Journal of King Saud University – Computer and Information Sciences 34 (2022) 4322–4334

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Knowledge graph and deep learning combined with a stock price prediction network focusing on related stocks and mutation points

Meiyao Tao [a], Shanshan Gao [a,b,c,*], Deqian Mao [a], Hong Huang [a]

[a] Shandong University of Finance and Economics, Jinan, China
[b] Shandong China-U.S. Digital Media International Cooperation Research Center, China
[c] Shandong Provincial Key Laboratory of Digital Media Techno, China

ARTICLE INFO

Article history:
Received 21 March 2022
Revised 11 May 2022
Accepted 17 May 2022
Available online 21 May 2022

Keywords:
Knowledge graph
Stock price prediction
Mutation point
Piecewise loss function

ABSTRACT

Due to the interaction of many factors in the stock market, stock price prediction has always been a challenging problem in the field of machine learning. In particular, the mutation factors of the stock market often have a great impact on subsequent predictions. The existing prediction models seldom consider the impacts of other stocks in the stock market and mutation points on the prediction accuracy of target stocks. Therefore, this paper presents a new knowledge graph and deep learning method combined with a stock price prediction network focusing on related stocks and mutation points. First, the target stock price features are obtained through the ConvLSTM network. Second, the knowledge graph is used to mine the hidden relationships between stocks to find the stocks relevant to the target stock to obtain the market information vector and the market information features through the ConvLSTM network. Then, we find the mutation points according to the price change range, construct the mutation point distance weight matrix according to the distance from each trading day to the mutation points, and obtain the mutation point information features through the graph convolutional network (GCN). Finally, the features of market information, mutation point information and target stock price are fused to jointly predict the future stock price. The experimental results on the A share of Shenzhen from 2010 to 2019 show that the algorithm has good robustness and that the prediction accuracy is effectively improved.

1. Introduction

Since stock trading is associated with high profits and extremely high risks, investors are eager to accurately predict future stock prices and to accordingly adjust investment plans in a targeted manner to obtain greater profits. The stock market is a complex nonlinear dynamic system that is affected by many factors, such as policies, social news events, the operation of companies and the psychological changes of investors. Therefore, the prediction of stock market trends or stock prices is an extremely difficult challenge.

To accurately predict stock prices, researchers have proposed many stock price prediction models, which can be roughly divided into three categories: traditional methods, machine learning and deep learning. Traditional solutions for stock forecasting are models based on time series analyses, such as the Kalman filter (Xu and Zhang, 2015) and autoregressive model and its extensions (Adebiyi et al., 2014), for example, the autoregressive model (AR), moving average model (MA), autoregressive moving average model (ARMA), and autoregressive integrated moving average model (ARIMA). Although the prediction accuracy of these traditional models has reached a general level, there are also limitations, such as seasonality and nonstationarity. Therefore, machine learning methods based on support vector machines (SVMs) and random forests are gradually being applied in the field of stock forecasting (Das, 2021). However, these traditional machine learning methods exhibit poor learning effects on complex high-dimensional data and the curse of dimensionality. To solve this problem, deep learning methods for stock forecasting have been developed (Jiang, 2021). With the success of deep neural networks in time series data modeling, long short-term memory (LSTM), convolutional neural networks (CNNs) and other models (Wei et al., 2017;

* Corresponding author at: Shandong University of Finance and Economics, Jinan, China.

E-mail address: gsszxy@aliyun.com (S. Gao).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2022.05.014
1319-1578/

Sinha et al., 2022; Polamuri et al., 2021) have become more effective solutions for predicting future stock prices.

With the development of the financial market, there are extensive connections between enterprises due to many relationships, which have different degrees of influence. Taking industry relations as an example, the development of enterprises is largely affected by their industry, so a shock to an industry inevitably affects the income of the enterprises in the industry. Although previous financial studies have shown that stock prices reflect all known information (Fama, 1965), most of the existing models ignore the impacts of other stocks on the target stock. They use only a single target stock to predict the stock price, which generates unsatisfactory prediction results. Considering that a variety of stocks in the stock market have the same or similar price trends and that interaction exists between stocks, we can more comprehensively represent stock market information by integrating one or more related stocks that have a direct or indirect relationship with the target stock. A knowledge graph can not only describe the direct relationships between entities but also describe the relationships hidden in the background knowledge and is widely used in the fields of financial fraud, recommendation systems (Wang et al., 2018; Wang et al., 2018; Wang et al., 2019), text analysis (Akimushkin et al., 2018; Santos et al., 2017) and fake news classification (Koloski et al., 2022) to better mine the hidden relevance between different objects. Therefore, this paper considers using a knowledge graph to mine the relevance between stocks.

According to the literature (Kang and Xu, 2007), changes in stock price can often be regarded as smooth and continuous processes, and the stock market operates stably under these conditions. However, because the stock market is easily disturbed by many factors, especially by some special cases, such as the adjustment of major national economic policies and emergencies in stock-related industries, stock prices may exhibit sudden changes in the short term, which poses risks to various economic activities in the stock market and increases the difficulty of stock price prediction. Fig. 1 is the result curve of stock price prediction using a graph convolutional network (GCN) model (Cerliani et al., 2020). Fig. 1 shows that the overall trend of the stock prediction results is roughly the same as that of the real price, but the accuracy of prediction results is decreased significantly near data points representing large inflections (the red points in the graph). We call these points in Fig. 1 mutation points. Due to the time continuity of stock price data, the stock price in a previous period has an impact on the future stock price, and old stock price information has a weaker impact on the stock price prediction than recent stock price information. Therefore, considering that the impact intensities of the mutation points on the stock price of each trading day are different, we set the weight according to the distance from the mutation point to each trading day. The closer the mutation point occurs to the trading day, the greater the impact weight is.

The existence of a mutation point increases the difficulty of stock price prediction. Most of the existing models use the mean square error (MSE) as the loss function, and the predicted value deviates greatly from the real value due to the influence of the mutation point. To solve the above problems, in this paper, we adopt a piecewise loss function and different forms of loss functions in different intervals to enhance the antinoise ability of the model and to improve the prediction accuracy. In summary, the main contributions of this paper can be summarized as follows:

- To account for the fact that a single stock cannot fully reflect the market information, we use a knowledge graph and graph embedding technology to mine the implicit relationships between stocks and to calculate their relevance. According to the relevance between stocks, the stocks relevant to the target stock are weighted to construct the corresponding market information matrix.
- Aiming at the problem of large errors near mutation points in stock price prediction, we obtain mutation points by using the range of price change, construct a mutation point weight matrix by using the distance between the mutation point and trading days, and use a GCN to better integrate the spatial and temporal relevance information.
- Due to the existence of mutation points in stock time series data, the error in the training process is too large. Therefore, we use a piecewise loss function and different penalty losses for different interval segments to improve the prediction performance of the model.
- By integrating the above methods, we propose a knowledge graph and deep learning combined with a stock price prediction network focusing on related stocks and mutation points. Experiments show that the model proposed in this paper performs well in the task of stock price prediction, and an ablation study proves the effectiveness of the important components in the model.

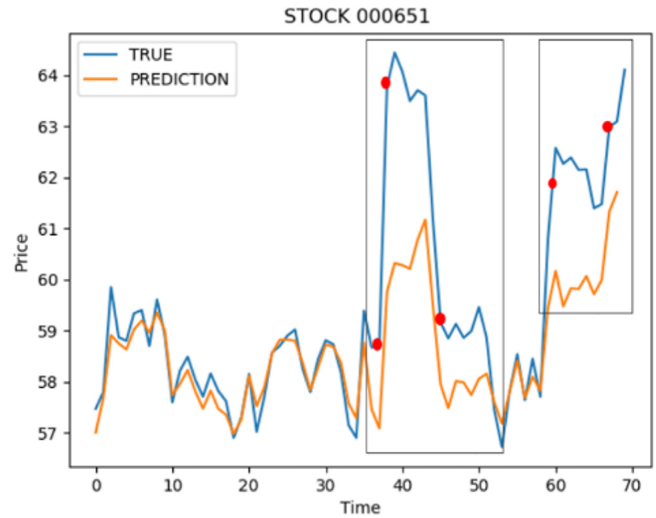


Fig. 1. Stock price prediction results of a GCN model. Red points indicate mutation points.

2. Related works

In this section, the existing work related to this paper is introduced.

2.1. Stock price forecasting method

Stock price forecasting has always been a popular research issue. With the maturity of computer and artificial intelligence technology, deep learning models are widely used in stock price forecasting. LSTM (Hochreiter and Schmidhuber, 1997) represents a special recurrent neural network (RNN) that can not only learn and store the context information of time series but also solve the problems of gradient disappearance and gradient explosion in the training process of long time series. Due to the good performance of LSTM models on time series, Chen et al. (2015) used LSTM to predict the return of Chinese stocks, and the return prediction accuracy increased from 14.3% to 27.2%. Wei et al. (2017) used a wavelet transform to denoise financial data, combined with stacked autoencoders and LSTM for financial time series prediction. Chen et al. (2018) used a deep learning network based on a convolutional neural network to predict the stock price trend of

China's stock market, and the result was better than the benchmark of an RNN model Gudelek et al. (2017) used the sliding window method to generate images by taking snapshots limited by the window in the time period of each day and then made predictions. Since both LSTM and CNNs exhibit good performance in stock forecasting, reference (Santika et al., 2021) combines LSTM and CNN algorithms to jointly learn the internal trends of time series data. The CNN is used to extract features, and the LSTM is used to learn long-term and short-term dependencies. On the basis of LSTM, reference (Shi et al., 2015) proposed a ConvLSTM network that replaces input-to-state and state-to-state transitions with a convolution structure on the basis of LSTM to better capture the spatiotemporal relevance of data. Wu et al. (2020) used a graph neural network to study multivariate time series for the first time, which proved the effectiveness of the graph neural network in time series data prediction. Cerliani et al. (2020) combined a graph convolutional neural network with a recursive structure to complete the task of sales forecasting.

Most of the above studies use single stock information for prediction, and few studies pay attention to target stocks and related stocks at the same time, ignoring the interaction between stocks. Nam and Seong (2019) paid attention to the importance of the target company and related companies and predicted stock trends based on the causal relationship between companies, which proved the importance of related companies. Chen et al. (2018) incorporated the related company information of the target company into stock price prediction and proposed a joint prediction model based on a graph convolution network. Long et al. (2020) proposed a deep neural network for predicting stock price trends by using desensitized transaction records and public market information. Sen and Mehtab (2021) designed an optimal portfolio by assigning the weight to its constituent stocks to achieve the best trade-off between return and risk and to obtain a higher rate of return. Inspired by this, we consider the impact of the target stock and the related stocks on the future stock price and assign different weights to different stocks according to the relevance degree.

2.2. Knowledge graph

The knowledge graph was first proposed by Google, and its essence is a semantic network that reveals the relationships between entities, describing objective things and their interrelationships in the form of graphs. The entities in the objective world correspond to the entities in the network, and the relationships between the entities constitute the edges in the network, which can be represented by (head entity, relationship, tail entity) triplets.

The construction of a knowledge graph mainly extracts entities, relationships and attributes from data of different sources and different structures, fuses the extraction results, and stores the fused knowledge in the graph database. Second, the entities and relations in the knowledge graph are embedded into a low-dimensional continuous vector space using graph embedding techniques (Wang et al., 2014; Yang et al., 2015) to generate representation vectors for downstream tasks.

Knowledge graphs have been applied in many fields because of their powerful knowledge reasoning abilities. Wang et al. (2018) integrated the knowledge graph into a recommender system and stimulated the propagation of user preferences on the set of knowledge entities by automatically and iteratively expanding the potential interests of users through the links in the knowledge graph to predict the final click probability. Cao et al. (2019) proposed a translation-based user preference model that transmits relational information in the knowledge graph to mine the implicit relationship between users and items and to further reveal users' preferences for consumption items. From the above research, it can be seen that knowledge graphs have the ability to mine hidden relevance for different objects. Inspired by this approach, we use a knowledge graph to mine the relevance between stocks.

2.3. ConvLSTM model

Although LSTM performs well on time sequence problems, it includes too much spatially redundant data and does not take the spatial relevance into account. Therefore, to better simulate the spatial–temporal relationships, the ConvLSTM network replaces input-to-state and state-to-state transitions with a convolution structure on the basis of LSTM and captures the basic spatial features through convolution in multidimensional data to obtain the spatiotemporal relevance of the time series. The input $\{X_1, X_2, \ldots, X_T\}$,memory unit $\{C_1, C_2, \ldots, C_T\}$, hidden state$\{H_1, H_2, \ldots, H_T\}$ and gating unit $\{i_t, f_t, \ldots, o_t\}$ of ConvLSTM are three-dimensional tensors. The structure of ConvLSTM is shown in Fig. 2, and its key steps are shown in Eq. (1).

$$\begin{aligned} f_t &= \delta(W_f * [X_t, H_{t-1}] + W_{cf} \circ C_{t-1} + b_f) \\ i_t &= \delta(W_i * [X_t, H_{t-1}] + W_{ci} \circ C_{t-1} + b_i) \\ C_t &= f_t \circ C_{t-1} + i_t \circ tan(W_c * [X_t, H_{t-1}] + b_c) \\ o_t &= \delta(W_o * [X_t, H_{t-1}] + W_{co} \circ C_{t-1} + b_o) \\ H_t &= o_t \circ tan(C_t) \end{aligned} \tag{1}$$

where σ represents the sigmoid function, ' $*$ ' represents the convolution operation, and ' $\circ$ ' represents the Hadamard product.

2.4. GCN model

GCNs (Kipf and Welling, 2016) have achieved great success in dealing with spatial dependencies between entities. They aim to fuse the information of nodes and their adjacent nodes to deal with the spatial dependencies between nodes. Each node changes its state until the final balance due to the influence of the adjacent points and further points, where the closer the adjacent points are, the greater their influence. This relevant information is stored in the adjacency matrix.

The essence of a GCN is a feature extractor. Suppose we have a batch of graph data that contain M nodes, in which each node has its own features, which form a matrix X $(X \in^{(M \times K)})$,where K is the index number. The relationship between nodes forms adjacency matrix A $\left(A \in R^{(M \times M)}\right)$. Its propagation method is shown in Eq. (2).

$$H^{(l+1)} = \delta\left(\widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}H^l W^l\right) \tag{2}$$

where H is the feature of each layer and H^l is the feature of layer L. For the input layer, H is X.

Based on the research of existing work, we take the ConvLSTM network as the basic model to extract stock price information to better learn the time and spatial information of stock price series and use a GCN to explore the impact of mutation points on stock prices. Knowledge graph and graph embedding technology is used to find the relevant stocks of the target stock to understand the information features of the stock market.

3. Model

This paper proposes a knowledge graph and deep learning combined with a stock price prediction network focusing on related stocks and mutation points. The network structure is shown in Fig. 3 and consists of three subnetworks: (1) The target stock price (TSP) subnetwork inputs the target stock price information into the ConvLSTM model to obtain the target stock price features. (2) The stock market information (SMI) subnetwork has the same network

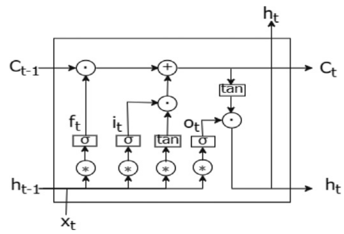


Fig. 2. ConvLSTM structure.

structure as the target stock price subnetwork, but its input is the market information vector composed of the target stock and its related stocks acquired through the knowledge graph to obtain market information features. (3) The mutation point information (MPI) subnetwork inputs the target stock price data and the mutation point distance weight matrix constructed according to the distance between each data point and the mutation point into the GCN model to obtain the features of the stock price mutation point. Finally, the features obtained by fusing these three subnetworks are used to predict the future price of the target stock.

3.1. Feature information network based on target stock and related stock

To predict the stock price of the next trading day, we pay attention to the impact of the target stock and related stocks on the future stock price and use the network model based on ConvLSTM to generate the target stock price feature vector and market information feature vector.

3.1.1. Target stock feature information subnetwork

According to existing research (Fama, 1965), all available information is reflected in the stock price, and the stock price can be predicted by using the basic price data. In this paper, we take the closing price as the prediction index and input the closing price of the past N trading days into the prediction network to predict the closing price of the target stock on the next trading day.

To better predict the future stock price, we describe stock price prediction as a prediction problem of spatiotemporal sequences, pay attention to the temporal and spatial information of stock price data, and use the ConvLSTM model, which can better capture the spatiotemporal relevance to more effectively extract the information features of stock price data. As shown in Fig. 4, we extract only the stock price features of the target stock, input the preprocessed closing price data of the target stock into the prediction network based on ConvLSTM, and extract its spatiotemporal features to obtain the feature vector of the target stock.

3.1.2. Market information subnetwork based on a knowledge graph and graph embedding technology

Since a knowledge graph can mine hidden associations between entities, we use a knowledge graph to find the relevant stocks of the target stock in the stock market and to obtain the market feature vector on this basis, as shown in Fig. 5. First, we use 1481 stocks in A share of Shenzhen to design and build a company knowledge graph with multiple entity and relationship types, which can be represented by triples (i,r,j), where $i,j \in E$ and $r \in R$. E represents a collection of entities, including companies, industries, concepts, provinces, cities, businesses, offices, and shareholders. R represents the relationships between entities, including location relationships, affiliations, and associations. Table 1 shows some triplet data.

After constructing the knowledge graph, first, we obtain the vector representation of each stock through node2vec (Grover et al., 2016). Second, the cosine similarity algorithm is used to calculate the similarity between the target stock and other stocks according to the representation vector of each stock, as shown in Eq. (3). Then, the top-k stocks with the highest similarity are selected as related stocks. After obtaining the relevant stocks, we calculate the relevance weight between them according to the similarity score between the relevant stocks and the target stocks, as shown in Eq. (4). Finally, the target stock and related stocks are weighted and summed according to the relevance weight to obtain the market information vector, as shown in Eq. (5).

$$e_t^i = cos(v_i, v_t) = \frac{v_i \cdot v_t}{|v_i| \times |v_t|} \tag{3}$$

$$\theta_t^i = \frac{exp(e_t^i)}{\sum_{i=1}^{k} exp(e_t^i)} \tag{4}$$

$$v_M = \alpha \times v_d^t + \beta \left(\sum_{j=1}^{k} \left(\theta_t^i \times v_d^j \right) \right) \tag{5}$$

where v_i and v_t in Eq. (3) are the embedded representation vector of the relevant stock i and the embedded representation vector of the target stock t, respectively, and e_t^i represents the cosine similarity between the relevant stock i and the target stock t. k in Eq. (4) represents the number of relevant stocks, θ_t^i represents the relevance weight between the relevant stock i and the target stock t, and v_d^t, v_d^j and v_M in Eq. (5) represent the index vector of the target stock, the index vector of relevant stocks and the market information vector, respectively.

Since the target stock price data are an integral part of the market information data, we can deduce that the market information data and the target stock price data have the same or similar temporal and spatial relationships. Therefore, to more comprehensively capture the market information data, the preprocessed market information vector is input into the network model based on ConvLSTM to obtain the spatiotemporal feature vector of market information.

3.2. Feature fusion subnetwork based on mutation point detection

Through the analysis of stock sequences, it is found that due to the existence of many influencing factors in the stock market, there may be one or more mutation points in a stock time series, which is nonlinear and unstable. Through the analysis of the existing stock price prediction models, it is found that previous models do not pay much attention to the mutation points in the stock price time sequence data. Based on this, we add the mutation point information subnetwork, as shown in Fig. 6.

Considering that our proposed model uses the price information of the previous N trading days to predict the closing price of the next trading day, we detect the mutation points within the range of these N trading days.

First, we calculate the price change range of adjacent trading days according to Eq. (6) and then calculate the average price change for N trading days. Because a mutation point in stock price data is a point with a sudden change, that is, the stock price at the mutation point exhibits a large rise and fall compared with the previous trading day. Therefore, we define data points satisfying Eq. (7) as the mutation points of a stock price sequence.

$$change_i = close_i - close(i-1) \tag{6}$$

$$|change_i| > w\mu \tag{7}$$

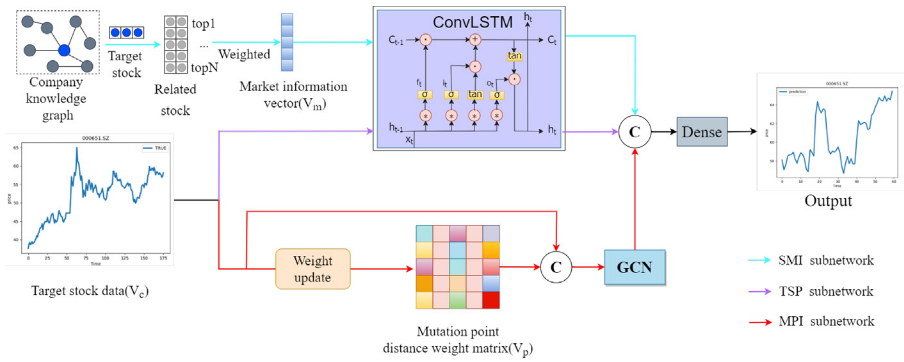


Fig. 3. Network structure.

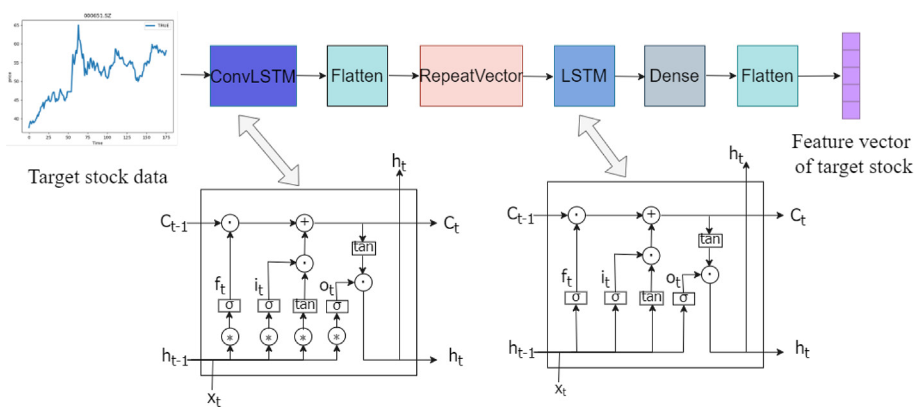


Fig. 4. Target stock feature information network framework.

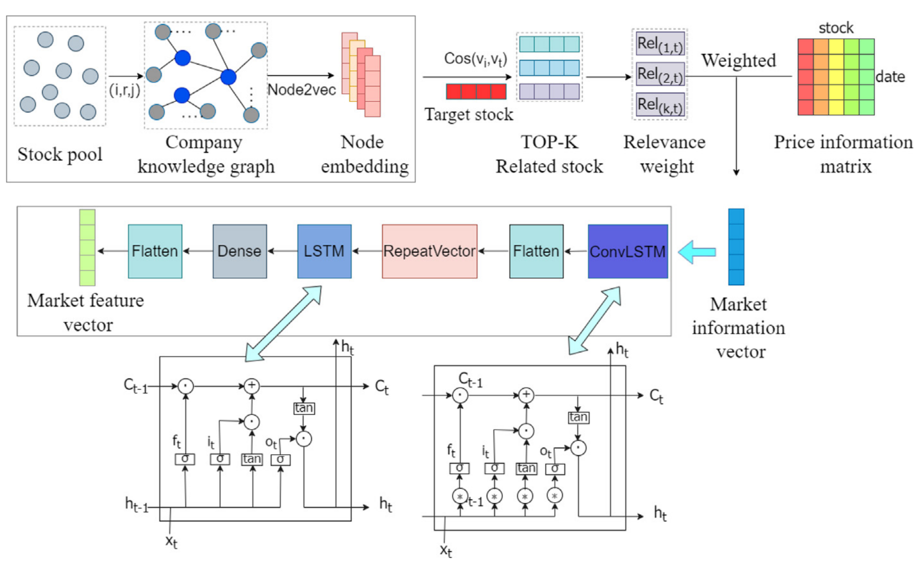


Fig. 5. Market information network framework.

where $change_i$ is the amount of change in the share price on trading day i and trading day i-1, μ is the average value of the amount of change in the stock price on trading day N, and w is obtained through a large number of experiments.

Second, we construct the mutation point distance weight matrix. We initialize the mutation point distance weight matrix $R^{(N \times N)}$ with zero. Then, the positions of the mutation points are obtained according to Eq. (7). Based on the obtained mutation

Table 1
Company knowledge graph triples.

i (Head entity)	r(relationship)	j(Tail entity)
000001.SZ	province	Guangdong
000001.SZ	industry	Bank
000001.SZ	concept	Subject stock of securities lending

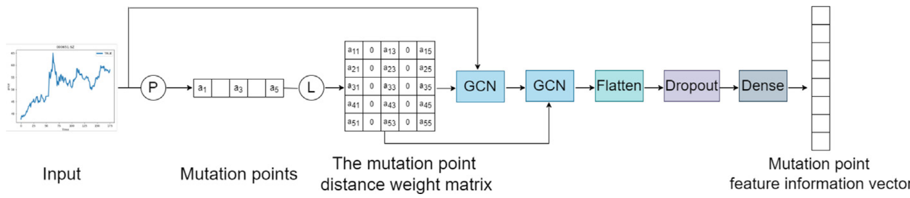


Fig. 6. Mutation point information subnetwork (Taking N = 5 and positions 1, 3 and 5 as mutation points).

point positions, the weight is updated according to the distance from each trading day to the mutations, where the shorter the distance is, the greater the influence weight is. The update process is shown in Eq. (8).

$$\begin{cases} 0, & (x_i - y_j) \notin P \\ N - (|x_i - y_j|), & (x_i - y_j) \in P \end{cases} \tag{8}$$

where $x_i (i = 1, 2 \ldots N)$ represents the x-coordinate of the matrix and $y_j (j = 1, 2 \ldots N)$ represents the y-coordinate of the matrix. P is the set of mutation points.

Finally, a GCN is used to extract the influence features of mutation points on stock price time series data. In the prediction network, we take the trading day as the node and the closing price as the features of the node. The stock closing price data are treated as a matrix X of dimensions $N \times 1$. The distance weight matrix of the mutation points shows the distance between each node and mutation points. The main idea of its construction is that points closer to mutation points are more affected by the mutation points. According to the core idea of the GCN, we regard the mutation point distance weight matrix as dimension $N \times N$ adjacency matrix A, which is input into the GCN together as input data to capture the mutation point features of stock price time series data.

3.3. Piecewise loss function

Stock prices are highly nonlinear and unstable because they are easily affected by many factors in the stock market, including enterprise performance, enterprise news and national economic policies. Most of the existing stock price prediction models based on deep learning use the MSE as the loss function, which cannot reasonably penalize the stock price near mutation points, resulting in a suboptimal solution of the model. Therefore, to avoid causing a large error between the actual value and the predicted value due to the influence of a mutation point, we aim to construct a reasonable loss function and then make the model converge to a better situation. According to existing knowledge, the MSE is an L2 loss function that can be derived everywhere and has fast convergence speed. When the MSE is used as the loss function, the gradient changes dynamically, but it is dominated by large error; that is, it is sensitive to outliers. If there are outliers, the MSE value is very large. The mean absolute error (MAE) is an L1 loss function. It is a continuous curve, but it is nondifferentiable at point 0, and the convergence speed is slow. When the MAE is used as the loss function, the gradient is the same for all points; that is, it is not sensitive to outliers.

Based on the above analysis, due to the existence of mutation points in the stock price, we adopt a piecewise loss function and combine the advantages of the MSE and MAE, as shown in Eq. (9).

$$\begin{cases} \frac{1}{N}\sum_{i=1}^{n}(y_t - y_p)^2, & if|x| < u \\ \frac{1}{N}\sum_{i=1}^{n}|y_t - y_p|, & otherwise \end{cases} \tag{9}$$

where $x = |y_t - y_p|$ represents the difference between the real value and the predicted value; u is obtained through a large number of experiments.

When the error between the real value and the predicted value is small, the gradient (the same as the MSE) decreases. When the error between the real value and the predicted value is large, the upper limit of the gradient (the same as the MAE) is 1. Compared with an L1 loss function, this piecewise loss function can converge faster. Compared with an L2 loss function, this piecewise loss function is less sensitive to outliers, and the gradient change is relatively small, which can allow the model to converge better during training.

4. Experiments

This section describes the data set and performance evaluation index used in this paper and presents the experimental studies and result analysis of the proposed model. To prove the universality of the model proposed in this paper, we choose Dong'e E-Jiao (000423.SZ), Yunnan Baiyao (000538.SZ), Gree Electric (000651.SZ) and Yanghe River share (002304.SZ) as the target stocks for the experiment, and in the experiment, we choose to use the price information of the previous 7 trading days to predict the closing price of the next day.

4.1. Dataset

The experimental data of this paper are stock data from the Shenzhen Stock Exchange obtained from crawling Tushare[1] (January 2010 to December 2019), including target stocks and corresponding related stocks. In the crawling data, the most representative close index is selected as the input data.

Through the analysis, we found that there were 2431 trading days in the Shenzhen stock market in China from 2010 to 2019. Although they were all traded in the Shenzhen Stock Exchange, the total trading days of each stock were different due to different conditions such as stock suspension. Therefore, to avoid destroying the actual model of price movement, we chose stocks with more than 90% of the total trading days for the experiment. For the closing index, we use the closing price of the previous trading day to fill in missing values. Table 2 shows some data from 000651.SZ as the target stock. After obtaining the complete data set, we divide the data set into training data and test data, of which 90% is used as training data and the remaining 10% is used as test data. After the data set is divided, we normalize the data; the calculation process is shown in Eq. (10). Then, the processed data are input into the model.

$$X^* = \frac{X - \mu}{\delta} \tag{10}$$

where x represents sample data, μ represents the mean, and δ represents the standard deviation.

[1] https://tushare.pro/

Table 2
Partial data set of 000651.

Trade-date	Close	Close1	...	Close9	Close10
2010–01-04	28.16	58.2	...	31.45	12.19
...	...	...	...	...	...
2019–12-31	65.58	89.43	...	133.01	4.31

4.2. Comparison with other models

To verify the accuracy of our proposed model, we use LSTM (Chen et al., 2015), gated recurrent unit (GRU) (Huynh et al., 2017), CNN (Chen et al., 2018), dual-stage attention-based RNN (DARNN) (Qin et al., 2017), generative adversarial network (GAN) (Romero, 2021), BiLSTM-Seq2Seq (BiLSTM-S) (Mootha et al., 2020), BiLSTM-Multitask (BiLSTM-M) (Mootha et al., 2020), ConvLSTM (Lin et al., 2020), CNN-LSTM (Santika et al., 2021), and Wasserstein GAN (WGAN) (Chen Chen and GaoFeng, 2021) models and our proposed model to forecast the prices of 000423.SZ, 000538.SZ, 000651.SZ, 002304.SZ. We use the MSE, MAE and mean absolute percentage error ($MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_t - y_p}{y_t}\right|$) to evaluate the above model, where y_t is the real value, y_p represents the predicted value, and the smaller the score of the evaluation index is, the better the performance of the prediction method. Tables 3–6 show the evaluation index scores of the target stock in each model, and Fig. 7 shows the prediction result curve of stock 000651.SZ. Fig. 7 (a) - (k) show the curves of the prediction results of each model and the real price. To more intuitively compare the effect differences of each model, we integrate the outputs of different models into the chart in Fig. 7 (l). See the appendix for more target stock forecast curve results.

Since the stock prices predicted by the same model are different, in Tables 3–6, we use the average of the results of multiple experiments to represent the performance of the model. LSTM and CNN exhibit high accuracy and good performance for each stock. By comparing the two basic networks of LSTM and CNN with the two network models of CNN-LSTM and ConvLSTM, we can see that the combination of LSTM and CNN can obtain higher prediction accuracy. Moreover, whether compared with the basic model or the improved models, such as GRU, BiLSTM-M, BiLSTM-S, WGAN, DARNN and GAN, the evaluation index score of the model proposed in this paper is the best. Therefore, our model is universal to stocks and can perform well for multiple stocks.

Fig. 7 and the appendix show the results of an experiment. The trend of the curve shows that the trend of the predicted value is generally consistent with that of the real value when using any of the above methods to predict the four stocks. The difference between the models is mainly reflected in the degree of fitting between the predicted value and the real value. Figs. 1 and 7 are the curves of the prediction results of the closing price of 000651.SZ. By comparing the two, we find that many previous methods have also better predicted the general trend of the stock price and obtained good prediction results. Their shortcomings are mainly reflected in the obvious decline in the prediction accuracy near the mutation point, and the degree of fitting is far lower than that of this method. Even the change directions predicted by some methods after the mutation point deviate, which effectively proves the effectiveness of this method.

In Fig. 7, LSTM, GRU, GAN and BiLSTM-M predict only the general trend of future stock prices, without predicting subtle fluctuations. The prediction results of BiLSTM-S are too high as a whole. GRU, DARNN, GAN, WGAN and CNN-LSTM exhibit large prediction delay problems. The prediction result of CNN and ConvLSTM is similar to that of our method, but its fitting degree near the mutation point is lower than that of our method.

Table 3
Index comparison for 000423.

Method	MSE	MAE	MAPE
LSTM	0.5651	0.4902	0.0145
GRU	0.7290	0.5974	0.0177
CNN	0.5459	0.5157	0.0153
DA-RNN	0.8784	0.6296	0.0183
GAN	0.8812	0.6856	0.0202
BiLSTM-M	0.8297	0.7551	0.0222
BiLSTM-S	0.5951	0.6951	0.0205
CNN-LSTM	0.4818	0.4547	0.0135
WGAN	0.7150	0.5880	0.0174
ConvLstm	0.5006	0.4654	0.0138
OURS	**0.4341**	**0.4426**	**0.0130**

Table 4
Index comparison for 000538.

Method	MSE	MAE	MAPE
LSTM	2.4530	1.0610	0.0128
GRU	3.0279	1.2167	0.0147
CNN	2.3323	1.0410	0.0125
DA-RNN	6.2782	1.9910	0.0241
GAN	3.6261	1.3485	0.0163
BiLSTM-M	4.1229	1.6468	0.0203
BiLSTM-S	2.6067	1.0637	0.0129
CNN-LSTM	2.3251	1.0483	0.0126
WGAN	3.5080	1.2770	0.0154
ConvLstm	2.2840	1.0640	0.0128
OURS	**2.1651**	**1.0282**	**0.0125**

Table 5
Index comparison for 000651.

Method	MSE	MAE	MAPE
LSTM	1.6564	0.9316	0.0160
GRU	2.5256	1.1722	0.0199
CNN	1.3501	0.8384	0.0144
DA-RNN	2.5014	1.1682	0.0203
GAN	1.9021	0.9793	0.0168
BiLSTM-M	1.7166	0.9284	0.0159
BiLSTM-S	1.3993	1.1321	0.0197
CNN-LSTM	1.2697	0.7894	0.0136
WGAN	1.6109	0.9592	0.0164
ConvLstm	1.1648	0.7550	0.0130
OURS	**1.0547**	**0.7060**	**0.0123**

Table 6
Index comparison for 002304.

Method	MSE	MAE	MAPE
LSTM	5.5949	1.6618	0.0151
GRU	9.1191	2.0800	0.0188
CNN	3.8020	1.4208	0.0129
DA-RNN	11.266	2.5021	0.0228
GAN	8.3632	2.0504	0.0187
BiLSTM-M	4.6754	1.5840	0.0144
BiLSTM-S	3.6557	1.7138	0.0158
CNN-LSTM	3.9302	1.3889	0.0126
WGAN	6.4381	1.4266	0.0164
ConvLstm	3.7484	1.4215	0.0129
OURS	**3.4193**	**1.3627**	**0.0125**

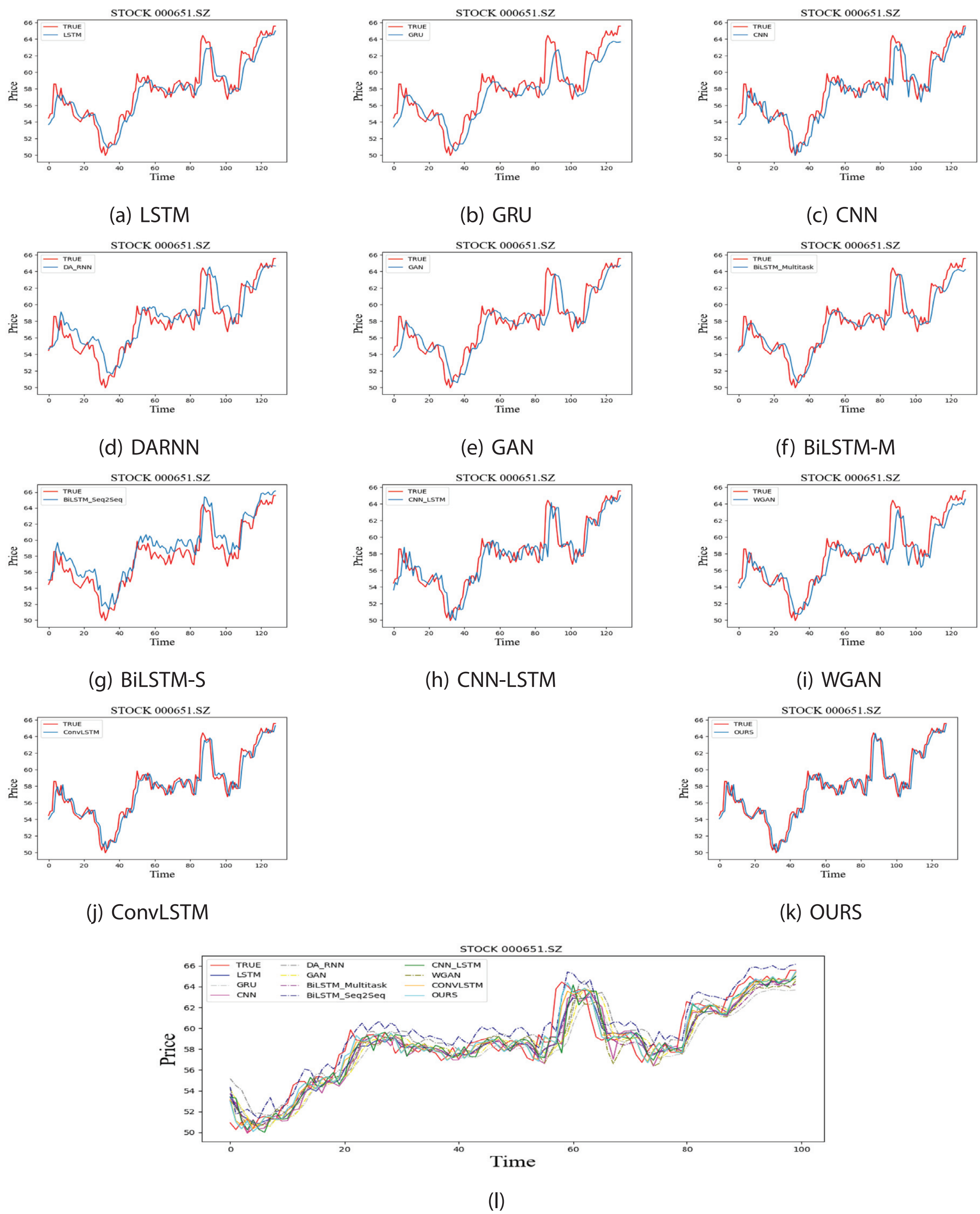


Fig. 7. Prediction results for 000651.

Figure B. 8 in the appendix shows that the delay of GRU, GAN and WGAN predicts the general trend of stock prices in the future without predicting subtle fluctuations. The fluctuation predicted by DARNN is greater than the real fluctuation, and the prediction results of BiLSTM-M and BiLSTM-S are lower as a whole. Compared with CNN-LSTM and ConvLSTM, with better results, our prediction results fit the real values better. Figure B. 9 in the appendix shows that GRU, DARNN, GAN, BiLSTM-M and WGAN predict the general trend of stock prices in the future without predicting subtle fluctuations. Compared with the model proposed in this paper, the predictions of LSTM, CNN, CNN-LSTM and ConvLSTM near the mutation point are not sufficiently accurate. Figure B. 10 in the appendix shows that GRU predicts only the general trend of stock price in the future, while DARNN and WGAN fluctuate only slightly on the basis of the general trend. Near the mutation point, LSTM, GAN, DARNN, GAN, WGAN and ConvLSTM cannot make good predictions. The overall prediction results of BiLSTM-S are too high. Compared with GRU, DARNN and GAN, our method has fewer

Table 7
Ablation experiment of 000651.

Related stock	Mutation point	Piecewise function	MSE	MAE	MAPE
			1.1648	0.7551	0.0130
		√	1.0911	0.7243	0.0125
√			1.0730	0.7163	0.0128
	√		1.0840	0.7188	0.0125
√		√	1.0599	0.7116	0.0124
	√	√	1.0644	0.7107	0.0124
√	√		1.0687	0.7125	0.0124
√	√	√	**1.0547**	**0.7060**	**0.0123**

Table 8
Ablation experiment of 002304.

Related stock	Mutation point	Piecewise function	MSE	MAE	MAPE
			3.7484	1.4215	0.0129
		√	3.5658	1.3892	0.0126
√			3.6086	1.3823	0.0126
	√		3.6123	1.3953	0.0128
√		√	3.5401	1.3794	0.0126
	√	√	3.5233	1.3767	0.0126
√	√		3.5487	1.3829	0.0126
√	√	√	**3.4193**	**1.3627**	**0.0125**

delay problems and more accurate prediction results. The experimental results and analysis show that compared with those of the previous models, the fitting degree of the proposed method is the best and can achieve satisfactory performance in the price prediction task of four target stocks.

To further analyze the results, we performed paired t tests to compare our method and other existing methods. According to Table A. 9 and the corresponding explanation in Appendix A, the findings indicate that the p value of our proposed method is greater than the significance level, with the largest p value compared with the other existing models. This means that the best rationale for accepting the original assumption that "there is no difference between the two sample sets" and for explaining that there is no significant difference between the predicted results and the real values is achieved by our method. Therefore, our method has higher prediction accuracy.

4.3. Ablation experiment

In this section, to verify the effectiveness of the relevant stock subnetwork, mutation subnetwork and piecewise loss function proposed by us, we conduct ablation experiments on 000651.SZ and 002304.SZ, and the above four evaluation indexes are used to evaluate the results. The experimental results are shown in Tables 7 and 8.

In Tables 7 and 8, the first line is the basic model. Lines 2–4 correspond to using only the piecewise loss function, adding only relevant stock information and adding only mutation point information, respectively. Lines 5–7 show the execution of pairwise combinations. The last line is the model proposed in this paper.

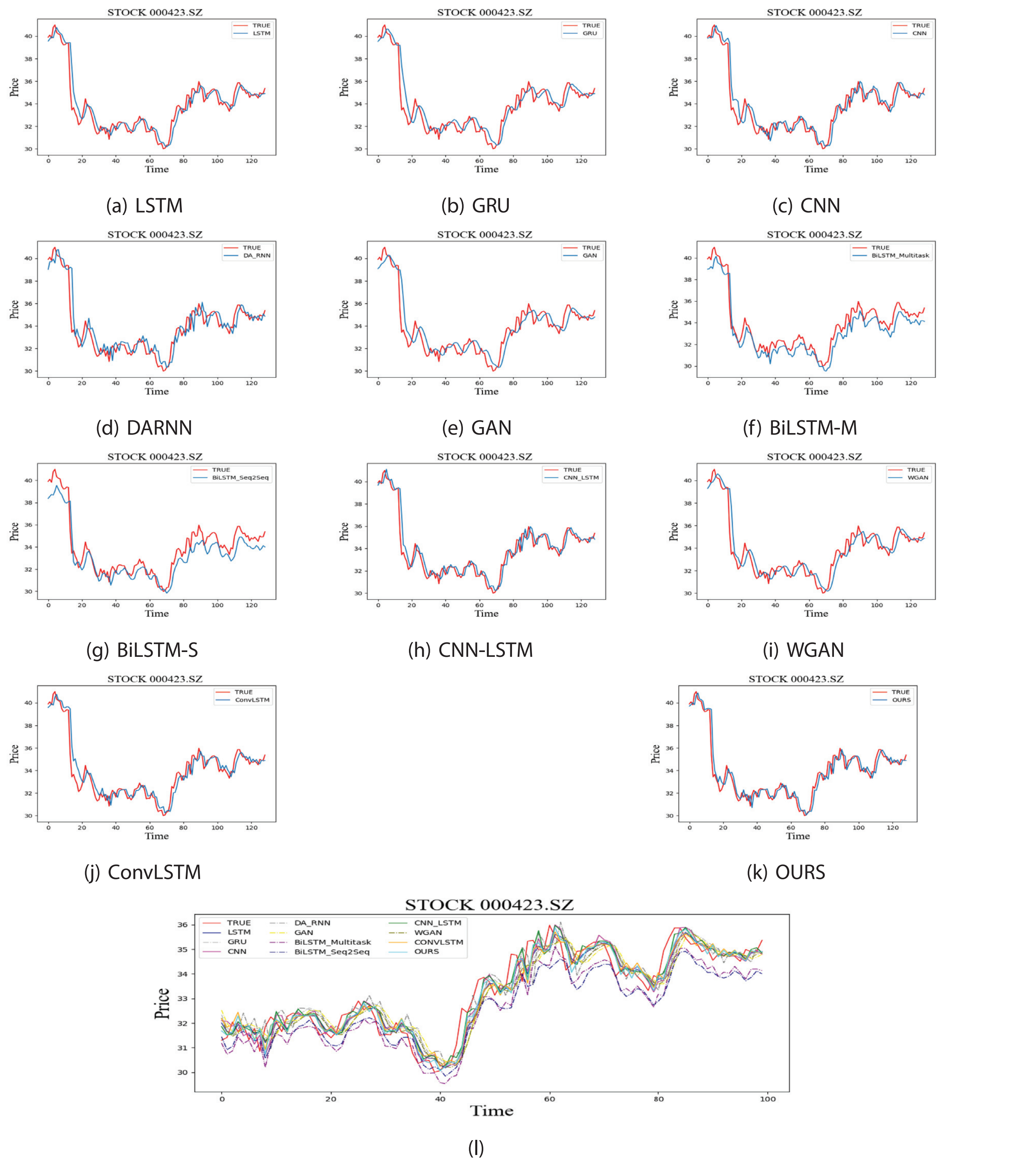
Table 7 shows that the performance of the model with only a single module added to the basic network is slightly better than that of the basic network, with increases of 0.0918 (7.88%), 0.0808 (6.94%) and 0.0737 (6.33%), respectively. Compared with the basic network, the performance of the pairwise combination model has been improved by 0.1049 (9.01%), 0.1004 (8.62%) and 0.0961 (8.25%). Compared with the basic network, our model improves by 0.1101 (9.45%). Table 8 shows that the performance of the model with only a single module added to the basic network is slightly better than that of the basic network, with increases of 0.1398 (3.73%), 0.1361 (3.63%) and 0.1826 (4.87%), respectively. Compared with that of the basic network, the performance of the pairwise combination model is improved by 0.2083 (5.56%), 0.2251 (6.01%) and 0.1997 (5.33%). Compared with the basic network, our model improves by 0.3291 (8.78%). Therefore, the effectiveness of each module in the model in this paper is further proven by experiments.

5. Conclusions

In this paper, first, considering that a single stock cannot fully reflect market information, we find the relevant stocks of the target stock through a knowledge graph and graph embedding technology. Then, the market information vector is weighted according to a similar weight and used as the input data of the ConvLSTM network to obtain the market features. Through the analysis of the prediction results, we find that the fitting effect of the model near mutation points is not good. Based on this, we propose the mutation point detection subnetwork, construct the mutation point distance weight matrix, and further obtain the stock price mutation features through the GCN. Finally, the features of stock price, market and mutation point are integrated to jointly predict the closing price of stocks. Additionally, considering that the exis-

Table A.9
Paired t test results.

	2-tailed paired t test	1-tailed paired t test
LSTM	0.003992	0.001996
GRU	1.29982E-05	6.49911E-06
CNN	0.005259	0.002629
DA-RNN	0.018765	0.009383
GAN	0.088239	0.04412
BiLSTM-M	0.056511	0.028256
BiLSTM-S	1.21447E-19	6.07233E-20
CNN-LSTM	0.215992	0.107996
WGANS	0.002989	0.001495
ConvLstm	0.08086	0.04043
OURS	0.226844	0.113422



(a) LSTM (b) GRU (c) CNN

(d) DARNN (e) GAN (f) BiLSTM-M

(g) BiLSTM-S (h) CNN-LSTM (i) WGAN

(j) ConvLSTM (k) OURS

(l)

Fig. B.8. Prediction results for 000423.

tence of a mutation point results in a large deviation between the real value and the predicted value, we use the piecewise loss function to eliminate the large error. The experimental results show that our model further improves the prediction accuracy.

6. Future work

In the future, we will combine different types of information as input (such as more market indicators, emotional data, and news data) to obtain more comprehensive features and achieve more accurate stock price predictions. Based on the combination of different types of information, we have preliminarily explored two methods based on the framework of this paper: (1) Other types of information are input together with "CLOSE" into the target stock price subnetwork as the input features of the target stock; (2) On the basis of the framework of this paper, a related indicator subnetwork is added and other types of information are utilized as the input data of this subnetwork. The preliminary experiments show that the prediction performance of (1) decreases. However, the predictive performance of (2) improves. We surmise that the existence of noise and missing values in the dataset affects the performance of the model. Therefore, we will continue to explore the influencing factors of stock price predictions, dataset processing methods, model improvements and other aspects to complete the purpose of accurate stock price predictions.

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.

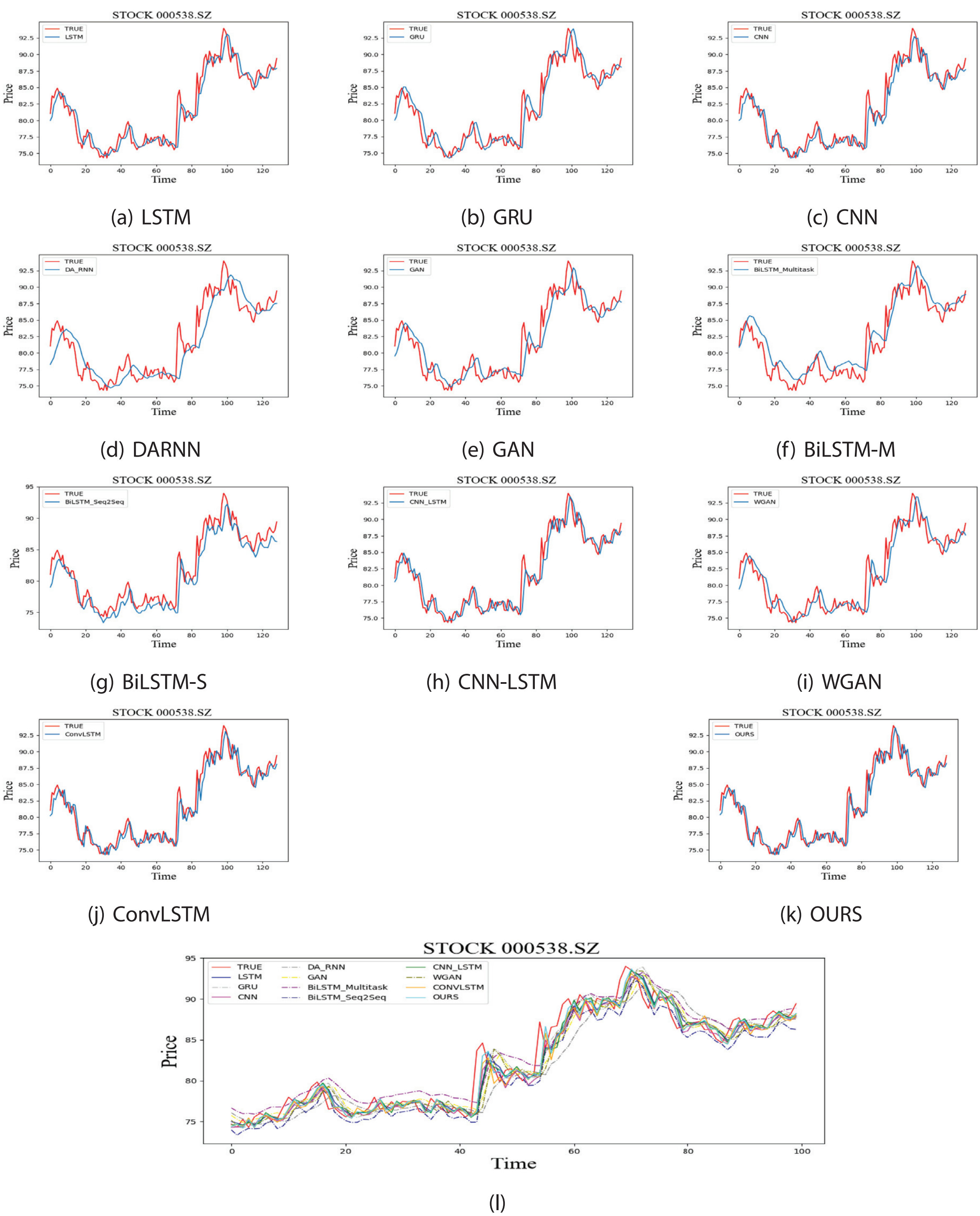


Fig. B.9. Prediction results for 000538.

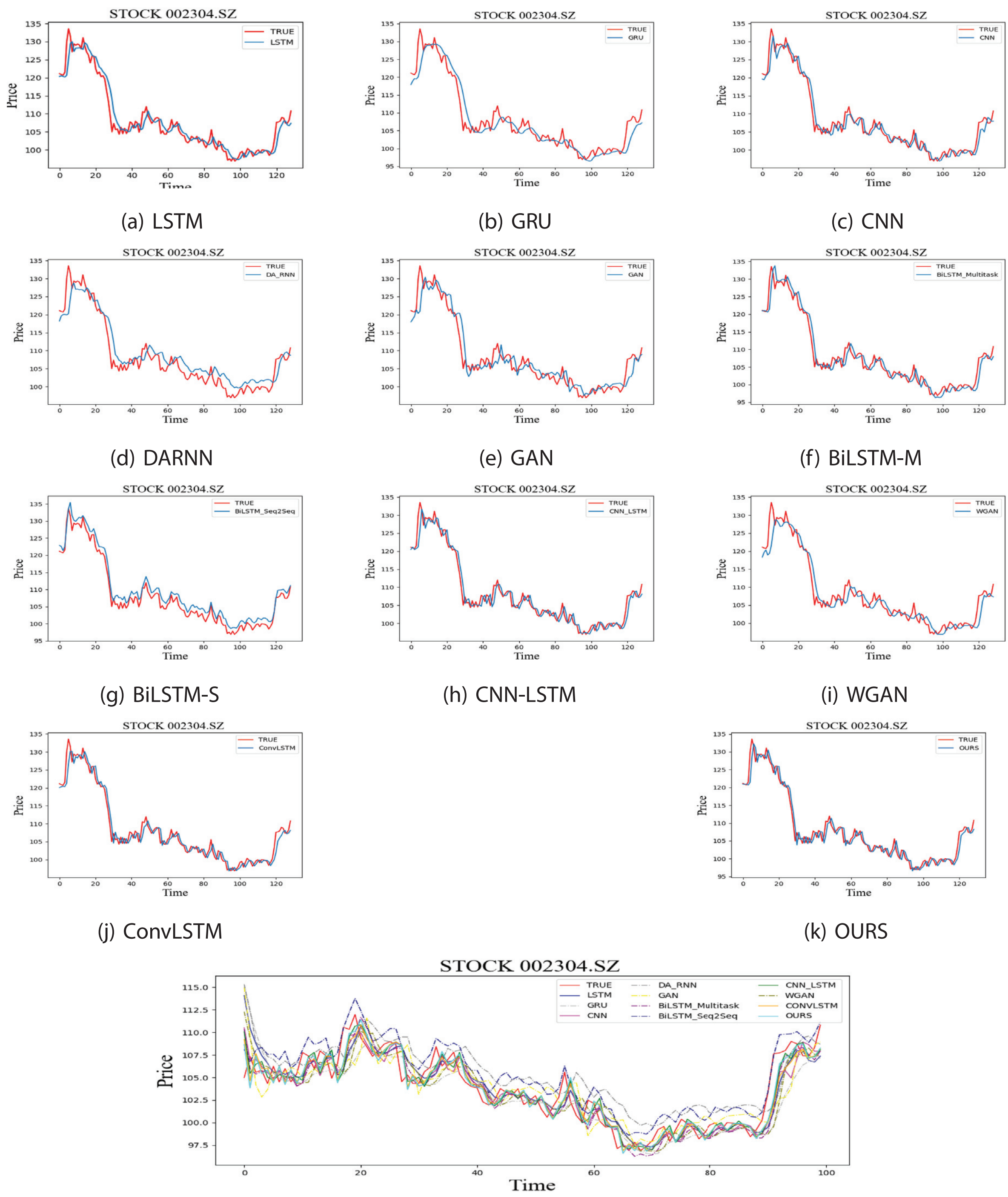
Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (U1909210), Natural Science Foundation &Key Research and Development Program of Shandong Province (ZR2020MF037, ZR2019MF016, 2020CXGC010110, 2021SFGC0102), The introduction and education plan of young creative talents in Colleges, Jinan scientific research leader studio (Z2020025) and Humanities and Social Sciences Project of Education Ministry (20YJA870013).

Appendix A

Hypothesis: There is no difference between the two sample sets. **Result analysis**: In regard to the probability of rejecting the original hypothesis, if the p value of the t test statistic is less than or equal to the significance level α, the original hypothesis is rejected; that is, there is a significant difference between the two hypotheses. Otherwise, the original hypothesis is accepted.

The result analysis of this paper: At a significance level of $\alpha = 0.05$, if $p - value > 0.05$, the original hypothesis is accepted,



(a) LSTM (b) GRU (c) CNN

(d) DARNN (e) GAN (f) BiLSTM-M

(g) BiLSTM-S (h) CNN-LSTM (i) WGAN

(j) ConvLSTM (k) OURS

(l)

Fig. B.10. Prediction results for 002304.

indicating that there is no significant difference between the predicted results and the real values. If $p - value < 0.05$, the original hypothesis is rejected, indicating that there is a significant difference between the predicted results and the real values. (The values in the table are the calculated p values.)

The table indicates that the two-tailed paired t test results achieve p values greater than 0.05 for the GAN, BiLSTM-M, CNN-LSTM, ConvLstm and OURS. Therefore, the original hypothesis is accepted, indicating that there is no significant difference between the predicted results and the real values. Moreover, the p value of OURS is the largest, achieving the most compelling reasons to accept the original hypothesis and to explain that there is no significant difference between the predicted results and the real values. For the one-tailed paired t test, the p values of CNN-LSTM and OURS are greater than 0.05, so the original hypothesis is accepted, indicating that there is no significant difference between the predicted results and the real values. Moreover, the p value of OURS is the largest, which substantiates accepting the original hypothesis and explaining that there is no significant difference between the predicted results and the real values. The two t test statistical methods in the above analysis demonstrate that the p value of the proposed method is the largest, which is the most reasonable explanation that there is no significant difference between the predicted results and the real values.

Appendix B

Figs. B.8–B.10 show the prediction results for 000423.SZ, 000538.SZ, 002304.SZ of each model, respectively, where (a) - (k) are the curves of the prediction results of each model and the real price. To more intuitively compare the effect differences with each model, we integrate the outputs of different models into (l).

References

Adebiyi, A.A., Adewumi, A.O., Ayo, C.K., 2014. Comparison of arima and artificial neural networks models for stock price prediction. J. Appl. Math. 2014 (1), 1–7.

Akimushkin, C., Amancio, D.R., Oliveira Jr, O.N., 2018. On the role of words in the network structure of texts: Application to authorship attribution. Physica A 495, 49–58.

Y. Cao, X. Wang, X. He, Z. Hu, T.-S. Chua, Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences, in: The world wide web conference, 2019, pp. 151–161.

M. Cerliani, Time series forecasting with graph convolutional neural network, https://towardsdatascience.com/time-series-forecasting-with-graph-convolutional-neural-network-7ffb3b70afcf/ (2020).

L.H. snd Chen Chen, H. GaoFeng, Stock price prediction using generative adversarial networks, Journal of Computer Science 17 (3) (2021) 188–196.

K. Chen, Y. Zhou, F. Dai, 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), IEEE, 2015, pp. 2823–2824.

S. Chen, H. He, Stock prediction using convolutional neural network, in: IOP Conference series: materials science and engineering, Vol. 435, IOP Publishing, 2018, p. 012026.

Y. Chen, Z. Wei, X. Huang, Incorporating corporation relationship via graph convolutional neural networks for stock price prediction, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 1655–1658.

S. Das, T.P. Sahu, R.R. Janghel, Stock market forecasting using intrinsic time-scale decomposition in fusion with cluster based modified csa optimized elm, Journal of King Saud University-Computer and Information Sciences.

Fama, E.F., 1965. The behavior of stock-market prices. J. Business 38 (1), 34–105.

A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, 2016, pp. 855–864.

M.U. Gudelek, S.A. Boluk, A.M. Ozbayoglu, A deep learning based stock trading model with 2-d cnn trend detection, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2017, pp. 1–8.

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

H.D. Huynh, L.M. Dang, D. Duong, A new model for stock price movements prediction using deep neural network, in: Proceedings of the Eighth International Symposium on Information and Communication Technology, 2017, pp. 57–62.

Jiang, W., 2021. Applications of deep learning in stock market prediction: recent progress. Expert Syst. Appl. 115537.

Kang, y., Xu, z., 2007. The application of cusp catastrophe theory in the stock price analyze. J. Harbin University Sci. Technol. 12 (1), 100–104.

T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907.

B. Koloski, T. Stepinik-Perdih, M. Robnik-Ikonja, S. Pollak, B. Krlj, Knowledge graph informed fake news classification via heterogeneous representation ensembles, Neurocomputing.

Lin, P., Zhou, J., Wen, Y., 2020. Sconv: A financial market trend forecast method based on emotional analysis. J. Computer Res. Dev. 57 (8), 1769–1778.

Long, J., Chen, Z., He, W., Wu, T., Ren, J., 2020. An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in chinese stock exchange market. Applied Soft Computing 91, 106205.

S. Mootha, S. Sridhar, R. Seetharaman, S. Chitrakala, Stock price prediction using bi-directional lstm based sequence to sequence modeling and multitask learning, in: 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE, 2020, pp. 0078–0086.

Nam, K., Seong, N., 2019. Financial news-based stock movement prediction using causality analysis of influence in the korean stock market. Decis. Support Syst. 117, 100–112.

S.R. Polamuri, K. Srinivas, A.K. Mohan, Multi-model generative adversarial network hybrid prediction algorithm (mmgan-hpa) for stock market prices prediction, Journal of King Saud University-Computer and Information Sciences.

Y. Qin, D. Song, H. Chen, W. Cheng, G.W. Cottrell, A dual-stage attention-based recurrent neural network for time series prediction.

R.A.C. Romero, Generative adversarial network for stock market price prediction.

I.W.K.G. Santika, S. Sa'adah, P.E. Yunanto, Gold price prediction using convolutional neural network-long short-term memory (cnn-lstm), Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control.

L.D. Santos, E. Corrêa, O. Oliveira, D.R. Amancio, L.L. Mansur, S. Aluísio, Enriching complex networks with word embeddings for detecting mild cognitive impairment from speech transcripts, arXiv e-prints.

J. Sen, S. Mehtab, Optimum risk portfolio and eigen portfolio: a comparative analysis using selected stocks from the indian stock market, arXiv preprint arXiv:2107.11371.

X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-C. Woo, Convolutional lstm network: A machine learning approach for precipitation nowcasting, Advances in neural information processing systems 28.

S. Sinha, S. Mishra, V. Mishra, T. Ahmed, Sector influence aware stock trend prediction using 3d convolutional neural network, Journal of King Saud University-Computer and Information Sciences.

Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 28, 2014.

H. Wang, F. Zhang, X. Xie, M. Guo, Dkn: Deep knowledge-aware network for news recommendation, in: Proceedings of the 2018 world wide web conference, 2018, pp. 1835–1844.

H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, M. Guo, Ripplenet: Propagating user preferences on the knowledge graph for recommender systems, in: Proceedings of the 27th ACM international conference on information and knowledge management, 2018, pp. 417–426.

H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, M. Guo, Multi-task feature learning for knowledge graph enhanced recommendation, in: The world wide web conference, 2019, pp. 2000–2010.

Wei, B., Jun, Y., Yulei, R., Boris, P., 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS ONE 12, (7) e0180944.

Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, C. Zhang, Connecting the dots: Multivariate time series forecasting with graph neural networks, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 753–763.

Xu, Y., Zhang, G., 2015. Application of kalman filter in the prediction of stock price. In: International Symposium on Knowledge Acquisition and Modeling (KAM). Atlantis press, pp. 197–198.

B. Yang, S.W.-t. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: Proceedings of the International Conference on Learning Representations (ICLR) 2015, 2015.