Neural Computing and Applications (2024) 36:15389–15406
https://doi.org/10.1007/s00521-024-09916-3

ORIGINAL ARTICLE

DeepAR-Attention probabilistic prediction for stock price series

Jiacheng Li[1] · Wei Chen[2] · Zhiheng Zhou[1] · Junmei Yang[1] · Delu Zeng[1]

Received: 30 November 2023 / Accepted: 23 April 2024 / Published online: 15 May 2024

Abstract

Stock price prediction is a significant research domain, intersecting statistics, finance, and economics. Accurately forecasting stock price trends has always been a focal point for many researchers. However, traditional statistical methods for time series prediction still lack accuracy. The existing deep learning-based methods for stock price prediction have significantly enhanced the accuracy of predicting individual stock prices. However, they are not effective in forecasting the probability range of future stock price trends. In this paper, to address these limitations, we propose a novel DeepAR model based on the attention mechanism (DeepARA) for both single-point and probabilistic predictions of stock prices. This enhances the accuracy and flexibility of stock price forecasting. Although the attention mechanism was initially developed for natural language processing, it has now found applications in time series forecasting, including the dynamics of the stock market. Attention allocates different weights to time points of varying importance, thereby enhancing the model's ability to capture fundamental market dynamics. We conducted multiple experiments in the Chinese stock market, involving 30 stocks across the top six sectors. Compared with baseline models, the DeepARA model demonstrates superior predictive capabilities.

Keywords Stock prediction · Deep learning · Recurrent neural networks · DeepAR · Attention mechanism

1 Introduction

The prediction of stock prices is a crucial topic in economics due to the random volatility of the stock market. It is often considered one of the most challenging subjects in effectively forecasting stock prices. Over the elongated course of capital markets, several prediction techniques have surfaced, comprising fundamental and technical analysis. Among them, temporal sequence analysis is getting more and more promising. For instance, in the early years, researchers predicted stock returns based on financial statements [1]. The Autoregressive Integrated Moving Average (ARIMA) method was been used to forecast the Taiwan stock market price [2]. Traditional machine learning methods are also extensively applied in stock price prediction, a method that combines Support Vector Machines (SVM) [3], and fractal feature selection was proposed to predict the direction of stock price indices [4]. The use of SVM for predicting stock returns based on information transmission across global markets was advocated [5]. In recent research, a fine-tuned SVM for stock price prediction was proposed [6]. BackPropagation Neural Networks (BPNN) [7] were applied to stock index forecasting [8]. Currently, with the rapid development of computer science and the improvement of software and hardware capabilities, the fields of deep learning applications are rising rapidly, and big data and deep learning methods have been applied to stock market predictions [9]. Considering that several factors need to be considered

✉ Delu Zeng
dlzeng@scut.edu.cn

Jiacheng Li
eejiacheng_li@mail.scut.edu.cn

Wei Chen
maweichen@mail.scut.edu.cn

Zhiheng Zhou
zhouzh@scut.edu.cn

Junmei Yang
yjunmei@scut.edu.cn

[1] School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, Guangdong province, China

[2] School of Mathematics, South China University of Technology, Guangzhou 510640, Guangdong province, China

when stock prices fluctuate, a deep learning approach was suggested [10] for predicting the stock market based on significant market events. Compared to foremost baseline models, this approach has demonstrated an improvement of nearly 6% in both S&P 500 index prediction and single stock price prediction.

Numerous studies have shown the effectiveness of employing deep learning models for anticipating stock market trends [11]. Unlike conventional linear algorithms such as AR, MA, and ARIMA that primarily forecast the price fluctuations of individual stocks, the adoption of advanced techniques like Long Short-Term Memory (LSTM) [12], Recurrent Neural Networks (RNN) [13], and Convolutional Neural Network (CNN) [14] can aid in identifying the underlying patterns present in the data. In the feature extraction module, RNN can serve multiple purposes such as performing one-step prediction on a time series, capturing temporal dynamics, reconstructing the input data, and maintaining local structure of the time series [15]. According to [16], using LSTM, RNN, and CNN can improve the accuracy of stock price prediction. Analyzing these types of trends and cycles can yield more profit for investors. LSTM, a variant of RNN, addresses the issue of gradient vanishing and explosion that commonly exists in RNN, making it an effective tool for modeling and predicting stock returns. LSTM was used to forecast China's stock returns, and it outperformed the random prediction method, improving the accuracy of stock return prediction from 14.3% to 27.2% [17]. Researchers are continuously improving deep learning methods for stock price series prediction [18].

With the development of recurrent neural network, researchers have focused on Gated Recurrent Unit Networks (GRU) [19], which are simpler variants of LSTM. An improved version of the GRU network was used to predict trading signals for Hong Kong, India, and US stock market indexes [20]. The GRU model was found to have higher accuracy in predicting these datasets. A novel stock market forecast model was proposed [21] that utilized deep learning techniques, factoring in investors' emotional tendencies and taking into account Empirical Model Decomposition (EMD) to systematically break down the intricate sequencing of stock prices. The empirical findings demonstrated that the enhanced LSTM-based model not only significantly enhanced the accuracy of predictions but also reduced any time lag. It would be significant to consider how the movement of stock prices could be impacted by the joint relationship between public news and social media. The critical role of incorporating event information with prediction models for achieving high accuracy in predictions is emphasized [22]. In order to tackle this complexity, researchers and practitioners have turned to Natural Language Processing (NLP) techniques to assist in the analysis of stock market data. The attention mechanism [23] has been proposed that has significantly contributed to the development of NLP. Attention mechanisms have also led to additional advances in time series prediction.

There is a considerable literature showing the effectiveness of deep learning frameworks for financial market forecasting. However, traditional structures only provide a single-point prediction of stock prices or indexes, ignoring the range of possible stock prices. DeepAR has been proposed [24] to generate precise probable predictions, and a feasible approach is to train a significant amount of relevant time series data with an autoregressive recurrent neural network model. This method allows for range prediction of stock prices, which can be extremely useful for investors. Furthermore, DeepAR was used to produce probabilistic temperature forecasting, demonstrating the versatility of this method in other applications [25].

Inspired by the DeepAR method, using DeepAR for the probabilistic forecasting of stock prices is considered a straightforward task. However, there has been a limited amount of research assessing DeepAR's capability in stock market prediction. Previous studies have mainly focused on probabilistic wind power forecasting [26]. In one such study, a novel model named NGOA-DeepAR was proposed for probabilistic wind power forecasting, and its performance was evaluated on two different datasets. A comparison involving a single DeepAR model and various hybrid models was made to precisely predict atmospheric concentrations [27]. With advancements in deep learning, the Transformer architecture [23], based on the attention mechanism, has been introduced. These advancements have significantly contributed to the progress in natural language processing techniques by eliminating the need for convolutional layers and recurrent networks. Consequently, this has facilitated the development of newer and more efficient models capable of processing and analyzing natural language data with greater accuracy. For example, applying the Transformer architecture with an attention mechanism to forecast stock market indexes resulted in superior prediction performance compared to traditional RNN-based models and their variants [28]. Additionally, a novel network utilizing a bidirectional GRU network for predicting stock price movements has been developed, incorporating both the attention mechanism and Reinforcement Learning (RL) [29]. This model surpassed recent advancements, achieving state-of-the-art performance.

In this paper, we introduce a novel approach named DeepARA, which synergizes the attention mechanism with the DeepAR model to enhance stock price prediction. DeepARA utilizes the attention mechanism to variably weigh peak prices and pivotal turning points in the stock market, advancing beyond mere single-point predictions. This method emphasizes the probabilistic distribution and directional shifts of stock prices. The primary contributions of this research are summarized as follows:

- We propose a novel architecture DeepARA that integrates DeepAR and attention mechanisms for predicting stock market trends.
- We integrate the attention mechanism into DeepAR for both single-point and probabilistic predictions of stock prices.
- We conducted experiments on 30 stocks across six industries in the Chinese stock market, selecting the top five stocks in each industry based on market capitalization. By comparing the performance of DeepARA with other baseline models, we were able to successfully demonstrate its superior performance. Furthermore, we carried out a comprehensive performance evaluation and compared our proposed method with relevant approaches to fully validate its effectiveness.

The subsequent sections are organized as follows: Section 2 reviews the relevant literature from previous studies. Section 3 introduces the background knowledge of DeepAR model. Section 4 provides a detailed description of our proposed DeepARA model. In Sect. 5, we elaborate on the experimental setup, including parameter settings, baseline models, evaluation metrics, autoregressive analysis, and an analysis of the experimental results. Finally, Sect. 6 concludes the paper.

2 Related work

There are two primary methods commonly utilized in analyzing and forecasting financial market trends: fundamental analysis and technical analysis. Fundamental analysis considers economic factors that might influence stock market movements, including both unstructured textual information, such as financial news and company reports, and macroeconomic factors [30]. In contrast, technical analysis views historical stock prices and trading data as time series [31]. A comprehensive review of machine learning techniques applied to stock market forecasting, discussing the state-of-the-art machine learning-based methods over the past two decades, was conducted [32]. Various approaches have been proposed by researchers to predict stock prices, drawing on these two distinct perspectives.

2.1 Fundamental analysis

The theory of fundamental analysis posits that a company's stock price is influenced by a range of internal and external political and economic factors. Furthermore, fundamental analysis takes into account broader economic trends, societal stability, and financial news to inform assessments of the stock market. This information serves to strengthen the basis of a company's fundamental analysis.

Financial statement analysis is a pivotal element of fundamental analysis in stock trading. For instance, a vast array of financial statements can be aggregated into a summary metric [1], aiding in adjusting stock positions to potentially boost profits based on financial statement analysis. The findings of [1] were later revisited [33], highlighting that, according to fundamental analysis theory, it is feasible to discern a company's stock's intrinsic value, which might not be fully reflected in its current stock prices. This enables systematic forecasting of abnormal returns by identifying undervalued or overvalued stocks. Recent studies have integrated social media and textual data to predict stock market trends. For example, Twitter data have been used to gauge public mood and, combined with historical DJIA values, to forecast stock market movements [34], achieving 75.56% accuracy using a novel cross-validation method. Additionally, a new approach for pinpointing crucial stock market events through social media text analysis and assigning relevant labels was introduced [35], enhancing the prediction of subsequent stock movements. Merging financial and textual information [36] has been shown to predict the next-day stock movements, establishing a corpus to underscore the significance of text analytics in stock price trends. Furthermore, two methodologies were developed to extract topic-sentiment features, demonstrating that sentiment analysis can significantly enhance stock prediction performance [37]. Consequently, text analysis has emerged as a crucial component of fundamental analysis.

The rapid advancements in deep learning in recent years have led to significant breakthroughs, with scientists applying fundamental analysis in combination with deep learning techniques to predict stock market trends. A method for predicting stock prices was proposed [38], utilizing the distributed representation of news articles and accounting for the correlation among multiple companies within the same industry. Various neural network models, including LSTM and CNN, have been applied to stock market opinions posted on StockTwits [39]. Their findings indicate that deep learning models are highly effective in financial sentiment analysis. More recently, a study incorporated components of fundamental analysis, such as the P/E ratio and profitability, alongside trading volume, to improve stock market trend prediction performance and minimize model errors [40]. That study employed a combination of LSTM and CNN methods, reflecting the ongoing innovation in this field.

In summary, fundamental analysis primarily focuses on predicting future stock market movements based on unstructured textual data. However, this method may not adequately address the subsequent volatility of stock prices. For instance, it might not distinguish between a 1% and a 10% increase in stock price. While efforts have been made to enhance the accuracy of fundamental analysis through the integration of deep learning with traditional analytical methods, there remains a challenge in accurately forecasting a specific range of future stock prices.

2.2 Technical analysis

In evaluating market price movements, technical analysis typically overlooks a company's internal and external attributes. Proponents of technical analysis maintain that any fundamental factors influencing market movements are quickly reflected in market prices [41]. Consequently, they focus on analyzing patterns in price, volume, breadth, and trading activity rather than subjective economic factors, believing this information suffices to predict future values [42]. Conversely, deep learning methods have shown effectiveness in forecasting future stock values by uncovering patterns in historical data [43]. The latest advancements in the application of deep learning for stock market forecasting were reviewed [44], with a categorization of various data sources, neural network architectures, and commonly used evaluation metrics being provided. Insights into potential future research directions were also offered. These methods do not rely solely on the immediate impact of fundamentals on stock prices. Instead, they focus on analyzing historical trends and patterns. Consequently, deep learning models can forecast future prices with a higher degree of accuracy and comprehensiveness than is possible with technical analysis alone.

In the early stages of stock price prediction research, traditional statistical methods like ARIMA were widely used. For instance, the ARIMA model was applied to forecast stock data for the New York Stock Exchange (NYSE) and the National Stock Exchange of India Ltd. (NSE) [45], showing its strong capability in short-term forecasting. However, a limitation of ARIMA is its dependence on stable data for time series prediction, with a struggle to capture patterns in highly volatile stock data, thus mainly yielding short-term trends. To address this, deep learning methods have been introduced. A comparative study between ARIMA and Artificial Neural Networks (ANN) highlighted the superior performance of neural network models [46]. More recently, advancements in RNN variants like LSTM and GRU have been significant. A deep learning model that integrates RNN and LSTM was developed [47], improving prediction rates by factoring in environmental elements, and it was observed to outperform DNN models by 15% in stock market prediction accuracy. With the evolution of neural network technology, researchers have started to blend various deep learning techniques for time series analysis. This includes the development of neural Ordinary Differential Equations (ODE) [48], which enhance traditional RNN with continuous-time hidden dynamics described by ordinary differential equations. An adaptive method for setting the integral interval of neural ODE systems was subsequently introduced [49], effectively accelerating the training and sampling processes. To enhance prediction accuracy, a stock price prediction model using a recurrent neural network was developed [50], building upon improvements to the Particle Swarm Optimization (PSO) algorithm. Furthermore, an algorithm for clustering banks with similar price trends was proposed [51], followed by training a LSTM model on these clustered stocks for both static and dynamic stock price predictions. The BiCuDNNLSTM-1dCNN model, combining bidirectional CUDNN-LSTM and 1D CNN, was also developed [52], demonstrating its effectiveness and reliability in aiding investors with their decision-making based on its forecasting results.

Building upon the concept of the attention mechanism, the transformative architecture known as the Transformer was introduced [23], achieving notable success in natural language processing. Following this approach, a novel LSTM model that incorporates attention mechanisms was developed [53]. This model is specifically designed to extract significant, low-correlation information from datasets, thereby enhancing its predictive accuracy by introducing additional input gates to filter out irrelevant data. The attention-enhanced LSTM model has demonstrated considerable potential in financial forecasting, especially in predicting stock prices. A hypergraph tri-attention network (HGTAN) comprising hierarchical attention was proposed [54] for stock trend prediction and investment simulation. Furthermore, a method termed CNN-BiLSTM-AM, which combines the attention mechanism with CNN and BiLSTM for stock price prediction, was introduced, and its effectiveness further validated [55]. In recent research, a Tensor Robust Principal Component Analysis (TRPCA) model [56], designed to integrate multi-modal and multi-time information, has been demonstrated to effectively enhance the performance of stock trend prediction.

While significant achievements have been made in predicting stock prices, it is crucial to acknowledge that, to the best of our knowledge, these models primarily provide single-point predictions. The use of probabilistic methods for predicting stock prices is a relatively novel area in the field of financial time series analysis. In this context, the innovative DeepAR architecture, renowned for its success in the probabilistic prediction of time series, was introduced [24]. Through experimental applications to U.S. highway traffic and electricity consumption time series datasets, the DeepAR model has been shown to effectively learn the Gaussian model from related time series, thereby substantially enhancing prediction accuracy. Incorporating the attention mechanism into the DeepAR framework, our paper introduces the DeepARA architecture for stock price prediction, representing a novel approach in the field.

3 Background

3.1 Preliminary definition

The goal of stock price prediction is to forecast future values based on historical data from a previous period. Using historical price data $Z = \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \ldots, \mathbf{z}_t \in \mathbb{R}^{d\times L}$ and associated covariates $\mathbf{x}_{i,1:T}$, we aim to predict the value of $\mathbf{z}_{t+1}$. This prediction is accomplished through a rolling forecast, employing a sliding window approach that starts the next prediction cycle at $\mathbf{z}_2$. Here, $\mathbf{z}$ represents the input stock price data, t denotes the time step, d represents the dimension of features, and L signifies the total length of the sequence, as depicted in Fig. 1. Predicting the green area through the values in the orange area, continuously sliding to forecast.

3.2 DeepAR model

DeepAR is a deep learning method that enables probabilistic prediction for time series data. In contrast to traditional single-point forecasting methods, DeepAR can generate probabilistic predictions along with single-point forecasts. This paper considers the DeepAR architecture for stock market prediction. The approach uses an autoregressive recurrent neural network to extract global patterns from the entire time series in the dataset, by leveraging historical data.

DeepAR, by focusing on predicting the probability distribution of data, facilitates the effortless incorporation of additional features beyond the primary prediction target. This probabilistic prediction approach surpasses traditional single-point prediction models in efficacy and utility. The advantages of this method are two-fold:

(i) Given the inherent randomness in many time series processes, generating a probability distribution yields predictions that are inherently more accurate.

(ii) This method allows for the quantification and comparison of uncertainty and associated risks. Consequently, DeepAR can deliver more precise forecasts with reduced manual intervention.

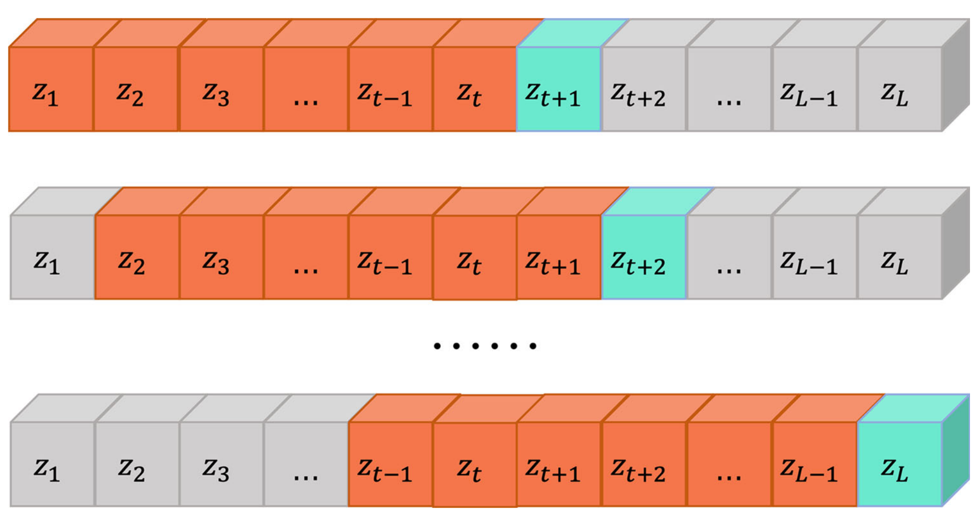
First, we define the value of the ith sequence at moment t_0 to divide the time points. Our objective is to obtain the joint conditional probability distribution $P(\mathbf{z}_{i,t_0:T}|\mathbf{z}_{i,1:t_0-1}, \mathbf{x}_{i,1:T})$. This means that we aim to model the future sequence $\mathbf{z}_{i,t_0:T}$ based on the available data $\mathbf{z}_{i,1:t_0-1}$ and the variable $\mathbf{x}_{i,1:T}$. The conditioning range and prediction range are denoted as $[1, t_0 - 1]$ and $[t_0, T]$, respectively.

The DeepAR model uses an autoregressive recurrent network architecture. We define the above distribution in the following likelihood form of

$$\begin{aligned} Q_\Theta(\mathbf{z}_{i,t_0:T}|\mathbf{z}_{i,1:t_0-1}, \mathbf{x}_{i,1:T}) &= \prod_{t=t_0}^{T} Q_\Theta(\mathbf{z}_{i,t}|\mathbf{z}_{i,1:t-1}, \mathbf{x}_{i,1:T}) \\ &= \prod_{t=t_0}^{T} p(\mathbf{z}_{i,t}|\theta(\mathbf{h}_{i,t}, \Theta)), \end{aligned} \tag{1}$$

herein, Q_Θ is the probability distribution under the set of parameters Θ, which is parameterized by the DeepAR model. Here, Θ represents all the parameters of the model. $\mathbf{z}_{i,t_0:T}$ represents future sequence data, $\mathbf{z}_{i,1:t_0-1}$ denotes past sequence data, and $\mathbf{x}_{i,1:T}$ signifies the covariates across the entire time series. $\mathbf{h}_{i,t}$ represents the output of the autoregressive recurrent network. The right hand side of the first line of Eq. (1) represents the original probability

Fig. 1 Sliding window to roll prediction



distribution in the form of an autoregressive probability product, and the second line represents the autoregressive probability in terms of a parameterized likelihood function. The function $h(\cdot)$ represents a recurrent neural network cell unit. We input the hidden layer $\mathbf{h}_{i,t-1}$ and data $\mathbf{z}_{i,t-1}$ of the previous moment and the known information $\mathbf{x}_{i,t}$ of the current moment to obtain the hidden layer $\mathbf{h}_{i,t}$ of that moment. Then, we transform $\mathbf{h}_{i,t}$ by the neural network $\theta(\cdot)$ into given distribution parameters and obtain the likelihood function $\ell(\mathbf{z}_{i,t}|\theta(\mathbf{h}_{i,t}, \Theta))$.

We can understand this autoregressive process within the sequence-to-sequence framework. First, we encode the conditioning range data using the encoder network to obtain the hidden layer output $\mathbf{h}_{i,t_0-1}$. Next, we use this output as the initialized hidden layer of the decoder network to transform the decoder output $\mathbf{h}_{i,t}$ into the distribution parameters. Then, we use the decoder iteration output $\mathbf{h}_{i,t}$ as the initialized hidden layer of the next decoder iteration to obtain the parameters of the distribution. In this way, the DeepAR model can provide us with the predicted probability distribution.

Here, we need to pay attention to the differences between the training process and the prediction process. The encoder part is the same for both, and we just need to input the conditioning range of data into the RNN in turn and obtain the hidden layer output $\mathbf{h}_{i,t-1}$. The difference lies in the decoder, as shown in Fig. 2, which is the framework of the DeepAR model. The left side is the encoder used for training, while the right side is the decoder used for prediction. The input section contains two inputs: one is the covariate $\mathbf{x}_{i,t-1}$ and the other is $\mathbf{z}_{i,t-2}$, which is the predicted value from the previous time point. This means that when we are predicting the $t-1$ step, we need to use the value from the $t-2$ time point. These inputs are then fed into a hidden layer, and $\mathbf{h}_{i,t-1}$ is also inputted into the next round, implementing the concept of autoregression.

In the training process depicted in Fig. 2, all data are known, and we can directly input the data for the prediction range. During training, the model considers the value of the prior time step $\mathbf{z}_{i,t-1}$ and the corresponding state $\mathbf{h}_{i,t-1}$ to update its parameters at each time step t. The neural network parameters are obtained by maximizing the log-likelihood objective function specified in Eq. (2):

$$\mathcal{L}_{(\theta)} = \sum_{i=1}^{N} \sum_{t=t_0}^{T} \log p\left(\mathbf{z}_{i,t} \mid \theta\left(\mathbf{h}_{i,t}\right)\right) \tag{2}$$

The form of $\theta\left(\mathbf{h}_{i,t}\right)$ is specific to the selected likelihood function $p(\mathbf{z}|\theta)$, which, in turn, is determined by the statistical properties of the data. The DeepAR algorithm utilizes a neural network that directly forcasts all the parameters θ. Specifically, in this paper, to parameterize the Gaussian likelihood, we use the mean and standard deviation, represented as $\theta = (\mu, \sigma)$. Individually, the mean is determined via an affine function of the network output, while the standard deviation is obtained by performing an affine transformation and soft additive activation to ensure $\sigma > 0$:

$$p_G(\mathbf{z} \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(\mathbf{z}-\mu)^2}{2\sigma^2}\right), \tag{3}$$

$$\mu\left(\mathbf{h}_{i,t}\right) = \mathbf{w}_{\mu}^{T}\mathbf{h}_{i,t} + b_{\mu} \tag{4}$$

$$\sigma\left(\mathbf{h}_{i,t}\right) = \log\left(1 + \exp\left(\mathbf{w}_{\mu}^{T}\mathbf{h}_{i,t} + b_{\sigma}\right)\right) \tag{5}$$

After the completion of the training process, we can input historical data of $t < t_0$ into the network to acquire the initial state $\mathbf{h}_{i,t_0-1}$. By utilizing the learned probability distribution, we can generate predictions by sampling from the initial state.

In the prediction process shown in Fig. 2, the values of the prediction range are unknown, so the true values of the previous time step cannot be directly input into the decoder. Instead, an estimate $\tilde{\mathbf{z}}_{i,t-1}$ can be obtained by sampling.

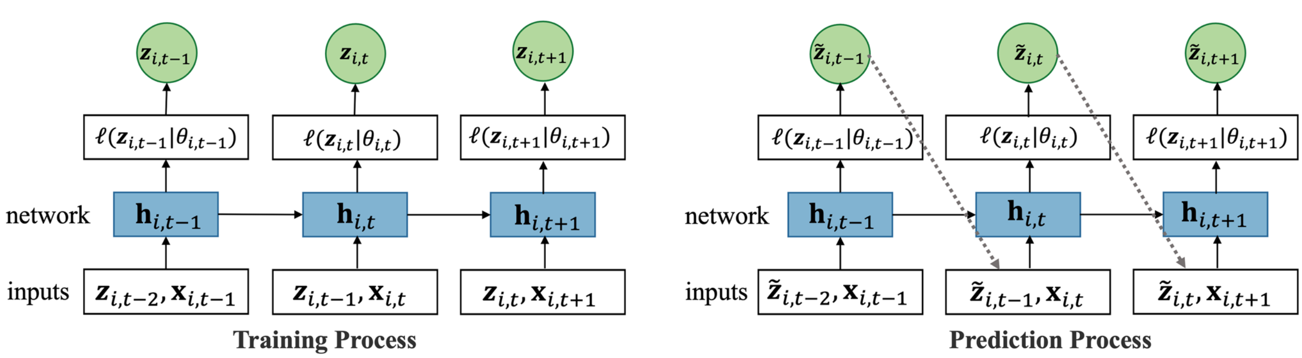


Fig. 2 Framework of the DeepAR model

The estimated value is then fed into the RNN at subsequent time steps, leading to a prediction range obtained through iterative computation.

4 The proposed architecture: DeepARA

In this section, we introduce DeepAR-Attention (DeepARA), a cutting-edge architecture that merges the predictive prowess of DeepAR with the precision of attention mechanisms to forecast stock price probabilities. This model transcends traditional single-point forecasts by offering a probabilistic insight into the dynamics of stock prices. Herein, we delve into the DeepARA architecture, highlighting its principal features and components for a thorough understanding.

4.1 DeepARA architecture

In recent years, attention mechanisms have become increasingly popular in various fields of machine learning and artificial intelligence [23]. One notable development is the Temporal Attention-Gated Model (TAGM) [57], which merges attention models with gated recurrent networks. This integration is designed to effectively manage noisy or unsegmented sequences by selectively focusing on pertinent features and filtering out irrelevant ones. TAGM accomplishes this by employing an attention mechanism to assess the significance of each input feature at every time step, subsequently using the attention weights to gate the input feature vector. This process yields a context vector for the gated recurrent network, enhancing performance in tasks like speech recognition and sentiment analysis. Another advancement is the TeaNet [28], a deep learning model tailored for stock prediction, leveraging transformer and attention mechanisms to predict stock movements. Recent research indicates that accurate stock price predictions and increased profits can be achieved using a small dataset of five-day stock prices combined with text representations. The incorporation of attention mechanisms into deep learning models has shown promise in enhancing performance in this domain. Consequently, we are exploring the integration of the DeepAR model with attention mechanisms to further this advancement.

The architecture of our proposed DeepAR-Attention (DeepARA) model is illustrated in Fig. 3. This innovative framework employs a weighted mechanism that prioritizes critical features while diminishing the influence of less significant ones, thereby enhancing the model's predictive precision.

Firstly, we acquire the raw stock price data, which is then normalized by our data preprocessing module. The processed data, along with the covariates, are fed into the DeepAR module to obtain the hidden state $\mathbf{h}_{i,t}$. Subsequently, this hidden state $\mathbf{h}_{i,t}$ is input into the attention module, where it undergoes a weighting computation to yield an updated state $\hat{\mathbf{h}}_{i,t}$. This new state $\hat{\mathbf{h}}_{i,t}$, in conjunction with the parameters Θ, is used to calculate the conditional probability distribution, culminating in our predictive output. The following sections will detail the operational principles of each module.

4.2 Date preprocessing block

Initially, the raw stock price data undergoe a preprocessing module for normalization, where we employ a linear normalization technique using the Max–Min normalization method. This approach proportionally scales the data, removing the constraints of data units, and converting the dimensions into pure numerical values. It is a form of linear normalization that linearly maps the original data so that they fall within the [0, 1] range. The transformation function is as follows:

$$\mathbf{z}_i = \frac{z_i - \min(z_i)}{\max(z_i) - \min(z_i)}, \tag{6}$$

where z_i represents the original stock price data, $\mathbf{z}_i$ is the transformed data, max and min are the maximum and minimum values at the given time point, respectively.

4.3 DeepARA attention module

In our study, we employ an approach that synergizes the attention mechanism with DeepAR. Following the preprocessing module, the transformed data $z_{i,t}$ and the covariates $\mathbf{x}_{i,t}$ pass through an autoregressive recurrent network structure to yield the hidden state $\mathbf{h}_{i,t}$:

$$\mathbf{h}_{i,t} = h(\mathbf{h}_{i,t-1}, \mathbf{z}_{i,t-1}, \mathbf{x}_{i,t}, \Theta), \tag{7}$$

h is represented by a multi-layer recurrent neural network architecture utilizing LSTM cells that are parameterized by Θ in Eq. (1). This process yields the hidden state $\mathbf{h}_{i,t}$. Subsequently, the attention mechanism takes the hidden state $\mathbf{h}_{i,t}$ as its input. Through transformations:

$$W_Q \mathbf{h}_{i,t} = Q, \tag{8}$$

$$W_K \mathbf{h}_{i,t} = K, \tag{9}$$

$$W_V \mathbf{h}_{i,t} = V. \tag{10}$$

we obtain the Query (Q), Key (K), and Value (V) components essential for calculating the attention scores $\alpha_{i,t}$. Next, we compute the attention distribution for each time step and, subsequently, perform a weighted summation of the respective outcomes:

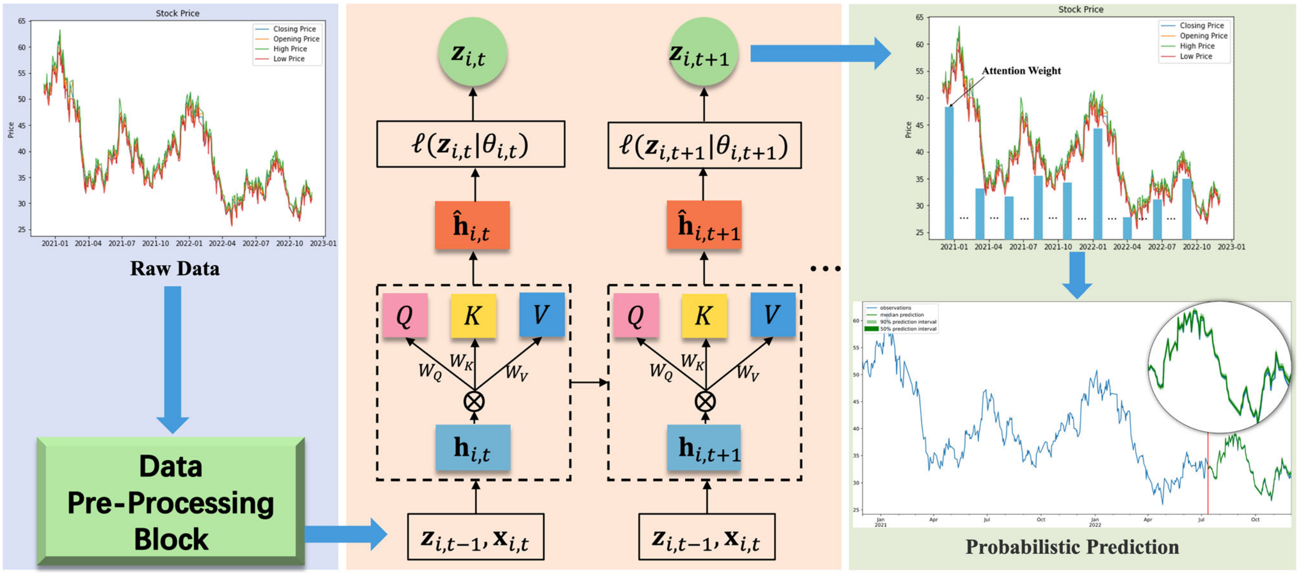


Fig. 3 DeepARA architecture: By computing attention weights for the hidden layers, important features are accorded higher significance, while the influence of redundant features is diminished, thereby ensuring enhanced predictive accuracy

$$attn_i = \sum_{i}^{N} softmax(s(q_i, k_j))v_j, \tag{11}$$

where $s(q_i, k_j)$ represents the score values obtained after the aforementioned dot product and scaling. To increase computational efficiency, we utilize matrix operations to calculate the attention output vectors for all positions in one go:

$$\alpha_{i,t} = \text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V. \tag{12}$$

Utilizing the attention scores, we compute the new weighted hidden state vector at each time step, denoted as $\hat{\mathbf{h}}_{i,t}$:

$$\hat{\mathbf{h}}_{i,t} = \sum_{i=1}^{N} \alpha_{i,t}\mathbf{h}_{i,t}, \tag{13}$$

where $\alpha_{i,t}$ is the attention score derived from Attention(Q, K, V), and $\mathbf{h}_{i,j}$ is the hidden state at time step t. The hidden state $\hat{\mathbf{h}}_{i,t}$ is used to parameterize the probability distribution of the target variable. Here, we opt for a Gaussian distribution, necessitating the estimation of the distribution's mean μ and standard deviation σ, as formalized in Eq. (3).

With the new hidden state $\hat{\mathbf{h}}_{i,t}$, the new distribution can be written in the following likelihood form:

$$Q_\Theta(\mathbf{z}_{i,t_0:T}|\mathbf{z}_{i,1:t_0-1}, \mathbf{x}_{i,1:T}) = \prod_{t=t_0}^{T} p(\mathbf{z}_{i,t}|\theta(\hat{\mathbf{h}}_{i,t}, \Theta)). \tag{14}$$

The DeepARA training procedure is shown in Algorithm 1, where $\hat{\mathbf{h}}_{i,t}$ is the hidden state after passing through the attention layer. Currently, different time points are assigned varying weights. The loss function of the current model is defined as Eq. (15):

$$\mathcal{L}_{(\theta)} = \sum_{i=1}^{N}\sum_{t=t_0}^{T} \log p\left(\mathbf{z}_{i,t} \mid \theta\left(\hat{\mathbf{h}}_{i,t}\right)\right). \tag{15}$$

In the ensuing training and prediction phases, these diverse weights are attributed to the inputs, culminating in the generation of the predictive results. In summary, the attention feedforward block employs the attention mechanism to learn the significance of various components of the input sequence and, subsequently, utilizes a feedforward network to generate the output by weighing the inputs.

Algorithm 1 DeepARA model training procedure.

Algorithm 1 DeepARA model training procedure.

Initialize model parameters Θ
for each training epoch **do**
  for each sequence i **do**
    Compute initial hidden state $\mathbf{h}_{i,t}$
    for each time step $t = t_0$ to T **do**
      $\mathbf{h}_{i,t} \leftarrow h(\mathbf{h}_{i,t-1}, \mathbf{z}_{i,t-1}, \mathbf{x}_{i,t}; \Theta)$ ▷ Forward pass
      $Q \leftarrow W_Q \mathbf{h}_{i,t}$
      $K \leftarrow W_K \mathbf{h}_{i,t}$
      $V \leftarrow W_V \mathbf{h}_{i,t}$
      $\alpha_{i,t} \leftarrow \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$ ▷ Attention scores
      $\hat{\mathbf{h}}_{i,t} \leftarrow \sum(\alpha_{i,t} \cdot \mathbf{h}_{i,t})$ ▷ Attention weighted hidden states
      $(\mu_{i,t}, \sigma^2_{i,t}) \leftarrow \mathrm{Linear}(\hat{\mathbf{h}}_{i,t}; \Theta)$ ▷ Predictive distribution parameters
      $p(\mathbf{z}_{i,t} | \hat{\mathbf{h}}_{i,t}) \leftarrow \mathrm{Gaussian}(\mu_{i,t}, \sigma^2_{i,t})$ ▷ Probability distribution
    end for
    $\mathcal{L}_{(\theta)} = \sum_{i=1}^{N} \sum_{t=t_0}^{T} \log p\left(\mathbf{z}_{i,t} \mid \theta\left(\hat{\mathbf{h}}_{i,t}\right)\right)$ ▷ Loss computation
    Perform backpropagation and update Θ
  end for
end for

4.4 Probabilistic prediction

Our model's probabilistic predictions are represented through confidence intervals, which are calculated based on the parameters of the predictive distribution, typically conditional probability distributions for each time point. A Gaussian distribution is employed, with the parameters mean μ and standard deviation σ predicted by the model. Once the model is trained, it provides a predicted mean and standard deviation for each time point's forecast. With these parameters, we can calculate the confidence intervals. Our model's confidence intervals range from 50% to 90%. For the 50% confidence interval, the corresponding Z-score is approximately 0.674, where the Z-score measures the distance from the mean of the distribution in units of the standard deviation, and *CI* means the confidence interval. Therefore, the 50% confidence interval can be expressed as:

$$CI_{50\%} = [\mu - 0.674 \times \sigma, \mu + 0.674 \times \sigma]. \tag{16}$$

For higher confidence levels, the Z-score for 90% is approximately 1.645. Consequently, the 90% confidence interval is given by:

$$CI_{90\%} = [\mu - 1.645 \times \sigma, \mu + 1.645 \times \sigma]. \tag{17}$$

As illustrated in Fig. 4, we present a line graph representation of the probabilistic prediction intervals, denoting confidence intervals using this methodology. The dark green shade indicates the 50% confidence interval, while the light green shade represents the 90% confidence interval. The central dark green line depicts the predicted values. This method of representation is consistently applied in the experimental section that follows.

This probabilistic forecasting method provides a measure of uncertainty for our model's predictions. In DeepARA, the prediction for each time point is not merely a single-point forecast but rather predicts a distribution. Confidence intervals surrounding the single-point forecast offer a range of uncertainty, representing the best estimate for future values. Thus, confidence intervals afford an intuitive understanding of the model's predictive uncertainty, enabling users to make more informed decisions based on these forecasts. In practical applications, especially when decisions must account for forecast uncertainty, confidence intervals are an invaluable tool.

In information theory, entropy is a standard measure of uncertainty. For probabilistic forecasting models, we typically desire the model's predictions to be as accurate as possible, implying that the predictive distribution should be concentrated (i.e., lower entropy) rather than dispersed (higher entropy). If the model's predicted distribution is closer to the true distribution of observed data, then the predictive entropy will be lower, indicating that the model's predictive uncertainty is reduced.

Specifically for the DeepAR and DeepARA models, the concept of Conditional Entropy (CE) can be utilized for demonstration. The conditional entropy

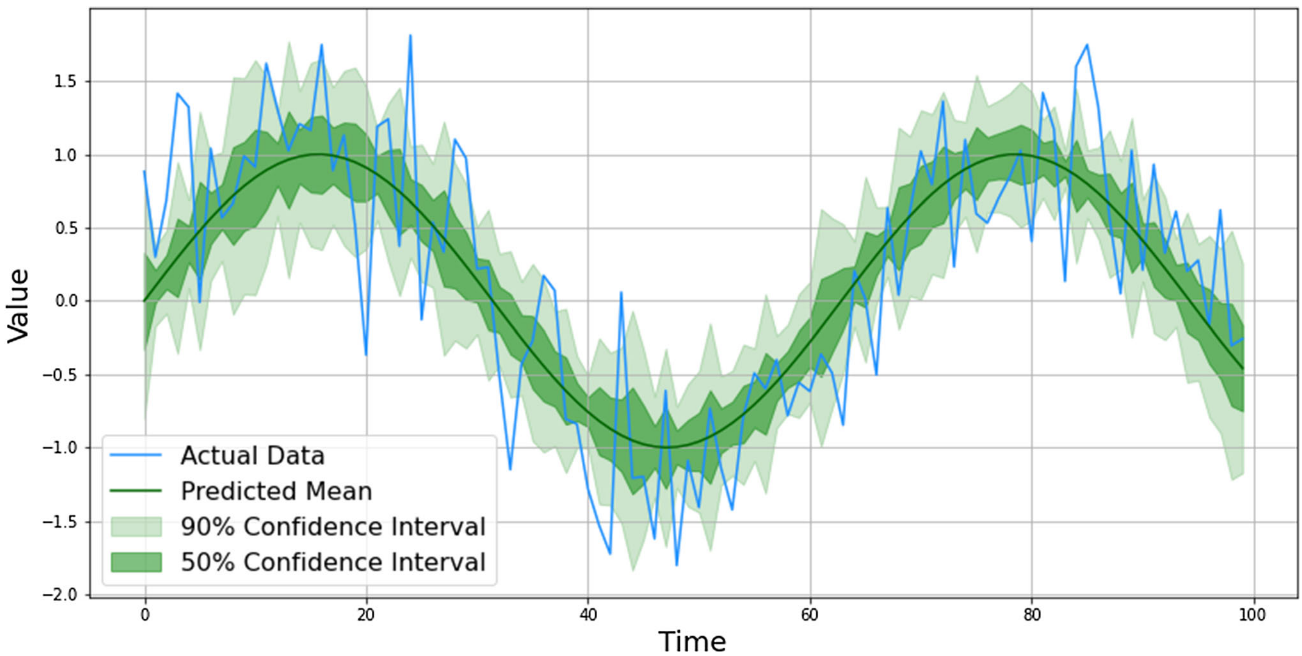


Fig. 4 Line plot with 50% to 90% confidence intervals

$CE(\mathbf{z}_{i,t}|\mathbf{z}_{i,1:t-1}, \mathbf{x}_{i,1:T})$ measures the uncertainty of the random variable $\mathbf{z}_{i,t}$ given the conditions set by the random variables $\mathbf{z}_{i,1:t-1}$ and $\mathbf{x}_{i,1:T}$. In the context of time series forecasting, $\mathbf{z}_{i,t}$ can be considered as the observation at a future time point, while $\mathbf{z}_{i,1:t-1}$ and $\mathbf{x}_{i,1:T}$ represent the known historical observations and covariates, respectively. Conditional entropy can be calculated using the following formula:

$$\begin{aligned} &CE(\mathbf{z}_{i,t}|\mathbf{z}_{i,1:t-1}, \mathbf{x}_{i,1:T}) \\ &= -\sum p(\mathbf{z}_{i,1:t-1}, \mathbf{x}_{i,1:T}) \\ &\quad \sum p(\mathbf{z}_{i,t}|\mathbf{z}_{i,1:t-1}, \mathbf{x}_{i,1:T}) \log p(\mathbf{z}_{i,t}|\mathbf{z}_{i,1:t-1}, \mathbf{x}_{i,1:T}), \end{aligned} \tag{18}$$

where $p(\mathbf{z}_{i,1:t-1}, \mathbf{x}_{i,1:T})$ represents the distribution of historical observations and covariates, and $p(\mathbf{z}_{i,t}|\mathbf{z}_{i,1:t-1}, \mathbf{x}_{i,1:T})$ is the conditional distribution of $\mathbf{z}_{i,t}$ given $\mathbf{z}_{i,1:t-1}$ and $\mathbf{x}_{i,1:T}$.

In our DeepARA model, through the attention mechanism, we selectively weight historical information to better predict the future, thereby influencing $p(\mathbf{z}_{i,t}|\mathbf{z}_{i,1:t-1})$. The attention mechanism enables the model to more accurately capture the historical information $(\mathbf{z}_{i,1:t-1}, \mathbf{x}_{i,1:T})$ that is most influential for predicting future observations $\mathbf{z}_{i,t}$. As a result, $p(\mathbf{z}_{i,t}|\mathbf{z}_{i,1:t-1})$ becomes more focused, leading to a lower CE, which signifies a reduction in the model's predictive uncertainty. Under such a mechanism, the DeepARA model is theoretically superior to DeepAR in performance.

5 Experiments

This section provides an overview of the experiments conducted to evaluate the effectiveness of the DeepARA model in stock price prediction. Initially, we outline the problem we aim to address, followed by an introduction to the benchmark models used for comparison. We then describe our experimental dataset, with a specific focus on datasets related to the highly volatile Chinese stock market. This choice is crucial for appropriately assessing DeepARA's performance in such a dynamic environment. Subsequently, we delve into the hyperparameter settings and evaluation metrics used to measure model performance. Finally, the section concludes with the presentation

Table 1 Stock dataset: Our dataset includes the stock symbol codes for the top five companies by market capitalization in six representative industries of the China stock market

Electronic	Healthcare	Coal	Food and Drink	Automotive	Electricity
002475.SZ	600276.SH	600050.SH	600809.SH	002594.SZ	300750.SZ
601138.SH	603259.SH	600188.SH	000858.SZ	600104.SH	601012.SH
603501.SH	300760.SZ	601225.SH	603228.SH	601633.SH	002812.SZ
000725.SZ	300122.SZ	601898.SH	600519.SH	000338.SZ	600438.SH
002371.SZ	600436.SH	000723.SH	000568.SH	601238.SH	600406.SH

Table 2 Basic information of the experimental stock data (002475.SZ)

Date	Open	High	Low	Close	Volume
12-01-2020	52.00	52.87	51.50	52.68	62055753.0
12-02-2020	52.63	52.63	51.08	51.60	67464338.0
12-03-2020	51.40	51.96	51.18	51.27	47744905.0
–	–	–	–	–	
11-29-2022	30.97	31.42	30.70	31.35	44090518.0
11-30-2022	31.02	31.34	30.52	30.91	55943047.0
12-01-2022	31.50	32.09	30.88	31.05	65869228.0

of our experimental results and an analysis of the errors encountered.

5.1 Baseline model

Here, we selected several baseline models for comparison, ensuring that all models utilized identical parameters and network layers. This approach was adopted to achieve the best possible outcomes, as determined by multiple experimental trials. Our selection of baseline models encompasses traditional time series forecasting methods along with various state-of-the-art variants applied to stock price prediction.

- SVM [3]: It is a traditional machine learning algorithm, also known as Support Vector Regression (SVR) when applied to regression problems. It operates by identifying the optimal hyperplane that best fits the sample data, focusing on minimizing the error within a certain threshold. The capability of SVM to handle high-dimensional data and its effectiveness even when the number of dimensions exceeds the number of samples make it a valuable comparative model in our experiments.
- RNN [13]: It is a fundamental model in deep learning, particularly renowned for its effectiveness in processing sequential data.
- LSTM [12]: It is a variant structure of the RNN that incorporates gated mechanisms and memory cells.
- BiLSTM [55]: Bidirectional Long Short-Term Memory is a variant of the traditional LSTM model. It processes data in both forward and backward directions, capturing information from past and future states of the sequence. This bidirectional approach allows for a more comprehensive understanding of the data, making BiLSTM

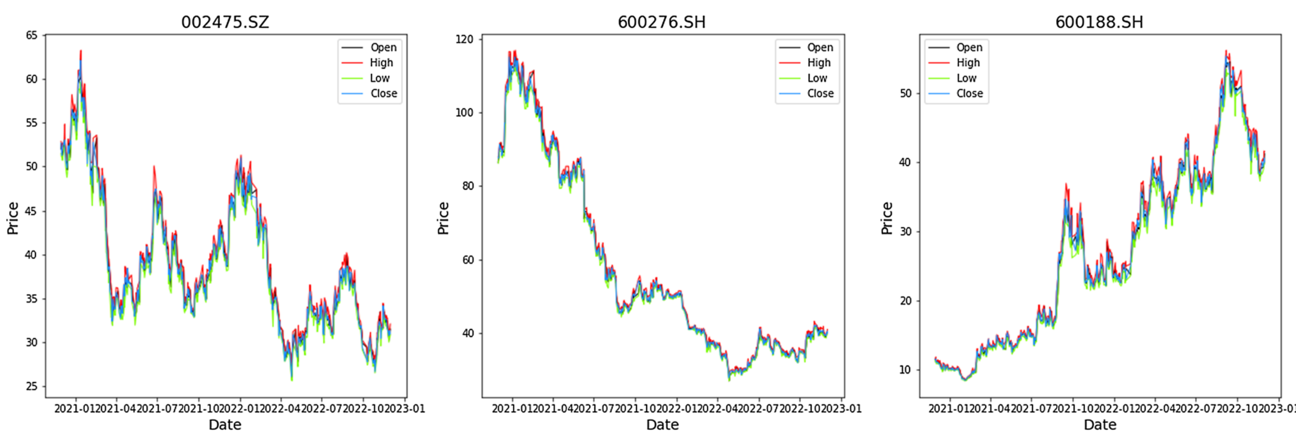


Fig. 5 Stock price raw data

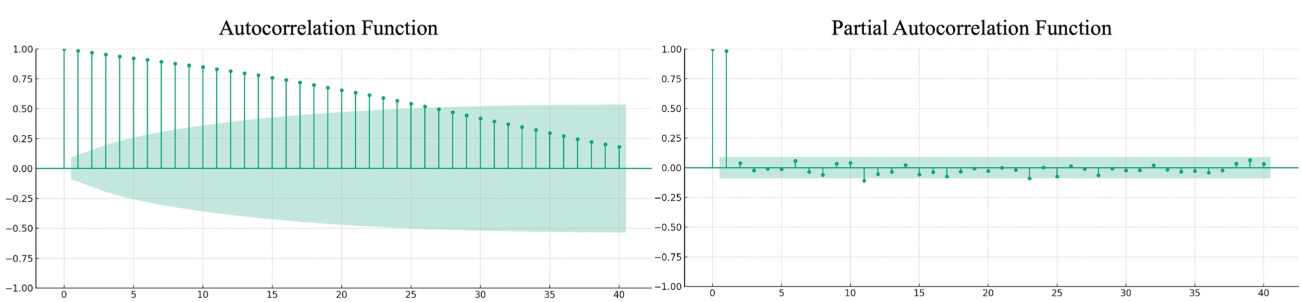


Fig. 6 ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots for stock price data: The ACF plot shows the correlation of the series with itself at different lags, while the PACF plot shows the partial correlation of the series with itself at different lags

Table 3 Results of three evaluation indicators in six datasets

Stock List	Model	MAE	RMSE	MAPE	Stock List	Model	MAE	RMSE	MAPE
Electronic	SVM	1.1787	2.1406	0.0560	Healthcare	SVM	2.3274	2.7825	0.0608
002475.SZ	RNN	0.7469	1.0583	0.0228	600276.SH	RNN	0.8385	1.0648	0.0227
	LSTM	0.7551	1.0533	0.0226		LSTM	1.0429	1.3639	0.0275
	BiLSTM	0.7599	1.0165	0.0228		BiLSTM	0.9528	1.2737	0.0252
	GRU	0.6749	0.9218	0.0202		GRU	0.7410	0.9755	0.0198
	CNN-LSTM	0.5254	0.5988	0.0161		CNN-LSTM	0.4435	0.5240	0.0119
	DeepAR	0.4351	0.3600	0.0091		DeepAR	0.6249	0.7364	0.0167
	DeepARA	**0.1835**	**0.2601**	**0.0062**		DeepARA	**0.3362**	**0.3823**	**0.0092**
Coal	SVM	2.8523	3.5207	0.0607	Food and Drink	SVM	9.4215	11.7765	0.0351
600188.SH	RNN	1.1509	1.4597	0.0262	600809.SH	RNN	4.9006	6.5995	0.0181
	LSTM	1.1184	1.4409	0.0258		LSTM	7.3710	9.2383	0.0269
	BiLSTM	1.1122	1.4200	0.0254		BiLSTM	5.3571	6.9183	0.0199
	GRU	1.0529	1.4261	0.0239		GRU	5.0515	6.9170	0.0188
	CNN-LSTM	0.7563	0.8834	0.0177		CNN-LSTM	3.7300	4.4221	0.0137
	DeepAR	**0.4918**	0.9101	0.0157		DeepAR	**2.1953**	**2.3830**	**0.0079**
	DeepARA	0.5562	**0.6506**	**0.0124**		DeepARA	4.4039	5.3155	0.0165
Automotive	SVM	7.9582	10.2177	0.0279	Electricity	SVM	18.6443	21.9715	0.0416
002594.SZ	RNN	5.5331	7.2615	0.0196	300750.SZ	RNN	11.3061	13.9979	0.0252
	LSTM	5.2001	6.9345	0.0184		LSTM	10.3719	12.7498	0.0228
	BiLSTM	5.7245	7.5788	0.0203		BiLSTM	11.6189	14.3010	0.0259
	GRU	5.3286	7.0216	0.0189		GRU	10.5738	13.0059	0.0233
	CNN-LSTM	4.0057	4.6731	0.0142		CNN-LSTM	7.7996	8.2888	0.0179
	DeepAR	5.6746	6.0875	0.0206		DeepAR	15.8704	17.1016	0.0363
	DeepARA	**3.8637**	**4.1560**	**0.0138**		DeepARA	**6.5135**	**7.0919**	**0.0151**

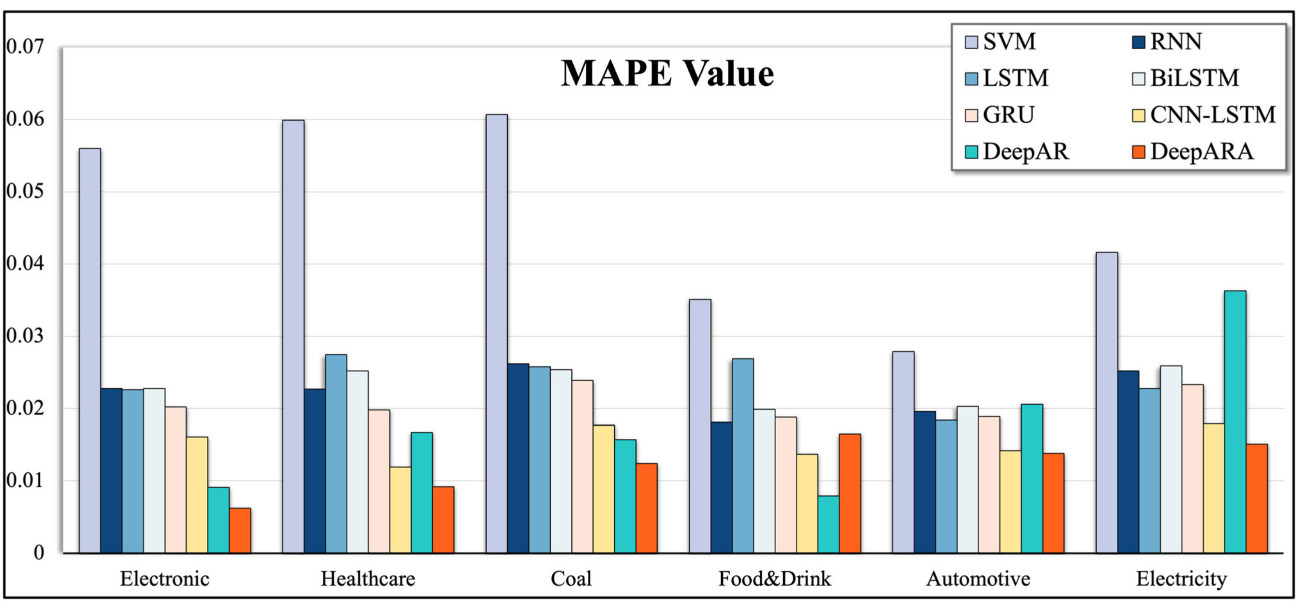


Fig. 7 The proposed DeepARA model is compared with baseline model on different industry test sets through their MAPE values

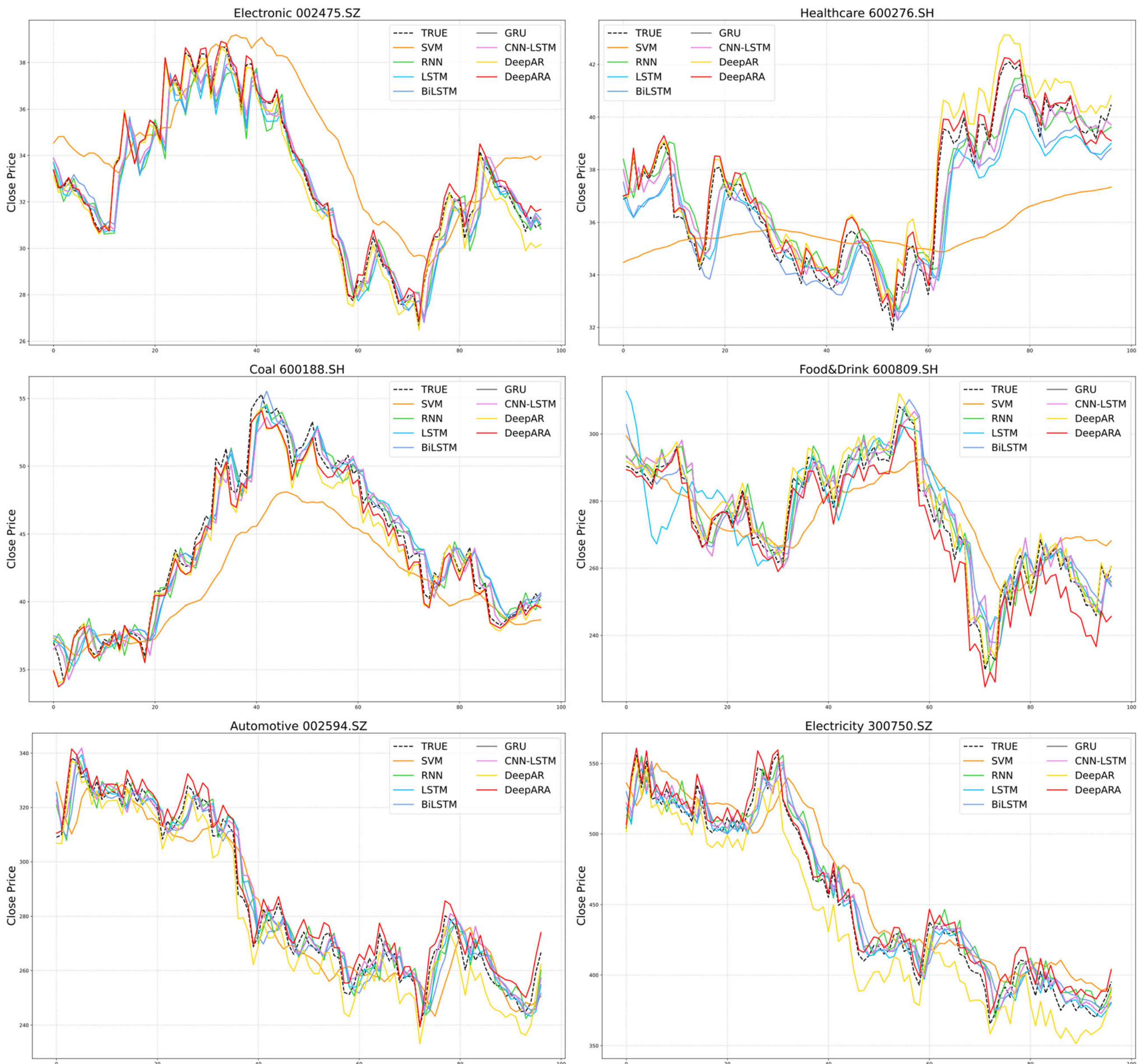


Fig. 8 The predicted curves of different models in the testing set

effective for tasks where context from both directions is important.

- GRU [19]: It is a streamlined version of the LSTM architecture. It simplifies the LSTM model by combining the forget and input gates into a single update gate, thereby reducing complexity while maintaining comparable performance, especially in tasks involving sequence modeling.
- CNN-LSTM [58]: The fusion of CNN and LSTM networks combines the robust feature extraction capabilities of CNN with the prowess of LSTM in predicting sequential data. This advanced modeling approach has become increasingly popular in recent years for predicting stock prices.
- DeepAR [24]: It is a probabilistic forecasting method using deep learning. It is particularly effective for time series data, predicting future values by learning patterns within the series and generating a range of possible future outcomes, often offering improved accuracy and uncertainty estimates compared to traditional methods.

5.2 Dataset

We procured daily closing prices of individual stocks for our experiments through the Baostock financial data

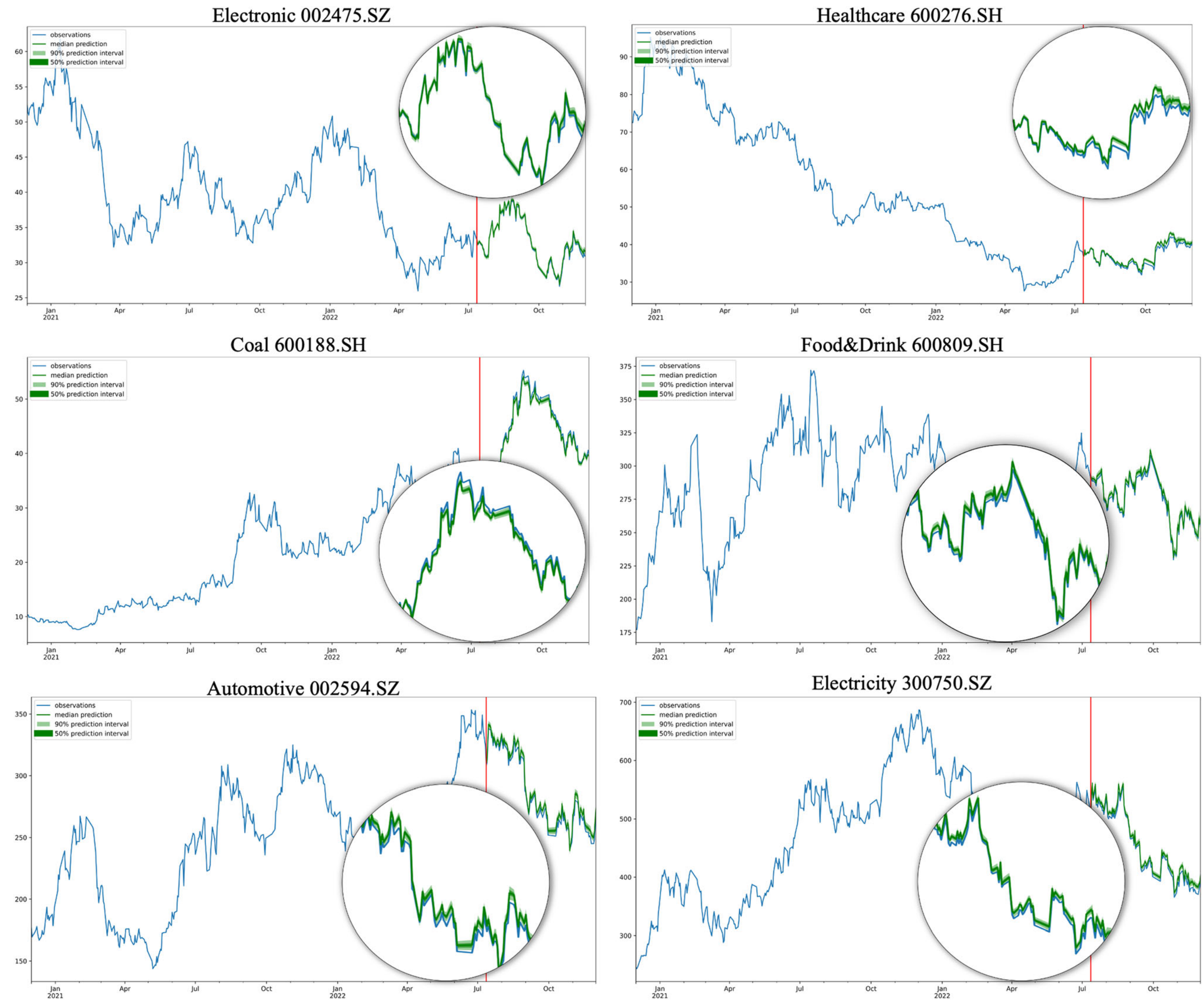


Fig. 9 DeepARA probability prediction results

interface. The dataset spans from December 1, 2020 to December 1, 2022 and includes 30 stocks across six different sectors. Within each sector, the top five companies were chosen based on their market capitalization. Table 1 provides a detailed list of the stock symbols and their corresponding sectors for these selected companies.

The dataset comprises 487 trading days' worth of stock data, with each day providing a quintet of crucial metrics: opening price, closing price, highest price, lowest price, and trading volume. For example, Table 2 presents a detailed snapshot of the electric power sector's stock, focusing on the ticker 002475.SZ. Furthermore, Fig. 5 displays the historical price fluctuations for three chosen stocks, illustrating the high, low, open, and close values. Our goal is to predict the closing price. We divided the dataset into two subsets: 80% for training and 20% for testing. The model training utilized daily price data from December 1, 2020 to July 11, 2022, spanning 390 days. The test phase evaluated performance from July 12, 2022 to December 1, 2022, over 97 days. We concentrated our examination on the price movements of individual stocks from 30 companies, selected for their notable volatility over the preceding two years and contrasted against broader stock indices. We anticipate that the model will incrementally refine its adaptations to capture the broader market trends over time.

5.3 Autoregressive analysis of stock price data

The autoregressive (AR) model relies on the linear relationships between historical data points. From Fig. 6 Autocorrelation Function (ACF) plot, it is evident that the autocorrelation of stock prices exists across various lags; however, this correlation diminishes as the lag increases (taking the data for Electronic 002475.SZ as an example). This indicates that while historical prices may have predictive value for future prices in the short term, the linear dependence may become unstable over longer periods, limiting the AR model's accuracy in forecasting long-term stock prices.

Although the ACF can reveal periodic characteristics and long-term dependencies within the data, it does not differentiate between direct and indirect causal relationships among these correlations. Consequently, ACF may overestimate the significance of certain lags, leading to model overfitting. From Fig. 6, Partial Autocorrelation Function (PACF) aids in identifying the number of AR terms that might be required in a model. However, in the presence of nonlinear relationships, seasonal effects, or external influencing factors within the data, PACF may be inadequate in capturing these complex features.

In this context, our DeepARA model becomes particularly crucial. Compared to traditional autoregressive models based on ACF and PACF analysis, deep learning models are capable of capturing nonlinear relationships in the data and addressing more complex patterns, such as long-term dependencies, periodic variations, and potential nonlinear interactions.

5.4 Hyperparameters setting

In all our experiments, we employed the ADAM optimizer along with early stopping and standard LSTM units. To determine the optimal hyperparameters, a series of experiments were conducted. We adopted a fixed batch size of 32 for mini-batch training and utilized mean squared error as the loss function to evaluate the discrepancy between the predicted and actual values. The DeepAR Estimator, with a learning rate set at 0.001, was used during training.

The DeepARA model features a confidence interval ranging from 50% to 90%, with the median value representing the single-point prediction. In its multi-head attention mechanism, the number of heads was set to eight, allowing each head to independently compute attention scores across the input sequence and capture different facets of the input data. Furthermore, we implemented a dropout rate of 0.1, a regularization strategy aimed at preventing overfitting and boosting the model's generalization capability. While higher dropout rates can potentially increase model robustness, they may also lead to diminished model performance.

5.5 Evaluation metrics

In the work of this paper, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) were selected as metrics for evaluating the accuracy and performance of our model's predictions.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\hat{y}_i - y_i|. \tag{19}$$

In the context of forecasting, $\hat{y}_i$ represents the predicted value, y_i is the true value, and N denotes the sample size. MAE is an abbreviation for Mean Absolute Error, which is a useful metric for avoiding mutual cancelation of errors and provides insights into the absolute errors in forecasting.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}, \tag{20}$$

where $\hat{y}_i$ indicates the predicted value, y_i refers to the true value, and N is the sample size. RMSE is a common metric for evaluating the accuracy of predictions, which refers to the calculation of the expected square deviation between the predicted and actual values. A lower RMSE suggests better model performance.

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N} \left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100\%. \tag{21}$$

In Eq. (21), $\hat{y}_i$ represents the predicted value, y_i is the true value, and N is the sample size. MAPE is a commonly used metric to evaluate the accuracy of a forecasting model. It is expressed as a percentage, and a lower value indicates better accuracy. However, MAPE has some limitations, such as its sensitivity to extreme values and its inability to handle zero or negative actual values. Furthermore, when actual values are very small or close to zero, MAPE can produce misleading results, leading to infinite or very large percentage errors. Therefore, it is important to use MAPE in combination with other metrics and to interpret the results with caution.

The three selected metrics evaluate the predictive accuracy of the model from distinct perspectives. A more optimal model will exhibit smaller values for these metrics, indicating better performance.

5.6 Analysis of experimental results

In this study, we compared the error metrics and predictive outcomes of our DeepARA model against several baseline models. Table 3 reports the values for three critical evaluation metrics. Metrics in bold indicate the best results. The smaller the evaluation metric, the more accurate the prediction. The values of MAE, RMSE, and MAPE listed in Table 3 clearly indicate that the probabilistic forecasting models DeepAR and DeepARA outperform the baseline methods, with our DeepARA model exhibiting the superior performance. Furthermore, our experiments revealed that among traditional single-point prediction models, the CNN-LSTM model shows the most promising results, surpassing the conventional machine learning method SVM, which in turn performs less effectively than RNN-based methods. We selected six industry-leading stocks from the Chinese stock market, characterized by their higher volatility relative to market indices. The results demonstrate that the error magnitude of basic single-point prediction models is higher than that of the DeepARA model. DeepARA excelled in the majority of dataset experiments, achieving the lowest evaluation metrics, thereby confirming its exceptional efficacy.

According to Fig. 7, the DeepARA model registers the lowest MAPE values in five distinct industries. However, in the Food and Drink sector, we observe that the MAPE value for DeepARA does not undercut that of DeepAR. This divergence may stem from the unique price volatility characteristics of the Food and Drink industry, where the timing of price pivots critically impacts forecast accuracy and includes a higher degree of noise or nonlinear features. Consequently, the introduction of the attention mechanism did not enhance the model's performance as anticipated but instead led to a decrease in the fit to the actual data. In this context, the foundational DeepAR model, due to its simplicity, is more resilient to noise and maintains effective trend prediction. This suggests that the attention mechanism may require further adjustment and optimization within specific industry contexts and data attributes to better capture the price movement patterns in the Food and Drink industry. Nonetheless, the overall performance of DeepARA remains advantageous compared to other benchmark models.

Figure 8 depicts the results of our single-point prediction experiments conducted within the testing set, which includes six representative datasets. As illustrated by the red line in Fig. 8, our model's forecasts closely align with the actual stock prices, highlighting the model's accuracy and superiority. In contrast to baseline models, our model's incorporation of attention mechanisms provides more granular precision in identifying peaks and inflection points. This accuracy is particularly pronounced in the stock predictions for Electronic 002475.SZ and Electricity 300750.SZ, where DeepARA demonstrates an increased responsiveness to price volatility contrasting the delayed response of traditional models. While the machine learning method SVM can grasp the overall trend of stock prices, it fails to predict specific prices, peaks, and critical turning points with precision. This deficiency is evident in the significant deviation of SVM predictions (represented by the orange line) from the actual stock prices. In comparison, RNN-based models show a higher accuracy in forecasting stock prices. Moreover, under the same experimental conditions, we observed that the LSTM and BiLSTM models have similar levels of predictive accuracy. However, as an advanced hybrid method, CNN-LSTM effectively enhances prediction precision with reduced errors in relation to the actual data, but their performance is still outshined by DeepARA, indicating that our model further improves forecasting capabilities.

The DeepARA model is capable of making single-point price predictions and also predicts the probability distribution of time series, offering a more comprehensive risk assessment than traditional RNN models. In the field of stock price forecasting, this probabilistic approach more closely aligns with the real-world probability distribution of stock prices, making it more meaningful than single-point predictions alone. Additionally, we used the median of this probability distribution as the basis for single-point prediction, to compare with baseline models, and found that it performs better than these baselines. Figure 9 displays the predictive results of the DeepARA model. Actual data from the training and test sets are shown with a blue line in the graph. The green shaded area indicates the confidence interval, denoting the range of probabilistic predictions for the test set, from 50% to 90% probability. Darker green represents a 50% confidence interval, while lighter green shows a 90% confidence interval. This probabilistic distribution method significantly improves the precision of our predictions. Our comparative analysis further shows that the DeepARA model surpasses all other methods in prediction accuracy across all datasets.

6 Conclusions

In this paper, we propose the DeepARA model, an innovative approach that integrates the DeepAR model with the attention mechanism. We conducted a performance evaluation of this approach against machine learning methods such as SVM and recurrent neural network techniques, including RNN, LSTM, BiLSTM, GRU, and CNN-LSTM specifically in the context of stock market forecasting. Our prediction experiments focus on stocks from six major industries in the China stock market, with results from six selected groups presented. The findings demonstrate that the DeepARA model outperforms traditional deep learning models in both single-point prediction accuracy and probabilistic prediction. This underscores the potential of the DeepARA architecture for financial time series forecasting. In practical terms, investors may be able to leverage the predictions made by the DeepARA model to achieve higher returns. The favorable performance of DeepARA is largely attributed to its integration of an attention mechanism, which assigns different weights to key moments within the model. It is important to note that financial time series, particularly those involving stock index trends, often display significant fluctuations. Consequently, DeepAR's probabilistic prediction capability enables the model to determine more accurate stock price ranges. This feature contributes to DeepARA's enhanced predictive performance compared to conventional RNN structures.

This paper demonstrates the effectiveness of the DeepARA model in predicting individual stock prices, as supported by empirical evidence. Future research will delve into the theoretical aspects of this domain. However, this study also has its limitations, as it solely focuses on the prediction of individual stocks in the Chinese stock market without considering the overall market trends. It is crucial to acknowledge the significant impact of government policies on the volatility of Chinese stock prices, which often exceeds that of the global market. Consequently, our future research aims to incorporate news events and policy information to enhance stock market predictions. Additionally, while this paper centers on mid-term predictions, investigating high-frequency stock price predictions for returns is also a promising avenue.

Acknowledgments This work was supported in part by grants from National Science Foundation of China (61571005) and the fundamental research program of Guangdong, China (2020B1515310023, 2023A1515011281).

Author contributions JL contributed to conception and design, data collection, software, analysis and interpretation of results, and writing—original and editing. WC contributed to data visualization, program modification, and writing—reviewing draft. ZZ contributed to writing—reviewing draft and supervision. JY contributed to writing—reviewing draft. DZ contributed to analysis and interpretation of results, writing—review draft, and supervision.

Data availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical approval Our study followed ethical guidelines and obtained informed consent from all participants regarding the data used.

References

1. Ou JA, Penman SH (1989) Financial statement analysis and the prediction of stock returns. J Account Econom 11(4):295–329
2. Wang J-H, Leu J-Y (1996) Stock market trend prediction using arima-based neural networks. In: proceedings of international conference on neural networks (ICNN'96), vol. 4, pp. 2160–2165. IEEE
3. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297
4. Ni L-P, Ni Z-W, Gao Y-Z (2011) Stock trend prediction based on fractal feature selection and support vector machine. Expert Syst Appl 38(5):5569–5576
5. Thenmozhi M, Sarath Chand G (2016) Forecasting stock returns based on information transmission across global markets using support vector machines. Neural Comput Appl 27:805–824
6. Dash RK, Nguyen TN, Cengiz K, Sharma A (2023) Fine-tuned support vector regression model for stock predictions. Neural Comput Appl 35(32):23295–23309
7. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536
8. Wang J-Z, Wang J-J, Zhang Z-G, Guo S-P (2011) Forecasting stock indices with back propagation neural network. Expert Syst Appl 38(11):14346–14355
9. Singh R, Srivastava S (2017) Stock prediction using deep learning. Multimedia Tools Appl 76(18):18569–18584
10. Ding X, Zhang Y, Liu T, Duan J (2015) Deep learning for event-driven stock prediction. In: Twenty-fourth international joint conference on artificial intelligence
11. Yu P, Yan X (2020) Stock price prediction based on deep neural networks. Neural Comput Appl 32:1609–1628
12. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
13. Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci 79(8):2554–2558
14. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
15. Ma Q, Li S, Zhuang W, Wang J, Zeng D (2020) Self-supervised time series clustering with model-based dynamics. IEEE Trans Neural Netw Learn Syst 32(9):3942–3955
16. Selvin S, Vinayakumar R, Gopalakrishnan E, Menon VK, Soman K (2017) Stock price prediction using lstm, rnn and cnn-sliding window model. In: 2017 international conference on advances in computing, communications and informatics (ICACCI), pp. 1643–1647. IEEE
17. Chen K, Zhou Y, Dai F (2015) A lstm-based method for stock returns prediction: a case study of china stock market. In: 2015 IEEE International Conference on Big Data (big Data), pp. 2823–2824. IEEE
18. Mukherjee S, Sadhukhan B, Sarkar N, Roy D, De S (2023) Stock market prediction using deep learning algorithms. CAAI Trans Intell Technol 8(1):82–94
19. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078
20. Shen G, Tan Q, Zhang H, Zeng P, Xu J (2018) Deep learning with gated recurrent unit networks for financial sequence predictions. Proced Comput Sci 131:895–903
21. Jin Z, Yang Y, Liu Y (2020) Stock closing price prediction based on sentiment analysis and lstm. Neural Comput Appl 32(13):9713–9729
22. Yoo PD, Kim MH, Jan T (2005) Machine learning techniques and use of event information for stock market prediction: A survey and evaluation. In: international conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06), vol. 2, pp. 835–841. IEEE
23. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inform Process Syst 30:17
24. Salinas D, Flunkert V, Gasthaus J, Januschowski T (2020) Deepar: probabilistic forecasting with autoregressive recurrent networks. Int J Forecast 36(3):1181–1191

25. Liao Y, Liang C (2021) A temperature time series forecasting model based on deepar. In: 2021 7th international conference on computer and communications (ICCC), pp. 1588–1593. IEEE
26. Arora P, Jalali SMJ, Ahmadian S, Panigrahi BK, Suganthan P, Khosravi A (2022) Probabilistic wind power forecasting using optimised deep auto-regressive recurrent neural networks. IEEE Trans Ind Inform 19(3):2814–2825
27. Jiang F, Han X, Zhang W, Chen G (2021) Atmospheric pm2.5 prediction using deepar optimized by sparrow search algorithm with opposition-based and fitness-based learning. Atmosphere 12(7):894
28. 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
29. Xu H, Chai L, Luo Z, Li S (2022) Stock movement prediction via gated recurrent unit network based on reinforcement learning with incorporated attention mechanisms. Neurocomputing 467:214–228
30. Chen Y-J, Chen Y-M (2013) A fundamental analysis-based method for stock market forecasting, 354–359
31. 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
32. Kumbure MM, Lohrmann C, Luukka P, Porras J (2022) Machine learning techniques and data for stock market forecasting: a literature review. Expert Syst Appl 197:116659
33. Greig AC (1992) Fundamental analysis and subsequent stock returns. J Account Econom 15(2–3):413–442
34. Mittal A, Goel A (2012) Stock prediction using twitter sentiment analysis. Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf) 15, 2352
35. Makrehchi M, Shah S, Liao W (2013) Stock prediction using event-based sentiment analysis. In: 2013 IEEE/WIC/ACM international joint conferences on web intelligence (WI) and intelligent agent technologies (IAT), vol. 1, pp. 337–342. IEEE
36. Lee H, Surdeanu M, MacCartney B, Jurafsky D (2014) On the importance of text analysis for stock price prediction. LREC 2014:1170–1175
37. Nguyen TH, Shirai K, Velcin J (2015) Sentiment analysis on social media for stock movement prediction. Expert Syst Appl 42(24):9603–9611
38. Akita R, Yoshihara A, Matsubara T, Uehara K (2016) Deep learning for stock prediction using numerical and textual information. In: 2016 IEEE/ACIS 15th international conference on computer and information science (ICIS), pp. 1–6
39. Sohangir S, Wang D, Pomeranets A, Khoshgoftaar TM (2018) Big data: deep learning for financial sentiment analysis. J Big Data 5(1):1–25
40. Nourbakhsh Z, Habibi N (2022) Combining lstm and cnn methods and fundamental analysis for stock price trend prediction. Multim Tools Appl 1–31
41. Cavalcante RC, Brasileiro RC, Souza VL, Nobrega JP, Oliveira AL (2016) Computational intelligence and financial markets: a survey and future directions. Expert Syst Appl 55:194–211
42. Teixeira LA, De Oliveira ALI (2010) A method for automatic stock trading combining technical analysis and nearest neighbor classification. Expert Syst Appl 37(10):6885–6890
43. Long W, Lu Z, Cui L (2019) Deep learning-based feature engineering for stock price movement prediction. Knowl Based Syst 164:163–173
44. Jiang W (2021) Applications of deep learning in stock market prediction: recent progress. Expert Syst Appl 184:115537
45. Ariyo AA, Adewumi AO, Ayo CK (2014) Stock price prediction using the arima model. In: 2014 UKSim-AMSS 16th international conference on computer modelling and simulation, pp. 106–112. IEEE
46. Adebiyi AA, Adewumi AO, Ayo CK (2014) Comparison of arima and artificial neural networks models for stock price prediction. J Appl Math 2014:4
47. Shin D-H, Choi K-H, Kim C-B (2017) Deep learning model for prediction rate improvement of stock price using rnn and lstm. J Korean Inst Inform Technol 15(10):9–16
48. Chen RT, Rubanova Y, Bettencourt J, Duvenaud DK (2018) Neural ordinary differential equations. Adv Neural Inform Process Syst 31:18
49. Du S, Luo Y, Chen W, Xu J, Zeng D (2022) To-flow: Efficient continuous normalizing flows with temporal optimization adjoint with moving speed. In: proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 12570–12580
50. Yang F, Chen J, Liu Y (2023) Improved and optimized recurrent neural network based on pso and its application in stock price prediction. Soft Comput 27(6):3461–3476
51. Chen Y, Wu J, Wu Z (2022) China's commercial bank stock price prediction using a novel k-means-lstm hybrid approach. Expert Syst Appl 202:117370
52. Kanwal A, Lau MF, Ng SP, Sim KY, Chandrasekaran S (2022) Bicudnnlstm-1dcnn-a hybrid deep learning-based predictive model for stock price prediction. Expert Syst Appl 202:117123
53. Li H, Shen Y, Zhu Y (2018) Stock price prediction using attention-based multi-input lstm. In: Asian conference on machine learning, pp. 454–469. PMLR
54. Cui C, Li X, Zhang C, Guan W, Wang M (2023) Temporal-relational hypergraph tri-attention networks for stock trend prediction. Patt Recogn 143:109759
55. Lu W, Li J, Wang J, Qin L (2021) A cnn-bilstm-am method for stock price prediction. Neural Comput Appl 33:4741–4753
56. Wang J, Hu Y, Jiang T-X, Tan J, Li Q (2023) Essential tensor learning for multimodal information-driven stock movement prediction. Knowl Based Syst 262:110262
57. Pei W, Baltrusaitis T, Tax DM, Morency L-P (2017) Temporal attention-gated model for robust sequence classification. In: proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6730–6739
58. Rostamian A, O'Hara JG (2022) Event prediction within directional change framework using a cnn-lstm model. Neural Comput Appl 34(20):17193–17205