc. Int. Conf. Commun. Inf. Comput. Technol. (ICCICT)*, Jun. 2021, pp. 1–6.

[17] X. Gong, X. Ye, W. Zhang, and Y. Zhang, "Predicting energy futures high-frequency volatility using technical indicators: The role of interaction," *Energy Econ.*, vol. 119, Mar. 2023, Art. no. 106533.

[18] X. Li, P. Wu, and W. Wang, "Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong," *Inf. Process. Manage.*, vol. 57, no. 5, Sep. 2020, Art. no. 102212.

[19] D. C. Yıldırım, I. H. Toroslu, and U. Fiore, "Forecasting directional movement of forex data using LSTM with technical and macroeconomic indicators," *Financial Innov.*, vol. 7, no. 1, pp. 1–36, Dec. 2021.

[20] S. Alonso-Monsalve, A. L. Suárez-Cetrulo, A. Cervantes, and D. Quintana, "Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators," *Expert Syst. Appl.*, vol. 149, Jul. 2020, Art. no. 113250.

[21] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019.

[22] G. Ding and L. Qin, "Study on the prediction of stock price based on the associated network model of LSTM," *Int. J. Mach. Learn. Cybern.*, vol. 11, no. 6, pp. 1307–1317, Jun. 2020.

[23] H. N. Bhandari, B. Rimal, N. R. Pokhrel, R. Rimal, K. R. Dahal, and R. K. C. Khatri, "Predicting stock market index using LSTM," *Mach. Learn. Appl.*, vol. 9, Sep. 2022, Art. no. 100320.

[24] S. Mehtab, J. Sen, and A. Dutta, "Stock price prediction using machine learning and LSTM-based deep learning models," in *Proc. 2nd Symp. Mach. Learn. Metaheuristics Algorithms, Appl.*, Chennai, India. Cham, Switzerland: Springer, 2020, pp. 88–106.

[25] N. Jing, Z. Wu, and H. Wang, "A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction," *Expert Syst. Appl.*, vol. 178, Sep. 2021, Art. no. 115019.

[26] M. S. Farahani and S. H. R. Hajiagha, "Forecasting stock price using integrated artificial neural network and Metaheuristic algorithms compared to time series models," *Soft Comput.*, vol. 25, no. 13, pp. 8483–8513, Jul. 2021.

[27] Y. Chen, R. Fang, T. Liang, Z. Sha, S. Li, Y. Yi, W. Zhou, and H. Song, "Stock price forecast based on CNN-BiLSTM-ECA model," *Sci. Program.*, vol. 2021, pp. 1–20, Jul. 2021.

[28] C. Zhang, N. N. A. Sjarif, and R. Ibrahim, "Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022," *WIREs Data Mining Knowl. Discovery*, vol. 14, no. 1, p. 1519, Jan. 2024.

[29] M. Jaggi, P. Mandal, S. Narang, U. Naseem, and M. Khushi, "Text mining of stocktwits data for predicting stock prices," *Appl. Syst. Innov.*, vol. 4, no. 1, p. 13, Feb. 2021.

[30] M. Müller, M. Salathé, and P. E. Kummervold, "COVID-Twitter-BERT: A natural language processing model to analyse COVID-19 content on Twitter," *Frontiers Artif. Intell.*, vol. 6, Mar. 2023, Art. no. 1023281.

[31] M. V. Koroteev, "BERT: A review of applications in natural language processing and understanding," 2021, *arXiv:2103.11943*.

[32] W. Li, R. Bao, K. Harimoto, D. Chen, J. Xu, and Q. Su, "Modeling the stock relation with graph network for overnight stock movement prediction," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 4541–4547.

[33] F. Zhao, X. Li, Y. Gao, Y. Li, Z. Feng, and C. Zhang, ‘‘Multi-layer features ablation of BERT model and its application in stock trend prediction,’’ *Expert Syst. Appl.*, vol. 207, Nov. 2022, Art. no. 117958.

[34] X. Weng, X. Lin, and S. Zhao, ‘‘Stock price prediction based on lstm and bert,’’ in *Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC)*, Sep. 2022, pp. 12–17.

[35] S. Mittal, A. Chauhan, and C. Nagpal, ‘‘Stock market prediction by incorporating news sentiments using bert,’’ in *Modern Approaches in Machine Learning & Cognitive Science: A Walkthrough*. Cham, Switzerland: Springer, 2022, pp. 35–45.

[36] A. Decelle, B. Seoane, and L. Rosset, ‘‘Unsupervised hierarchical clustering using the learning dynamics of restricted Boltzmann machines,’’ *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 108, no. 1, Jul. 2023, Art. no. 014110.

[37] A. Glielmo, B. E. Husic, A. Rodriguez, C. Clementi, F. Noé, and A. Laio, ‘‘Unsupervised learning methods for molecular simulation data,’’ *Chem. Rev.*, vol. 121, no. 16, pp. 9722–9758, Aug. 2021.

[38] P. O. Pinheiro, A. Almahairi, R. Y. Benmalek, F. Golemo, and A. Courville, ‘‘Unsupervised learning of dense visual representations,’’ in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2020, pp. 4489–4500.

[39] Y. Du, S. Li, Y. Sharma, J. B. Tenenbaum, and I. Mordatch, ‘‘Unsupervised learning of compositional energy concepts,’’ in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2021, pp. 15608–15620.

[40] A. Decelle and C. Furtlehner, ‘‘Restricted Boltzmann machine: Recent advances and mean-field theory,’’ *Chin. Phys. B*, vol. 30, no. 4, Apr. 2021, Art. no. 040202.

[41] A. Ibrahim, E.-S.-M. El-Kenawy, M. M. Eid, A. A. Abdelhamid, M. El-Said, A. H. Alharbi, D. S. Khafaga, W. A. Awad, R. Y. Rizk, N. Bailek, and M. A. Saeed, ‘‘A recommendation system for electric vehicles users based on restricted Boltzmann machine and WaterWheel plant algorithms,’’ *IEEE Access*, vol. 11, pp. 145111–145136, 2023.

[42] V. Upadhya and P. S. Sastry, ‘‘An overview of restricted Boltzmann machines,’’ *J. Indian Inst. Sci.*, vol. 99, no. 2, pp. 225–236, Jun. 2019.

[43] T. Leangarun, P. Tangamchit, and S. Thajchayapong, ‘‘Stock price manipulation detection using deep unsupervised learning: The case of Thailand,’’ *IEEE Access*, vol. 9, pp. 106824–106838, 2021.

[44] J. Wang and S. Zhu, ‘‘A multi-factor two-stage deep integration model for stock price prediction based on intelligent optimization and feature clustering,’’ *Artif. Intell. Rev.*, vol. 56, no. 7, pp. 7237–7262, Jul. 2023.

[45] H. Liu, T. Zhao, S. Wang, and X. Li, ‘‘A stock rank prediction method combining industry attributes and price data of stocks,’’ *Inf. Process. Manage.*, vol. 60, no. 4, Jul. 2023, Art. no. 103358.

[46] R. Devrani, P. Mahajan, and D. Kumar, ‘‘The smart predictive analysis of stock market performance using restricted Boltzmann machines,’’ in *Proc. Int. Conf. Optim. Comput. Wireless Commun. (ICOCWC)*, Jan. 2024, pp. 1–6.

[47] I. Prima and D. Ahmad, ‘‘Analisis conditional restricted boltzman machine untuk memprediksi Harga saham Bank syariah Indonesia,’’ *Jurnal Lebesgue, Jurnal Ilmiah Pendidikan Matematika, Matematika dan Statistika*, vol. 4, no. 1, pp. 409–416, Apr. 2023.

[48] X. Zhang, N. Gu, J. Chang, and H. Ye, ‘‘Predicting stock price movement using a DBN-RNN,’’ *Appl. Artif. Intell.*, vol. 35, no. 12, pp. 876–892, Oct. 2021.

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, ‘‘Attention is all you need,’’ 2017, *arXiv:1706.03762*.

[50] Q. Zhang, C. Qin, Y. Zhang, F. Bao, C. Zhang, and P. Liu, ‘‘Transformer-based attention network for stock movement prediction,’’ *Expert Syst. Appl.*, vol. 202, Sep. 2022, Art. no. 117239.

[51] C. Li and G. Qian, ‘‘Stock price prediction using a frequency decomposition based GRU transformer neural network,’’ *Appl. Sci.*, vol. 13, no. 1, p. 222, Dec. 2022.

[52] S. Wang, ‘‘A stock price prediction method based on BiLSTM and improved transformer,’’ *IEEE Access*, vol. 11, pp. 104211–104223, 2023.

[53] X. Hu, ‘‘Stock price prediction based on temporal fusion transformer,’’ in *Proc. 3rd Int. Conf. Mach. Learn., Big Data Bus. Intell. (MLBDBI)*, Dec. 2021, pp. 60–66.

[54] D. Daiya and C. Lin, ‘‘Stock movement prediction and portfolio management via multimodal learning with transformer,’’ in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 3305–3309.

[55] A. Bozanta, S. Angco, M. Cevik, and A. Basar, ‘‘Sentiment analysis of StockTwits using transformer models,’’ in *Proc. 20th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2021, pp. 1253–1258.

[56] A. T. Haryono, R. Sarno, and K. R. Sungkono, ‘‘Transformer-gated recurrent unit method for predicting stock price based on news sentiments and technical indicators,’’ *IEEE Access*, vol. 11, pp. 77132–77146, 2023.

[57] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, ‘‘A dual-stage attention-based recurrent neural network for time series prediction,’’ 2017, *arXiv:1704.02971*.

[58] X. Zheng, T. Wang, W. Cheng, A. Ma, H. Chen, M. Sha, and D. Luo, ‘‘Parametric augmentation for time series contrastive learning,’’ 2024, *arXiv:2402.10434*.

[59] D. Luo, W. Cheng, Y. Wang, D. Xu, J. Ni, W. Yu, X. Zhang, Y. Liu, Y. Chen, and H. Chen, ‘‘Time series contrastive learning with information-aware augmentations,’’ in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, vol. 37, no. 4, pp. 4534–4542.

[60] F. R. Johnston, J. E. Boyland, M. Meadows, and E. Shale, ‘‘Some properties of a simple moving average when applied to forecasting a time series,’’ *J. Oper. Res. Soc.*, vol. 50, no. 12, p. 1267, Dec. 1999.

[61] J. Bollinger, *Bollinger on Bollinger Bands*. New York, NY, USA: McGraw-Hill, 2002.

[62] S. Gumparthi, ‘‘Relative strength index for developing effective trading strategies in constructing optimal portfolio,’’ *Int. J. Appl. Eng. Res.*, vol. 12, no. 19, pp. 8926–8936, 2017.

[63] Z. Zhang, S. Zohren, and S. Roberts, ‘‘DeepLOB: Deep convolutional neural networks for limit order books,’’ *IEEE Trans. Signal Process.*, vol. 67, no. 11, pp. 3001–3012, Jun. 2019.

[64] A. Arévalo, J. Niño, G. Hernández, and J. Sandoval, ‘‘High-frequency trading strategy based on deep neural networks,’’ in *Proc. Int. Conf. Intell. Comput.* Cham, Switzerland: Springer, Jan. 2016, pp. 424–436.

XIAOBIN LU received the B.S. degree in financial statistics from Chinese University of Hong Kong, Shenzhen, in 2020, and the M.S. degree in data science from The University of Sydney, in 2024, where he is currently pursuing the M.S. degree in philosophy (engineering). His research interests include machine intelligence applied to forex trading, stock market prediction, and the development of deep learning models for time-series forecasting.

JOSIAH POON is currently a Senior Lecturer with the School of Computer Science, The University of Sydney, with an extensive background in data analytics and artificial intelligence. His research interests include data analytics, AI, and their applications in healthcare.

MATLOOB KHUSHI is currently an Associate Professor of artificial intelligence (AI) with over two decades of experience in both industry and academia. His research interests include AI for FinTech and life sciences. He has been recognized as a top 2% scientist in AI (Stanford/Elsevier, 2024).