Expert Systems With Applications 202 (2022) 117239

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Transformer-based attention network for stock movement prediction

Qiuyue Zhang [a], Chao Qin [b], Yunfeng Zhang [b,c,*], Fangxun Bao [d], Caiming Zhang [e,f], Peide Liu [a]

[a] School of Management Science and Engineering, Shandong University of Finance and Economics, Jinan, 250014, China
[b] School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, 250014, China
[c] Shandong Key Laboratory of blockchain Finance, Shandong University of Finance and Economics, Jinan, 250014, China
[d] School of Mathematics, Shandong University, Jinan, 250100, China
[e] School of Software, Shandong University, Jinan, 250101, China
[f] Shandong Co-Innovation Center of Future Intelligent Computing, Yantai, 264025, China

ARTICLE INFO

Keywords:
Stock movement prediction
Deep learning
Transformer
Attention

ABSTRACT

Stock movement prediction is an important field of study that can help market traders make better trading decisions and earn more profit. The fusion of text from social media platforms such as Twitter and actual stock prices is an effective but difficult approach for stock movement prediction. Although some previous methods have explored this approach, there are still difficulties with the temporal dependence of financial data and insufficient effectiveness of fusing text and stock prices. To solve these problems, we propose the novel Transformer Encoder-based Attention Network (TEANet) framework, which is based on precise description through small-sample feature engineering and uses a small sample of 5 calendar days to capture the temporal dependence of financial data. In addition, this deep learning framework uses the Transformer model and multiple attention mechanisms to achieve feature extraction and effective analysis of financial data to achieve accurate prediction. Extensive experiments on four datasets demonstrate the effectiveness of our framework. Further simulations show that an actual trading strategy based on our proposed model can significantly increase profit and has practical application value.

1. Introduction

Stock movement prediction has attracted the attention of both investors and researchers for decades due to its great value in seeking to maximize stock profit (Hu et al., 2018). Early approaches mainly relied on historical stock prices and time series analysis methods (Akaike, 1969). However, stock movement prediction is quite a challenging issue because of the highly volatile and nonstationary nature of the stock market. Moreover, the stock market is affected by random noise generated by participants with different viewpoints, making its movements complicated and difficult to predict. The efficient market hypothesis (EMH) argues that stock prices are driven by all observable information and relevant news (Fama, 1965; Fama et al., 1969). This classical theory opens the door to predicting financial market trends and movements, and many experts have dedicated themselves to improving the accuracy of such predictions, which may lead to better trading decisions.

Based on Fama's hypothesis, the scientific community has developed a large number of different ways to predict the stock market (Cavalcante et al., 2016). The most common method of stock market prediction in the early literature was to take stock prices or indicators extracted from them as input (Aguilar-Rivera et al., 2015; Atsalakis & Valavanis, 2009). It was believed that all new information, such as news and social media discourse, is fully reflected in the stock price and that it is thus sufficient to predict the stock market simply by analyzing the patterns of price movements. The related indicators have been widely studied and used as signals to buy or sell stocks, reflecting the current state of the stock market (Farias Nazário et al., 2017; Yang et al., 2019). However, research has shown that the effectiveness of using price or indicator data alone to make trading decisions is limited (Park & Irwin, 2007).

Since it is difficult to establish a model to comprehend stock fluctuations, the most commonly used information is related to macroeconomic time series, such as GDP, interest rates, currency exchange rates, and consumer price indices (Boyacioglu & Avci, 2010). Other sources of information include general financial news reports; however, their unstructured nature and discontinuous behavior make them difficult to use. Consequently, natural language processing (NLP) techniques have been applied to address this complexity. Several studies have focused on news and social media analytics and have gradually formed the

* Corresponding author at: School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, 250014, China.
E-mail addresses: 201106007@mail.sdufe.edu.cn (Q. Zhang), 182115011@mail.sdufe.edu.cn (C. Qin), yfzhang@sdufe.edu.cn (Y. Zhang), fxbao@sdu.edu.cn (F. Bao), czhang@sdu.edu.cn (C. Zhang), liupd@sdufe.edu.cn (P. Liu).

https://doi.org/10.1016/j.eswa.2022.117239
Received 29 November 2020; Received in revised form 1 April 2022; Accepted 10 April 2022
Available online 21 April 2022

research field of stock market prediction based on natural language-based financial forecasting (NLFF). Among the noteworthy studies of this kind, some studies have made use of public news to predict future stock movements (Bustos & Pomares-Quimbaya, 2020; Huynh et al., 2017; Kraus & Feuerriegel, 2017; Li et al., 2020). Moreover, social media is a more time-sensitive information source than public news; accordingly, the use of texts such as tweets to predict stock movements has recently received considerable attention. In addition, the data expressed on social media are able to directly reflect investors' attitudes, which are among the main influencing factors for stock movement prediction. Consequently, many recent studies have used only tweets to predict stock movements while exploring the feature information of these texts at a deeper level (Araci, 2019; Liu et al., 2019; Sohangir & Wang, 2018). Nevertheless, the importance of temporal dependence for movement prediction remains a critical issue, as data closer to the target trading day have a greater impact on the prediction results.

In the above research on stock prediction, a few studies have combined NLP with historical stock prices to realize stock market prediction. Tweets collected on social media were combined with actual stock price data, and the time window for judging stock trends was narrowed (Wu et al., 2018; Xu et al., 2020; Xu & Cohen, 2018). Comparative experiments showed that this combined method is superior to analyzing stock prices or tweets alone. The main reason is that social media comments and actual stock market prices are both factors that are closely related to the stock market and directly reflect stock trends. Therefore, once the temporal dependence problem has been solved, extracting features from text and prices simultaneously and then integrating them can fundamentally improve the effectiveness of stock prediction. However, there are few frameworks that use this approach to realize stock market prediction; therefore, it is a problem worth studying to develop a more effective method of analyzing financial data.

Considering the above research results, some challenges still remain in the task of stock movement prediction: the stock market is a time series problem, leading to temporal dependence in financial data, and the effectiveness of a large amount of historical information will thus be reduced over time; moreover, on the premise that the temporal dependence problem can be solved, there remains the question of how to more accurately analyze and predict the movements of stocks based on financial data such as tweets and historical stock prices. In essence, considering the conditions of traders and the actual prices of stocks together can enable a more comprehensive prediction of stock movements. To effectively solve the above problems, we propose the novel Transformer encoder-based attention network (TEANet) architecture, which is a small-sample feature engineering network framework based on a precise description that includes a *feature extractor* and a *concatenation processor*. We use a small sample of only 5 calendar days as training data to solve the temporal dependence problem for financial data. On this basis, we are inspired by the field of NLP and use the *feature extractor*, which includes a Transformer encoder, to extract deep text features and use the *concatenation processor* to further analyze tweets and stock prices to integrate the influence of these diverse factors for stock movement prediction. In this way, the framework can be used to effectively predict stock movements on given trading days based on the features learned from the portfolio model.

For comprehensive experiments, we choose several evaluation metrics to verify the effectiveness of the proposed method in various situations. First, we investigate the feasibility of TEANet by comparing its predictive performance with that of other baseline models. Second, we conduct an ablation study of the integrated TEANet framework, including ablation of the input data and the model components. Then, visual experiments are presented to demonstrate the effects of various attention mechanisms. Through the performance comparison with related methods, the effectiveness of the method proposed in this paper is fully demonstrated. Furthermore, stock trading simulations are used to illustrate the potential application of the proposed model in actual market trading. Finally, we analyze the performance of the TEANet model under special circumstances to detect weaknesses of this method.

In summary, the contributions of our work include the following:

- A small-sample feature engineering network framework based on a precise description is proposed, which can effectively solve the problems currently faced in stock movement prediction, namely, the temporal dependence of financial data, and needs to further improve the effectiveness of fusing tweets and stock price data.
- The fusion of the Transformer model and various attention mechanisms is introduced for the first time for stock movement prediction to construct TEANet, in which the Transformer model is used to extract deep features of small samples and multiple attention mechanisms are used to capture dependencies and obtain key information. The integration of these approaches enables more accurate stock movement prediction.
- In experiments, the proposed TEANet method is found to be significantly superior to several state-of-the-art baselines, demonstrating that the proposed method is more robust and practical than these baselines and can be successfully applied in the financial industry.

The rest of this paper is structured as follows. After this brief introduction, in Section 2, a survey of existing work is presented, with a focus on predicting stock movements. Section 3 describes the background related to our work. Detailed descriptions of our proposed framework are provided in Section 4. Section 5 explains how the data used in the experiments were collected and discusses the results. We conclude the paper and identify future directions of research in Section 6.

2. Related work

Prediction refers to the use of present and past data to anticipate the future. Supposing that there exists a time series $x_{t-1}, x_{t-2}, \ldots, x_{t-p}$, the purpose is to predict future values $y_{t-1}, y_{t-2}, \ldots, y_{t-p}$ based on the collected data (Lund, 2007). Many methods are currently used for stock market prediction, mainly based on machine learning (ML) algorithms, which can be divided into more specific classes of algorithms as follows: traditional ML, deep learning (DL), and hybrid approaches (Lee & Kim, 2020).

As early as 40 years ago, finance was one of the most researched applications of ML. To date, thousands of studies have been published in various areas of finance, and researchers' interest in this topic has not waned. Especially in the field of stock prediction, many traditional ML methods have been widely applied (Barak et al., 2017; Chen & Hao, 2017; Gorenc Novak & Velušček, 2016). Bahrammirzaee (2010) demonstrated the application of artificial neural networks (ANNs) and expert systems to financial markets. Zhang and Zhou (2004) reviewed the current popular techniques for text data mining related to the stock market, mainly including genetic algorithms (GAs), rule-based systems, and neural networks (NNs). Meanwhile, a number of papers have proposed specific ML techniques. Among them, evolutionary algorithms are relatively common, including GAs and particle swarm optimization (PSO), which are generally used for financial optimization purposes, such as optimizing stock prediction algorithms. Chalup and Mitschele (2008) proposed the use of kernel methods, including principal component analysis (PCA) and support vector machines (SVMs), to predict the movements of the stock market. However, one of the main limitations of traditional ML methods is that they cannot extract deep features of financial data to bridge the gap in predictive power between machines and humans.

In recent years, predictive approaches based on DL have been actively studied (Ding et al., 2020; Feng et al., 2019; Wu et al., 2018). Patil et al. (2020) proposed a novel approach based on graph theory that utilizes information about the spatial–temporal relationships between different stocks by modeling the stock market as a

complex network. This graph-based approach was used in conjunction with two other techniques to create two hybrid DL models based on correlations of historical stock prices and a causal graph based on financial news reports over time. Li et al. (2020) proposed an LSTM-based model that used four different sentiment dictionaries to conduct a five-year stock market prediction experiment on stock technical indicators and textual news data from Hong Kong. This article adopts a unique method text and data fusion, and gives full play to the effects of both. Sohangir and Wang (2018) proposed using stock Twitter data to make financial predictions via DL methods, such as CNN, to help investors make decisions. This method is more innovative than previous analysis methods and provides inspiration for later solutions. Vargas et al. (2018) used financial news titles and a set of price indicators to estimate intraday directional movements. These papers focused on architectures such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), which have yielded good financial prediction results with the help of traditional NLP methods. Liu et al. (2019) proposed a capsule network model based on a Transformer encoder (CapTE) that uses the Transformer encoder to extract deep semantic features of social media texts and then captures the structural relationships of those texts through the capsule network. In most current research, attention mechanisms have begun to be used as the main structures for solving financial market prediction problems, especially temporal attention mechanisms, to focus on key positions that have greater impact on the results. For example, Hu et al. (2018) developed a hybrid attention network (HAN) to predict stock trends based on sequences of recent related news items, which can significantly increase the annualized return. (Xu et al., 2020) also proposed a stock price prediction network (SMPN) with the help of tweets and stock prices that is based on incorporative attention mechanisms and multilevel local features. Wang et al. (2020) conducted an in-depth analysis of the timeliness and target sensitivity of stock investment reviews as well as the reliability of investors, public stock commentaries and their application in stock movement prediction. That paper proposed a new framework based on expert dynamic attitude aggregation for stock movement prediction, in which scores are mainly obtained through temporal attention to realize decision-making. Therefore, before we can reach the next narrative curve in the framework of using NLP to solve financial forecasting problems, we have reason to expect to witness different means of competition for a long time. As seen from the abovementioned research, DL frameworks show great potential in financial fields, and the experimental results produced to date are quite satisfactory, providing excellent reference value for our research.

Another major means of improving stock market prediction is to incorporate additional strategies that can be used to assist in predicting market movements. Hybrid methods, namely, DL methods combined with other auxiliary algorithms, have demonstrated great improvements in predictive ability (Chen et al., 2018; Lee & Soo, 2018). Feng et al. (2019) proposed a new DL method for predicting stock movements in which the aim is to predict whether stock prices will rise or fall in the near future. The key novelty lies in the use of adversarial training to improve the generalization ability of NN prediction models and solve the problems of data overfitting and insufficiency to obtain reliable models. Xu and Cohen (2018) proposed a new deep generative model combining text and price signals to address the complexity of stock data. In contrast to discriminant or topic modeling, this model introduced recursive, continuous potential variables to better accommodate randomness, and variational autoencoders (VAEs) were used for in-depth reasoning. The above research suggests that improved hybrid methods can be more suitable for predicting stock fluctuations than a DL framework alone.

In summary, using an improved hybrid DL framework to analyze stock-related financial data is an effective way to improve the results of stock market forecasting. There is a need for such a combined framework that can not only extract deep features of text and stock prices but also effectively fuse the extracted features, thereby comprehensively improving the accuracy of prediction and increasing the application value. However, the current DL frameworks still have room for improvement in realizing such integrated analysis based on text and stock prices. In this paper, we propose the novel framework TEANet, which utilizes the currently lesser known Transformer model and various attention mechanisms to collaboratively predict stock movements on a target trading day. Specifically, we hypothesize that using deeper textual semantic information will be more advantageous than using only historical stock prices. Accordingly, we wish to design an effective network structure that can process time series financial data and select and learn the necessary information from the corresponding text. Therefore, we adopt the Transformer architecture, which is completely different from traditional NNs. In particular, it has the following two advantages: it is not troubled by long-term dependence problems and does not rely on past hidden states to capture previous words, so there is no risk of loss of past information, and it avoids recursion and instead processes each sentence as a whole, thus allowing parallel calculations and making it possible to reduce the training time. Considering the temporal dependence of financial data, we use a small sample of 5 calendar days adjacent to the target trading day to make predictions, thereby reducing the impact of historical data with low relevance on the prediction results. Accordingly, we use an improved temporal attention mechanism to analyze financial data from the perspectives of dependence and information content to extract features that are more beneficial for predicting the stock market. In addition, we adopt an auxiliary prediction strategy, referred to as the *objective-level temporal auxiliary* strategy, in which the stock situation on previous trading days is used to assist in predicting the movements on the target trading day. Overall, the TEANet architecture described above serves as the basis for a small-sample feature engineering network framework based on precise description to realize effective and practical stock prediction.

3. Background

In this section, through detailed theoretical analysis, we will reveal the internal process of the Transformer model and several attention mechanisms, which are the basic components of TEANet.

3.1. Transformer

The Transformer is a classic NLP product proposed by the Google team that is superior to RNNs and CNNs for machine translation tasks. This model mainly relies on an attention mechanism, and it has an advantage in its ability to be effectively parallelized, as measured by the minimum number of sequential operations required. Radford et al. (2018) performed various ablation studies and analyzed the effectiveness of the Transformer by comparing it with a similar framework using the long short-term memory (LSTM) architecture; it was found that the experimental effect was greatly reduced when using the LSTM architecture. Because the Transformer overcomes the limitation of RNN models that the relevant calculations cannot be performed in parallel and the number of operations required to calculate the association between two positions does not increase with distance, unlike in a CNN, we use the Transformer architecture as the main architecture for TEANet.

Fig. 1 depicts the whole Transformer structure, where x represents the input data. The model consists of two stacked encoders and decoders with basically the same composition. The input to an encoder first enters a self-attention layer, which helps the encoder view other words in the input sequence as it encodes a word. The output is then passed to a fully connected feedforward NN with the simplest possible network structure, in which each neuron is arranged hierarchically. The feedforward NN of each encoder has the same number of parameters, but the functions are independent. There are residual connections around the two submodels, and layer normalization is performed thereafter. To consider the order of the input words, the Transformer adds a

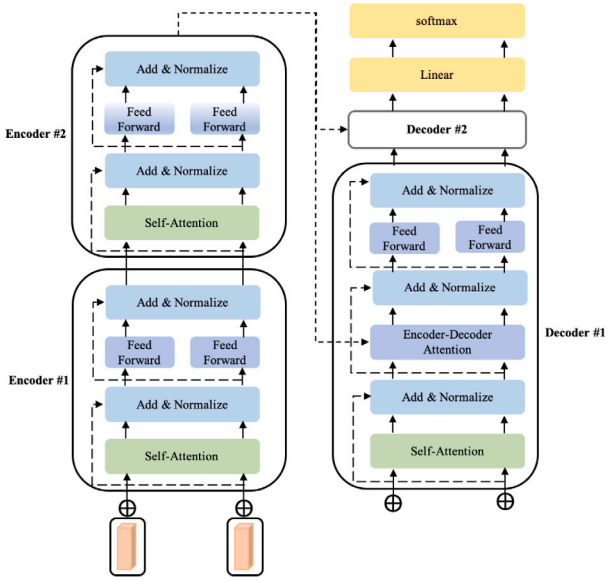


Fig. 1. The architecture of Transformer.

new position vector to the word embedding so that it can capture the position of each word or the distances between different words in the sequence. The final output of a decoder is a vector list of floating-point numbers, which is transformed into words with the help of linear and softmax layers. The linear layer is a simple fully connected NN layer that takes the output generated by the decoder stack and projects it as a logarithmic vector. Connected to the linear layer is a softmax layer, which converts scores into probabilities and selects the index with the highest probability; then, the corresponding word is provided as the output.

3.2. Attention mechanisms

Inspired by the visual attention mechanism of the fovea, a selective attention mechanism focusing on the relevant parts of the input has been proposed based on measuring the sensitivity of the output to the input variance (Pei et al., 2017). Such an attention mechanism not only fundamentally improves model performance but also enables stronger interpretability (Xu et al., 2015). The classical attention mechanisms include traditional attention, multihead attention and temporal attention, which are widely used in text classification, machine translation, image recognition, video processing and other fields (Araabi & Monz, 2020; Dosovitskiy et al., 2021; Fan et al., 2021; Gabeur et al., 2020; Wu et al., 2020).

$$\alpha_i = softmax(f(key_i, q)) \tag{1}$$

$$att((K,V),q) = \sigma_{i=1}^{N} \alpha_i X_i \tag{2}$$

$$attention((K,V),Q) = att((K,V),q_1) \oplus \ldots \oplus att((K,V),q_M) \tag{3}$$

In our work, we adopt four classical attention mechanisms, as mentioned above, which have many similarities. First, traditional attention is essentially an addressing process. Given a task-related query vector q, the attention value is calculated from the attention distribution with the corresponding key, to which the calculated value is then attached. When the input information is $X = [x_1, x_2, \ldots, x_N]$, the architecture is as described in Fig. 2 and formulas (1)–(2). The function f is an indispensable scoring mechanism that includes three operations: sum, dot product and scale dot product. Second, self-attention

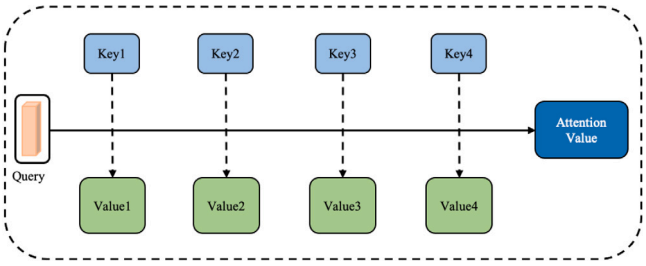


Fig. 2. The architecture of traditional attention mechanism.

is a special attention mechanism that is the same as the traditional attention mechanism except that Q=K. On this basis, in multihead attention, information is selected from the input through multiple queries $Q = [q_1, q_2, \ldots, q_M]$ to be calculated in parallel. Each attention head focuses on a different part of the input information, and the results are then concatenated. Accordingly, the multihead attention mechanism is realized as shown in formula (3), which enables the model to process data as in the traditional attention mechanism, where $\oplus$ denotes elementwise concatenation. Finally, the goal of temporal attention is to estimate the saliency and correlation of each sequential observation. The significance score should not only be based on the input observations in the current time step but should also consider the information of adjacent observations in both directions. The specific architecture varies with the particular application and is mainly used to learn the attention weights for each step, while the process for each individual step is essentially the same as the traditional attention mechanism. Due to the importance of timeliness in the financial field, temporal attention is widely used in stock market analysis. We have further improved the original model, and the specific implementation process is explained in Section 4.

In summary, both the Transformer model and various attention mechanisms have been demonstrated to effectively process data and have achieved certain results for various tasks. Due to the strong temporal dependence of financial data in the stock market context and the room for further improvement in analyzing such data, we propose the TEANet framework for the task of predicting the movements of a small sample of stocks based on tweets and stock prices, effectively achieving accurate prediction and improving profit.

4. Methodology

In this section, we first formalize the stock movement prediction problem. Then, we propose our framework based on the two design principles discussed in the background analysis (Section 3). Accordingly, we propose the TEANet architecture, which consists of a *feature extractor* and a *concatenation processor*.

4.1. Problem formulation

We wish to predict the movement of stock s on trading day td. We use the tweet corpus T and the historical prices in a lag period $[d - \delta d, d - 1]$, where δd is a fixed lag size. The output is a judgment on the binary movement direction, where 1 denotes rise and 0 denotes falls:

$$y = 1(p_{td}^{c} > p_{td-1}^{c}) \tag{4}$$

where p_{td}^{c} denotes the adjusted closing price, which is adjusted in response to actions that affect the movement of the stock, such as dividends and splits. The adjusted closing price has also been used for stock movement prediction prior to our work (Xie et al., 2013).

As explained in Section 1, we can predict future market evolution based on historical financial data. Accordingly, there may be a certain

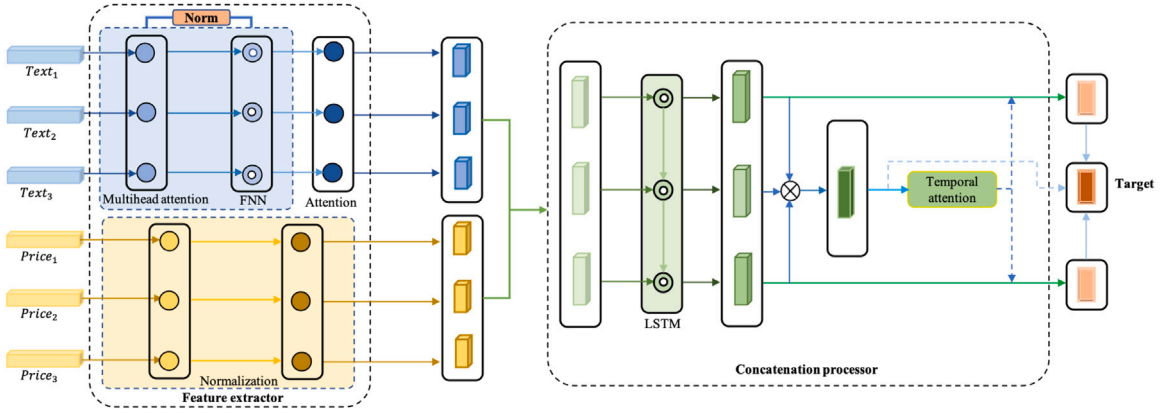


Fig. 3. The architecture of TEANet.

lag with respect to the target trading day *td*, allowing us to also simulate and predict other target days close to *td*. We can predict the movements not only on *td* itself but also on other trading days within the lag period. For example, if we choose 23/09/2020 as the target trading day, then 18/09/2020 and 22/09/2020 are the calendar days representing the endpoints of the lag period (the lag period is usually 5 days); thus, we mainly capture the relationships between the predictions within this sample range. However, considering the actual financial market conditions, we ignore nontrading days in the calculation process to effectively organize and utilize the input data. For this reason, determining the correlations between text and stock prices is critical for the handling of the text corpus. Finally, we can predict a series of movements $z = [z_1, z_2, \ldots, z_T]$, where the target trading day is z_T and the rest are auxiliaries.

4.2. The proposed model: TEANet

To obtain more valuable information from text, we employ a Transformer encoder to encode the text and obtain the representation to serve as the input to the traditional attention mechanism. Through the attention model, we capture key information that affects the stock market. At the same time, the stock prices are preprocessed, and the results are then combined to form the input to the next submodels, i.e., an LSTM model and a temporal attention model. Finally, we obtain a probability as the prediction result. Figs. 3 and 4 show the architecture of the proposed model and the detailed steps of the entire prediction process.

To handle cases in which multiple stocks are discussed in the same message, the location of the mentioned stock symbol *s* is added to the text message. Formally, for a particular trading day in the message corpus, we represent the word sequence of each message k, $k \in [1, K]$, and the input word embedding matrix is $E = [e_1; e_2; \ldots; e_L]$, where L represents the length of the current message corpus.

4.2.1. Feature extractor

Here, we will introduce the composition of the *feature extractor* in detail, including the processing of text and stock prices. As seen in Fig. 3, each layer of the Transformer encoder is composed of two submodels: one is a multihead attention mechanism composed of h self-attention heads, and the other is a feedforward NN. The stock positions in the text dataset have been marked during preprocessing, so we no longer make use of the position coding vector in the Transformer model. The stock price processing procedure is relatively simple; we mainly adopt the normalization strategy.

Multihead attention: The kernel component of the multihead attention mechanism is scaled dot-product attention. For the embedding of the kth message on a trading day, we first create a query, key and value vector and generate the Query (Q), Key (K) and Value (V) matrix by multiplying the word embedding by the three training matrices created during the training process. Second, to obtain a more stable gradient during the training process, we calculate the queries and each key vector for dot multiplication and divide by $\sqrt{d_k}$. Then, we adopt the softmax function to normalize the output value to obtain a score, which directly determines the importance of the current word in its context. Finally, the input vector for the feedforward NN is obtained by multiplying each value vector by the corresponding score and accumulating the results. The matrix of outputs and the framework are shown in formula (5) and Fig. 4:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{5}$$

Generally, it is difficult for a single attention function to capture sufficient information to improve the results of stock movement prediction. As shown in Fig. 5, the mechanism implements a linear transformation of the word embeddings. For a single trading day, the queries are equal to the keys, consistent with the nature of a self-attention mechanism. From these queries, keys and values, we generate the output value matrix by executing the attention function in parallel. Because the h heads run in parallel, their results are concatenated in series and projected to produce the final value of the next step, which will then be input to the feedforward NN as a single matrix vector. Consequently, the advantage of the multihead attention mechanism is that it extends the ability of the attention model to focus on different positions and provides multiple "representation subspaces" in the attention layer. The multihead attention function as follows:

$$MultiHead(Q, K, V) = Concat(head_1, head_2, \ldots, head_h)W^O \tag{6}$$

where $W^O \in R^{hd_v \times d_{model}}$ represents the weight matrix. In this work, we employ $h = 5$ parallel attention layers, or heads. For each of these, we use $d_k = d_v = \frac{d_{model}}{h} = 10$. Because the dimensionality of each head is decreased, the total computational cost is similar to that of a single head with full dimensionality.

Addition and normalization: A certain detail of the encoder architecture should be mentioned: around each submodel (multihead attention mechanism or feedforward NN) in each encoder, there is a residual connection, followed by a "layer normalization" step (denoted by Norm in Fig. 3). The purpose is to stabilize the distribution and impose constraints on the space for expressing the ambiguity of each word so as to reduce the variance of the data in various dimensions.

Feedforward NN: Each submodel in the encoder contains a fully connected feedforward neural network (FNN) layer that acts equally on each position. It is made up of two linear transformations, with a ReLU activation in between.

$$FFN = max(0, xW^1 + b^1)W^2 + b^2 \tag{7}$$

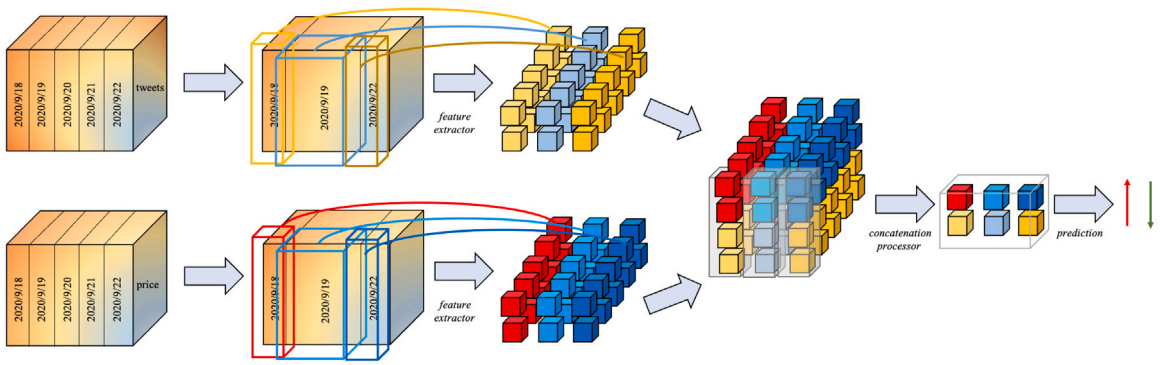


Fig. 4. The detailed implementation processes.

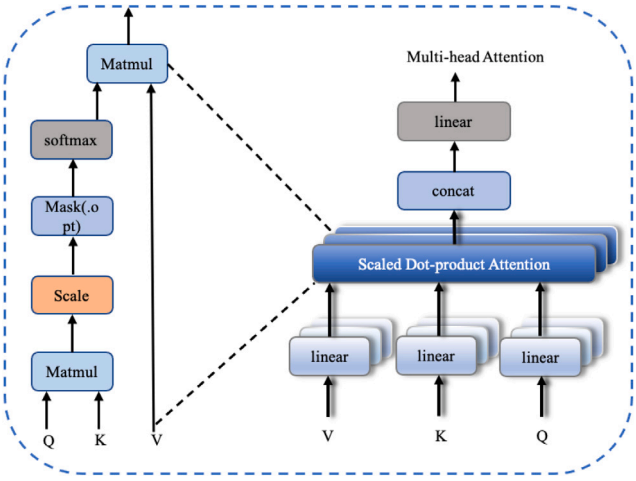


Fig. 5. The architecture of multihead attention.

While the linear transformations are the same for all different positions, different parameters are used between different layers. The dimensionality of both the input and the output is $d_{model} = 50$.

All message embeddings for the tdth trading day are combined to construct a matrix $M \in R^{d_{model} \times K}$. Thus, the process of deep feature extraction is completed; however, the attention is still balanced. We now need an attention mechanism to identify important information in the message. Specifically, we make use of the softmax function to nonlinearly project the matrix M to u to generate the normalized attention weights. These attention weights correspond to the information matrix used to obtain the corpus embedding.

$$u = softmax({w_u}^T tanh(W_m^M)) \tag{8}$$

$$c = Mu^T \tag{9}$$

where $w_u \in R^{d_{model} \times d_{model}}$ and $W_m \in R^{d_{model} \times 1}$ are parameters.

Relying on text alone is not necessarily sufficient to predict stock movements; therefore, we also consider historical stock prices. However, stock movements are determined by continuous changes in price rather than the simple absolute values of the closing and opening prices. Thus, instead of directly feeding the raw price vector for trading day td into the network, we employ a normalization strategy to obtain an adjusted closing price. The price adjustment formula is shown below:

$$p_{td} = [p_{td}^c, p_{td}^h, p_{td}^l] \tag{10}$$

$$p_a = \frac{p_{td}}{p_{td-1}^c} - 1 \tag{11}$$

where p_{td}^c, p_{td}^h and p_{td}^l denote the closing price, highest price and lowest price vectors, respectively.

4.2.2. Concatenation processor

The problem of how to integrate text with corresponding price data to obtain satisfactory experimental results is worthy of in-depth discussion. In this paper, we adopt the method of concatenation, as shown in formula (12). The adjusted stock prices are fused with tweet features to form the input to the LSTM and temporal attention submodels.

$$x = [c, p_a] \tag{12}$$

LSTM: In accordance with the nature of time series data, we use an RNN with LSTM units to recursively extract features. The LSTM architecture is improved by the RNN structure, which solves the problems of vanishing gradient and learning long-term dependence from the original data. In particular, the RNN structure and the associated transformations are used to analyze time series data by means of sequential processing in various fields. An LSTM unit includes gates and cell states, where the cell states are controlled by the three gates: a forget gate, an input gate, and an output gate. Fig. 6 depicts the structure of an LSTM unit, where x_t represents the input data; h_t and c_t represent the output value and cell state, respectively, at time point t; and f_t, i_t and o_t correspond to the above three gates. The first step is to determine the information to be discarded from the cell state, which is handled through the sigmoid unit of f_t. The next step is to add new information to the cell state and update the old information. Finally, we identify the characteristics of the cell state that need to be output. Formulas (13)–(18) describe the details of these LSTM operations.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{13}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{14}$$

$$\widetilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{15}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{16}$$

$$O_t = \sigma(W_O \cdot [h_{t-1}, x_t] + b_O) \tag{17}$$

$$h_t = O_t * tanh(C_t) \tag{18}$$

The use of LSTM units in RNNs has led to good results in many financial applications. Therefore, the fused features of tweets and price data are obtained based on the above calculation process. However, the emergence of attention mechanisms has represented another step forward for many researchers. Thus, we analyze the fused information again to perform the prediction task with the help of temporal attention.

Temporal attention: As described in Section 4.1, while predicting the movements on the target trading day, we can also predict the movements on other trading days during the lag period, which play a supplementary role in determining the conditions of the final target

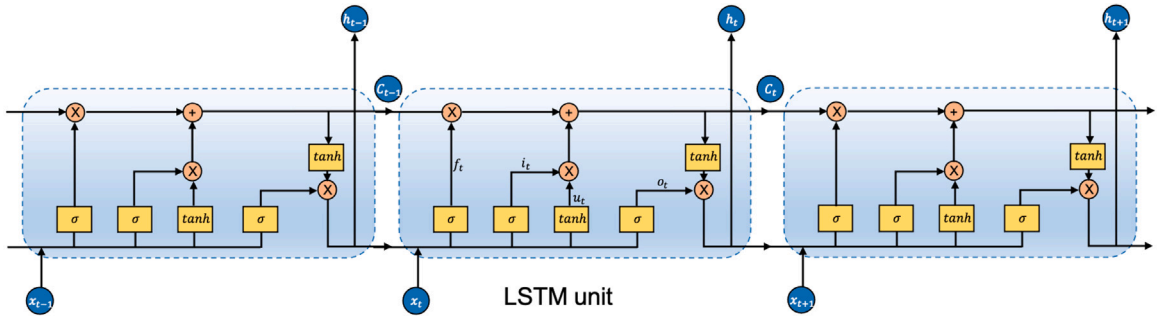


Fig. 6. The architecture of LSTM.

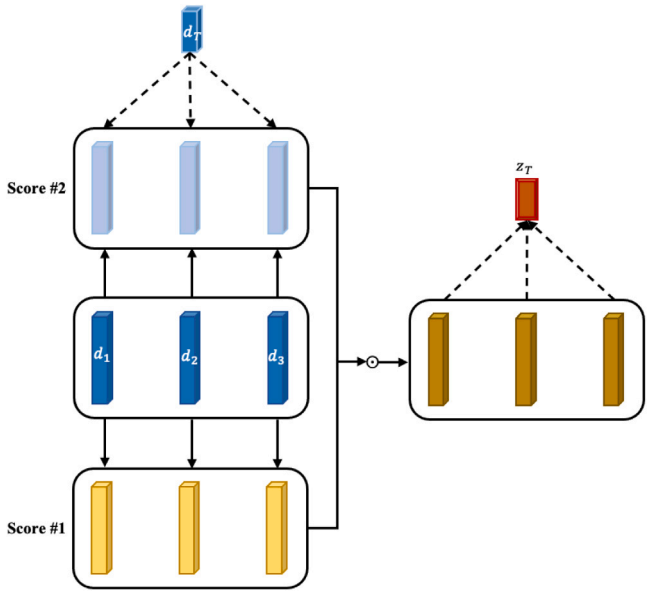


Fig. 7. The architecture of temporal attention.

trading day, providing *objective-level temporal auxiliary* information. We therefore introduce a temporal attention mechanism to realize prediction for the target and auxiliary trading days, thus yielding a sequence of predictions $z = [z_1, z_2, \ldots, z_T]$.

Temporal attention is essential for various stock market analysis tasks, and consequently, improved temporal attention mechanisms have been developed by many researchers based on different tasks. During the training and prediction experiments presented in this paper, the contributions of the obtained hidden features are unequal. Accordingly, we improve the original temporal attention model by dividing it into two subprocessors. As shown in Fig. 7, the temporal attention model calculates the contribution weights of the data by means of an information score and a dependency score, corresponding to Score #1 and Score #2, respectively. We integrate multiple deterministic composite features to form the input to the temporal attention model.

$$d_t = tanh(W_d[x, h_t] + b_d) \tag{19}$$

where W_d denotes the weight matrix and b_d is the bias. Then, d_t is nonlinearly projected to the corresponding information score and dependency score. The results of the temporal attention model are generated as expressed in formulas (20)–(22). The information score is calculated to evaluate each historical trading day according to its own information quality, while the dependency score represents the relationship between the target trading day and each auxiliary trading day.

$$v_i' = w_i^T tanh(W_{d,i} D) \tag{20}$$

$$v_m' = d_T^T tanh(W_{d,m} D) \tag{21}$$

$$v = softmax(v_i' \odot v_m') \tag{22}$$

where $W_{d,i}, W_{d,m} \in R^{d_d \times d_d}$ and $w_i \in R^{d_d \times 1}$ are the weight parameters. In these formulas, D represents the set of all auxiliary trading days, i.e., all days except the target day. Considering that we mainly wish to predict the target trading day, d_T is reused as the final information representation. Then, we employ the softmax function to obtain the final normalized attention weights $v \in R^{1\times(T-1)}$. Finally, the stock movements on all trading days are predicted as follows:

$$z_t = softmax(W_y d_t + b_y), t < T \tag{23}$$

$$z_T = softmax(W_T[Zv^T, d^T] + b^T) \tag{24}$$

where W_y and W_T are weight matrices, b_y and b_T are biases, and T represents the set of prediction results for all auxiliary trading days.

5. Experiments

In this section, we first describe the datasets used and the preprocessing steps followed. Second, the experimental setup is described in detail. Then, we present comprehensive experiments and analyses conducted to evaluate the performance of our proposed TEANet framework. Systematic trading simulations are performed to verify the effectiveness of our framework in an actual market scenario. Finally, we conduct an error analysis of the proposed method to find its deficiencies.

5.1. Dataset and preprocessing

Table 1 shows the data periods used in our work; dataset 1,[1] dataset 2,[2] dataset 3[3] and dataset 4[4] are available on GitHub. Stock traders often post personal opinions on the development trend of the current stock market on social platforms such as Twitter. The tweets of dataset 1 and dataset 2 are collected on Twitter, reflecting the current investors' perception of stock market. Based on the degree of discussion, the 88 stocks with the highest capital scale rankings were selected for the time period from January 1, 2014, to January 1, 2016, in dataset, where tweets were derived from valid critical text data on stocks published by stock traders on Twitter. In addition, the dataset is currently recognized financial data used for stock price prediction, and it has been widely used in many research results (Xu et al., 2020; Xu & Cohen, 2018). To further verify the applicability of the method proposed in this paper,

[1] https://github.com/yumoxu/stocknet-dataset
[2] https://github.com/wuhuizhe/CHRNN
[3] https://www.kaggle.com/
[4] https://github.com/ShreyamsJain/Stock-Price-Prediction-Model

Table 1
Basic statistics of the datasets.

Dataset	Training period	Validation period	Test period
Dataset 1	Jan. 1, 2014–Aug. 1, 2015	Aug. 1, 2015–Oct. 1, 2015	Oct. 1, 2015–Jan. 1, 2016
Dataset 2	Jan. 1, 2017–Nov. 1, 2017	Nov. 1, 2017–Dec. 1, 2017	Dec. 1, 2017–Jan. 1, 2018
Dataset 3	Jan. 1, 2018–Aug. 1, 2018	Aug. 1, 2018–Oct. 1, 2018	Oct. 1, 2018–Jan. 1, 2019
Dataset 4	Jan. 1, 2008–Jan. 1, 2015	Jan. 1, 2015–Jan. 1, 2016	Jan. 1, 2016–Jan. 1, 2017

we chose the historical prices and tweets of 47 popular stocks to form dataset 2 for the time period from January 1, 2017, to January 1, 2018, where the tweets in dataset 2 have been applied to stock prediction by Wu et al. (2018). To verify the practicability of TEANet in predicting stock movements, we also consider dataset 3, which is composed of news headlines and stock prices of 10 stocks for the time period from January 1, 2018, to January 1, 2019. Similarly, the dataset 4 is also composed of news headline and historical stock prices, where are selected from the 8-year Dow Jones Inc Top 25 between 2008 and 2016.

Next, we take dataset 1 as an example to introduce the processing of text data and stock price data; the processing methods of the other three datasets are the same. Research shows that among a large amount of stock data, some of these data will exhibit only minimal ratios of change. To address this problem, Hu et al. (2018) proposed that upper and lower limits on stock price changes should be specified for stock trend prediction tasks. Since our goal is to perform binary classification of stock movements, we set two specific thresholds: −0.5% and 0.55%. For samples with movement ratios of $\leq -0.5\%$ and $>0.55\%$, 0 and 1 were used to represent fall and rise, respectively, and we simply removed 38.72% of the selected targets with movement percentages between the two thresholds. Two thresholds were selected to balance the two classes, and finally 26,614 prediction targets were obtained for the entire dataset 1, of which the two classes account for 49.78% and 50.22%. Preprocessed dataset 1 was then divided by time to obtain a training set, validation set and test set, which is a common method of dividing time series data to avoid pollution of the dataset. Dataset 1 consists of two parts: a tweet dataset and a historical price dataset. The tweets were preprocessed by querying regexes composed of NASDAQ ticker symbols using the Natural Language Toolkit (NLTK) package. Historical stock prices were extracted from Yahoo Finance to establish the corresponding dataset.[5]

5.2. Training setup

For the experiments reported below, we use 32 mixed samples in one batch. The word embedding size is 50, and the GloVe embedding algorithm is used, while the maximum number of historical calendar days is set to 5. The maximum numbers of messages and words in a single message are set to 30 and 40, respectively. The conventional Adam optimizer is used to train the model, with an initial learning rate of 0.001. Furthermore, the decay rate is set to 0.96, and the decay step is set to 100. The whole method is implemented using TensorFlow (version 1.14.0).

5.3. Measures of prediction performance

We use two metrics to measure prediction performance, namely, *Accuracy* and the Matthews correlation coefficient (*MCC*), of which the ranges are [0,100] and [−1,1], respectively. *Accuracy* is widely used in various fields, and this paper is used to measure the accuracy of two-class. *MCC* is essentially the correlation coefficient between the observed and predicted binary classes. As shown in formula (25), *tp*, *fp*, *tn* and *fn* represent the numbers of samples classified as true positives, false positives, true negatives, and false negatives, respectively. *MCC* is calculated as follows:

$$MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}} \tag{25}$$

5.4. Verifying the effectiveness of TEANet

The experiments are divided into two sets: the purpose of one set of experiments is to verify the feasibility of the proposed model by comparing the performance of TEANet with that of baseline models, while the purpose of the other set of experiments is to analyze all the components of TEANet in detail by constructing variants of TEANet, (i.e. an ablation study).

5.4.1. Comparison with baseline models

We selected several classical and representative baseline methods for comparison with TEANet. These models have many similarities, and they all use basic NN architectures to solve the stock market prediction problem.

- ARIMA: An advanced technical analysis method that uses only price signals, called the autoregressive integrated moving average method (Goodman, 1965).
- TSLDA: A classical model for predicting stock price movements using sentiments on social media, which can capture a theme and the sentiment regarding that theme simultaneously (Nguyen & Shirai, 2015).
- HAN: A hybrid attention network for predicting stock trends based on sequences of recent related news items by imitating the learning process of human beings. This model includes news-level attention and temporal attention mechanisms, which are used to focus on key information in the news (Hu et al., 2018).
- CH-RNN: A novel cross-modal attention-based hybrid recurrent neural network, which consists of two submodules: one makes use of an improved RNN structure to obtain trend representations for different stocks, and the other uses an RNN to model social texts (Wu et al., 2018).
- StockNet: An NN model that uses a VAE to encode input stock data to capture their randomness and analyze the importance of different time steps using temporal attention. The dataset used is the same as that used for our model (Xu & Cohen, 2018).
- Adv-LSTM: A novel NN prediction model with adversarial training, which is highly expressive for sequential data. This model includes a feature mapping layer, an LSTM layer, a temporal attention mechanism, and a prediction layer; it can extract deep text features and capture the dependencies between texts. Again, the dataset used is the same as that used for our model (Feng et al., 2019).
- CapTE: A state-of-the-art DL network model that uses a Transformer encoder to extract deep semantic features of social media texts and then captures the structural relationships of these texts through a capsule network (Liu et al., 2019).

As seen in Table 2, TEANet achieves the best results in most cases. Compared to the baselines, TEANet exhibits improvements of more than 20% and 104% on the four datasets in terms of *Accuracy* and *MCC*, respectively. CapTE scores the highest among all baselines. The histogram in Fig. 8 more intuitively shows the changes in *Accuracy* and *MCC*, indicating that the proposed method achieves great improvement. Since predicting stock movements is a challenging task and even a minor improvement could lead to enormous profits, 56% accuracy is generally considered a satisfactory result for binary stock movement prediction (Nguyen & Shirai, 2015). Although they are slightly better

[5] https://finance.yahoo.com/industries

Table 2
Performance of baselines and TEANet in accuracy and MCC.

Models	Dataset 1		Dataset 2		Dataset 3		Dataset 4	
	Accuracy (%)	MCC	Accuracy (%)	MCC	Accuracy (%)	MCC	Accuracy (%)	MCC
ARIMA	51.39	−0.0205	51.19	−0.0255	51.37	−0.0205	51.41	−0.0177
TLSDA	54.07	0.0653	53.88	0.0614	54.10	0.0661	54.22	0.0691
HAN	57.14	0.0723	57.02	0.0693	56.99	0.0711	57.19	0.0724
CH-RNN	59.15	0.0945	59.15	0.0945	59.00	0.0893	59.21	0.0913
StockNet	58.23	0.0807	57.93	0.0787	58.24	0.0804	58.30	0.0831
Adv-LSTM	57.20	0.1483	57.22	0.1395	57.31	0.1501	57.33	0.1611
CapTE	64.22	0.3481	64.18	0.3405	**64.29**	0.3502	**64.22**	0.3511
TEANet	**65.16**	**0.3637**	**64.37**	**0.3481**	64.18	**0.3504**	64.20	**0.3514**

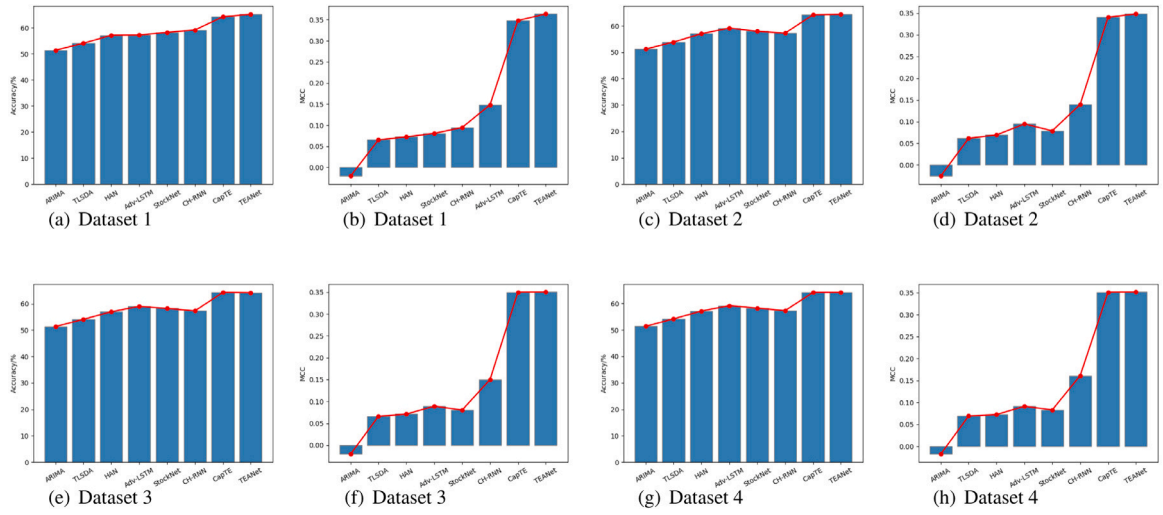


Fig. 8. Comparison of Accuracy and MCC.

than random guessing, the classical analysis techniques (ARIMA and TLSDA) do not produce satisfactory results. In general, the experimental results of the other 5 baseline models are satisfactory.

Next, we compare the baseline models and TEANet in terms of their frameworks on the four datasets. First, the performance of all models on dataset 1 was originally used to verify the performance of TEANet, so the results are more representative. In the text feature extraction stage, bidirectional gated recurrent unit (BGRU) and LSTM structures are adopted in StockNet and Adv-LSTM, respectively, while TEANet uses a Transformer encoder; the subsequent architecture is basically the same. The results indicate that Transformer based approaches have a significant performance advantage over RNN-based approaches. There are two fundamental reasons for this phenomenon: the Transformer model is not troubled by long-term dependence, and the multihead attention mechanism provides information about the relationships between different words. CapTE also uses a Transformer encoder to extract deep text features, and overall, the experimental results of this method are indeed significantly better than those of the other baseline methods. However, the results of TEANet are still superior to those of this other Transformer-based method, mainly because the input data for CapTE consist only of social media texts rather than including actual stock market price changes. Moreover, as seen by comparing the overall prediction results of the baseline models, there is still much room for improvement in the performance of the HAN model and the Adv-LSTM model. The fundamental reasons are the lack of practical applicability of the submodels and the overly one-sided nature of the reflected sentiments. Thus, choosing an appropriate submodel has a crucial influence on the experimental results. Finally, the RNN-based CH-RNN model achieves satisfactory results. This model adopts a variety of different types of attention mechanisms and analyzes the target text in a bidirectional manner, thus demonstrating the advantages of attention mechanisms in obtaining key information.

Table 2 also shows the performance of TEANet and other baseline models on dataset 2, dataset 3 and dataset 4. TEANet outperforms the other methods in most cases. The data attributes contained in the four datasets are similar, and we can infer that the experimental results obtained by TEANet are indeed comparable. The above results are based on the combination framework adopted in this paper, which is similar to the analysis of dataset 1 and is not described here.

Compared to all the other models, TEANet yields significantly improved results in terms of both evaluation indicators, suggesting that the overall architecture of TEANet can achieve satisfactory prediction results by solving the problem of temporal dependence and fusing text and stock price data.

5.4.2. Ablation study

To perform a detailed analysis of all of the primary components of TEANet, in addition to the fully equipped TEANet, we also construct the following four variants. Among them, "TEANet (W/O text)" means that the input data do not include the text corpus. Moreover, we propose variants with alternative parameter values in which the length of the lag period is modified to 7 and 10 days, named TEANet (lag_size_7) and TEANet (lag_size_10), respectively.

- TEANet (W/O text): TEANet with the original parameters using only historical prices as input data.
- TEANet (W/O price): TEANet with the original parameters using only text as input data.
- TEANet (lag_size_7): TEANet with alternative parameter settings corresponding to a lag period of 7 calendar days.
- TEANet (lag_size_10): TEANet with alternative parameter settings corresponding to a lag period of 10 calendar days.

Table 3 shows the performance of the different TEANet variants. We can see that TEANet (W/O text) yields the worst results, showing

Table 3
Performance of TEANet variations in accuracy and MCC.

Models	Dataset 1		Dataset 2		Dataset 3		Dataset 4	
	Accuracy (%)	MCC	Accuracy (%)	MCC	Accuracy (%)	MCC	Accuracy (%)	MCC
TEANet (W/O text)	59.99	0.1697	58.92	0.1597	59.93	0.1688	58.61	0.1615
TEANet (W/O price)	64.77	0.3513	63.70	0.3223	**64.77**	**0.3511**	63.54	0.2438
TEANet (lag_size_7)	64.98	0.3569	63.08	0.3265	64.18	0.2269	**64.33**	0.2579
TEANet (lag_size_10)	62.57	0.2102	61.52	0.2012	61.54	0.1833	60.57	0.1753
TEANet	**65.16**	**0.3637**	**64.37**	**0.3481**	64.18	0.3504	64.20	**0.3514**

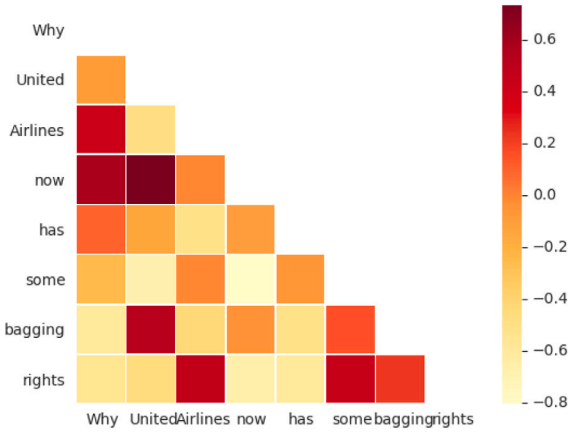


Fig. 9. Visualization of attention scores. The darker color in the picture indicates that the word has more weight and has a greater effect in the context; the lighter color indicates that the weight is smaller and the effect is relatively smaller. The words containing emotional color and substantial information play a more important role in the final prediction result, while the role of auxiliary words is relatively small.

Table 4
Profit comparison between TEANet and CapTE.

Stock	TEANet	Return(%)	CapTE	Return(%)
AAPL	$2113	21.13	$2046	20.46
ABBV	$1680	16.80	$1524	15.24
BAC	$1927	19.27	$1766	17.66
CELG	$3130	31.30	$2581	25.81
CVX	$1988	19.88	$2017	20.17
DIS	$1547	15.47	$1396	13.96
GOOG	$1752	17.52	$1567	15.67
INTC	$3527	35.27	$2399	23.99
ORCL	$1858	18.58	$1849	18.49
PFE	$2147	21.47	$2125	21.25
WMT	$3057	30.57	$2841	28.41
XOM	$2043	20.43	$2103	21.03
Average return (%)		22.31		20.18

that stock movements are affected by a variety of factors other than historical price. Even compared to the baseline models described above, TEANet (W/O text) is unsatisfactory, demonstrating the necessity of extracting text features as input. By contrast, TEANet (W/O price) produces exceptionally competitive results, similar to those of TEANet. Consequently, it can be concluded that social media texts contain a large amount of market information, and several studies have shown that this kind of information is useful when predicting stock prices and even for other financial tasks (Bustos & Pomares-Quimbaya, 2020). The performance results of TEANet (W/O text) and TEANet (W/O price) confirm the positive effects of text and historical prices, respectively, for stock movement prediction.

The length of the lag period is typically set to between 3 and 10 days. The trading day is taken as the basic unit in TEANet, so the value in terms of calendar days has a more direct influence on the experimental results. However, there are limits to using three calendar days to guarantee the presence of more than one trading day in the lag period, as in the case of movement prediction for Monday. As illustrated in Table 3, experiments with lag periods of 7 and 10 calendar days do not produce better results than those achieved with a lag period of 5. We believe that there are two major reasons: many random factors affect the stock market, and changes on the target trading day are more likely to be affected by the prices on adjacent trading days; moreover, investor comments tend to reflect current market conditions and are not suitable for predicting long-term trading behavior.

5.5. Effects of attention mechanisms

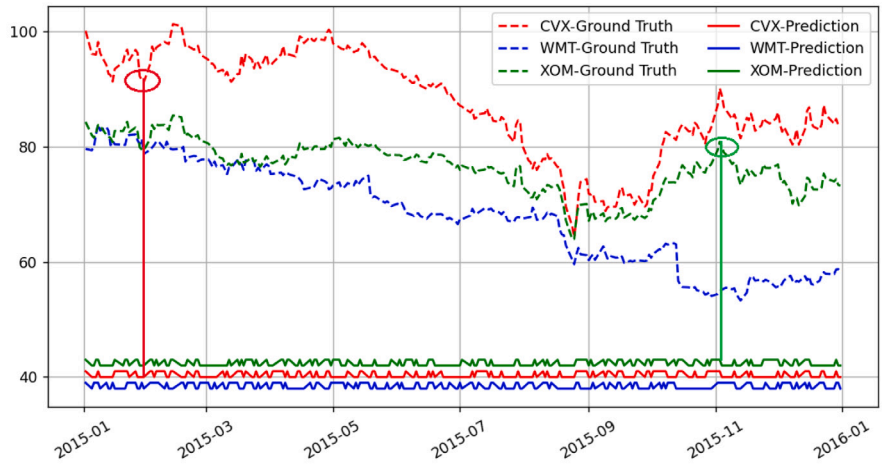
To qualitatively analyze the attention mechanisms in our model, we choose to use tweets and stock prices in the datasets to represent the capabilities of the multihead attention and temporal attention mechanisms, respectively.

5.5.1. Visualization of multihead attention

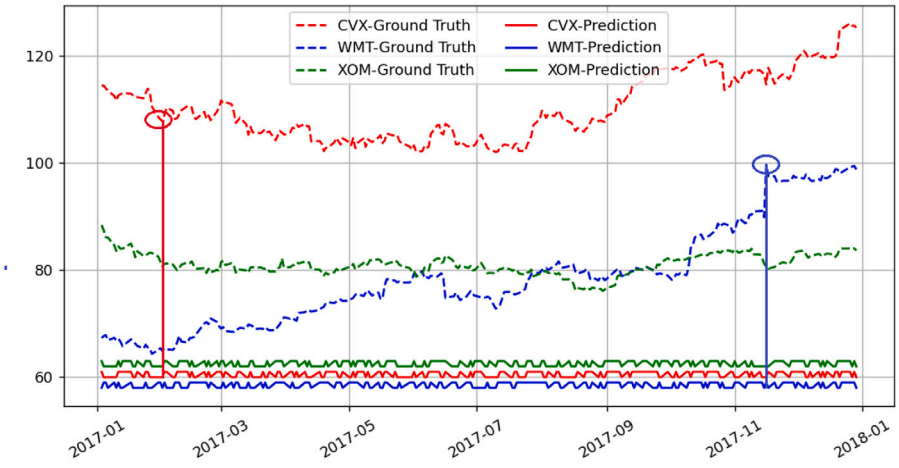
Since the tweet attention value is constructed by considering the embedding vector as part of the input, we can obtain the attention value of each tweet in the dataset, which can be used to verify whether our framework can effectively select important information and filter out noncritical information. By analyzing the highest and lowest attention scores, we can determine which parts of the tweets are considered important. Fig. 9 shows the attention scores for a random tweet in the test set, where a darker color indicates greater importance of the corresponding word. From this figure, we can see that when analyzing a commentary sentence “Why United Airlines now has some bagging rights”, the attention mechanism can basically determine the more critical words in comparison, especially “now”, “United”, “Airlines”, “bagging”, the information they hide plays a greater role in the prediction of the stock market. Therefore, words with low attention scores clearly provide no indication of future trends, such as “has”, “sum”, they play a supporting role, whereas a higher value indicates information that is more relevant to the current stock and can be explicitly used to predict the movement of the market, such as the words mentioned above. These cases clearly demonstrate that the multihead attention mechanism does distinguish between critical information and irrelevant information and illustrate why it can improve performance.

5.5.2. Demonstration of temporal attention

To further illustrate the influence of the information and dependency scores used in the temporal attention mechanism on the actual stock market, we refer to the relationship between stock price changes and tweet information. In Fig. 10, we plotted the actual upward and downward movements of the three stocks over time in 2015 and 2017 with virtual lines, in which the obvious changes were marked with circles, indicating that the stock fluctuated greatly on a trading day. With the help of the complete TEANet framework, the prediction results of three stocks in the corresponding time intervals are shown in the dotted line drawn. We can observe that most of the prediction results are consistent with the actual stock movement trends. For example, when a stock company has a scandal or the government issues a policy, it will have an impact on stock prices falling or rising. The circles in the figure mark several trading days with large changes in stock prices.



(a)



(b)

Fig. 10. Comparison between the real stock market price and the predicted stock movements, where the circles mark the obvious changes in the real stock, and the vertical line corresponds to the predicted movement results. The prediction results are taken from the textual information and the time-dependent changes of the stock market, which confirms the two considerations of time-series attention: information score and dependency score.

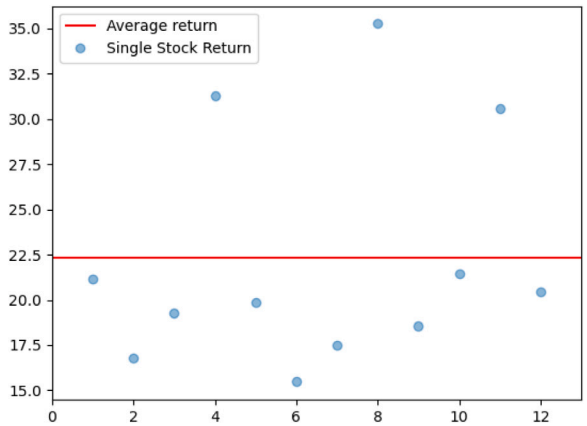


Fig. 11. Visualization of individual stock return and average return.

The real tweets corresponding to these obvious changes make mostly correct predictions about the expected stock situation in advance, and the key information can be effectively calculated by means of the temporal attention mechanism. Therefore, it can be concluded that the movements of stocks are affected not only by historical stock prices but also by tweets, which directly proves that our proposed improved temporal attention mechanism is comprehensive. In reality, investors can carefully synthesize relevant investor comments and prices to better assess their impact on an underlying stock. Therefore, an ideal framework to mimic this analysis process should integrate and interpret such information over a continuous period of time rather than analyzing these factors separately.

5.6. Market trading simulation

We adopt a market simulation strategy proposed by Ding et al. (2020) to evaluate the stock prediction performance of TEANet through a standard profit method. If the model shows that a stock will rise the next day, the trader will buy that stock at its opening price of $10,000.

Fig. 12. Comparison of different prediction models and their changes in profit.

Relative to the opening price, we set a threshold for price fluctuations of 2% to indicate whether the stock should be traded before the end of the day. That is, if the current price of the stock is above 2% of the opening price, the trader will sell immediately. Otherwise, the trader will need to sell the stock at the closing price at the end of the day. Conversely, if the model predicts that the stock price will fall, the trader will buy the stock when it is possible to buy it at 1% less than the short price; otherwise, the trader will buy the stock at the closing price.

In Table 4, we show the profits for 12 randomly selected stocks achieved with TEANet and CapTE over 44 trading days (or 60 calendar days), where the maximum profit for DIS exceeds 30%. These results undeniably prove that TEANet can obtain higher profits and has a superior practical application value to CapTE. We note that if there is no review from the previous day, the model cannot accurately predict the stock movement for the day because no information on the subjective attitudes of investors is available. In this case, not only the *Accuracy* of the evaluation results and the *MCC* will be degraded but also the actual profit. At the same time, we considered the return of a single stock and the market average return of the selected stocks (Lam, 2004). According to Table 4 and Fig. 11, the return of 12 single stocks we selected and the calculated market average return are shown. The monthly return of the historical real stock market was 8.35%, the selected market average return obtained by TEANet was 11.15%, and the excess return calculated by the two was positive.

Fig. 12 shows the daily profits for 12 stocks. It can be intuitively seen that TEANet is superior to CapTE and can avoid market risks as much as possible. Therefore, the profit is consistent with the overall performance analysis, and our proposed framework can achieve the best profit results among all compared methods. Overall, the method proposed in this paper has great potential value for the actual task of predicting the movements of stocks to help investors increase their profit.

5.7. Error analysis

We compare the predictions of TEANet and CapTE and analyze the cases in which movements are incorrectly predicted by TEANet but well predicted by CapTE. We summarize two situations: First, a tweet is written by a trader with a negative sentiment that leads to a malicious comment on the corresponding stock. For example, consider the tweet

"The phone is so terrible, so is AAPL"; this comment is actually talking about phones, but the trader's sentiment also extends to the stock. Second, a tweet may talk about the state of the market a long time ago. For example, consider the tweet "The 2006 EPIDEMIC in the US has seriously affected the trend of BBL stock". The epidemic under discussion occurred in 2006 and is not relevant to the current stock market. However, in such cases, it is difficult for our model to obtain correct predictions without introducing the relevant information.

6. Conclusion and future work

In this paper, we have proposed a novel DL model called TEANet, which can use historical stock prices from 5 calendar days in combination with text representations to predict stock movements by means of a Transformer encoder and attention mechanisms. To solve the problems of temporal dependence in financial data and insufficient effectiveness in fusing information from text and stock prices, an architecture consisting of a *feature extractor* and a *concatenation processor* is adopted. The model has been trained on small samples of text and stock prices, and the extracted features can accurately describe the state of the stock market. For the *feature extractor*, the structure consists of a Transformer encoder, attention mechanisms and a normalization strategy to effectively extract features and learn key information through relevant reviews of tweets and preprocessing of stock prices. The *concatenation processor* then processes the fused features to capture the temporal dependency. The overall framework realizes effective processing and analysis of text and stock prices to improve the accuracy of stock movement prediction.

In this study, five main sets of experiments were performed to verify the validity of the proposed model. First, we compared the prediction performance with that of various baseline models to investigate model feasibility. Second, four different variants of the TEANet model were constructed and tested to analyze the impact of different components. Third, the effects of the attention mechanisms were intuitively investigated to confirm the contributions of the multihead attention and temporal attention mechanisms to the overall performance. Then, trading simulations were performed as a profitability test. Finally, we performed an error analysis to analyze TEANet from various perspectives. The results indicate that the proposed model successfully achieves the stock movement prediction task with satisfactory experimental performance.

Through comparisons with state-of-the-art methods, we find that the proposed method is more suitable for practical application and can help traders avoid financial risks and make more favorable decisions. However, we still face challenges in further improving the practical application value of TEANet. Because the stock market is very complex, the use of knowledge graphs and other strategies to study the relationships between different stocks is one of the main research problems that we will address in the future.

CRediT authorship contribution statement

Qiuyue Zhang: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing. **Chao Qin:** Methodology, Software, Investigation, Writing – review & editing. **Yunfeng Zhang:** Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing. **Fangxun Bao:** Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing. **Caiming Zhang:** Resources, Visualization, Formal analysis. **Peide Liu:** Project administration, Writing – review & editing, Validation.

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.

Acknowledgments

The authors are thankful for the anonymous referee's constructive comments. This work was supported by the National Natural Science Foundation of China (Grant Nos. 61972227), the Natural Science Foundation of Shandong Province (Grant Nos. ZR2019MF051 and ZR201808160102) and in part by the Fostering Project of Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions.

References

Aguilar-Rivera, R., Valenzuela-Rendón, M., & Rodríguez-Ortiz, J. J. (2015). Genetic algorithms and Darwinian approaches in financial applications: A survey. *Expert Systems with Applications*, *42*(21), 7684–7697. http://dx.doi.org/10.1016/j.eswa.2015.06.001.

Akaike, H. (1969). Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics*, *21*(1), 243–247.

Araabi, A., & Monz, C. (2020). Optimizing transformer for low-resource neural machine translation. In *Proceedings of the 28th international conference on computational linguistics* (pp. 3429–3435). International Committee on Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.coling-main.304, URL: https://aclanthology.org/2020.coling-main.304.

Araci, D. (2019). FinBERT: Financial sentiment analysis with pre-trained language models. arXiv, URL: http://arxiv.org/abs/1908.10063.

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques - part II: Soft computing methods. *Expert Systems with Applications*, *36*(3 PART 2), 5932–5941. http://dx.doi.org/10.1016/j.eswa.2008.07.006.

Bahrammirzaee, A. (2010). A comparative survey of artificial intelligence applications in finance: Artificial neural networks, expert system and hybrid intelligent systems. *Neural Computing and Applications*, *19*(8), 1165–1195. http://dx.doi.org/10.1007/s00521-010-0362-z.

Barak, S., Arjmand, A., & Ortobelli, S. (2017). Fusion of multiple diverse predictors in stock market. *Information Fusion*, *36*, 90–102. http://dx.doi.org/10.1016/j.inffus.2016.11.006.

Boyacioglu, M. A., & Avci, D. (2010). An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: The case of the Istanbul stock exchange. *Expert Systems with Applications*, *37*(12), 7908–7912. http://dx.doi.org/10.1016/j.eswa.2010.04.045.

Bustos, O., & Pomares-Quimbaya, A. (2020). Stock market movement forecast: A systematic review. *Expert Systems with Applications*, *156*, http://dx.doi.org/10.1016/j.eswa.2020.113464.

Cavalcante, R. C., Brasileiro, R. C., Souza, V. L., Nobrega, J. P., & Oliveira, A. L. (2016). Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications*, *55*, 194–211. http://dx.doi.org/10.1016/j.eswa.2016.02.006.

Chalup, S. K., & Mitschele, A. (2008). Kernel methods in finance. In *International handbooks information system*, *Handbook on information technology in finance* (pp. 655–687). Springer, http://dx.doi.org/10.1007/978-3-540-49487-4_27.

Chen, Y., & Hao, Y. (2017). A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. *Expert Systems with Applications*, *80*, 340–355. http://dx.doi.org/10.1016/j.eswa.2017.02.044.

Chen, W., Yeo, C. K., Lau, C. T., & Lee, B. S. (2018). Leveraging social media news to predict stock index movement using RNN-boost. *Data and Knowledge Engineering*, *118*, 14–24. http://dx.doi.org/10.1016/j.datak.2018.08.003.

Ding, Q., Wu, S., Sun, H., Guo, J., & Guo, J. (2020). Hierarchical multi-scale Gaussian transformer for stock movement prediction. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence* (pp. 4640–4646). http://dx.doi.org/10.24963/ijcai.2020/640.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International conference on learning representations*. URL: https://openreview.net/forum?id=YicbFdNTTy.

Fama, E. F. (1965). The behaviour of stock market prices. *Journal of Business*, *38*(1), 34–105.

Fama, E. F., Fisher, L., Jensen, M. C., & Roll, R. (1969). The adjustment of stock prices to new information. *International Economic Review*, *10*(1), 1–21.

Fan, R., Chu, W., Chang, P., Xiao, J., & Alwan, A. (2021). An improved single step non-autoregressive transformer for automatic speech recognition. In *Interspeech* (pp. 2–6). URL: http://www.seas.ucla.edu/spapl/paper/ruchao_IS_2021.pdf.

Farias Nazário, R. T., e Silva, J. L., Sobreiro, V. A., & Kimura, H. (2017). A literature review of technical analysis on stock markets. *Quarterly Review of Economics and Finance*, *66*, 115–126. http://dx.doi.org/10.1016/j.qref.2017.01.014.

Feng, F., He, X., Wang, X., Luo, C., Liu, Y., & Chua, T.-S. (2019). Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems*, *37*(2), 1–30.

Gabeur, V., Sun, C., Alahari, K., & Schmid, C. (2020). Multi-modal transformer for video retrieval. In *Computer vision–ECCV 2020: 16th European conference* (pp. 214–229). Springer.

Goodman, L. S. (1965). Reviewed work: Smoothing, forecasting, and prediction of discrete time series by robert goodel brown. *Journal of Marketing Research*, *2*(3), 314–315, URL: http://www.jstor.org/stable/3150192.

Gorenc Novak, M., & Velušček, D. (2016). Prediction of stock price movement based on daily high prices. *Quantitative Finance*, *16*(5), 793–826. http://dx.doi.org/10.1080/14697688.2015.1070960.

Hu, Z., Liu, W., Bian, J., Liu, X., & Liu, T. Y. (2018). Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *WSDM 2018 - proceedings of the 11th ACM international conference on web search and data mining, vol. 2018-Febua* (pp. 261–269). http://dx.doi.org/10.1145/3159652.3159690, arXiv:1712.02136.

Huynh, H. D., Dang, L. M., & Duong, D. (2017). A new model for stock price movements prediction using deep neural network. In *ACM international conference proceeding series, vol. 2017-Decem* (pp. 57–62). http://dx.doi.org/10.1145/3155133.3155202.

Kraus, M., & Feuerriegel, S. (2017). Decision support from financial disclosures with deep neural networks and transfer learning. *Decision Support Systems*, *104*, 38–48. http://dx.doi.org/10.1016/j.dss.2017.10.001, arXiv:1710.03954.

Lam, M. (2004). Neural network techniques for financial performance prediction: integrating fundamental and technical analysis. *Decision Support Systems*, *37*(4), 567–581. http://dx.doi.org/10.1016/S0167-9236(03)00088-5.

Lee, S. W., & Kim, H. Y. (2020). Stock market forecasting with super-high dimensional time-series data using convlstm, trend sampling, and specialized data augmentation. *Expert Systems with Applications*, *161*, Article 113704. http://dx.doi.org/10.1016/j.eswa.2020.113704.

Lee, C. Y., & Soo, V. W. (2018). Predict stock price with financial news based on recurrent convolutional neural networks. In *Proceedings - 2017 conference on technologies and applications of artificial intelligence* (pp. 160–165). http://dx.doi.org/10.1109/TAAI.2017.27.

Li, X., Wu, P., & Wang, W. (2020). Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong. *Information Processing and Management*, *57*(5), Article 102212. http://dx.doi.org/10.1016/j.ipm.2020.102212.

Liu, J., Liu, X., Lin, H., Xu, B., Ren, Y., Diao, Y., & Yang, L. (2019). Transformer-based capsule network for stock movements prediction. In *Proceedings of the first workshop on financial technology and natural language processing* (pp. 66–73). URL: https://aclanthology.org/W19-5511.

Lund, R. (2007). Time series analysis and its applications: With r examples. *Journal of the American Statistical Association*, *102*(479), 1079. http://dx.doi.org/10.1198/jasa.2007.s209.

Nguyen, T. H., & Shirai, K. (2015). Topic modeling based sentiment analysis on social media for stock market prediction. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing, vol. 1* (pp. 1354–1364). http://dx.doi.org/10.3115/v1/p15-1131.

Park, C. H., & Irwin, S. H. (2007). What do we know about the profitability of technical analysis? *Journal of Economic Surveys*, *21*(4), 786–826. http://dx.doi.org/10.1111/j.1467-6419.2007.00519.x.

Patil, P., Wu, C. S. M., Potika, K., & Orang, M. (2020). Stock market prediction using ensemble of graph theory, machine learning and deep learning models. In *ICSIM '20: the 3rd international conference on software engineering and information management.*

Pei, W., Baltrušaitis, T., Tax, D. M., & Morency, L. P. (2017). Temporal attention-gated model for robust sequence classification. In *Proceedings - 30th IEEE conference on computer vision and pattern recognition, vol. 2017-Janua* (pp. 820–829). http://dx.doi.org/10.1109/CVPR.2017.94, arXiv:1612.00385.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. Preprint, URL: https://www.bibsonomy.org/bibtex/273ced32c0d4588eb95b6986dc2c8147c/jonaskaiser.

Sohangir, S., & Wang, D. (2018). Finding expert authors in financial forum using deep learning methods. In *Proceedings - 2nd IEEE international conference on robotic computing, 2018-Janua* (pp. 399–402). http://dx.doi.org/10.1109/IRC.2018.00082.

Vargas, M. R., Dos Anjos, C. E. M., Bichara, G. L. G., & Evsukoff, A. G. (2018). Deep leaming for stock market prediction using technical indicators and financial news articles. In *International joint conference on neural networks* (pp. 1–8).

Wang, H., Wang, T., & Li, Y. (2020). Incorporating expert-based investment opinion signals in stock prediction: A deep learning framework. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 01* (pp. 971–978). http://dx.doi.org/10.1609/aaai.v34i01.5445.

Wu, Z., Nguyen, T.-S., & Ong, D. C. (2020). Structured self-attention weights encode semantics in sentiment analysis. In *Proceedings of the third blackboxNLP workshop on analyzing and interpreting neural networks for NLP* (pp. 255–264). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.blackboxnlp-1.24, URL: https://aclanthology.org/2020.blackboxnlp-1.24.

Wu, H., Zhang, W., Shen, W., & Wang, J. (2018). Hybrid deep sequential modeling for social text-driven stock prediction. In *2018 association for computing machinery* (pp. 1627–1630). http://dx.doi.org/10.1145/3269206.3269290.

Xie, B., Passonneau, R. J., Wu, L., & Creamer, G. G. (2013). Semantic frames to predict stock price movement. In *Proceedings of the 51st annual meeting of the association for computational linguistics, vol. 1* (pp. 873–883).

Xu, K., Ba, J. L., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R. S., & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *32nd international conference on machine learning, vol. 3* (pp. 2048–2057).

Xu, H., Chai, L., Luo, Z., & Li, S. (2020). Stock movement predictive network via incorporative attention mechanisms based on tweet and historical prices. *Neurocomputing*, *418*, 326–339. http://dx.doi.org/10.1016/j.neucom.2020.07.108.

Xu, Y., & Cohen, S. B. (2018). Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th annual meeting of the association for computational linguistics, vol. 1* (pp. 1970–1979). http://dx.doi.org/10.18653/v1/p18-1183.

Yang, L., Xu, Y., Lok, T., Ng, J., & Dong, R. (2019). Leveraging BERT to improve the FEARS index for stock forecasting. In *Proceedings of the first workshop on financial technology and natural language processing* (pp. 54–60).

Zhang, D., & Zhou, L. (2004). Discovering golden nuggets: Data mining in financial application. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, *34*(4), 513–522. http://dx.doi.org/10.1109/TSMCC.2004.829279.

Qiuyue Zhang received the M.E. degree from the Department of Computer Technology, Shandong Normal University, Jinan, China, in 2020. She is currently pursuing the Ph.D. degree with the School of Management Science and Engineering, Shandong University of Finance and Economics, Jinan, China. Her current research interests include financial data analysis and natural language processing.

Chao Qin received the B.E. degree from the Department of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, China, in 2017. He is currently pursuing the M.E. degree with the School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, China. His current research interests include data analysis and data mining.

Yunfeng Zhang received the B.E. degree in computational mathematics and application software from the Shandong University of Technology, Jinan, China, in 2000, and the M.S. degree in applied mathematics and the Ph.D. degree in computational geometry from Shandong University, Jinan, in 2003 and 2007, respectively. He is currently a Professor with the Shandong Provincial Key Laboratory of Digital Media Technology, Shandong University of Finance and Economics. His current research interests include computer-aided geometric design, digital image processing, computational geometry, and function approximation.

Fangxun Bao received the M.Sc. degree from the Department of Mathematics, Qufu Normal University, Qufu, China, in 1994, and the Ph.D. degree from the Department of Mathematics, Northwest University, Xián, China, in 1997. He is currently a Full Professor with the Department of Mathematics, Shandong University, Jinan, China. His research interests include computer-aided geometric design and computation, computational geometry, and functional approximation.

Caiming Zhang received the B.S. and M.E. degrees in computer science from Shandong University in 1982 and 1984, respectively, and the Ph.D. degree in computer science from the Tokyo Institute of Technology, Japan, in 1994. He is currently a Professor, a Doctoral Supervisor, and the Dean of the School of Computer Science and Technology, Shandong University. He is also the Dean and a Professor with the School of Computer Science and Technology, Shandong Economic University. From 1997 to 2000, he held a visiting position at the University of Kentucky, USA. His research interests include CAGD, CG, information visualization, and medical image processing.

Peide Liu received the B.S. and M.S. degrees in signal and information processing from Southeast University, Nanjing, China, in 1988 and 1991, respectively, and the Ph.D. degree in information management from Beijing Jiaotong University, Beijing, China, in 2010. He is currently a Professor with the School of Management Science and Engineering, Shandong University of Finance and Economics, Shandong, China. He is an Associate Editor of the Journal of Intelligent and Fuzzy Systems, the editorial board of the journal Technological and Economic Development of Economy, and the members of editorial board of the other 12 journals. He has authored or coauthored more than 200 publications. His research interests include aggregation operators, fuzzy logic, fuzzy decision making, and their applications.