Neurocomputing 470 (2022) 66–75

Contents lists available at ScienceDirect

Neurocomputing

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

News-driven stock prediction via noisy equity state representation

Heyan Huang [a,b], Xiao Liu [a,b], Yue Zhang [c,d,*], Chong Feng [a,b]

[a] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
[b] Southeast Academy of Information Technology, Beijing Institute of Technology, Fujian 351100, China
[c] School of Engineering, Westlake University, Zhejiang 310024, China
[d] Institute of Advanced Technology, Westlake Institute for Advanced Study, Zhejiang 310024, China

ARTICLE INFO

Article history:
Received 1 April 2021
Revised 6 September 2021
Accepted 29 October 2021
Available online 2 November 2021
Communicated by Zidong Wang

Keywords:
Stock prediction
News events modeling
Recurrent neural network
Equity state
Random noise

ABSTRACT

News-driven stock prediction investigates the correlation between news events and stock price movements. Previous work has considered effective ways for representing news events and their sequences, but rarely exploited the representation of underlying equity states. We address this issue by making use of a recurrent neural network to represent an equity state transition sequence, integrating news representation using contextualized representations as inputs to the state transition mechanism. Thanks to the separation of news and equity representations, our model can accommodate additional input factors. We design a novel random noise factor for modeling influencing factors beyond news events, and a future event factor to address the delay of news information (e.g., insider trading) and reduce the learning difficulties. Results show that the proposed model outperforms strong baselines in the literature.

1. Introduction

Stock movement prediction [1,2] is a central task in computational quantitative finance. With recent advances in deep learning and natural language processing (NLP), event-driven stock prediction has received increasing research attention [3,2]. The goal is to predict the movement of stock prices according to financial news. Previous work adopts a relatively simple model on the stock movement process, casting price change as a response to a set of news. The prediction model can therefore be viewed as variation of a classifier that takes news as input and yields a movement direction output. Investigations have focused on news representation, where bag-of-words [4], named entities [5], event structures [1] or neural representation features [2,6–9] are considered.

Intuitively, news events carry information on important changes of company management, market, revenue and other factors, which can affect the fundamental states of equities, and thereby can consequently impact the stock price, as shown in Fig. 1. Properly representing news events is key to modeling such impact on the market. However, the stock market movement can also be influenced by accumulated effects of fundamental changes over time, the overall market sentiment, and other volatile factors, which can be considered as noise to analytical models. These factors have been relatively less modeled by existing work on event-driven stock prediction. For example, although there has been work modeling long-term event impacts by representing event sequences [2], little work has considered representing fundamental equity states directly.

To address these issues, we consider representing the equity state directly using a recurrent neural network over time and propose the stock movement prediction network using Noisy Equity State representation (NES). At each time step, the equity state reflects the current stock price trend, and can be used directly for predicting the next movement. The advantage of separating news representation from equity state representation is that factors beyond news can be modeled as additional input in the recurrent state transition process. Although such factors can be calculated using external tools such as sentiment classification over tweet data, we simply treat them as a random noise factor. The reason is twofold. First, in practice, noise is inevitable in stock prediction and no single mathematical model can perfectly fit the stock price movement distribution. Second, for fair comparison with existing work on news-driven stock prediction, no additional input should be used on top of standard benchmark input settings.

The input to each recurrent equity state transition consists of a news factor and a noise factor. The news factor consists of three components, namely past news within thirty days, present news

* Corresponding author at: School of Engineering, Westlake University, Zhejiang 310024, China.

E-mail addresses: hhy63@bit.edu.cn (H. Huang), xiaoliu@bit.edu.cn (X. Liu), yue.zhang@wias.org.cn (Y. Zhang), fengchong@bit.edu.cn (C. Feng).

https://doi.org/10.1016/j.neucom.2021.10.092
0925-2312/

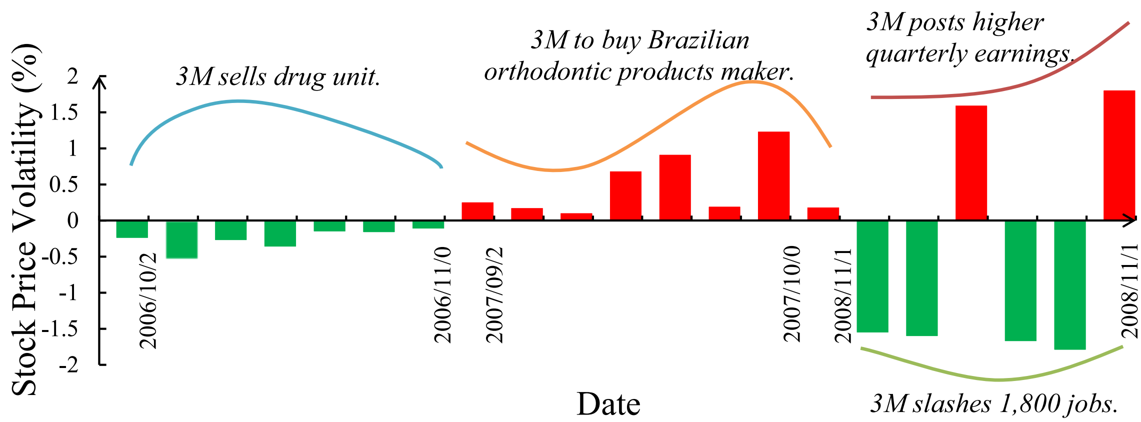


Fig. 1. The volatility and the news impact on `3 M Company` stock price. The green bars mean the stock price is lower than the previous trading day, while the red bars mean higher. Each curve indicates the time-varying impact of individual news. Over the first and the second period, there was only one event. And in the third period, there were two events affecting the stock price movements simultaneously. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

between two trading days, and future news within seven trading days. We use real historical news for the future news component for training, and predicted future news according to the current equity state during testing. The motivation for modeling future news is to not only address the negative effect of the delayed news release and insider trading on the prediction accuracy, but also reduce the learning difficulties when purely fitting to the historical news and stock trends. Following previous work on encoding stock news headlines [1,2,6,10,11,8], we represent each news event using the contextualized representation of the news title, and aggregate news representations by using the current equity state as a query to conduct attention. The noise factor is integrated into the model by using the current equity state to derive a normal noise distribution specific to the trading day, and then sampling a noise vector.

Compared with existing work, our model has three main potential advantages. First, the relative importance of news can be easily visualized using attention. Second, insider trading effect is explicitly handled. Third, noise effect is integrated into the model. All three benefit result directly from the direct representation of equity state. Experiments over the benchmark of Ding et al. [2] show that our method outperforms strong baselines, giving the best reported results in the literature. To our knowledge, we are the first to explicitly model both events and fundamental equity states for news-driven stock movement prediction, and the first to consider noise factors using a neural random sample module.

2. Related work

Our work is related with two strands of existing work on stock prediction, including: i) methods on price movements correlation and ii) explainable approaches for stock prediction.

2.1. Modeling price movements correlation

Most existing work on event-driven stock prediction treats the representation of news events independently using bag-of-words [4], named entities [5], semantic frames [3], event structures [1], event embeddings [2] or knowledge bases [6,12]. In contrast, work on time-series based stock prediction [13,14,7,15] aims to capture continuous movements of prices themselves. There has also been work modeling the correlations between samples by sparse matrix factorization [16], hidden Markov model [15], Transformers [17] and Bi-RNNs [10,11]. Some methods [7,18] also learn from both textual news and historical price data and apply information fusion from multiple data sources [19]. Some work models the correlations among different stocks by pre-defined correlation graph [20], tensor factorization [21], reinforcement learning [22,23], dynamic financial distress prediction models [24] and graph convolutional networks [25,8]. Our work is different in that we use only news events as inputs, and our recurrent states are additionally designed to accommodate noise.

2.2. Explainable stock prediction

Rationalization is an important problem for news-driven stock price movement prediction, which is to find the most important news event along with the model's prediction. Factorization, such as sparse matrix factorization [16] and tensor factorization [21], is a popular method where results can be traced back upon the input features. Our attention-based module achieves a similar goal yet has linear time complexity on feature size. There are also methods focusing on other techniques. Gite et al. [26] use sentiments from news headlines as key features. Rezaei et al. [27] apply ensemble decomposition-based algorithms for different frequency spectra in analyzing financial time series. Chang et al. [28] use attention for characterizing the influence of individual news within a trading day for predicting the cumulative abnormal return in a three-day window. Xu et al. [7] train a deep generative model jointly exploiting stock news and historical prices. Sawhney et al. [9] jointly learn models from historical prices, stock news, and inter stock relations through hierarchical attention representations. Yang et al. [18] apply dual-layer attention to predict the stock movement by using news published in the previous six days. Each day's news embeddings and seven days' embeddings are summed by the layer. Our work is different from existing attention-based methods [28,18,9] in that our news events attention is based on dynamic queries, which is more strongly related to noisy recurrent states. In contrast, their attention is not query-based and tends to output the same result for each day even if the previous day's decision is changed.

3. Method

Following previous work [1,2], the task is defined as a binary classification task. Formally, given a trading day t, the input is a history news set about a targeted stock or index and the output is a label $y \in \{+1, -1\}$ indicating whether the adjusted closing price[1] p_t will be greater than p_{t-1} $(y = +1)$ or not $(y = -1)$. Please noted that a calendar day is not always a trading day in stock market

[1] https://yhoo.it/3i5UkRh

and we mean a *day* is a *calendar day* by default.

The structure of our model is shown in Fig. 2. The equity state, which reflects the fundamentals of a stock, is modeled using LSTM [29], which serves as a basis for making prediction. News events are separately represented as inputs to the state transition process. For each trading day, we consider the impact fusion of news events in the day as well as the past news events via neural attention mechanism [30]. Considering the impacts of insider trading, we also involve actual future news in the training procedure and estimate the future impact in the evaluating procedure. To model the high stochasticity of stock markets, we sample an additive noise using a neural module.

Considering the general principle of sample independence, building temporal connections between individual trading days in the training set can harm training [7] and we find it easy to overfit. Additionally, it is worth noticing that LSTM usually takes several steps to reach a stable hidden state. To address this issue, we extended the time span of a single trading day to T previous continuous trading days $(t-T+1, t-T+2, \ldots, t-1, t)$, which we call a trading sequence. Trading sequences are used as the basic training elements in this paper.

3.1. Recurrent equity state transition

The binary classification task is based on the equity state of each trading day. Formally, we denote the equity state using $\mathbf{z}$. In each trading day, the recurrent state transition can be written as

$$\mathbf{z}_t = f(\mathbf{z}'_t) \tag{1}$$

$$\mathbf{z}'_t = \overrightarrow{\mathrm{LSTM}}(\boldsymbol{v}_t, \mathbf{z}_{t-1}) \tag{2}$$

where $\mathbf{v}_t$ represents the news event on the trading day t and f is the function for leveraging the noise effect. Non-linear compositional effects of multiple events can also be captured in a time window because of the LSTM representation. To predict the current day stock price movement, we use the sequential state $\mathbf{z}_t$ to make binary classification

$$\hat{p}_t = \mathrm{softmax}(\mathbf{W}^y \mathbf{z}_t) \tag{3}$$

$$\hat{y}_t = \underset{i\in\{+1,-1\}}{\overset{i}{\mathrm{argmax}}}\, \hat{p}_t(\hat{y}_t = i|t) \tag{4}$$

where $\hat{p}_t$ is the estimated probabilities for up $(+1)$ and down (-1) movements, $\hat{y}_t$ is the predicted label, $\mathbf{W}^y$ is the classification parameter and t is the input trading day.

3.2. Modeling news events

We separately represent long-term and short-term impact of news events for each day t in a trading sequence. For short-term impact, we use news articles published after the previous trading day $t-1$ and before the trading day t as the set of present news. Similarly, for long-term impact, we use news articles within thirty calendar days as the set of past news.

Each news article is simply regarded as a separate news event. We extract the headline and use the pretrained contextualized embedding model to transform it to a V-dim representation vector. Note that this paper only obtains event representations from headlines because news headlines carry more representative features and less distracting information than the news content. In this paper, we employ ELMo [31] to learn the contextualized representations. ELMo is a pretrained language representation model with multiple encoding layers that has been shown to improve many NLP tasks. Comparing with bigger pretrained language models like BERT [32], we choose ELMo for two reasons: i) ELMo is light-weighted and consumes less resources when finetuning; ii) ELMo

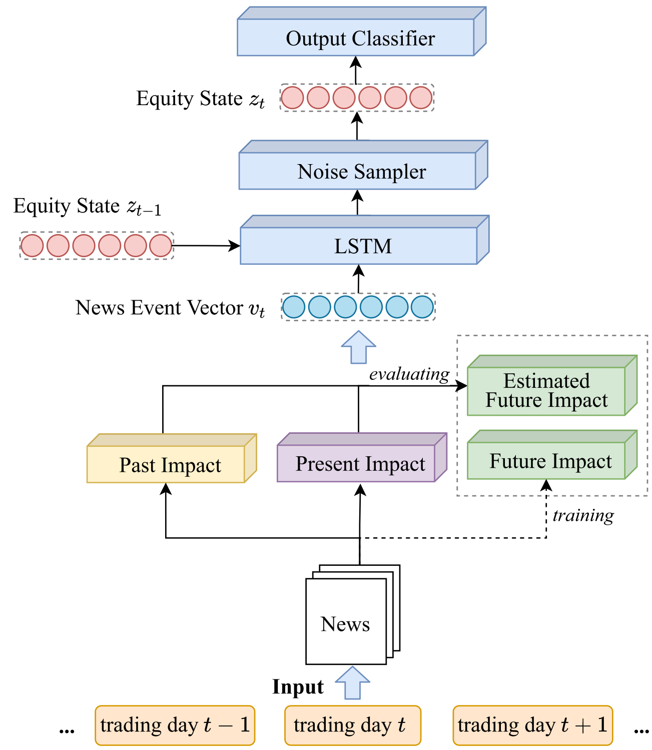


Fig. 2. The model framework of NES with a trading day t in the trading sequence. The solid lines with arrows are used both in the training and the evaluating procedures, while the dotted one is only used in the training procedure.

is character-based, which is more suitable for the genre of financial news where there will be abbreviations and short phrases. We concatenate the bidirectional output hidden states of the first and the last word in the last ELMo encoding layer to get the contextualized representation of a news article.

By stacking the contextualized representations of news articles, we obtain two embedding matrices $\mathbf{C}'_t$ and $\mathbf{B}'_t$ for the present and past news events in the trading day t, respectively, as

$$\mathbf{e} = \left[\overrightarrow{\mathrm{ELMo}}(\mathbf{h})^{-1},\ \overleftarrow{\mathrm{ELMo}}(\mathbf{h})^{0}\right] \tag{5}$$

$$\mathbf{C}'_t = \left[\mathbf{e}_c^1 \oplus \mathbf{e}_c^2 \oplus \ldots \oplus \mathbf{e}_c^{l_c}\right] \tag{6}$$

$$\mathbf{B}'_t = \left[\mathbf{e}_b^1 \oplus \mathbf{e}_b^2 \oplus \ldots \oplus \mathbf{e}_b^{l_b}\right] \tag{7}$$

where $[,]$ is the vector concatenation operation, $[\oplus]$ is the vector stacking operation, l_c is the size of present news set and l_b is the size of past news set.

To make the model more numerically stable and avoid overfitting, we apply the over-parameterized method of [33] to the news event embedding matrices

$$\mathbf{C}_t = \sigma(\mathbf{U}\mathbf{C}'_t) \odot \tanh(\mathbf{V}\mathbf{C}'_t) \tag{8}$$

$$\mathbf{B}_t = \sigma(\mathbf{U}\mathbf{B}'_t) \odot \tanh(\mathbf{V}\mathbf{B}'_t) \tag{9}$$

where $\odot$ is element-wise multiplication, $\sigma(\cdot)$ is the sigmoid function, and $\mathbf{U}$ and $\mathbf{V}$ are model parameters.

Due to the unequal importance fact of news events with regard to the stock price movement in day t, we use scaled dot-product attention [30] to capture the influence of news $\mathbf{C}_t$ and $\mathbf{B}_t$ to the current day stock movement. Formally, we first transform the last trading day's equity state $\mathbf{z}_{t-1}$ to a query vector $\mathbf{q}_t$, and then calculate two attention score vectors γ_t and β_t for the present and past news events, respectively as

$$\mathbf{q}_t = \tanh(\mathbf{W}^q \mathbf{z}_{t-1}) \tag{10}$$

$$\gamma_t = \text{softmax}\left(\frac{\mathbf{C}_t^i \mathbf{q}_t}{\sqrt{V}}\right) \tag{11}$$

$$\boldsymbol{\beta}_t = \text{softmax}\left(\frac{\mathbf{B}_t^i \mathbf{q}_t}{\sqrt{V}}\right) \tag{12}$$

We sum the news event embedding matrices to obtain two final news representation vectors $\mathbf{c}_t$ and $\mathbf{b}_t$ on the trading day t according to the weights γ_t and $\boldsymbol{\beta}_t$, respectively, as

$$\mathbf{c}_t = \tanh\left(\sum_{i=1}^{l_c} \gamma_t^i \mathbf{C}_t^i\right) \tag{13}$$

$$\mathbf{b}_t = \tanh\left(\sum_{i=1}^{l_b} \beta_t^i \mathbf{B}_t^i\right) \tag{14}$$

3.3. Capturing future impact

We find that news events can exert an influence on the stock price movement *before* being released, which can be attributed to news delay or insider trading [34,35] factors. Additionally, future news may introduce potential useful patterns which benefit the task of stock prediction. To this end, we propose a novel future event module to represent backward news influence. In this paper, we define future news events as those that are published within seven calendar days starting from the trading day t.

Similarly to the past and present news events, we first use ELMo [31] to encode and obtain the contextualized representations for future news event headlines. Then the contextualized representations are stacked to form an embedding matrix $\mathbf{A}_t'$. We also adapt the over-parameterized method and sum the stacked embedding vectors by scaled dot-product attention. Formally, the future news events impact vector $\mathbf{a}_t$ on the trading day t is calculated as

$$\mathbf{A}_t = \sigma(\mathbf{U}\mathbf{A}_t') \odot \tanh(\mathbf{V}\mathbf{A}_t') \tag{15}$$

$$\boldsymbol{\alpha}_t = \text{softmax}\left(\frac{\mathbf{A}_t^i \mathbf{q}_t}{\sqrt{V}}\right) \tag{16}$$

$$\mathbf{a}_t = \tanh\left(\sum_{i=1}^{l_a} \alpha_t^i \mathbf{A}_t^i\right) \tag{17}$$

where l_a is the size of future news set.

The above steps can be used to reduce overfitting in the training procedure, where the future event module is trained over gold "future" data over historical events. While the problem is that, in the evaluating procedure, future news events are not accessible. To address this issue, we use a non-linear transformation to estimate a future news events impact vector $\hat{\mathbf{a}}_t$ with the past and present news events impact vectors $\mathbf{b}_t$ and $\mathbf{c}_t$ as

$$\hat{\mathbf{a}}_t = \tanh(\mathbf{W}^a[\mathbf{c}_t, \mathbf{b}_t]) \tag{18}$$

where $[,]$ is the vector concatenation operation and $\mathbf{W}^a$ is model parameter. We concatenate the aforementioned three types of news event vectors $\mathbf{a}_t, \mathbf{b}_t$ and $\mathbf{c}_t$ to obtain a final news event input $\mathbf{v}_t$ for the recurrent state transition on trading day t as

$$\mathbf{v}_t = \begin{cases} [\mathbf{c}_t, \mathbf{b}_t, \mathbf{a}_t], & \text{for training} \\ [\mathbf{c}_t, \mathbf{b}_t, \hat{\mathbf{a}}_t], & \text{for evaluating} \end{cases} \tag{19}$$

3.4. Leveraging noise

All factors beyond input news articles, such as sentiments, expectations and noise are explicitly modeled as noise using a random factor. Considering the high stochasticity and nondeterminacy of noise, we sample a random factor $\mathbf{n}_t$ for each trading day independently from a normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_t)$ parameterized by $\mathbf{z}_t'$ as

$$\boldsymbol{\sigma}_t = \sqrt{\exp(\tanh(\mathbf{W}^\sigma \mathbf{z}_t'))} \tag{20}$$

Following Eq. (1), the noise factor $\mathbf{n}_t$ will facilitate the transition hidden states $\mathbf{z}_t'$ to obtain the equity state $\mathbf{z}_t$. The original form of the computation process is shown in Fig. 3(a).

In addition, to facilitate back-propagation in training, we use reparameterization for normal distributions [36,37], drawing a sample random factor from a parameter-free distribution to obtain the noisy recurrent state $\mathbf{z}_t$ as

$$\mathbf{z}_t = \tanh(\mathbf{z}_t' + \boldsymbol{\sigma}_t \boldsymbol{\epsilon}_t) \tag{21}$$

$$\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \tag{22}$$

Therefore, the final computation process is shown in Fig. 3(b).

3.5. Training objective

For training, we consider two main terms for defining the loss function. The first term is a cross entropy loss for the stock movement probabilities $\hat{p}_t$ and gold labels y_t, and the second term is the mean squared error between the estimated future impact vector $\hat{\mathbf{a}}_t$ and the true future impact vector $\mathbf{a}_t$ stated in Section 3.3. Formally, the loss for a trading sequence containing T trading days with standard L_2 regularization is calculated as

$$L_{ce} = \sum_{t=1}^{T} -\log(1 - \hat{p}_t(y_t|t)) \tag{23}$$

$$L_{mse} = \frac{1}{V}\sum_{t=1}^{T}\sum_{i=1}^{V}(\hat{\mathbf{a}}_t^i - \mathbf{a}_t^i)^2 \tag{24}$$

$$L_{total} = L_{ce} + \theta L_{mse} + \lambda \|\Phi\|_2^2 \tag{25}$$

where θ is a hyper-parameter which indicates the relative importance of L_{mse} to L_{ce}, Φ is the set of trainable parameters in the model and λ is the regularization weight.

4. Experiments

4.1. Dataset

Following previous work [2,6,10,11], we use the public financial news dataset released by Ding et al. [1], which is crawled from Reuters and Bloomberg over the period from October 2006 to November 2013, and follow their method for splitting the dataset to make fair comparisons. Experiments are conducted on predicting the Standard & Poor's 500 stock (S&P 500) index and a set of selected individual stocks, obtaining prices from Yahoo Finance[2]. There are 358,122 documents and 1425 trading days in the training set, 96,299 documents and 169 trading days in the development set, and 99,030 documents and 191 trading days in the test set. Detailed statistics of the training, development and test sets are shown in Table 1. This dataset is also bigger than that used in other methods [7,9,8] with which we compare. We report the final results after tuning hyper-parameters in the development experiments.

4.2. Experimental settings

We choose the hyper-parameters according to grid search in development experiments. The hyper-parameters and corresponding ranges for searching are shown in Table 2. We use mini-batch

[2] https://finance.yahoo.com/

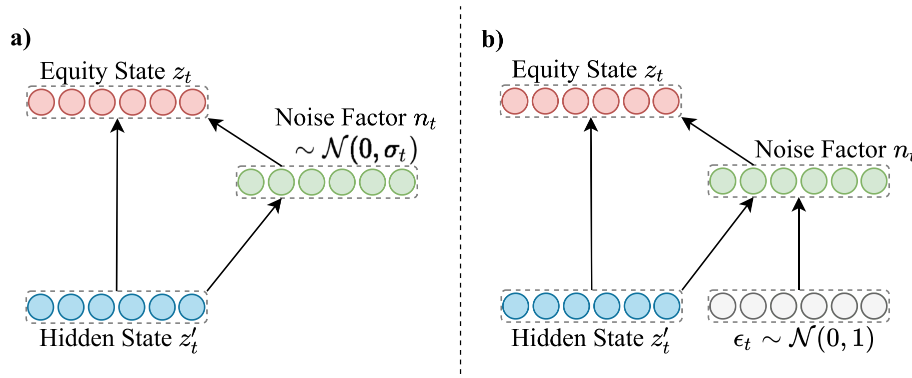


Fig. 3. Leveraging the noise factor in the equity state. (a) In the original form, the noise factor is sampled from a normal distribution. (b) In the reparameterized form, the noise factor is deterministic, and the randomness comes from other variables.

Table 1
Statistics of the datasets.

	Training	Development	Test
#Documents	358,122	96,299	99,030
#Samples	1425	169	191
Time span	10/20/2006–06/18/2012	06/19/2012-02/21/2013	02/22/2013-11/21/2013

and SGD with momentum to update the parameters. Following previous work [3,1,7,9], we adopt Precision (p), Recall (r), F1 score (F_1), Accuracy (acc) and Matthews Correlation Coefficient (MCC) [38] to evaluate model performances in S&P 500 index prediction and individual stock prediction. MCC is applied because it avoids bias due to data skew. Given the confusion matrix which contains true positive (TP), false positive (FP), true negative (TN) and false negative (FN) values, the above chosen metrics are calculated as

$$p = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{26}$$

$$r = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{27}$$

$$F_1 = \frac{2 \times p \times r}{p + r} \tag{28}$$

$$\mathrm{acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{29}$$

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{30}$$

4.3. Development experiments

We report a set of development experiments demonstrating the selection of important hyperparameters to the model.

4.3.1. Initializing noisy recurrent states

As the first set of development experiments, we compare different ways to initialize the noisy equity states of NES. For each trading day, we compare the results whether states transitions are modeled or not as well as the initialization methods of the equity states. The following four baselines are designed:

- NES_SING_R: randomly initializing the states for each single trading day.
- NES_SING_Z: initializing the states as zeros for each single trading day.

Table 2
Hyper-parameters setting.

Name	Value	Range
batch size	16	[1, 64]
learning rate lr	0.005	[1, 64]
SGD momentum μ	0.9	[0, 0.99]
dropout rate r	0.3	[0, 0.8]
MSE loss weight θ	0.4	[0, 2.0]
regularization weight λ	0.0005	[0, 0.01]
news embedding dimension V	256	[50, 1024]
recurrent state dimension D	100	[50, 1024]

- NES_SEQ_R: randomly initializing the first states for each trading sequence only.
- NES_SEQ_Z: initializing the first states as zeros for each trading sequence only.

The random initialization method returns a tensor filled with random numbers from the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{1})$. The results on the S&P 500 index are shown in Table 3. We can see that modeling recurrent sequences (NES_SEQ_R and NES_SEQ_Z) gives better result than treating each trading day separately (NES_SING_R and NES_SING_Z), which shows that modeling equity states can capture long-term and non-linear compositional effects of multiple events. In addition, NES_SING_R and NES_SING_Z both achieves worse performances than some baselines in Section 4.4, which also reminds us of that the LSTM-based recurrent state transition is essential in NES. From another perspective, comparing with the models NES_SING_R and NES_SING_Z also demonstrates the strengths of our basic representations of news events in isolation. In particular, we can also see that using only the basic news events representations is not sufficient for index prediction, while a combination with the equity state transition can give better results. This is consistent with previous reports on news-driven stock prediction [2]. By comparing the results of NES_SEQ_R and NES_SEQ_Z, we decide to use zero initialization in the remaining experiments.

4.3.2. Study on trading sequence length

We use the development set to find a suitable length T for a trading sequence, searched from $\{1, 3, 5, 7, 9, 11, 13, 15\}$. The S&P 500 index prediction results and time per training epoch on the development set are shown in Fig. 4. We can see that the F1 score, Accuracy and MCC are positively correlated with T. When $T \geqslant 7$, the growth of MCC becomes slower, while the running time per training epoch is nearly linear with respect to T. We therefore choose the hyper-parameter $T = 7$ for the remaining experiments.

Table 3
Results on initializing the noisy equity states.

	Precision	Recall	F1	Accuracy	MCC
NES_SING_R	68.34%	50.21%	57.89%	62.91%	0.3704
NES_SING_Z	68.81%	50.03%	57.94%	63.63%	0.3672
NES_SEQ_R	**69.42%**	57.92%	63.15%	67.94%	0.5141
NES_SEQ_Z	69.18%	**58.13%**	**63.18%**	**68.51%**	**0.5392**

4.4. Main results

We compare our approach with the following strong baselines on predicting the S&P 500 index. The first group contains methods for modeling price movements correlation only using news texts.

- CNN [2] uses event embeddings as input and convolutional neural network to model a sequence of events.
- CNN + KB [6] empowers event embeddings with knowledge bases, and adopts convolutional neural networks as the event sequence model.
- LSTM [10] uses a fully connected model and character-level embedding input with LSTM to encode news sequences.
- RNN [11] uses recurrent neural networks with skip-thought vectors to represent news text.
- LSTM-RGCN [8] uses LSTM to encode news texts and learns node representations for each stock or index in a trading day.

The second group is two existing explainable methods on stock movement prediction with multi-source input. Please noted that our work only uses news texts as inputs, but StockNet [7] and MAN-SF [9] also receive historical prices as input. To make a fair comparison, we also report the results of disabling the price encoder in these two models (denoted as StockNet_NP and MAN-SF_NP).

- StockNet [7] is a deep generative model. The market information consists of Bi-GRU output vectors for stock news and normalized price vectors containing the adjusted closing, highest and lowest price on a trading day.
- StockNet_NP is the model without concatenating the normalized price vectors in StockNet.
- MAN-SF [9] adopts GRUs and attention mechanisms to encode normalized price vectors and stock news. And it uses graph attention networks [39] to learn node representations for each stock or index before classification.
- MAN-SF_NP is the model without the price encoders in MAN-SF.

Table 4 shows the test set results. The table shows that NES achieves the best results among all the models making predictions based on news input only, giving at least 2.97%, 0.61% and 6.68% performance gain of F1/Accuracy/MCC, respectively. It demonstrates the advantage of separately representing equity state sequences and integrating noise features. When comparing with CNN [2] and the knowledge-enhanced method CNN + KB [6], we find that modeling the correlations between trading days in trading sequences can better capture the compositional effects of multiple news events. In addition, by comparing with LSTM [10], RNN [11] and LSTM-RGCN [8], we also find that modeling the noise by using a state-related random factor is very beneficial and effective to meet the goal to deal with the high market stochasticity.

In the second group, when only using stock news as the input, NES has 2.70%, 2.89% and 1.43% performance gain of F1/Accuracy/MCC, respectively, outperforming StockNet_NP [7] and MAN-SF_NP [9] in all the metrics, which shows the effectiveness of our model. However, if the historical prices are involved, the performances of both StockNet_NP [7] and MAN-SF_NP [9] improve a lot and exceed that of NES. It shows that the historical prices are essential information that incorporates well with stock news texts. We leave the investigation of integrating historical prices into NES as our future work.

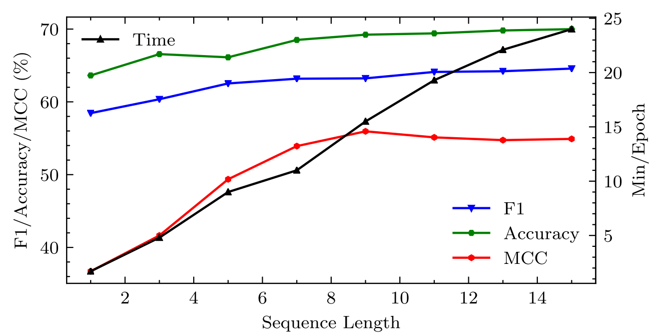


Fig. 4. Results of different T.

4.5. Analysis

We explore the effects of news events and random noise by comparing with some variations on the test set. The S&P 500 index prediction results of the compared variations are shown in Table 5.

4.5.1. News events impact

We first ablate the past news (w/o Past News), the present news (w/o Present News) and the future news (w/o Future News) in turn. Without using the past news events as input, the result becomes the lowest. This shows that history news articles contain the most significant amount of relevant news events. In addition, since we set the trading sequence length $T = 7$ and only use the news between two trading days as the present news, those ablated news published before the date range will not be involved in our model, while the ablated present or past news will be a part of the input on adjacent trading days.

We also observe that using future news events is more effective than using the present news events. On the one hand, it confirms the importance of involving the future news events in NES, which can deal with the news latency or insider trading factors to some extent. Furthermore, from the computational perspective, introducing such future information helps to reduce the learning difficulties and the information gap. In addition, such information avoids purely fitting to the noise that is difficult to generalize and may guide the model to learn beneficial patterns for this task. On the other hand, the results also benefit from the recurrent state transition nature of the model, as the future news impact on the $(t-1)$-th day can be carried forward to the t-th day in the equity state to compensate for the absence of the present news events.

In addition, to better illustrate the performance gap between the estimated and the golden future impact, we design another variation using golden future news events in both training and testing (w/ Golden Future News). As we can see from the results, using golden news events brings 1.57%, 1.90% and 2.92% gain of

Table 4
Test set results on predicting S&P 500 index. * denotes that the model learns from multi-source input other than only stock news.

	Precision	Recall	F1	Accuracy	MCC
CNN [2]	68.42%	54.45%	60.64%	64.21%	0.4035
CNN + KB [6]	68.63%	57.35%	62.49%	66.93%	0.5072
LSTM [10]	68.17%	57.82%	62.57%	63.34%	0.5096
RNN [11]	68.92%	58.30%	63.17%	64.55%	0.5132
LSTM-RGCN [8]	69.20%	58.93%	63.65%	63.87%	0.5081
StockNet_NP [7]	69.04%	59.34%	63.82%	65.45%	0.5398
*StockNet [7]	69.92%	62.31%	65.90%	68.06%	0.5534
MAN-SF_NP [9]	68.77%	59.10%	63.57%	63.35%	0.5265
*MAN-SF [9]	70.10%	62.25%	65.94%	69.11%	0.5613
NES	**69.74%**	**61.82%**	**65.54%**	**67.34%**	**0.5475**

Table 5
Results of the compared variations. w/o denotes without and w/ denotes with.

	F1	Δ_{F_1}%	Accuracy	Δ_{acc}%	MCC	Δ_{MCC}%
NES	65.54%	–	67.34%	–	0.5475	–
w/o Past News	62.83%	−4.13	62.17%	−7.68	0.4421	−19.25
w/o Present News	64.11%	−2.18	64.73%	−3.88	0.4823	−11.91
w/o Future News	63.61%	−2.94	64.58%	−4.10	0.4781	−12.68
w/ Golden Future News	66.57%	+1.57	68.62%	+1.90	0.5635	+2.92
w/o Noise-All	63.19%	−3.59	63.90%	−5.11	0.4608	−15.84
w/o Noise-Testing	62.74%	−4.27	64.04%	−4.90	0.4583	−16.29
w/ Delayed Noise	63.37%	−3.31	64.04%	−4.90	0.4641	−15.23

performances for F1, Accuracy and MCC, respectively. Comparing with NES without future news, using golden news events makes further efforts to reduce the learning difficulties. We think that the future news may introduce potential useful patterns that benefit stock movement prediction. But the performance gap indeed exists. The reason may be that our non-linear transformation estimator for future news impacts still suffers from the divergence between training and testing.

4.5.2. The noise factor

Since our model samples noises in both training and testing, we compare with variations without sampling noises in all the phases (w/o Noise-All) and only in testing (w/o Noise-Testing). When disabling the noise factor in all the phases, the effect of the noise factor is lower only to modeling the past news events but higher than the other news impacts, which demonstrates the effectiveness of the noise factor module. This shows the advantage of considering factors beyond news events. It also directly benefits from our representation of equity states, which provides a basis for calculating a noise distribution conditioned on accumulated history.

If only disabling the noises in testing, the F1 score and MCC drop 4.27% and 16.29%, respectively, which is more than the performance degradation of not sampling noises in all the phases. The reason may be that although learning with sampled noises gives the model ability to generalize better on the dataset, testing without noises makes the divergence between training and testing bigger than disabling the noise factor in all the phases.

To show the influence of the noise factor, we also compare with the variation of sampling noises only before the final classification illustrated in Fig. 2 (w/ Delayed Noise). By delaying sampling the noise factor, NES_CN has 3.31%, 4.90% and 15.23% performance degradation for F1/Accuracy/MCC, respectively. We think that NES samples the noise factor to form the equity state $\mathbf{z}_t$ for each trading day, with the help of state transitions shown in Eq. (1) and (2), realizing accumulation of the sampled noises from the previous several trading days. But the variation of sampling noises only before the final classification faces the problem of fit the noises alone in each trading day, which is more computationally challenging to achieve a similar goal.

4.6. Predicting individual stock movements

Other than predicting the S&P 500 index, we also investigate the effectiveness of our approach on individual stock prediction using the test set. In total, ten randomly chosen stocks are evaluated, out of which seven stocks receive accuracies above 50% and three stocks below 50%. In particular, stocks with less company news tend to perform worse than those with more company news except for some specific stocks, which have news articles not highly related to their financial conditions. For better illustration, we count the amounts of individual company related news events by name matching for each company, and select five well known companies from four different sectors with the most news, `Apple Inc.`, `Citigroup`, `Boeing Company`, `Google` and `Wells Fargo`. The sectors are decided by the Global Industry Classification Standard. For each company, we prepare not only news events about itself, but also news events about the whole sector. We use company news, sector news and all financial news to predict individual stock price movements, respectively. The experimental results and news statistics are listed in Table 6.

Individual stock prediction by only using company news dramatically outperforms that using sector news and all news, demonstrating a negative correlation between total used amounts of news events and model performance. The main reason may be that company-related news events can more directly affect the volatility of company shares. In contrast, sector news and all news contain many irrelevant news events, which obstruct NES's learning the underlying stock price movement trends.

Note that Ding et al. [2,6] and Yang et al. [18] also reported results on individual stocks. But we cannot directly compare our results with them because the existing methods used different data split, and Ding et al. [2,6] reported only development set results. This is reasonable since the performance of each model can vary from stock to stock over the S&P 500 chart and comparison over the whole index is more indicative.

4.7. Case study

To look into what news event contributes the most to our prediction result, we further analyze the test set results of the model

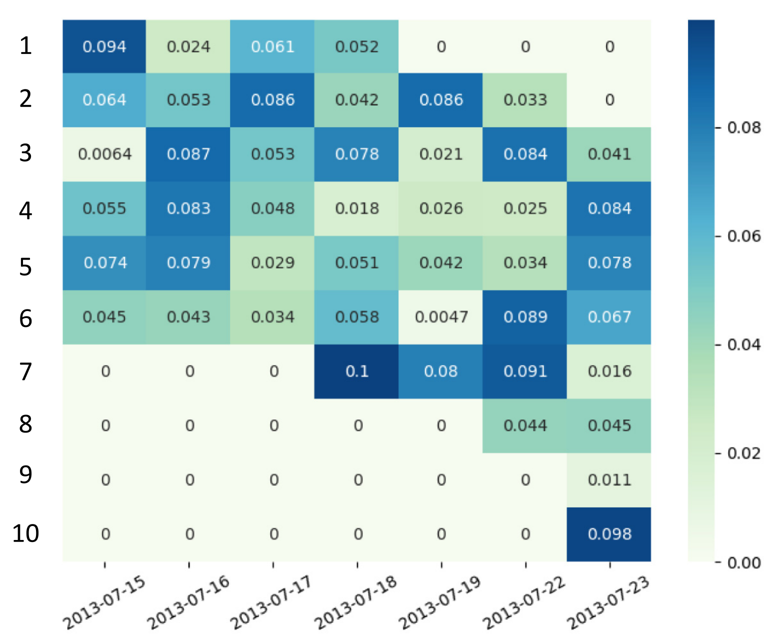
Table 6
Test set results of individual stock price movement prediction.

Stock	Sector	Company News				Sector News				All News		
		#docs	F1	Accuracy	MCC	#docs	F1	Accuracy	MCC	F1	Accuracy	MCC
Apple Inc.	Information Technology	2,398	67.53%	69.21%	0.5632	12,812	64.62%	64.35%	0.3861	58.27%	56.14%	0.2355
Citigroup	Financials	2,058	63.41%	63.57%	0.5193	117,659	57.79%	56.29%	0.3021	55.34%	55.15%	0.1852
Boeing Co.	Industrials	1,870	65.37%	66.25%	0.4423	17,969	62.58%	61.35%	0.2719	60.02%	57.23%	0.1824
Google	Communication Services	1,762	64.81%	66.13%	0.3717	13,344	60.89%	60.47%	0.2644	61.08%	58.41%	0.1387
Wells Fargo	Financials	845	62.97%	61.64%	0.3944	117,659	58.92%	57.34%	0.1294	56.06%	54.64%	0.0823

a)

No.	Date	News Events
1	2013-06-17	Apple Joins Facebook, Microsoft in Outlining Data Requests
2	2013-06-21	Apple Wins Suit Against Samsung in Japan on Screen Effects
3	2013-06-24	Apple Falls Below $400 Amid IPhone Slump, Worker Exits
4	2013-06-25	Samsung Beats Apple in Japanese Patent Suit on Syncing
5	2013-07-02	Apple plans Nevada solar farm in clean energy push for data centers
6	2013-07-10	Apple Faces Damages Trial Over E-Book Antitrust Violation
7	2013-07-17	Apple May Delay Introduction of IPhone 5S, Commercial Times Says
8	2013-07-19	Apple Said to Buy HopStop, Pushing Deeper Into Maps
9	2013-07-21	High-End Smartphone Boom Ending as Price Drop Hits Apple
10	2013-07-22	Apple Developer Website Taken Down After Hacker Attack

b)



Gold	1	1	1	1	-1	1	-1
Pred	1	1	1	-1	1	1	-1

Fig. 5. Given (a) company news related to `Apple Inc.` in the trading sequence [07/15/2013, 07/23/2013], predict `Apple Inc.`'s stock price movements. (b) Attention visualization and results comparison of the trading sequence.

for Apple Inc.'s stock price movements only using company news, which achieves the best results among the five selected companies mentioned before. As shown in Fig. 5, we take the trading sequence from 07/15/2013 to 07/23/2013 for illustration. The table on the top half shows the top-ten news events, while attention visualization and results are shown on the bottom chart. The news events listed in Fig. 5(a) are ranked by the attention scores from the past news events, which are the most effective news according to the ablation study. There are some zeros in the attention heat map in Fig. 5(b) because these news do not belong to the corresponding trading days.

We can find that the News Event 1 has been correlated with the stock price rises on 07/15/2013, but for the next two trading days, its impact fades out. On 07/18/2013, the News Event 7 begins to show its impact. However, NES pays more attention to it compared with other events, which leads to the incorrect prediction that the stock price decreases. On the next trading day, our model infers that the impact of the News Event 2 is bigger than that of the News Event 7, which leads to an incorrect prediction again. From these findings, we can see that NES tends to pay more attention to a new event when it first occurs, which offers us a potential improving direction in the future.

5. Conclusion

We investigated explicit modeling of equity state sequences in news-driven stock prediction by using an LSTM to model the recurrent state transition, adding news impact and noise impact by using attention and noise sampling, respectively. Results show that our method is highly effective, giving the best performance on a standard benchmark for stock index movement prediction. To our knowledge, we are the first to explicitly model both events and noise factors for news-driven stock movement prediction.

CRediT authorship contribution statement

Heyan Huang: Supervision, Funding-acquisition, Writing-review-editing. **Xiao Liu:** Conceptualization, Methodology, Software, Validation, Formal-analysis, Investigation, Data-curation, Writing-original-draft, Visualization. **Yue Zhang:** Investigation, Writing-review-editing, Funding-acquisition. **Chong Feng:** Supervision.

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.

Acknowledgements

We would like to thank the anonymous reviewers for their thoughtful and constructive comments. This work was supported by the Joint Funds of the National Natural Science Foundation of China (Grant No. U19B2020) and the funding from Rxhui Inc. (https://rxhui.com).

References

[1] X. Ding, Y. Zhang, T. Liu, J. Duan, Using structured events to predict stock price movement: an empirical investigation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1415–1425, https://doi.org/10.3115/v1/d14-1148.

[2] X. Ding, Y. Zhang, T. Liu, J. Duan, Deep learning for event-driven stock prediction, in: Proceedings of the 24th International Joint Conference on Artificial Intelligence, 2015, pp. 2327–2333. url:http://ijcai.org/Abstract/15/329.

[3] B. Xie, R.J. Passonneau, L. Wu, G. Creamer, Semantic frames to predict stock price movement, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, 2013, pp. 873–883. url:https://www.aclweb.org/anthology/P13-1086/.

[4] S. Kogan, D. Levin, B.R. Routledge, J.S. Sagi, N.A. Smith, Predicting risk from financial reports with regression, in: Proceedings of the 2009 Conference of the North American Chapter of the Association of Computational Linguistics: Human Language Technologies, 2009, pp. 272–280. url:https://www.aclweb.org/anthology/N09-1031/..

[5] R.P. Schumaker, H. Chen, Textual analysis of stock market prediction using breaking financial news: the azfin text system, ACM Tran. Inf. Syst. 27 (2) (2009) 12:1–12:19. doi:10.1145/1462198.1462204..

[6] X. Ding, Y. Zhang, T. Liu, J. Duan, Knowledge-driven event embedding for stock prediction, in: Proceedings of the 26th International Conference on Computational Linguistics, 2016, pp. 2133–2142. url:https://www.aclweb.org/anthology/C16-1201/.

[7] Y. Xu, S.B. Cohen, Stock movement prediction from tweets and historical prices, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 1970–1979, https://doi.org/10.18653/v1/P18-1183.

[8] W. Li, R. Bao, K. Harimoto, D. Chen, J. Xu, Q. Su, Modeling the stock relation with graph network for overnight stock movement prediction, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, 2020, pp. 4541–4547, https://doi.org/10.24963/ijcai.2020/626.

[9] R. Sawhney, S. Agarwal, A. Wadhwa, R.R. Shah, Deep attentive learning for stock movement prediction from social media text and company correlations, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020, pp. 8415–8426, https://doi.org/10.18653/v1/2020.emnlp-main.676.

[10] L. dos Santos Pinheiro, M. Dras, Stock market prediction with deep learning: a character-based neural language model for event-based trading, in: Proceedings of the Australasian Language Technology Association Workshop, 2017, pp. 6–15. url:https://www.aclweb.org/anthology/U17-1001/..

[11] P. Lin, X. Mo, G. Lin, L. Ling, T. Wei, W. Luo, A news-driven recurrent neural network for market volatility prediction, in: Proceedings of the 4th IAPR Asian Conference on Pattern Recognition, 2017, pp. 776–781, https://doi.org/10.1109/ACPR.2017.35.

[12] J. Long, Z. Chen, W. He, T. Wu, J. Ren, An integrated framework of deep learning and knowledge graph for prediction of stock price trend: an application in chinese stock exchange market, Appl. Soft Comput. 91 (2020), https://doi.org/10.1016/j.asoc.2020.106205 106205.

[13] R. Levine, S. Zervos, Stock market development and long-run growth, World Bank Econ. Rev. 10 (2) (1996) 323–339, https://doi.org/10.1093/wber/10.2.323.

[14] Y. Amihud, Illiquidity and stock returns: cross-section and time-series effects, J. Finan. Markets 5 (1) (2002) 31–56, https://doi.org/10.1016/S1386-4181(01)00024-6.

[15] X. Zhang, Y. Li, S. Wang, B. Fang, P.S. Yu, Enhancing stock market prediction with extended coupled hidden markov model over multi-sourced data, Knowl. Inf. Syst. 61 (2) (2019) 1071–1090, https://doi.org/10.1007/s10115-018-1315-6.

[16] F.M.F. Wong, Z. Liu, M. Chiang, Stock market prediction from WSJ: text mining via sparse matrix factorization, in: Proceedings of the 2014 IEEE International Conference on Data Mining, 2014, pp. 430–439, https://doi.org/10.1109/ICDM.2014.116.

[17] Q. Ding, S. Wu, H. Sun, J. Guo, J. Guo, Hierarchical multi-scale gaussian transformer for stock movement prediction, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, 2020, pp. 4640–4646, https://doi.org/10.24963/ijcai.2020/640.

[18] L. Yang, Z. Zhang, S. Xiong, L. Wei, J. Ng, L. Xu, R. Dong, Explainable text-driven neural network for stock prediction, in: Proceedings of the 5th IEEE International Conference on Cloud Computing and Intelligence Systems, 2018, pp. 441–445, https://doi.org/10.1109/CCIS.2018.8691233.

[19] I.K. Nti, A.F. Adekoya, B.A. Weyori, A novel multi-source information-fusion predictive framework based on deep neural networks for accuracy enhancement in stock market prediction, J. Big Data 8 (1) (2021) 1–28, https://doi.org/10.1186/s40537-020-00400-y.

[20] Y. Peng, H. Jiang, Leverage financial news to predict stock price movements using word embeddings and deep neural networks, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 374–379, https://doi.org/10.18653/v1/n16-1041.

[21] X. Zhang, Y. Zhang, S. Wang, Y. Yao, B. Fang, P.S. Yu, Improving stock market prediction via heterogeneous information fusion, Knowl.-Based Syst. 143 (2018) 236–247, https://doi.org/10.1016/j.knosys.2017.12.025.

[22] Y. Deng, F. Bao, Y. Kong, Z. Ren, Q. Dai, Deep direct reinforcement learning for financial signal representation and trading, IEEE Trans. Neural Networks Learn. Syst. 28 (3) (2017) 653–664, https://doi.org/10.1109/TNNLS.2016.2522401.

[23] X. Wu, H. Chen, J. Wang, L. Troiano, V. Loia, H. Fujita, Adaptive stock trading strategies with deep reinforcement learning methods, Inf. Sci. 538 (2020) 142–158, https://doi.org/10.1016/j.ins.2020.05.066.

[24] J. Sun, H. Li, H. Fujita, B. Fu, W. Ai, Class-imbalanced dynamic financial distress prediction based on adaboost-svm ensemble combined with SMOTE and time weighting, Inf. Fusion 54 (2020) 128–144, https://doi.org/10.1016/j.inffus.2019.07.006.

[25] W. Chen, M. Jiang, W. Zhang, Z. Chen, A novel graph convolutional feature based convolutional neural network for stock trend prediction, Inf. Sci. 556 (2021) 67–94, https://doi.org/10.1016/j.ins.2020.12.068.
[26] S. Gite, H. Khatavkar, K. Kotecha, S. Srivastava, P. Maheshwari, N. Pandey, Explainable stock prices prediction from financial news articles using sentiment analysis, PeerJ Comput. Sci. 7 (2021), https://doi.org/10.7717/peerj-cs.340 e340.
[27] H. Rezaei, H. Faaljou, G. Mansourfar, Stock price prediction using deep learning and frequency decomposition, Expert Syst. Appl. 169 (2021), https://doi.org/10.1016/j.eswa.2020.114332 114332.
[28] C. Chang, Y. Zhang, Z. Teng, Z. Bozanic, B. Ke, Measuring the information content of financial news, in: Proceedings of the 26th International Conference on Computational Linguistics, 2016, pp. 3216–3225. url:https://www.aclweb.org/anthology/C16-1303/..
[29] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.
[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 2017, pp. 5998–6008. url:https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html..
[31] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 2227–2237, https://doi.org/10.18653/v1/n18-1202.
[32] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171–4186. doi:10.18653/v1/n19-1423..
[33] S. Merity, Single headed attention RNN: stop thinking with your head, CoRR abs/1911.11423. arXiv:1911.11423..
[34] H.N. Seyhun, Why does aggregate insider trading predict future stock returns, Quart. J. Econ. 107 (4) (1992) 1303–1331, https://doi.org/10.2307/2118390.
[35] C. Dang, S. Foerster, Z.F. Li, Z. Tang, Analyst talent, information, and insider trading, J. Corp. Finan. 67 (2021), https://doi.org/10.1016/j.jcorpfin.2020.101803 101803.
[36] A. Srivastava, C.A. Sutton, Autoencoding variational inference for topic models, in: Proceedings of the 5th International Conference on Learning Representations, 2017. url:https://openreview.net/forum?id=BybtVK9lg..
[37] M. Kim, V. Pavlovic, Recursive inference for variational autoencoders, in: Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, 2020. url:https://proceedings.neurips.cc/paper/2020/hash/e3844e186e6eb8736e9f53c0c5889527-Abstract.html..
[38] B.W. Matthews, Comparison of the predicted and observed secondary structure of t4 phage lysozyme, Biochim. Biophys. Acta 405 (2) (1975) 442–451, https://doi.org/10.1016/0005-2795(75)90109-9.
[39] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: Proceedings of the 6th International Conference on Learning Representations, 2018. url:https://openreview.net/forum?id=rJXMpikCZ..

Heyan Huang received her Ph.D. degree from China Academy of Sciences, Institute of computer technology in 1989. And now she is a Professor in the School of Computer Science and Technology in Beijing Institute of Technology, and the director of Research Center of High Volume Language Information Processing and Cloud Computing. Her current research interests mainly focus on natural language processing and machine translation.

Xiao Liu received his B.E. degree in computer science and technology from Beijing Institute of Technology, Beijing, in 2016. Currently, he is a Ph.D. candidate in the School of Computer Science and Technology, Beijing Institute of Technology. His research interests include natural language processing, information extraction, and event extraction.

Yue Zhang is currently an Associate Professor with Westlake University, Hangzhou, China. His research interests include natural language processing and computational finance. He has been working on statistical parsing, text synthesis, natural language synthesis, machine translation, information extraction, sentiment analysis, and stock market analysis intensively. He won the best paper awards of IALP 2017 and COLING 2018. He serves the editorial board for the Transactions of the Association of Computational Linguistics (Action Editor), ACM Transactions on Asian and Low-Resource Language Information Processing (Associate Editor), and IEEE TRANSACTIONS ON BIG DATA (Associate Editor), and as area chairs of COLING 2014/18, NAACL 2015/19, EMNLP 2015/17/19, and ACL 2017/18/19.

Chong Feng received the PhD degree from the University of Science and Technology of China, in 2005. And now he is an Associate Professor with the School of Computer Science and Technology in Beijing Institute of Technology. His current research interests include information extraction and sentiment analysis.