Expert Systems With Applications 237 (2024) 121424

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Series decomposition Transformer with period-correlation for stock market index prediction

Zicheng Tao [a], Wei Wu [a,b,*], Jianxin Wang [a,c]

[a] School of Computer Science and Engineering, Central South University, Changsha, 410083 Hunan, China
[b] Xiangjiang Laboratory, Changsha, 410205 Hunan, China
[c] BANK OF SANXIANG, Changsha, 410023 Hunan, China

ARTICLE INFO

Keywords:
Stock price prediction
Financial time series
Transformer
Deep learning

ABSTRACT

Stock price forecasting has been always a difficult and crucial undertaking in the field of finance. In the last few decades, deep learning models based on RNNs and LSTMs have dominated the research, where the stock price data are modeled as time series data. However, the high volatility of stock prices and the decay of information learned from historical data prevented these models from achieving more accurate predictions in this problem. Recently, Transformer has been gradually applied in time series prediction, but the methods aim to feed the highly-uncertain social media information as the additional auxiliary information into Transformer, rather than improving the ability to extract features from historical series. In this paper, we propose a Series Decomposition Transformer with Period-correlation (SDTP), which uses the period-correlation mechanism and series decomposition layers to further discover relation between historical series and learn the changing trends in the stock market for high forecasting accuracy and generalizability. The extensive experimental results show that the proposed SDTP model generally outperforms the state-of-the-art methods on a collection of datasets.

1. Introduction

As the world economy continues to grow, an increasing number of researchers from different fields are devoted to this area. Among many problems in financial markets, stock price forecasting is a challenging and very important task, because it helps the researchers to analyze stock market changes and to make the correct investment decisions about the stock market (Heaton, Polson, & Witte, 2017). For individual investors, accurate stock price forecasting can have profound influence on their investments and asset management. For large companies and organizations, accurate stock price forecasting can help them plan their capital and develop sound business strategies (Chen, Wu, & Wu, 2022). However, the changes in stock prices are not simply linear, because they are often influenced by many market and extra-market factors, e.g., historical stock price, economic factors of the companies, the sentiment of investors, political influence from home and abroad, etc (Kohara, Fukuhara, & Nakamura, 1996). Many researchers have made huge attempts to solve this problem.

Statistical methods were first used to solve this problem including Moving Average (Wu & Chau, 2013), Auto Regressive Moving Average (ARMA) (Rounaghi & Zadeh, 2016), Auto Regressive Integrated Moving Average (ARIMA) (Zhang, Zhang, & Feng, 2016), Auto Regressive Conditional Heteroskedasticity (ARCH) (Baldauf & Santoni, 1991), and Generalized Auto Regressive Conditional Heteroskedasticity (GARCH) (Yamaguchi, 2008), etc. The approaches tend to model under the ideal assumptions, which make them efficient in terms of time, but it is difficult to discover the correlated information in data from large stock markets. To this end, some researchers resort to the machine learning techniques, e.g., K-Nearest Neighbor (KNN) (Alkhatib, Najadat, Hmeidi, & Shatnawi, 2013), decision tree (Nair, Mohandas, & Sakthivel, 2010), Support Vector Machine (SVM) (Lin, Guo, & Hu, 2013), random forest (Khaidem, Saha, & Dey, 2016) and XGBoost (Yun, Yoon, & Won, 2021). Although the methods can perform well in some stock prediction (Vadlamudi, 2017), they are hardly extended to complex stock markets due to limited capability of feature extraction. By contrast, the deep learning techniques, e.g., Deep Neural Network (DNN) (Ticknor, 2013), Convolutional Neural Network (CNN) (Dingli & Fournier, 2017; Wu, Li, Srivastava, Tasi, & Lin, 2021), Recurrent Neural Network (RNN) (Hochreiter, Bengio, Frasconi, & Schmidhuber, 2001; Jiang, Pan, Jiang, & Long, 2018) and Long Short-Term Memory (LSTM) (Budiharto, 2021; Shah, Campbell, & Zulkernine, 2018; Wu, Wu, Hung, Hassan, & Fortino, 2020), show the ability to learn the hidden patterns from the huge historical data in the complex stock markets. However, the high volatility of stock prices and the decay of information learned from historical data kept the models

* Corresponding author at: School of Computer Science and Engineering, Central South University, Changsha, 410083 Hunan, China.
E-mail addresses: taozicheng@csu.edu.cn (Z. Tao), wei.wu@csu.edu.cn (W. Wu), jxwang@mail.csu.edu.cn (J. Wang).

https://doi.org/10.1016/j.eswa.2023.121424
Received 1 February 2023; Received in revised form 17 July 2023; Accepted 31 August 2023
Available online 9 September 2023

from attaining more accurate results. Motivated by the success of the Transformer models in other fields (Devlin, Chang, Lee, & Toutanova, 2018; Dosovitskiy et al., 2020; Zhou et al., 2021), an increasing number of researchers adopt the Transformer technique to solve the financial time series forecasting problems (Ding, Wu, Sun, Guo, & Guo, 2020; Li, Lv, Liu, & Zhang, 2022; Liu et al., 2019; Sonkiya, Bajpai, & Bansal, 2021; Zhang et al., 2022). However, in the above methods, Transformer is mainly used to analyze social information and obtain sentiment information as the additional auxiliary information, instead of improving the ability to extract features from historical series. Unfortunately, the social information from multiple sources is difficult to obtain and has a high level of uncertainty, because the information is easily impacted by many factors, e.g., the collection methods, the data sources and the commenters. Therefore, these uncertainties may make the methods perform unstably across different stock markets.

In this paper, we propose a Series Decomposition Transformer with Period-correlation (SDTP) to predict the closing price of the stock index, which has a better capability of feature extraction from the time series data and generalization to other stock markets. Unlike the traditional attention methods that update information by directly aggregating all the nodes, we first estimate the period of the series by selecting the time lags of the time lag series with the highest similarities to the series itself and then divide the series into sub-series based on the periods with the highest similarities and finally, update information by aggregating the related nodes in the period-based sub-series. This is due to the fact that the nodes in the corresponding positions in the similar period-based sub-series have similar properties. Consequently, the proposed SDTP method is more reasonable for stock data and acquires more accurate prediction. We observe that the stock markets are highly fluctuating, so it is difficult to analyze stock data directly. To this end, we decompose the series into the trend and seasonal parts, which helps the model learn the complicated changing patterns of the financial series without relying on highly-uncertain semantic information, so it can perform well across different stock markets. We conduct extensive empirical evaluation of the proposed SDTP model and a number of the state-of-the-art methods on a collection of well-known stock markets. In summary, our contributions are three-fold:

1. We propose a novel Transformer-based model called SDTP, which uses series decomposition layers and period-correlation mechanism to handle stock timing series data.
2. To the best of our knowledge, this is the first endeavor to adopt Transformer to forecast the stock closing price of the following day based on the historical price, rather than making predictions by feeding highly-uncertain social text into Transformer.
3. The extensive experimental results show that the proposed SDTP model outperforms the state-of-the-art methods in different stock markets, which demonstrates that it is a more appropriate choice for stock price prediction.

The remaining part of this paper is structured as follows. The literature review is introduced in Section 2. We introduce the preliminary knowledge in Section 3. Our proposed model is explained in Section 4. Section 5 describes the results of the experiment. We conclude this paper in Section 6.

2. Related works

Research into stock forecasting has become increasingly active in the last decades because the financial markets are very important for the country and they permeate all aspects of everyday lives. However, considering the vast volume of stock data and the intricacy of the stock market, stock price prediction has remained a challenging problem so far. The problem of predicting stock prices can be modeled as the time-series problem: we predict the future values $x_{t+1}, x_{t+2}, \dots, x_{t+3}$ based on the known time series, $x_1, x_2, \dots, x_t$. In general, there are three types of stock price forecasting methodologies: conventional statistical models, machine learning models, and deep learning models.

2.1. Statistical models

The statistical approaches are based on a subjective model, which gives empirical prediction via past data. The commonly used methods of statistical learning are Auto Regressive Moving Average (ARMA) (Rounaghi & Zadeh, 2016), Auto Regressive Integrated Moving Average (ARIMA) (Zhang et al., 2016), Auto Regressive Conditional Heteroskedasticity (ARCH) (Baldauf & Santoni, 1991) and Generalized Auto Regressive Conditional Heteroskedasticity (GARCH) (Yamaguchi, 2008), etc. Because of their simplicity and low complexity, the models are frequently employed in a variety of sectors. Ariyo, Adewumi, and Ayo (2014) used ARIMA to predict the stock price from New York Stock Exchange (NYSE) and Nigeria Stock Exchange (NSE), and show that ARIMA is potential for short-term forecasting. Faced with the substantial volatility of stock markets, Babu and Reddy (2014) proposed a linear mixed model based on ARIMA and GARCH, which aims to preserve the trend of the data. Singh, Parmar, and Kumar (2021) used discrete wavelet transformation combined with autoregressive to forecast the weekly and daily closing prices of the BSE100 and S&P Sensex index. The above methods tend to model under the ideal assumptions, but the stock market is a complicated system where numerous variables and uncertainties, e.g., noise, policies and human manipulation, interact with each other. Therefore, the methods are not practical.

2.2. Machine learning models

The machine learning approaches have become popular in financial data analysis, because they achieve better performance by fitting the data distribution. Alkhatib et al. (2013) used K-Nearest Neighbor (KNN) with non-linear regression method to assist investors in making better-investing selections. Nair et al. (2010) used technical indicators to extract features from historical data and then used decision trees for feature selection, and a rough set-based system was used to induce rules from the extracted features. Lin et al. (2013) explored SVM to develop a forecasting model for the stock price movement. The model outperformed some traditional models in accuracy on the Taiwan stock market. To reduce market and investment risks, Khaidem et al. (2016) proposed an ensemble learning model based on a random forest classifier to predict the returns of a stock. However, the above methods generally ignore the key indicators and feature engineering in the financial markets. To this end, Yun et al. (2021) presented a predictive model based on ga-xgboost, which enhanced feature expansion, data preprocessing, and ideal feature choosing. Mohanty, Parida, and Khuntia (2021) developed a hybrid model based on Auto Encoder (AE) and Kernel Extreme Learning Machine (KELM) to improve the accuracy of stock forecasting. Although the methods can represent nonlinear relationships well, it is not enough to help investors in the complex and large stock markets.

2.3. Deep learning models

With the huge increasing of the computing power, the deep learning techniques, e.g., Convolutional Neural Network (CNN) (LeCun et al., 1989), Recurrent Neural Network (RNN) (Rumelhart, Hinton, & Williams, 1986), Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), Transformer (Vaswani et al., 2017), etc., have witnessed much success in a wide range of fields and also provide a new way of solving stock prediction problems. Dingli and Fournier (2017) used the CNN model to forecast the price direction of the following day with the current price. Wu, Li, et al. (2021) used an array as the input graph for the CNN framework which includes all the relevant information about the stock. Since the RNN model can use the historical information when propagating forward, it has been widely used for time series data. Jiang et al. (2018) adopted an RNN-based model, which combines attention mechanisms to analyze intraand cross-domain relations, to capture the interactions of time series in financial data. Unfortunately, Hochreiter et al. (2001) discovered that the RNN models might cause gradients to disappear or explode, which is dangerous when dealing with long time sequence dependencies. Consequently, LSTM, which incorporates gate logic units into RNN, is proposed as a solution to this issue, and has become the most used model in predicting stocks in recent years. Shah et al. (2018) pointed out that LSTM was more suitable for long-term forecasting by predicting the daily and weekly fluctuations of the BSE Sensex index in India. Budiharto (2021) proposed a data analytic model based on LSTM for predicting stock prices on the Indonesian stock exchange. Note that LSTM learns historical information by iteratively passing information from the previous node to the following node, but decay is unavoidable in the forward propagation process, which prevents it from capturing the shifting patterns of time series. By contrast, Transformer learns information from all nodes in the entire series simultaneously, which avoids information loss during the iterative training. Therefore, a growing number of academics are turning to Transformer to tackle financial time series forecasting problems.

Liu et al. (2019) firstly applied Transformer to the financial problems. Ding et al. (2020) used Multi-Scale Gaussian Prior to enhance the locality of Transformer. Li et al. (2022) established an attention model based on Transformer that captures the dependence of the financial series by using social information and stock prices. Zhang et al. (2022) used LSTM and Transformer to predict stock movements by integrating the text information into the real stock prices. Sonkiya et al. (2021) first used a Transformer-based model, BERT (Devlin et al., 2018), to perform sentiment analysis on news and headlines, and then used Generative Adversarial Network (GAN) (Goodfellow et al., 2020) to predict stock prices with the technical indicators, the national stock indices, the selected commodities, the historical prices, and the sentiment scores. These methods, however, do not improve the ability to extract features from the historical series, but rather aid prediction by employing Transformer to analyze social data and further obtain sentiment information. We would like to note that the social information from numerous sources is difficult to obtain and highly-uncertain, which makes the methods perform unstably across different stock markets.

To get a deep learning model with excellent prediction accuracy across different stock markets, we present a novel deep learning model SDTP to anticipate the stock price of the following day via the previous price. To better discover the relationship between financial series and get more accurate forecasts, we use period-correlation instead of the traditional attention mechanism. In addition, we add series decomposition layers inside the SDTP model, which decomposes the financial time series to learn the complex patterns without utilizing highly-uncertain semantic information, so the SDTP model can perform well in different stock markets.

3. Preliminaries

3.1. Problem definition

Forecasting stock prices aims to predict the closing price of the following day, $\hat{x}_{I+1}$, by utilizing data from the information of the previous period, $X = (x_1, x_2, \dots, x_I)$, where $X \in \mathbb{R}^{I\times d}$, d represents the feature latitude and I represents the input time length (usually 5 or 10 days Zhang et al., 2022). The task is run on a rolling manner, with the next prediction starting at x_2.

3.2. Transformer

The Transformer model has had a huge influence on the NLP field, making a huge performance improvement in the NLP tasks. The Transformer model can substantially outperform the RNN model on the large-scale datasets. Fig. 1 depicts the architecture of the Transformer model (Vaswani et al., 2017), which is divided into two parts: the encoder and the decoder. The input of the encoder consists of node embedding with position embedding and is then passed through the self-attention layer, which helps the encoder utilize all the node information in the entire series simultaneously and is then followed by residual connection to avoid network decline and layer normalization to normalize the input. By residual connection and normalization of the output of the forward propagation network, we obtain the final output of the encoder. The network of the decoder is similar to the encoder, except that mask training is used to enhance the learning capacity of the model, and information from the encoder is utilized as the extra features for the self attention layer. Finally, the scores are transformed into the probabilistic outputs by softmax.

Table 1
Definition of symbols.

Symbol	Definition
X_{en}	The input of encoder
X_{ent}	The trend part of X_{en}
X_{ens}	The seasonal part of X_{en}
X_{en}^{l}	The output of the lth encoder
$S_{en}^{l,i}$	The seasonal part of the ith series decomposition layer in the lth encoder
X_{det}	The trend part of the decoder input
X_{des}	The seasonal part of the decoder input
X_{de}^{l}	The output of the lth decoder
$S_{de}^{l,i}$	The seasonal part of the ith series decomposition layer in the lth encoder
$T_{de}^{l,i}$	The trend part of the ith series decomposition layer in the lth encoder
τ	Time lag
Q, K, V	Query, key, and value arrays

4. The proposed model: SDTP

To deal with the complex changing patterns of the financial series, we propose the SDTP model with the series decomposition layer and period-correlation mechanism. We will introduce these two components carefully in Section 4.1 and Section 4.2, respectively. In order to help understand the operation process of the entire model, we describe the process of encoder and decoder in Section 4.3 and Section 4.4, respectively. Some of the key symbols are listed in Table 1.

4.1. Series decomposition layer

Rozeff and Kinney Jr (1976) provides evidence for the existence of seasonality in the stock market, and points out that the trend is a detectable change in the stock market time series. It is difficult to learn the complex temporal patterns directly in the financial series, and we are inspired by the work of Cleveland (1990), Taylor and Letham (2018), Wu, Xu, Wang, and Long (2021), which decomposes the series into the trend and seasonal parts. To control the consistent length of the series, we pad the series by the $Padding$ operation. The method uses $AvgPool$ to estimate the overall trend of the series and then subtracts the trend component from the series to obtain the seasonal component, which can accurately represent the process of time series changes. We add series decomposition layers inside the Transformer model, which can extract information from the intermediate hidden variables. For the input length-L series $X \in \mathbb{R}^{L\times d}$, the series decomposition is performed as follows:

$$\begin{aligned} X_t &= AvgPool(Padding(X)) \\ X_s &= X - X_t \end{aligned} \tag{1}$$

where X_s and X_t represent the season and trend parts of X, respectively. The $AvgPool(\cdot)$ means using moving averages to estimate the trend. $Padding(\cdot)$ aims to keep the data length consistent, and we implement it by copying the data at the end. For convenience we simplify the above formula as SeriesDecomp(X).

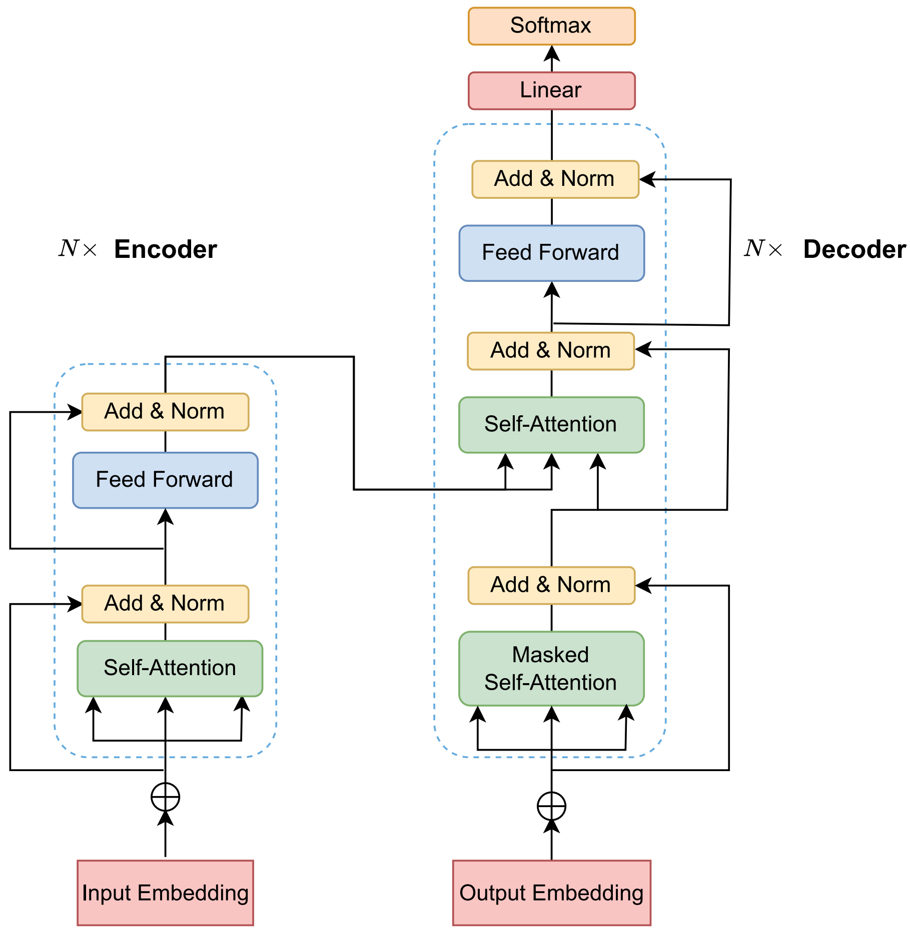

Fig. 1. The architecture of the original Transformer model.

4.2. Period-correlation mechanism

Traditional self-attention directly uses the features of all nodes to update the information without taking the periodicity between the sub-series into account. Observing that the nodes in the corresponding positions in the period-based sub-series usually have similar properties to each other (Xu, Li, Qian, & Wang, 2022), we are inspired by Wu, Xu, et al. (2021) to use period-correlation instead of traditional self-attention, which can detect the periodic relationship between the series more reasonably. We first calculate the similarities between the series itself and the time lag series of it. Then we choose the m time lags with the highest similarities to estimate the period and then divide the series into sub-series based on the periods. We normalize the similarity scores of the sub-series by $SoftMax$, and finally, aggregate the sub-series by $SoftMax$ scores to get the period-correlation. For a discrete-time series X_T, the period-correlation is calculated by:

$$\begin{aligned} &S_{xx}(\tau) = \frac{1}{L}\sum_{T=1}^{L} X_T X_{T-\tau} \\ &\tau_1,\dots,\tau_m = Topm_{\tau\in\{1,\dots,L\}}(S_{Q,K}(\tau)) \end{aligned} \tag{2}$$

where $X_{T-\tau}$ denotes the τ lag series of X_T, and $S_{xx}(\tau)$ denotes the similarity between the series X_T and $X_{T-\tau}$. $Topm(\cdot)$ means choosing the first m arguments of maximum similarities, where $m = \lfloor c \times \log L \rfloor$ and c is a hyper-parameter. Q, K and V represent query, key, and value arrays, respectively, which are the same as those in traditional self-attention.

$$\begin{aligned} &\hat{S}_{Q,K}(\tau_i) = SoftMax(S_{Q,K}(\tau_i)), i \in \{1,\dots,m\} \\ &Period\text{-}Correlation(Q,K,V) = \sum_{i=1}^{m} Roll(V,\tau_i)\hat{S}_{Q,K}(\tau_i) \end{aligned} \tag{3}$$

where $S_{Q,K}$ denotes the similarity between series Q and K, and $Roll(V,\tau)$ denotes series V after time lag τ, during which the element that is moved out of the first position is brought back to the last position.

4.3. Encoder

The input of the encoder is the original time series $X_{en} \in \mathbb{R}^{I\times d}$, where I denotes the time length. As shown in Fig. 2, the encoder models the seasonality and trend of the series through the series decomposition operation, and then puts the output into the decoder as supplementary information for prediction. Suppose that there are N encoder layers. The calculation process is as follows:

$$\begin{aligned} &S_{en}^{l,1}, \_ = SeriesDecomp(Period\text{-}Correlation(X_{en}^{l-1}) + X_{en}^{l-1}) \\ &S_{en}^{l,2}, \_ = SeriesDecomp(FeedForward(S_{en}^{l,1}) + S_{en}^{l,1}) \end{aligned} \tag{4}$$

where $FeedForward$ is the simplest neural network, X_{en}^{l}, $l \in \{1,\dots,N\}$ is the output of the lth encoder, "_" represents the trend part that is not used, $S_{en}^{l,1}$ and $S_{en}^{l,2}$ denote the seasonal parts in the first and the second series decomposition layer in the lth encoder, respectively, and $S_{en}^{l,2}$ is equal to the output of the lth encoder X_{en}^{l}.

4.4. Decoder

The input of the decoder is divided into two parts: the seasonal part $X_{des} \in \mathbb{R}^{(I+1)\times d}$ and the trend part $X_{det} \in \mathbb{R}^{(I+1)\times d}$. They are initialized by the seasonal part X_{ens} of the encoder with 0 padding and the trend part X_{ent} of the encoder with the mean of series padding, respectively. Assume that M decoder layers exist. The following is the

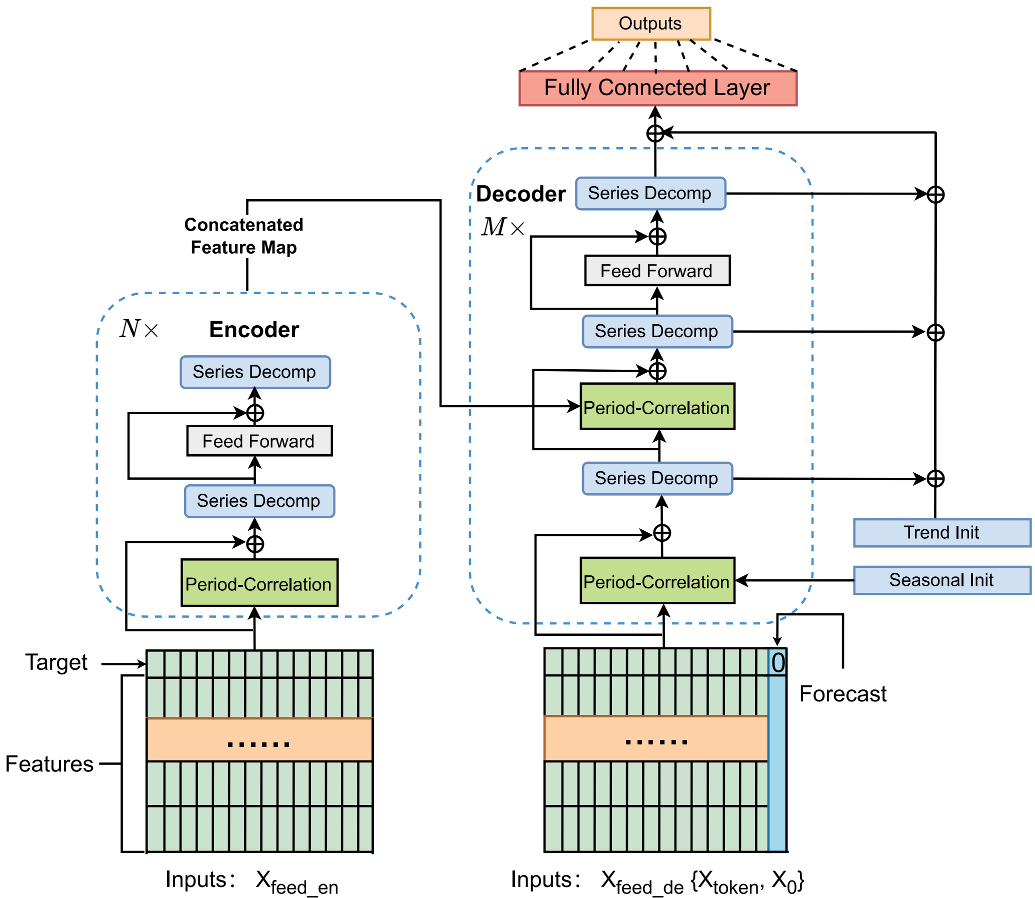

Fig. 2. The architecture of the SDTP model.

initial formula:

$$\begin{aligned} X_{ens}, X_{ent} &= SeriesDecomp(X_{en}) \\ X_{des} &= Concat(X_{ens}, X_0) \\ X_{det} &= Concat(X_{ent}, X_{Mean}) \end{aligned} \tag{5}$$

The information of the encoder as well as the trend and season portions learned, are used by the decoder to produce predictions. The period-correlation mechanism and the series decomposition module inside the model are able to find period-based dependencies in the series, exploit past information and eliminate interfering information. The decoder is calculated as follows:

$$\begin{aligned} S_{de}^{l,1}, T_{de}^{l,1} &= SeriesDecomp(Period\text{-}Correlation(X_{de}^{l-1}) + X_{de}^{l-1}) \\ S_{de}^{l,2}, T_{de}^{l,2} &= SeriesDecomp(Period\text{-}Correlation(S_{de}^{l,1}, X_{en}^{N}) + S_{de}^{l,1}) \\ S_{de}^{l,3}, T_{de}^{l,3} &= SeriesDecomp(FeedForward(S_{de}^{l,2}) + S_{de}^{l,2}) \\ T_{de}^{l} &= T_{de}^{l-1} + W_{l,1} \times T_{de}^{l,1} + W_{l,2} \times T_{de}^{l,2} + W_{l,3} \times T_{de}^{l,3} \end{aligned} \tag{6}$$

where X_{de}^{l}, $l \in \{1, \dots, M\}$ is the output of the lth decoder, $S_{de}^{l,i}$ and $T_{de}^{l,i}$, $i \in \{1, 2, 3\}$ denote the seasonal part and the trend part of the ith series decomposition layer in the lth encoder, respectively, $S_{de}^{l,3}$ is equal to the output of the lth decoder X_{de}^{l}, $W_{l,i}$, $i \in \{1, 2, 3\}$ denotes the projection matrix of the ith obtained trend part $T_{de}^{l,i}$, and T_{de}^{l} denotes the weighted sum of trends in the lth decoder including the trend of the $(l-1)$-th decoder and the sub-trend parts in the lth decoder. The final prediction result can be calculated by:

$$Prediction = W_S \times X_{de}^{M} + T_{de}^{M} \tag{7}$$

where X_{de}^{M} and T_{de}^{M} denote the seasonal part and trend part of the last decoder layer, respectively, and W_S denotes the projection matrix that projects X_{de}^{M} into the target dimension.

5. Experiments

5.1. Data

Since the fluctuations of a single stock price are influenced by many external factors and do not correctly reflect the changes of the whole market, in this experiment we use the stock market index, which is a weighted sum of the representative stocks selected as the financial level of the whole market by the financial institutions. Stock indices are important for investors to make decisions and analyze the market. Particularly, we select three well-known stock indices, the Shanghai Composite Index (SH) from July 1, 1991 to June 30, 2020, and the Shenzhen Component Index (SZ) and the Hang Seng Index (HSI) from July 31, 1991 to May 31, 2022. These datasets have been widely used in many research studies for stock price forecasting (He, Khushi, Tran, & Liu, 2021; Lu, Li, Wang, & Qin, 2021; Yan et al., 2020). All data are collected from datayes. The three stock indices have collected data for 7083, 7505, and 7345 trading days, respectively, over more than 30 years. During this period, China has experienced an important period of rapid economic development since the reform and opening up, as well as the winter period brought by the global financial crisis and the current period of stable economic growth. Our dataset covers a time horizon of nearly 30 years and can include many situations in the stock market, so the results obtained on these three datasets are persuasive. Each piece of data includes eight features: Volume, Turnover, Change, Change rate, Highest price, Lowest price, Open price, and Close price (the HSI index does not have the first two features). Volume indicates the whole number of the stocks traded during the day. Turnover indicates the whole value of the stocks traded during the day. Change indicates the price of a stock compared to that on the previous day. Change rate indicates the percentage difference

Table 2
Data samples of the SH index.

Date	Volume	Turnover	Change	Change rate	Highest price	Lowest price	Open price	Close price
1991/7/1	2294000	12469884	−0.71	−0.5161	138.62	136.56	136.64	136.85
1991/7/2	283800	3794100	−0.89	−0.6503	135.96	135.69	135.91	135.96
1991/7/3	271500	1818504	−0.69	−0.5075	135.96	134.98	135.28	135.27

Table 3
Comparisons of the SDTP model and the baselines on three stock indices.

Datasets	SH				SZ				HSI			
Models	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2
RNN	26.822	35.801	0.848	0.9751	150.870	204.151	1.342	0.9931	259.935	348.273	1.012	0.9784
LSTM	24.361	34.331	0.800	0.9770	144.466	194.805	1.271	0.9937	258.704	349.577	1.002	0.9782
BiLSTM (Siami-Namini, Tavakoli, & Namin, 2020)	23.409	33.579	0.786	0.9780	144.302	193.466	1.265	0.9938	259.209	347.573	1.003	0.9785
CNN-LSTM (Lu, Li, Li, Sun, & Wang, 2020)	23.195	32.640	0.774	0.9792	144.979	192.438	1.288	0.9938	258.676	345.834	1.002	0.9787
CNN-BiLSTM (Lu et al., 2021)	22.715	32.065	0.760	0.9800	143.118	191.727	1.270	0.9939	258.015	345.945	0.998	0.9787
CNN-BiLSTM-AM (Lu et al., 2021)	21.952	31.694	0.741	0.9804	142.247	188.854	1.244	0.9941	257.800	345.555	0.993	0.9787
SDTP	**21.731**	**31.604**	**0.732**	**0.9805**	**133.736**	**180.484**	**1.164**	**0.9946**	**256.024**	**345.411**	**0.992**	**0.9788**

between the price of the current day and the prior day. Highest price indicates the highest stock price traded on that day. Lowest price indicates the lowest stock price traded on that day. Open price indicates the first stock price traded on that day. Close price indicates the last stock price traded on that day. Table 2 shows data samples of the Shanghai Composite Index.

5.2. Evaluation metrics

Since forecasting stock prices is a regression task, we selected four indicators commonly used in stock price forecasting to evaluate the results: the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). The metrics are calculated as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{x}_i - x_i| \tag{8}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{x}_i - x_i)^2} \tag{9}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{x}_i - x_i|}{x_i} \times 100\% \tag{10}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{\sum_{i=1}^{n}(\overline{x}_i - \hat{x}_i)^2} \tag{11}$$

where the $\overline{x}_i$ denotes the average price of x_i, $\hat{x}_i$ denotes the predicted price, and x_i denotes the actual price. MAE reflects the magnitude of the actual prediction error. The prediction result will be better when the MAE value is lower. The RMSE denotes a quantitative evaluation metric. The smaller the value of RMSE, the better the prediction result. The prediction result will be better when MAPE has a lower value. R^2 indicates the similarity between the predicted value and the actual value. The higher the degree of similarity, the closer the R^2 value is to 1, and vice versa. For the investors, RMSE can help them assess the overall accuracy of a forecasting model. In the real-world investing, a lower RMSE means more accurate forecasts, which can help the investors make more informed investment decisions. Unlike RMSE, MAE provides the average error of the model's forecasts without interference from the outliers, and thus the investors can use the MAE to understand the average predictive accuracy of a model. MAPE is significant for the investors as it provides a relative magnitude of the prediction error. Consequently, the investors can assess the forecast bias of a model based on MAPE by comparing it to the actual value. A higher R^2 value means that the model is able to explain the actual data better, thus providing the investors with more reliable forecasts.

5.3. Experiment settings

We divide the three collected datasets into two parts: the training set and the testing set. We use the closing prices of the last 1000 days as the testing set for evaluation and the remaining data as the training set for training. We compare the proposed SDTP model with six state-of-the-art deep learning models: RNN, LSTM, BiLSTM (Siami-Namini et al., 2020), CNN-LSTM (Lu et al., 2020), CNN-BiLSTM (Lu et al., 2021) and CNN-BiLSTM-AM (Lu et al., 2021). We carry out the experiments following the settings of the work (Lu et al., 2021): The loss function is MAE, which is less sensitive to outliers, and provides an intuitive measure of the average prediction error. The batch size is 64, the time step is 5 days, the epoch is 100, the learning rate is 0.001, and the optimizer is Adam. We use Z-score for the input data as follows:

$$x_i' = \frac{x_i - \overline{x}}{s} \tag{12}$$

where x_i denotes the raw input data, $\overline{x}$ denotes the average of the input data, s denotes the standard deviation of the input data, and x_i' denotes the value after Z-score.

5.4. Results

We present the findings of two studies in this section. In Section 5.4.1, we summarize the prediction results on the three major stock indices. We also compare with the competitive method (Wang, Tang, Xiong, & Zhang, 2021) on the eight stock indices in Section 5.4.2. In Section 5.4.3, we conduct an ablation study of our proposed SDTP model for the critical components. We analyze the impact of the length of the lag period in Section 5.4.4.

5.4.1. Results on three stock market indices

We compare the proposed SDTP model with six deep learning models on three different stock index datasets, and evaluate them in terms of four general evaluation metrics, i.e., MAE, RMSE, MAPE and R^2.

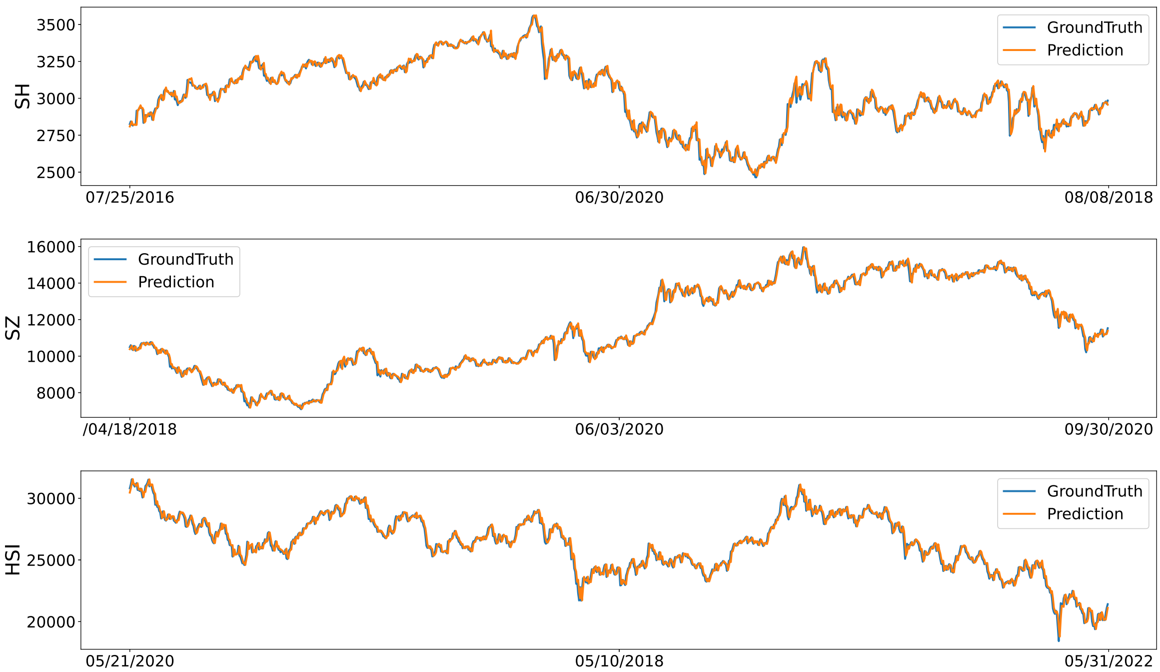

Fig. 3. Plots of predicted index price and ground truth price for three stock markets during the test period.

Table 3 reports the results on the SH, SZ, and HSI indices. Compared with the simple RNN models, the LSTM models achieve a large improvement in all three evaluation metrics, which illustrates that the LSTM models can powerfully capture the information in historical data and discover long-term dependencies in stock data. Furthermore, the hybrid models generally perform better than the single model. For instance, on the SH index, adding CNN and AM to BiLSTM makes MAE drop from 23.409 to 21.952, the RMSE drop from 33.579 to 31.694, the MAPE drop from 0.786 to 0.741, and the R^2 increase from 0.9780 to 0.9804. This is not surprising because the hybrid models can leverage the advantages of different models and mine information from different perspectives. Our proposed SDTP model achieves the improvement compared with the CNN+BiLSTM+AM model in three metrics in all datasets. Fig. 3 shows the degree of fit between the true and predicted values. The SDTP model has a good fit on the three datasets, which also implies a smaller error. Consequently, the investors can make more accurate judgements via the predicted outcomes from the SDTP model. The results are in line with our expectations because the proposed SDTP model can better simulate the financial series by series decomposition layers and find the relation between the series with the period-correlation mechanism, which helps it learn the complex patterns of the financial time series and work on different datasets.

5.4.2. Supplementary study

To further demonstrate the feasibility of our proposed SDTP model for the stock price prediction, we conduct the experiments with the Reservoir Computing (RC) (Wang et al., 2021) model to predict the closing prices of the following day in eight different stock indices, including DJI, the Nasdaq Composite Index, the SSE Index, the FTSE 100 Index, the Nikkei 225 Index, the NYSE Composite Index, the CAC40 index, and the S&P500 Index. The above data are well-known indices from the world's major economies, and they have been widely used as benchmarks in the experiments predicting stock indices. Each stock market has its own unique property. For example, the DJI represents the traditional industrial sector of the U.S. stock market, including companies in manufacturing, infrastructure, energy, etc., while the Nasdaq Composite Index consists of numerous technology companies covering computer technology, software development, Internet services, electronic equipment manufacturing and other emerging areas. The models that perform well on some markets might not work on others with different styles. To further verify the generalization of the SDTP model, validation in various stock markets is necessary. The first 80% of trading days make up the training set, while the left 20% do the testing set in each dataset. Table 4 shows detailed information about the datasets. We follow the previous experiment settings and train the SDTP model with the same metrics.

On the eight datasets, the SDTP model outperforms the competitor in terms of almost all evaluation metrics, as shown in Table 5. In particular, on the S&P500 and NYSE datasets, it achieves significant 5.63% and 6.75% decreases of MAE, 2.02% and 3.78% decreases of RMSE, 6.75% and 7.27% decreases of MAPE, respectively. On the DJI dataset, the SDTP model is slightly inferior to the competitor on RMSE, which might be due to the volatility of this dataset with a big variance. The experiment again demonstrate that the SDTP model performs very well in predicting the stock closing price of the following day, which can assist investors in making decisions across different stock markets.

5.4.3. Ablation study

For a comprehensive study of the components of our model, two variants are constructed. "Trans_corr" means the SDTP model without the decomposition layers. "Trans_dec" means the SDTP model without the period-correction mechanism.

Table 6 reports the results on three datasets. Transformer_corr model performs worst, which demonstrates that it is infeasible to directly utilize historical data to simulate the complex stock markets. Also, Transformer_dec performs worse, which shows the importance of mining the relationship between historical data. The positive effects of the period-correlation mechanism and decomposition layer on stock price prediction are confirmed by the performance of Trans_corr and Trans_dec, respectively. The experiment shows that the components can effectively improve the performance of our model in stock prediction.

5.4.4. Parameter sensitivity study

In order to assess the sensitivity of the SDTP model to the parameters, we conducted a sensitivity analysis on the key parameter (i.e., the lag length of trading days). Since the SDTP model adopts the trading day as the fundamental unit, the length of the trading days can reflect the periodicity of the time series and then directly impact the final

Table 4
The descriptive statistics of eight stock indices.

Datasets	Name	Time period	Total days	Mean	Std.dev.
S&P500	S&P 500 Index	01/04/2010–12/28/2018	2263	1856.50	520.45
CAC40	CAC 40 Index	01/04/2010–12/28/2018	2298	4298.21	698.69
NYSE	New York Stock Exchange	12/31/2009–12/28/2018	2264	9859.53	1807.08
DJI	Dow Jones Industrial Average	12/31/2009–12/28/2018	2264	16 624.57	4447.15
NASDAQ	Nasdaq Composite Index	12/31/2009–12/28/2018	2264	4386.49	1622.66
FTSE	Financial Times Stock Exchange	01/04/2010–12/31/2018	2272	6429.09	702.92
N225	Nikkei Stock Average	01/04/2010–12/28/2018	2208	15 141.53	4759.24
SSE	Shanghai Stock Exchange	01/04/2010–12/28/2018	2188	2786.43	555.17

Table 5
Comparisons of the RC model and the SDTP model on eight stock indices.

Models	RC (Wang et al., 2021)				SDTP			
Datasets	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2
S&P500	15.80	23.25	0.60	0.9769	**14.91**	**22.78**	**0.56**	**0.9808**
CAC40	32.98	43.38	0.63	0.9438	**31.50**	**42.25**	**0.60**	**0.9478**
NYSE	67.96	96.13	0.55	0.9669	**63.37**	**92.50**	**0.51**	**0.9713**
DJI	155.11	**225.01**	0.64	0.9759	**145.78**	225.80	**0.60**	**0.9842**
NASDAQ	53.24	78.09	0.76	0.9761	**51.04**	**75.75**	**0.72**	**0.9857**
FTSE	41.03	54.02	0.56	0.9501	**39.83**	**52.16**	**0.54**	**0.9524**
N225	160.77	229.35	0.75	0.9733	**159.64**	**224.86**	**0.74**	**0.9751**
SSE	22.07	30.76	0.74	0.9874	**21.28**	**29.71**	**0.70**	**0.9887**

Table 6
Ablation analysis on three stock indices.

Datasets	SH				SZ				HSI			
Models	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2
Trans_corr	47.342	83.457	1.549	0.8727	372.301	469.691	3.406	0.9637	269.166	360.490	1.043	0.9769
Trans_dec	23.472	34.200	0.784	0.9786	137.509	184.136	1.221	0.9944	263.528	353.442	1.022	0.9778
SDTP	**21.731**	**31.604**	**0.732**	**0.9805**	**131.648**	**178.253**	**1.164**	**0.9947**	**256.024**	**345.411**	**0.992**	**0.9788**

Table 7
Parameter sensitivity on three stock indices.

Datasets	SH				SZ				HSI			
Lag_size	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2
10	22.709	32.832	0.760	0.9803	131.731	178.640	1.162	0.9947	258.468	348.598	1.000	0.9784
7	22.243	32.538	0.744	**0.9806**	**129.036**	**175.059**	**1.139**	**0.9949**	257.607	349.534	0.999	0.9783
5 (original)	**21.731**	**31.604**	**0.732**	0.9805	131.648	178.253	1.164	0.9947	**256.024**	**345.411**	**0.992**	**0.9788**

forecast results. In the previous studies (Lu et al., 2021; Sonkiya et al., 2021; Zhang et al., 2022), the lag length usually ranges from 5 to 10 days (i.e., one to two weeks of trading day time). The longer time intervals cannot reflect the rapid fluctuations of the stock market and the shorter time intervals cannot capture the overall recent trend. To further analyze the effect of different lag lengths on model predictions, we compare our proposed SDTP (lag_5) model with the lag length of 5 days with two variants, SDTP (lag_7) and SDTP (lag_10), which use 7 and 10 days as the lag length, respectively. On three datasets, SDTP (lag_5) and SDTP (lag_7) almost perform better than SDTP (lag_10), as shown in Table 7. However, SDTP (lag_5) does not perform as well as SDTP (lag_7) on the SZ index, which indicates that it is not the case that the shorter the lag length used, the better the results. In other words, the optimal lag lengths are not the same on all datasets. In our experiments, the lag time serves as a representation of the period, with its length reflecting the duration of the cycle time. However, each market has a specific structure and trading rules, which further lead to unique cycle time. Consequently, the proper lag period time is distinct in different markets and needs to be carefully adjusted according to the specific market. Additionally, Table 7 shows that the SDTP model with the longest time interval performed poorly on all three datasets. We believe that the cyclical information on stocks is short-term, which is largely due to the high degree of randomness and frequent fluctuations in the stock market.

As is known to all, the stock market exhibits conspicuous cyclical characteristics encompassing short-term fluctuations, seasonal variations and long short-term trends. Optimal selection of the lag length helps capture these periodicity patterns, which in turn improves the performance and reliability of the forecasting model. Besides, the periodicity patterns offer valuable insights into stock prices and market behavior, which enables the investors to discern opportune moments for buying and selling while facilitating the formulation of well-informed investment strategies. Therefore, it is important to note that the optimal lag length is not constant and needs carefully adjusting to specific market conditions.

6. Conclusion

Correctly predicting the stock price is of great importance for studying the financial markets. In this paper, we propose a novel deep learning model dubbed SDTP, which has a better capability of feature extraction from the time series data and stock price prediction across different stock markets by adopting the series decomposition layer and period-correlation mechanism to get the inherent periodicity and relation between the series. We conduct extensive experiments of the proposed SDTP algorithm and a collection of the state-of-the-art deep learning methods on a number of datasets. The experimental results demonstrate that the proposed SDTP algorithm clearly outperforms the state-of-the-art competitors.

CRediT authorship contribution statement

Zicheng Tao: Methodology, Experiment, Writing – original draft. **Wei Wu:** Data curation, Writing – review & editing. **Jianxin Wang:** Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors are thankful for the anonymous referee's constructive comments. This work was supported by the National Natural Science Foundation of China (No. 62302528), the Open Project of Xiangjiang Laboratory, China (No. 22XJ03010), the Natural Science Foundation of Changsha, China (No. kq2202104) and the High Performance Computing Center of Central South University, China.

References

Alkhatib, K., Najadat, H., Hmeidi, I., & Shatnawi, M. K. A. (2013). Stock price prediction using k-nearest neighbor (kNN) algorithm. *International Journal of Business, Humanities and Technology*, *3*(3), 32–44.

Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Stock price prediction using the ARIMA model. In *2014 UKSim-AMSS 16th international conference on computer modelling and simulation* (pp. 106–112).

Babu, C. N., & Reddy, B. E. (2014). Selected Indian stock predictions using a hybrid ARIMA-GARCH model. In *2014 International conference on advances in electronics computers and communications* (pp. 1–6). IEEE.

Baldauf, B., & Santoni, G. (1991). Stock price volatility: some evidence from an ARCH model. *The Journal of Futures Markets (1986-1998)*, *11*(2), 191.

Budiharto, W. (2021). Data science approach to stock prices forecasting in Indonesia during Covid-19 using long short-term memory (LSTM). *Journal of Big Data*, *8*(1), 1–9.

Chen, Y., Wu, J., & Wu, Z. (2022). China's commercial bank stock price prediction using a novel K-means-LSTM hybrid approach. *Expert Systems with Applications*, *202*, Article 117370.

Cleveland, S. (1990). A seasonal-trend decomposition procedure based on Loess (with discussion). *Journal of Official Statistics*, (6), 3.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Ding, Q., Wu, S., Sun, H., Guo, J., & Guo, J. (2020). Hierarchical multi-scale Gaussian transformer for stock movement prediction.. In *International joint conferences on artificial intelligence* (pp. 4640–4646).

Dingli, A., & Fournier, K. S. (2017). Financial time series forecasting-a deep learning approach. *International Journal of Machine Learning and Computing*, *7*(5), 118–122.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144.

He, J., Khushi, M., Tran, N. H., & Liu, T. (2021). Robust dual recurrent neural networks for financial time series prediction. In *Proceedings of the 2021 SIAM international conference on data mining* (pp. 747–755).

Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, *33*(1), 3–12.

Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

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

Jiang, X., Pan, S., Jiang, J., & Long, G. (2018). Cross-domain deep learning approach for multiple financial market prediction. In *2018 international joint conference on neural networks* (pp. 1–8). IEEE.

Khaidem, L., Saha, S., & Dey, S. R. (2016). Predicting the direction of stock market prices using random forest. arXiv preprint arXiv:1605.00003.

Kohara, K., Fukuhara, Y., & Nakamura, Y. (1996). Selective presentation learning for neural network forecasting of stock markets. *Neural Computing and Applications*, *4*(3), 143–148.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., et al. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541–551.

Li, Y., Lv, S., Liu, X., & Zhang, Q. (2022). Incorporating transformers and attention networks for stock movement prediction. *Complexity*, *2022*.

Lin, Y., Guo, H., & Hu, J. (2013). An SVM-based approach for stock market trend prediction. In *The 2013 international joint conference on neural networks* (pp. 1–7). IEEE.

Liu, J., Lin, H., Liu, X., Xu, B., Ren, Y., Diao, Y., et al. (2019). Transformer-based capsule network for stock movement prediction. In *Proceedings of the first workshop on financial technology and natural language processing* (pp. 66–73).

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

Lu, W., Li, J., Wang, J., & Qin, L. (2021). A CNN-BiLSTM-AM method for stock price prediction. *Neural Computing and Applications*, *33*(10), 4741–4753.

Mohanty, D., Parida, A. K., & Khuntia, S. S. (2021). Financial market prediction under deep learning framework using auto encoder and kernel extreme learning machine. *Applied Soft Computing*, *99*, Article 106898.

Nair, B. B., Mohandas, V., & Sakthivel, N. (2010). A decision tree-rough set hybrid system for stock market trend prediction. *International Journal of Computer Applications*, *6*(9), 1–6.

Rounaghi, M. M., & Zadeh, F. N. (2016). Investigation of market efficiency and financial stability between S&P 500 and London stock exchange: monthly and yearly forecasting of time series stock returns using ARMA model. *Physica A. Statistical Mechanics and its Applications*, *456*, 10–21.

Rozeff, M. S., & Kinney Jr, W. R. (1976). Capital market seasonality: The case of stock returns. *Journal of Financial Economics*, *3*(4), 379–402.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Shah, D., Campbell, W., & Zulkernine, F. H. (2018). A comparative study of LSTM and DNN for stock market forecasting. In *2018 IEEE international conference on big data* (pp. 4148–4155). IEEE.

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2020). The performance of LSTM and BiLSTM in forecasting time series. In *2019 IEEE international conference on big data* (pp. 3285–3292).

Singh, S., Parmar, K. S., & Kumar, J. (2021). Soft computing model coupled with statistical models to estimate future of stock market. *Neural Computing and Applications*, *33*(13), 7629–7647.

Sonkiya, P., Bajpai, V., & Bansal, A. (2021). Stock price prediction using BERT and GAN. arXiv preprint arXiv:2107.09055.

Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, *72*(1), 37–45.

Ticknor, J. L. (2013). A Bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications*, *40*(14), 5501–5506.

Vadlamudi, S. (2017). Stock market prediction using machine learning: A systematic literature review. *American Journal of Trade and Policy*, *4*(3), 123–128.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Wang, W.-J., Tang, Y., Xiong, J., & Zhang, Y.-C. (2021). Stock market index prediction based on reservoir computing models. *Expert Systems with Applications*, *178*, Article 115022.

Wu, C., & Chau, K.-W. (2013). Prediction of rainfall time series using modular soft computingmethods. *Engineering Applications of Artificial Intelligence*, *26*(3), 997–1007.

Wu, J. M.-T., Li, Z., Srivastava, G., Tasi, M.-H., & Lin, J. C.-W. (2021). A graph-based convolutional neural network stock price prediction with leading indicators. *Software - Practice and Experience*, *51*(3), 628–644.

Wu, J. M.-T., Wu, M.-E., Hung, P.-J., Hassan, M. M., & Fortino, G. (2020). Convert index trading to option strategies via LSTM architecture. *Neural Computing and Applications*, 1–18.

Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, *34*, 22419–22430.

Xu, Z., Li, Q., Qian, L., & Wang, M. (2022). Multi-sensor fault diagnosis based on time series in an intelligent mechanical system. *Sensors*, *22*(24), 9973.

Yamaguchi, K. (2008). Reexamination of stock price reaction to environmental performance: A GARCH application. *Ecological Economics*, *68*(1–2), 345–352.

Yan, B., Aasma, M., et al. (2020). A novel deep learning framework: Prediction and analysis of financial time series using CEEMD and LSTM. *Expert Systems with Applications*, *159*, Article 113609.

Yun, K. K., Yoon, S. W., & Won, D. (2021). Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process. *Expert Systems with Applications*, *186*, Article 115716.

Zhang, Q., Qin, C., Zhang, Y., Bao, F., Zhang, C., & Liu, P. (2022). Transformer-based attention network for stock movement prediction. *Expert Systems with Applications*, *202*, Article 117239.

Zhang, G., Zhang, X., & Feng, H. (2016). Forecasting financial time series using a methodology based on autoregressive integrated moving average and Taylor expansion. *Expert Systems*, *33*(5), 501–516.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., et al. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence, vol. 35, no. 12* (pp. 11106–11115).