International Journal of Forecasting 36 (2020) 1541–1562

Contents lists available at ScienceDirect

International Journal of Forecasting

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

The role of text-extracted investor sentiment in Chinese stock price prediction with the enhancement of deep learning

Yelin Li [a], Hui Bu [b,*], Jiahong Li [c], Junjie Wu [a,d]

[a] *Department of Information Systems, School of Economics and Management, Beihang University, Beijing 100191, China*
[b] *Department of Finance, School of Economics and Management, Beihang University, Beijing 100191, China*
[c] *JD Digits, Beijing 100176, China*
[d] *Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China*

A R T I C L E I N F O

Keywords:
Textual data
Text mining
Naïve Bayes classification algorithm
Deep learning method
Stock price forecasting

A B S T R A C T

Whether investor sentiment affects stock prices is an issue of long-standing interest for economists. We conduct a comprehensive study of the predictability of investor sentiment, which is measured directly by extracting expectations from online user-generated content (UGC) on the stock message board of *Eastmoney.com* in the Chinese stock market. We consider the influential factors in prediction, including the selections of different text classification algorithms, price forecasting models, time horizons, and information update schemes. Using comparisons of the long short-term memory (LSTM) model, logistic regression, support vector machine, and Naïve Bayes model, the results show that daily investor sentiment contains predictive information only for open prices, while the hourly sentiment has two hours of leading predictability for closing prices. Investors do update their expectations during trading hours. Moreover, our results reveal that advanced models, such as LSTM, can provide more predictive power with investor sentiment only if the inputs of a model contain predictive information.

1. Introduction

Whether investor sentiment affects stock prices is a question of long-standing interest for economists. Numerous authors have considered the possibility that a significant presence of sentiment-driven investors can cause prices to depart from their fundamental values. The "noise trader" theories of Black (1986) and De Long, Shleifer, Summers, and Waldmann (1990) suggest that if some investors trade on a "noisy" signal that is unrelated to the fundamentals, asset prices will deviate from their intrinsic values. Currently, scholars increasingly accept the notion that stock prices are driven by two types of investors: noise traders and arbitrageurs (Shleifer & Summers, 1990). The classic argument against the effects of sentiment is that these effects are eliminated by rational traders who seek to exploit the arbitrage opportunities created by mispricing. However, if rational traders cannot fully exploit these opportunities, then sentiment effects become more likely. Therefore, we conduct a comprehensive study of the effectiveness of investor sentiment in the Chinese stock market by considering as many influential factors in prediction as possible, including the direct measurement of investor sentiment via messages from internet stock message boards, selection of different text classification algorithms, comparison of price prediction models, configuration of different time horizons, and information update schemes.

To ensure that individuals' expectations toward future returns, which are identified via online posted messages, are employed as the investor sentiment indicator, we improve the text mining method to obtain the text-extracted investor sentiment measure for the Chinese stock market. This is the first contribution of this study. Using collected messages from the internet stock message board

* Corresponding author.
E-mail address: buhui@buaa.edu.cn (H. Bu).

https://doi.org/10.1016/j.ijforecast.2020.05.001
0169-2070/

of *Eastmoney.com*, which is the largest and most representative stock message board in China, we classify each piece of user-generated content (UGC) into three types of expectations—positive, negative, and neutral—by using the Naïve Bayes classification algorithm, under the consideration of the difficulty of dictionary-based approaches in processing online posted messages, especially those with Chinese words. We further improve several aspects of the sentiment extraction method: First, following Bu, Xie, Li, and Wu (2018), we focus on extracting expectations for the future stock price changes from UGC rather than the emotions of users when they posted messages on the stock forum. Following the definition of finance field, positive sentiment indicates a bullish expectation about future returns or the tendency to buy; negative sentiment indicates a bearish expectation about future returns or the tendency to sell; neutral sentiment indicates a neutral and stable expectation about future returns or no tendency to trade. Second, we compare different algorithms to solve the three-class classification problem, including the direct three-class Naïve Bayes classification algorithm and the three binary Naïve Bayes classification algorithms with a voting algorithm. The comparison results show that the direct three-class Naïve Bayes classification algorithm, which tends to classify messages into positive and negative categories and therefore could benefit subsequent indicator construction, is a better choice. Third, we improve the method for constructing the investor sentiment indicator by combining the ideas of message bullishness (Antweiler & Frank, 2004) and investor attention, which gives the novel bullishness and attention indicator. Fourth, the issue of repeated posting messages, which may be related to robot posting, is carefully discussed. We compare different message aggregation methods to construct the investor sentiment indicators, including the total message aggregation and non-repeated message aggregation approaches. The results reveal that the repeated posting issue is not relevant for our study due to the small ratio of repeated messages in the text sample. Last, it is notable that text-extracted investor sentiment indicators are not limited by time frequency and time horizon. Sentiment indicators can be constructed at any frequency—monthly, weekly, daily, hourly, or even at a very high-frequency level. Sentiment indicators can also express the expectations at any time interval set as needed, which is one reason why we adopt this approach to measure investor sentiment.

Many studies have examined investor sentiment's predictive power and effects on stock returns (e.g., Antweiler & Frank, 2004; Kim & Kim, 2014; Tetlock, 2007) with inconsistent results. For example, Antweiler and Frank (2004) discover that online messages help predict market volatility, while their effect on stock returns is statistically significant but economically small based on high frequency data for the year 2000. They also suggest that disagreement among the posted messages is associated with increased trading volume. Tetlock (2007) finds that high media pessimism predicts downward pressure on market prices followed by a reversion to fundamentals and that unusually high or low pessimism predicts a high market trading volume. However, Kim and Kim (2014) obtain no evidence that investor sentiment forecasts future stock returns at either the aggregate level or individual firm level or that investor sentiment from internet message postings has the power to predict volatility or trading volume. Bu et al. (2018) discover that investor sentiment extracted from internet stock message boards does not have the power to predict daily returns, volatility, or trading volume in the Chinese stock market. However, investor sentiment does have a significant positive effect on contemporaneous stock price. Moreover, Bu et al. (2018) uncover that text-extracted investor sentiment constructed by using messages before the market opens could predict the open price of stock market.

Given the inconsistency of the results reported in the literature, we are interested in several questions: Is the examination of the investor sentiment's predictive power related to the forecasting models, forecasting horizon, and update of information? These aspects motivate the second and most important contribution of this study—to provide more comprehensive evidence about whether individuals' posted messages contain information that can predict future stock returns. First, we investigate whether different forecasting models lead to different forecasting performances, which can yield different conclusions about the predictive power of investor sentiment. If the predictive power of investor sentiment is strong, we can expect that all models can reveal correct results despite their varied predictive abilities. However, if the predictive power is weak, some models may not be able to report the correct results. When no models can work, investor sentiment has no predictive power, or the predictive power is too weak to be applied in practice. This study introduces a deep learning model, which is referred to as *long short-term memory* (LSTM), to construct forecasts for stock returns. We consider LSTM to be a very suitable model to describe the characteristics of the stock market, because LSTM shows remarkable results in tasks which have strong noise, such as speech recognition (Graves, Mohamed, & Hinton, 2013) and time series prediction (Wang, Zhang, Tang, Wu, & Xiong, 2019). The forecasting performance of LSTM is compared with that of several benchmark models, including the support vector machine (SVM), logistic regression, and Naïve Bayes model, to discuss the issue of investor sentiment's predictive power.

Based on a sample from January 1, 2009, to October 31, 2014, we develop an LSTM model that is based on investor sentiment extracted from internet stock message boards and market data to conduct out-of-sample forecasts for the open and closing prices of the CSI 300 index in the Chinese stock market. The results show that daily investor sentiment can adequately predict the subsequent trading day's market open prices, while the predictive information for the daily closing price is weak. The results confirm that an advanced model such as LSTM can extract more effective information from inputs, which should be carefully considered in the discussion of the predictive power of investor sentiment.

Second, we investigate how many hours in advance investor sentiment can forecast and whether investors update their sentiment. We incorporate hourly investor sentiment indicators in the models and compare them with models that only utilize daily inputs. We discover that incorporating hourly sentiment yields minimal improvement for forecasting open prices because daily input variables have enough information. Our empirical results reveal that we can use the previous day's information to predict the open prices, which implies that the information in investor sentiment after the market closes does not change substantially. However, for forecasting closing prices, hourly investor sentiment has two hours of leading predictive power. This result reveals that investors incorporate new information and update expectations during trading hours, which equips the more recent investor sentiment with more predictive power for closing prices. Our results provide evidence of how individuals' expectations affect stock prices in the Chinese stock market.

The remainder of this paper is organized as follows: Section 2 reviews the literature about measurement of investor sentiment and financial market forecasting with machine learning methods. Section 3 describes the data collection and method for constructing the text-extracted investor sentiment indicators employed in this study. The statistics of the constructed text-extracted investor sentiment indicators and stock returns are also discussed. Section 4 explains how to employ the LSTM model to explore the predictive power of investor sentiment and provides comparisons of different models and different information update schemes. Section 5 concludes the paper.

2. Literature review

2.1. Measurement of investor sentiment

In the literature, the meaning of investment sentiment has received considerable attention. Bu and Pi (2014) compare the definitions of investor sentiment in prior studies, such as those by Baker and Wurgler (2006), Berger and Turtle (2012), Brown and Cliff (2004), and Hribar and McInnis (2012), and suggest that although these studies define sentiment in slightly different ways, the essence of their expressions is the same. Brown and Cliff (2004) note that sentiment represents market participants' expectations relative to a norm: a bullish (bearish) investor expects returns to be above (below) average, regardless of how the "average" is defined. Baker and Wurgler (2006) indicate that sentiment is a belief about future cash flows and investment risks and is not justified by the facts at hand. Hribar and McInnis (2012) consider that sentiment reflects errors in investors' expectations about future payoffs. Berger and Turtle (2012) consider that sentiment is the general optimism or pessimism toward future returns. We find that these definitions have two key points: (1) expectation, which indicates the beliefs and judgments about assets' future returns and (2) errors in expectations. This study discusses the key question of how to explore investors' expectations about future returns.

The traditional methods used to measure investor sentiment in the literature are mainly based on direct surveys or the construction of a sentiment index using market variables. Baker and Wurgler (2006, 2007) create an aggregate sentiment index based on six sentiment proxies, including the closed-end fund discount, share turnover, number of initial public offerings (IPOs), first-day IPO return, share of equity issues relative to debt issues, and dividend premium. Many scholars —for example, Berger and Turtle (2012)—adopt Baker and Wurgler's monthly sentiment index. We note that the investor sentiment index constructed using observable market variables as proxies is limited by the available data frequency of proxy variables, which is usually expressed in monthly frequency. With the development of text mining techniques, some researchers have begun to directly extract opinions and beliefs from online text data, such as data from internet stock message boards, social media platforms, news websites, and online searches. Antweiler and Frank (2004) investigate the information content of the *Yahoo! Finance* and *Raging Bull* internet stock message boards and measure the messages' bullishness using computational linguistic methods. Kim and Kim (2014) follow the same method to measure investor sentiment based on the stock message boards of *Yahoo! Finance*. Tetlock (2007) extracts media pessimism from the daily content of a popular *Wall Street Journal* column as the proxy for investor sentiment, which is further extended by Tetlock, Saar-Tsechansky, and Macskassy (2008) to address the impact of negative words in all *Wall Street Journal* and *Dow Jones News Service* stories about individual S&P 500 firms from 1980 to 2004. Data on online search volumes are also employed to measure investor sentiment (Chen, De, Hu, & Hwang, 2014) and investor attention (Da, Engelberg, & Gao, 2011; Joseph, Wintoki, & Zhang, 2011). Most studies of the text-extracted measure of investor sentiment focus on the United States market. Evidence about other markets is limited. This paper complements the literature by extending the text-extracted investor sentiment analysis to the Chinese stock market, which is an important emerging market, to provide new empirical evidence of investor sentiment predictive power.

2.2. Machine learning methods in financial time series

Machine learning methods and artificial neural networks are well suited for forecasting highly volatile financial time series with nonlinearity, temporal correlation, and strong noise. Artificial neural networks (ANNs) and recurrent neural networks (RNN) are usually employed for forecasting asset prices, especially stock prices (Kamijo & Tanigawa, 1990) and futures prices (Kohzadi, Boyd, Kermanshahi, & Kaastra, 1996), as well as their volatility (Hamid & Iqbal, 2004; Tino, Schittenkopf, & Dorffner, 2001). Compared with other types of neural networks, the most important advantage of an RNN is its particular network structure, including circulations, which enables it to consider information persistence—that is, the temporal characteristics of time series as input data. Other types of neural networks need to flatten the input time series without considering their characteristics. However, an RNN cannot perfectly address the long-term dependence problem. To address this problem, a particular type of RNN, namely LSTM, is proposed. LSTM is a deep learning method that has shown remarkable results in tasks such as artificial handwriting generation (Graves, 2013), language forecasting (Sundermeyer, Schlüter, & Ney, 2012), and speech recognition (Graves et al., 2013).

Traditional machine learning methods have insufficient capability to extract suitable features and capture the nonlinear nature of complex tasks in financial markets. Recently, some researchers have begun to develop deep learning methods for financial markets, such as an autoencoder composed of stacked restricted Boltzmann machines for stock return forecasting (Takeuchi & Lee, 2013), continuous restricted Boltzmann machines for foreign exchange rate forecasting (Shen, Chao, & Zhao, 2015), a spatial neural network model for predicting the limit order books (Sirignano, 2016), and deep neural networks (DNNs) for high-frequency futures prices forecasting (Dixon, Klabjan, & Bang, 2016). Deep learning methods can be combined with other techniques to solve the time series forecasting problems of stock markets. For example, Bao, Yue, and Rao (2017) combine wavelet transforms, stacked autoencoders, and LSTM for forecasting daily closing prices of market indices. Li, Shang, and Wang (2018) apply convolutional neural network method and latent Dirichlet allocation method to extract and group online texts to forecast crude oil prices.

Recently, deep learning methods have been utilized in the discussion of the predictability of investor attention and investor sentiment in the stock market, especially based on the measures of investor sentiment or investor attention constructed by text mining techniques. Xiong, Nichols, and Shen (2015) apply LSTM neural networks to model S&P 500's volatilities by incorporating Google Domestic Trends and choosing 25 target words to represent the public interest in various macroeconomic factors, which can represent investor attention. Feuerriegel and Fehrer (2016) use a recurrent autoencoder and information extracted from daily headline financial news to forecast stock market movements in Germany. Ding, Zhang, Liu, and Duan (2015) propose a deep convolutional neural network, in which the events are extracted from news, to model short-term and long-term influences of events on stock price movements.

3. Construction of investor sentiment indicators

3.1. Data and message classification using the Naïve Bayes classification algorithm

To construct investor sentiment indicators, we need to extract expectations from internet messages. We collect online messages from the internet stock message board of *Eastmoney.com*, which is the largest and most representative stock message board in China. *Eastmoney.com* is an important and famous finance and economics website in China; it has the most monthly unique visitors among all the financial information websites in China according to *Iresearch*, an authoritative Chinese consulting organization.[1] We adopt the CSI 300 index as a proxy for the whole Chinese A-share stock market, because it is the most important stock market index in China that describes stock movements in both the Shanghai Stock Exchange and Shenzhen Stock Exchange. We collect all the internet messages related to all component stocks of the CSI 300 index posted on the stock message board of *Eastmoney.com*[2] using the Web crawler Scrapy. Because the components of CSI 300 change with time, we collect approximately 18 million messages on all the constituent stocks in the CSI 300 index from January 1, 2009, to October 31, 2014. Each message includes 12 terms, as shown in Table 1. From 2009 to 2014, a daily average of approximately 6000–8000 text messages are posted on the stock message board of *Eastmoney.com*.

Our first job is to extract the expectations and beliefs about future stock returns from each message. There are two main methods to conduct text mining: the dictionary-based approach and corpus-based approach (Liu & Zhang, 2012). Processing online posted messages, especially those with Chinese words, is difficult using the dictionary-based approach (Zhao, Qin, & Liu, 2010). Thus, this study adopts the corpus-based approach and Naïve Bayes classification algorithm, which provide a simple, robust, and efficient way to solve the classification problem of high-dimensional, sparse, and short texts (Zhao, Dong, Wu, & Xu, 2012). Antweiler and Frank (2004) test the robustness of the Naïve Bayes classification algorithm by comparing it with the SVM classifier to classify individual messages; their results reveal that these two algorithms produce similar results. Moreover, they prefer the Naïve Bayes classification algorithm because of its low complexity. In the Naïve Bayes classification algorithm, a given text message is split into a group of words, which yields a "bag of words" for every text. We only retain the verbs, nouns, and adjectives. For a given message d, this step can be denoted by $d = (tf_1, \ldots, tf_i, \ldots, tf_n)$, where tf_i is the text feature. The probability that a text feature vector belongs to a specific category c_j is

$$P(tf_i|c_j) = \frac{TF(tf_i, c_j) + 1}{\sum_{c_j \in C}(TF(tf_i, c_j) + 1)}, \tag{1}$$

where $TF(tf_i, c_j)$ denotes the frequency of text feature tf_i that belongs to c_j. The denominator adopts Laplace smoothing to avoid the situation of zero. The probability that message d belongs to c_j can be estimated as

$$P(c_j|d) = P(c_j)P(d|c_j)/P(d) \propto P(c_j)\prod_{i=1}^{n} P(tf_i|c_j). \tag{2}$$

Therefore, the classification of message d is determined by

$$c_d = \underset{c_j \in C}{arg\ max} \left\{ P(c_j)\prod_{i=1}^{n} P(tf_i|c_j) \right\}. \tag{3}$$

In this study, we need to classify messages into three categories: positive, negative, and neutral, that is, $c \in \{pos, neu, neg\}$, according to the expectations or beliefs expressed in the messages. Positive sentiment in a message means that the stock price mentioned in the message is expected to rise in the near future, or it indicates the poster's tendency to buy this stock. Negative sentiment in a message means that the stock price mentioned in the message is expected to fall in the near

[1] See from the link: https://index.iresearch.com.cn/pc.

[2] The internet stock message board of *Eastmoney.com* is available at http://guba.eastmoney.com/.

Table 1
Message terms of the collected internet posted messages.

Message terms	Notes
stock_id	It is the ticker symbol, which is unique for each stock
user_id	If the user is anonymous,this term is "none".
user_url	If the user is anonymous,this term is "none".
stockholder	True or False
stock_name	
content	
em_info	There are six types, including news, data, research report, notice, stockholder, and none. Each message is labeled to a type.
replies	
title	
post_id	
clicks	
releaseTime	

future, or it indicates the poster's tendency to sell this stock. Neutral sentiment means that the stock price is expected to remain unchanged in the near future with no obvious expectation and poster has no tendency to trade. Our definition of investor sentiment is different from the definition of emotion-based sentiment that cares about poster's emotional happiness or unhappiness. The emotion-based sentiment is usually applied in the text mining analysis field. For example, for the message "The market will fall sharply tomorrow. I have already sold all my stocks today. Ha-ha!", the linguistic emotion is happy but the expectation for the future market is negative. To ensure that we extract the posters' actual expectations or beliefs about future stock prices, we invite 10 financial experts and financial analysts to label the texts. We randomly choose 5000 messages as our training sample. Each message is labeled by at least two analysts, and we review and compare the labeling results. If one text has completely different labeling results, we invite a third analyst to label the text again, and then we consider the majority as the final label of this message. After the labeling process, messages are classified into three categories, $c \in \{pos, neu, neg\}$, which yields 1586 positive labels, 1765 negative labels, and 1649 neutral labels.

To address the multiclass classification problem, we utilize different strategies to set up the Naïve Bayes classification algorithm. For example, we can directly employ a three-class Naïve Bayes classification algorithm, or we can adopt three binary classification algorithms and then decide the category of messages via a voting mechanism according to the results of these binary classification algorithms. This study compares both of these classification approaches. We utilize a five-hold, cross-validation method to test the out-of-sample classification accuracy of these algorithms. In the k-fold cross-validation, dataset D is randomly split into k mutually exclusive subsets (folds), $D_1, D_2, \ldots, D_k$, of approximately equal size. Of the k subsamples, a single subsample is retained as a validation set for testing the algorithm, and the remaining $k-1$ subsamples are employed as the training set. The cross-validation process is then repeated k times, with each k subsample employed exactly once as the validation set. The k results can then be averaged to produce a single estimation. The validation results of these two types of classification algorithms are reported in Table 2.

Panel A of Table 2 shows the results of the three-class Naïve Bayes classification algorithm. The prediction accuracy of the positive and negative categories is relatively high with $f_{pos} = 0.6848$ and $f_{neg} = 0.6639$. In addition, we find that the precision of the positive and neutral classes is higher than that of the negative class, while the recall of the negative class is higher than that of the other two classes, especially the neutral class. This classification algorithm has the highest precision for the positive category—71.49%—and has the highest recall for the negative category—84.99%. The neutral category's precision exceeds 70% but its recall is low, which causes the F-score to be lower than 50% for the neutral category. This result implies that the three-class Naïve Bayes classification algorithm tends to classify the messages into the positive or negative category. Panel B of Table 2 reports the results of the binary-class classification and voting algorithm. The F-scores of the three categories are 64.38%, 64.32%, and 58.64%. Although the binary classification and voting algorithm improves the precision of the negative category and the recall and F-score of the neutral category, it causes a decrease in the F-score of the positive and negative categories compared with the three-class algorithm. This finding implies that the binary classification and voting algorithm tends to classify messages with complex opinions into the neutral category, which leads to a more balanced accuracy than the direct three-class classification algorithm. This result may be due to the voting mechanism. As this study considers the positive and negative categories more than the neutral category and the three-class Naïve Bayes classification algorithm has better accuracy, we prefer the three-class Naïve Bayes classification algorithm in our study.

Our Naïve Bayes classification algorithm is more appropriate than similar methods in the literature, even when considering the difficulty of classifying Chinese text. Most studies examine the in-sample accuracy of the Naïve Bayes classification algorithm. For example, Antweiler and Frank (2004) train the Naïve Bayes classification algorithm based on 1000 manually classified messages, and report that the precision of the "buy" or "sell" category is approximately 72.3%. Das and Chen (2007) report that the Naïve Bayes algorithm has 50% in-sample classification accuracy for text messages from *Yahoo! Finance*. Kim and Kim (2014) classify the messages into two categories—"buy" and "sell"—using the Naïve Bayes algorithm based

Table 2
Accuracy of the Naïve Bayes classification algorithms using five-fold cross-validation tests.

Panel A: Three-class Naïve Bayes classification algorithm									
Trial ID	Positive category			Negative category			Neutral category		
	p	r	f	p	r	f	p	r	f
1	74.73%	70.47%	72.54%	53.35%	83.8%	65.2%	73.05%	38.87%	50.74%
2	68.65%	63.6%	66.03%	55.31%	82.51%	66.23%	67.36%	35.53%	46.52%
3	72.59%	67.38%	69.89%	55.74%	86.62%	67.83%	71.43%	36.26%	48.1%
4	68.98%	62.83%	65.76%	53.93%	85.41%	66.12%	67.9%	36.42%	47.41%
5	72.49%	64.34%	68.17%	54.07%	86.59%	66.57%	75.18%	37.59%	50.12%
Average	71.49%	65.72%	68.48%	54.48%	84.99%	66.39%	70.98%	36.93%	48.58%
Panel B: Multiple binary Naïve Bayes classifications and voting algorithm									
Trial ID	Positive category			Negative category			Neutral category		
	p	r	f	p	r	f	p	r	f
1	60.09%	51.89%	55.69%	63.78%	63.28%	63.53%	51.60%	58.76%	54.95%
2	65.49%	53.62%	58.96%	59.93%	58.18%	59.04%	46.23%	56.54%	50.87%
3	66.23%	54.51%	59.80%	60.23%	55.12%	57.56%	45.68%	58.96%	51.48%
4	62.04%	53.71%	57.58%	57.20%	53.07%	55.06%	47.80%	56.86%	51.94%
5	91.39%	88.41%	89.87%	89.47%	83.51%	86.39%	80.19%	88.07%	83.95%
Average	69.05%	60.43%	64.38%	66.12%	62.63%	64.32%	54.30%	63.84%	58.64%

Note: p, r, and f denote the precision, recall, and F-score, respectively, $f = 2pr/(p + r)$.

on the sample of 2,000 randomly selected "buy" messages and 2,000 "sell" messages. They report that the mean of the 200 in-sample hitting percentages is 86.3% by repeating the in-sample training test 200 times. Kim and Kim (2014) also report that the mean of 200 out-of-sample hitting percentages is 62.7% by repeating out-of-sample test 200 times based on the randomly selected 2000 out-of-sample "buy" messages and 2000 out-of-sample "sell" messages.

3.2. Investor sentiment indicators construction based on the extracted expectations

In this section, we discuss the method to construct the investor sentiment indicators based on the messages' classification results. Following Antweiler and Frank (2004), Bu et al. (2018), and Kim and Kim (2014), we adopt the messages' bullishness indicator, which is defined as Eq. (4):

$$B_t = \frac{M_t^{pos} - M_t^{neg}}{M_t^{pos} + M_t^{neg}}, \tag{4}$$

where $M_t^c = \sum_{i \in D(t)} w_i x_i^c$ denotes the weighted sum of messages of type $c \in \{pos, neu, neg\}$ in the time interval $D(t)$, where x_i^c is an indicator variable that is equal to one when message i is type c and zero otherwise, and w_i is the weight of the message. Antweiler and Frank (2004) reveal that the alternative weighting schemes do not affect any of their research conclusions and apply the simplicity of equal weighting. Therefore, we employ equal weighting in this study. With the equal weighting scheme, M_t^c can be calculated as the number of messages of different categories for simplicity. Antweiler and Frank (2004) define another message bullishness indicator, as shown in Eq. (5),

$$B_t^* = \ln\left[\frac{1 + M_t^{pos}}{1 + M_t^{neg}}\right], \tag{5}$$

and provide the relationship between these two indicators: $B_t^* \approx B_t \ln\left(1 + \left(M_t^{pos} + M_t^{neg}\right)\right)$. They note that the second measure shown in Eq. (5) considers the number of traders that express a particular sentiment. Therefore, they prefer the second indicator. Their results show that measure B_t^* appears to outperform the other alternatives in their study.

The messages' bullishness indicators B_t and B_t^* in Antweiler and Frank (2004) only include the numbers of positive and negative expectation messages without considering neutral texts. Even if the neutral messages do not contain obvious sentimental information, they contribute to the total investor attention and are valuable. Therefore, we propose a new investor sentiment indicator B_t^{Att}, which combines message bullishness and investor attention, as defined in Eq. (6),

$$B_t^{Att} = B_t \ln\left(1 + M_t\right), \tag{6}$$

where M_t is the total amount of UGC messages on the internet stock message boards at the time interval $D(t)$, which is a proxy for investor attention. M_t changes with the investor attention on stocks but is not affected by the message classification algorithms. Thus, B_t^{Att} is decided by two parts: relative bullishness expressed by B_t and investor attention.

The investor sentiment indicators can be calculated at different time horizons—for example, daily or hourly. The daily investor sentiment for day t is extracted according to the posted messages from 15:00 of the previous trading day to 15:00 of the current trading day t. The hourly investor sentiment indicators are calculated at the end of each one-hour period using the classified messages released within this hour. For example, the investor sentiment at 16:00 contains posted messages from 15:00 to 16:00.

Moreover, we consider the repeated message problem that may be related to the robot posting issue in the message aggregation part. We examine the repeated messages in our sample. We count the number of total messages ($MTotal_t^c$), number of repeated messages ($MRep_t^c$), and number of messages after removing the repeated messages ($MNonRep_t^c$). Then, we examine some statistics

Table 3
Statistics of total, repeated, and non-repeated messages in the sample.

	Three-class classification algorithm				Binary classification and voting algorithm			
	Positive	Negative	Neutral	Total	Positive	Negative	Neutral	Total
$MTotal_t^c$ in the whole sample	4470090	6162796	7888145	18521031	4432228	4455184	9633619	18521031
Daily average of $MTotal_t^c$	3163.55	4361.50	5582.55	13107.59	3136.75	3152.99	6817.85	13107.59
Daily percentage of $MTotal_t^c$	24.14%	33.27%	42.59%	100.00%	23.93%	24.05%	52.01%	100.00%
$MNonRep_t^c$ in the whole sample	4195422	5871475	7376489	17443386	4143909	4198354	9101123	17443386
Daily average of $MNonRep_t^c$	2969.16	4155.33	5220.45	12344.93	2932.70	2971.23	6440.99	12344.93
Daily percentage of $MNonRep_t^c$	24.05%	33.66%	42.29%	100.00%	23.76%	24.07%	52.18%	100.00%
$MRep_t^c$ in the whole sample	274668	291321	511656	1077645	288319	256830	532496	1077645
Daily average of $MRep_t^c$	194.39	206.17	362.11	762.66	204.05	181.76	376.85	762.66
Repeated ratio of all messages	6.14%	4.73%	6.49%	5.82%	6.51%	5.76%	5.53%	5.82%

Table 4
Descriptive statistics of daily investor sentiment indicators, returns, and trading volume of market index.

Panel A: Three-class Naïve Bayes classification algorithm												
Statistics	R_t	Vol_t	Total messages aggregation					Only non-repeated messages aggregation				
			M_t^{pos}	M_t^{neg}	B_t	B_t^*	B_t^{Att}	M_t^{pos}	M_t^{neg}	B_t	B_t^*	B_t^{Att}
No. of Obs.	1412	1413	1413	1413	1413	1413	1169	1413	1413	1413	1413	1169
Mean	0.0003	6.7919	3164	5583	−0.2777	−0.5719	−0.0098	2969	5220	−0.2758	−0.5675	−0.0087
Median	0.0004	6.1332	3035	5402	−0.2805	−0.5763	−0.0164	2843	5025	−0.2788	−0.5727	−0.0144
Maximum	0.0668	23.917	7167	12162	−0.1125	−0.2260	0.2992	6727	11333	−0.1111	−0.2230	0.2997
Minimum	−0.0711	2.1901	1329	2119	−0.4160	−0.8856	−0.3243	1251	1974	−0.4158	−0.8851	−0.3190
Std. Dev.	0.0149	2.9335	971.98	1607.08	0.0483	0.1045	0.0933	909.08	1514.31	0.0470	0.1016	0.0918
Skewness	−0.1996	1.1747	0.8310	0.5710	0.2619	0.1837	0.2384	0.8167	0.6094	0.2550	0.1783	0.2242
Kurtosis	5.0940	4.9876	3.7901	3.1125	2.9179	2.9818	3.0455	3.7246	3.1904	2.9375	2.9050	2.9991
Jarque–Bera	264.5***	557.6***	198.4***	77.31***	16.55***	8.80**	5.96*	187.1***	89.29***	15.54***	8.07**	5.96*
ADF test	−10.44***	−3.34*	−3.72**	−3.79**	−6.54***	−6.54***	−6.11***	−3.56**	−3.69**	−6.55***	−6.56***	−6.11***
Panel B: Binary Naïve Bayes classification and voting algorithm												
Statistics	R_t	Vol_t	Total messages aggregation					Only non-repeated messages aggregation				
			M_t^{pos}	M_t^{neg}	B_t	B_t^*	B_t^{Att}	M_t^{pos}	M_t^{neg}	B_t	B_t^*	B_t^{Att}
No. of Obs.	1412	1413	1413	1413	1413	1413	1413	1413	1413	1413	1413	1413
Mean	0.0003	6.7919	2808	2822	−0.0041	−0.0082	−0.0357	2636	2664	−0.0065	−0.0129	−0.0577
Median	0.0004	6.1332	2680	2732	−0.0060	−0.0120	−0.0569	2526	2577	−0.0088	−0.0176	−0.0813
Maximum	0.0668	23.917	6272	6136	0.1873	0.3791	1.7725	5963	5810	0.1845	0.3732	1.7055
Minimum	−0.0711	2.1901	1161	1073	−0.1955	−0.3960	−1.9084	1114	1014	−0.1819	−0.3678	−1.6814
Std. Dev.	0.0149	2.9335	880.13	840.39	0.0584	0.1171	0.5485	823.53	799.60	0.0572	0.1148	0.5338
Skewness	−0.1996	1.1747	0.8354	0.5933	0.1599	0.1601	0.1831	0.8346	0.6192	0.1676	0.1674	0.1885
Kurtosis	5.0940	4.9876	3.7149	3.1067	3.0254	3.0527	3.1267	3.7239	3.1408	3.0161	3.0412	3.1020
Jarque–Bera	264.5***	557.6***	193.5***	83.34***	6.04**	6.16**	8.75**	193.97***	91.19***	6.61**	6.66**	8.91**
ADF test	−10.44***	−3.34*	−3.74**	−3.82**	−7.05***	−7.05***	−7.03***	−3.59**	−3.68**	−7.08***	−7.08***	−7.06***

Note: *, **, and *** represent that the statistic is significant at the 10% level, 5% level, and 1% level, respectively.

including messages in our whole sample, daily average numbers, and daily percentage ratio of different categories for the above variables. We report the statistics in Table 3. We discover that the repeated ratio of all messages is only 5.82%. This small quantity of repeated messages implies that the difference is minimal whether considering repeated messages or not in each category. We construct investor sentiment indicators using these two types of message aggregation methods—that is, aggregating all messages in a specific time interval and aggregating only the non-repeated messages.

3.3. Analysis of investor sentiment indicators and stock returns

Based on 5000 randomly selected messages as our training sample, we train our chosen Naïve Bayes classification algorithm and then obtain all the classification results for all messages. We obtain the investor sentiment indicators, including the two message bullishness indicators (B_t and B_t^*) and a new investor sentiment indicator (B_t^{Att}) at the daily and hourly horizons using different message aggregation methods based on the message classification results of different algorithms. To investigate the intertemporal predictive power of investor sentiment for stock prices, we adopt the CSI 300 index as a proxy for the Chinese A-share stock market. The study sample includes data from January 1, 2009, to October 31, 2014. The beginning date of our study sample is limited by the availability of the collected messages on the internet stock message board for the component stocks in the CSI 300 index. We define stock returns as log price returns using the closing prices of the market index—that is, $R_t = \ln(P_t/P_{t-1})$. Table 4 provides the descriptive statistics of daily returns, trading volume, number of classified messages, and constructed investor sentiment indicators according to different classification algorithms using the all messages' aggregation approach or only non-repeated messages' aggregation approach. The Jarque–Bera statistics in Table 4 reveal that none of the daily variables

Table 5
Correlation of daily investor sentiment indicators and stock returns.

Panel A: Three-class Naïve Bayes classification algorithm

	R_t	Vol_t	M_t	B_t(Total)	B_t^*(Total)	B_t^{Att}(Total)
R_t	1.0000					
Vol_t	0.1407	1.0000				
M_t	−0.0310	0.4999	1.0000			
B_t (Total)	0.1894	0.1625	0.1795	1.0000		
B_t^*(Total)	0.1889	0.1608	0.1760	0.9998	1.0000	
B_t^{Att}(Total)	0.1993	0.0748	0.0088	0.9834	0.9837	1.0000
$MNonRep_t$	−0.0253	0.5297	0.9928	0.1638	0.1604	−0.0029
B_t (Non-repeated)	0.1970	0.1897	0.1766	0.9884	0.9883	0.9717
B_t^* (Non-repeated)	0.1965	0.1882	0.1735	0.9882	0.9884	0.9720
B_t^{Att} (Non-repeated)	0.2061	0.0949	0.0014	0.9714	0.9718	0.9888

Panel B: Binary Naïve Bayes classification and voting algorithm

	R_t	Vol_t	M_t	B_t	B_t^*	B_t^{Att}
B_t (Total)	0.2201	0.1625	0.1448	1.0000		
B_t^*(Total)	0.2201	0.1608	0.1448	1.0000	1.0000	
B_t^{Att}(Total)	0.2216	0.1602	0.1486	0.9995	0.9995	1.0000
$MNonRep_t$	−0.0207	0.4367	0.8688	0.1268	0.1268	0.1293
B_t (Non-repeated)	0.2287	0.1307	0.1367	0.9884	0.9876	0.9876
B_t^* (Non-repeated)	0.2286	0.1306	0.1367	0.9884	0.9876	0.9876
B_t^{Att} (Non-repeated)	0.2304	0.1297	0.1393	0.9884	0.9887	0.9886

Note: M_t is the number of total messages. $MNonRep_t$ is the number of non-repeated messages. "Total" in the bracket indicates the message aggregation method, which aggregates the total messages in each category. "Non-repeated" in the bracket presents the other message aggregation method, which aggregates only the non-repeated messages in each category.

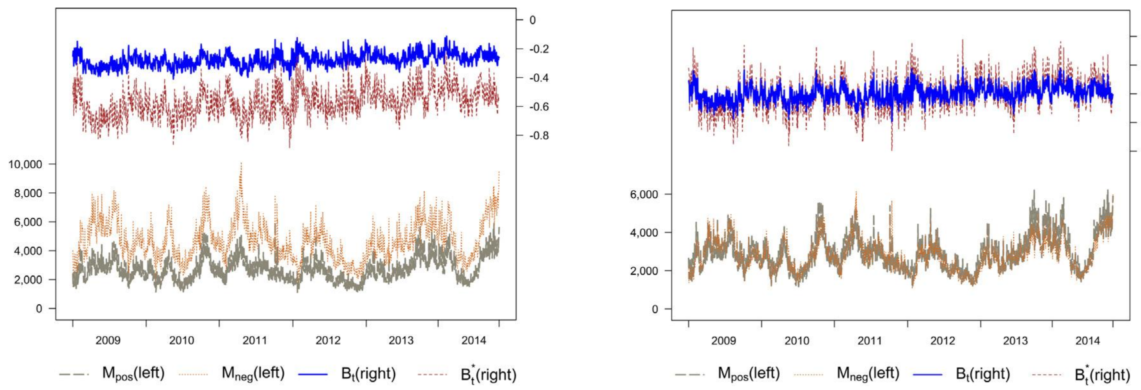
follows a normal distribution. The augmented Dickey–Fuller (ADF) statistics show that stock returns and all investor sentiment indicators are stationary series at the 1% significance level. Moreover, Table 5 provides the correlations between these variables. The correlation between different investor sentiment indicators based on the results of the three-class classification algorithm and stock market returns is 0.19, and the correlation between different investor sentiment indicators based on the results of the binary classification algorithm and stock returns is 0.22. If we consider only the non-repeated messages, the correlation between the investor sentiment indicator and the stock returns increases slightly. The correlations between any two of the three investor sentiment indicators are close to 1, which implies no big difference among these three sentiment indicators. The correlation between the trading volume of the CSI 300 index (Vol_t) and the total number of messages (M_t) is 0.49, which is much higher than the correlation between the trading volume and investor sentiment indicators. These results show that trading activity increases with investor attention, and that there is a positive correlation between sentiment and stock returns.

From the results of Tables 4 and 5, we determine that the difference between the two types of message aggregation methods is very small. Thus, we only report the results of investor sentiment indicators constructed by using the trained three-class and multiple binary classification algorithms based on aggregating total messages in the following sections to conserve space.[3] Fig. 1 describes the daily message bullishness indicators (B_t and B_t^*), and Fig. 2 shows the daily investor sentiment indicator (B_t^{Att}). To present the difference between the investor sentiment by combining message bullishness indicators B_t and investor attention, we also include B_t for comparison in Fig. 2. Furthermore, we calculate and obtain the hourly investor sentiment indicators using the messages of all component stocks in the CSI 300 index from January 1, 2009, to October 31, 2014. To further clarify the hourly investor sentiment indicators and investigate the intraday pattern of investor sentiment indicators, we calculate the average of each hour's investor sentiment indicators across the whole sample and report them in Fig. 3. The trading hours of the Chinese stock market are from 9:00 to 11:30 and from 13:00 to 15:00. Fig. 3 presents results from the first nontrading hour after the market closes to the last trading hour of the next day.

Figs. 1–3 clearly indicate differences between the two types of message classification algorithms. The three-class classification algorithm classifies many messages into the negative category, which yields negative investor sentiment indicators most of the time. This is because the precision and recall for the negative category is low and high, respectively, as shown in Table 2. When we change the classification algorithm to the proposed binary classification and voting algorithm, the number of messages in the negative category decreases to a level that is similar to that of the positive category. This is because the binary classification and voting algorithm have higher precision and decrease the recall of the negative category, as shown in Table 2. This finding leads the investor sentiment indicators to fluctuate near zero. We will further discuss the classification algorithms in the following sections to determine which method can provide more expectation information about future stock returns.

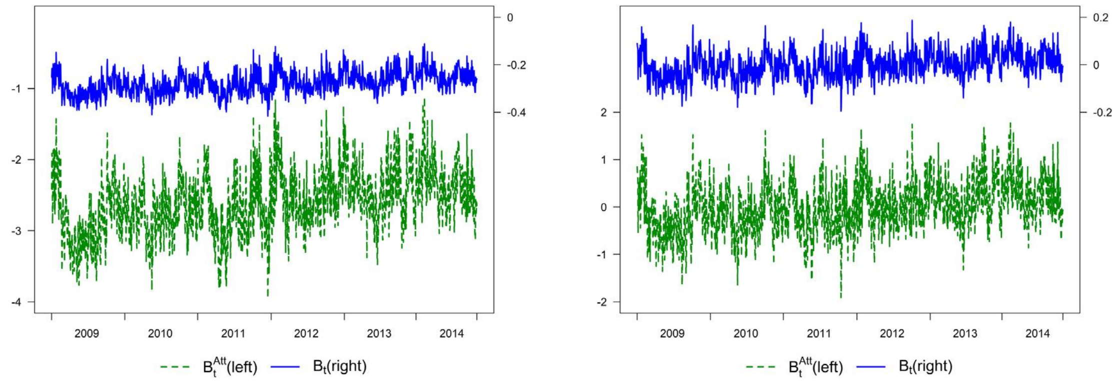
Moreover, Fig. 3 reveals some intraday patterns of investor sentiment. We discover a stable sentiment even

3 The results of investor sentiment indicators constructed by using the trained classification algorithms based on only the non-repeated messages may be provided upon request.



(A) Three-class classification algorithm (B) Binary classification and voting algorithm

Fig. 1. Message bullishness indicators and the number of classified messages.[4]



(A) Three-class classification algorithm (B) Binary classification and voting algorithm

Fig. 2. Daily investor sentiment indicators B_t^{Att} and B_t.

with a rising trend during the nontrading hours before the market opens. Investors' expectations are volatile during the trading hours of the day and they have a tendency to sell during the afternoon trading hours (from 13:00 to 15:00). Investors seem to discuss their tendency to trade during the midday break of the Chinese stock market with greater detail, as shown in the changes in investor sentiment indicators at 13:00. These results imply that investors may be affected by bad information more easily and form negative expectations during the trading hours of the Chinese stock market.

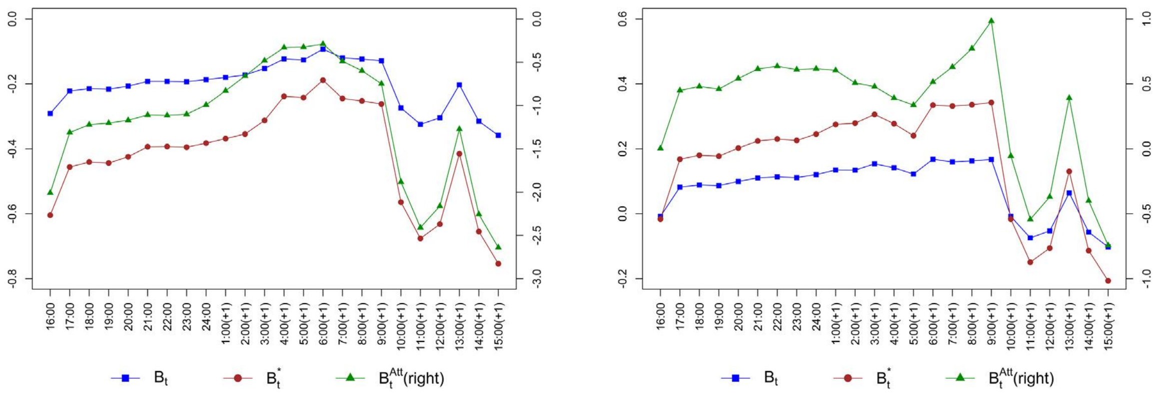
4. Exploring the predictive power of investor sentiment under the LSTM method

4.1. LSTM model

To investigate whether different forecasting methods produce different results of the investor sentiment's power of predicting stock returns, we introduce the LSTM method in our discussion. LSTM is a deep RNN method that can capture the temporary features of time series input and solve the unknown long-term reliance problems of complex financial time series data. We adopt the LSTM model to forecast future price changing directions of the stock market index. By using the constructed investor sentiment indicators from the classified messages posted on the internet stock message board about component stocks in the CSI 300 index, according to the Naïve Bayes models, and transaction data, such as open prices, closing prices, and trading volume of the stock market index, we set up the LSTM models to obtain forecasts of changes in the direction of daily open prices and daily closing prices of the market index. We illustrate the whole framework of the empirical design in Fig. 4.

LSTM is a different type of RNN architecture that was introduced by Hochreiter and Schmidhuber (1997) in conjunction with an appropriate gradient-based learning algorithm. LSTM is designed to overcome the error backflow problems in conventional backpropagation through time or real-time recurrent learning. Hochreiter and Schmidhuber (1997) state that LSTM can learn to bridge minimal

4 Investor sentiment indicators described in the figures are that constructed by using the trained classification algorithms based on total messages. The investor sentiment indicators obtained based on non-repeating messages are not reported to save space.



(A) Three-class classification algorithm (B) Binary classification and voting algorithm

Fig. 3. Hourly investor sentiment indicators.

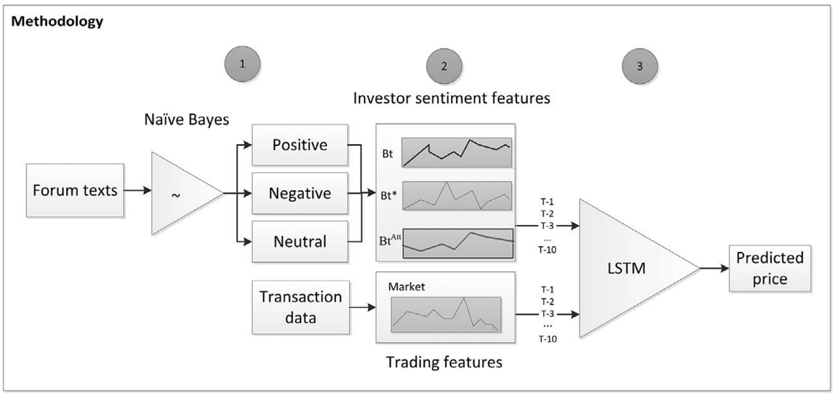


Fig. 4. Framework of the empirical design. (Note: LSTM is long short-term memory model).

time lags in excess of 1000 discrete-time steps by enforcing a constant error flow through constant error carousels within special units. Compared with standard RNNs, LSTM is better at storing and accessing information (Graves, 2013), and can solve the long-term dependency problem, which remains unsolved by RNNs in practice (Bengio, Simard, & Frasconi, 1994; Hochreiter, 1991).

To construct an architecture that allows for a constant error flow through special, self-connected units without the disadvantages of the naïve approach of constant error flow, the LSTM model introduces a multiplicative input gate unit to protect the memory contents stored in j from perturbation by irrelevant inputs. The model also introduces a multiplicative output gate unit to protect other units from perturbation by the currently irrelevant memory contents stored in j. The LSTM architecture employs these purpose-built memory cells to store information; they are better at exploiting long-range dependencies in the data. The architecture of a single LSTM memory cell is illustrated in Fig. 5. The hidden layer function of the LSTM is implemented by the following composite function, as discussed by Graves (2013):

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i), \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f), \\ c_t &= f_t c_{t-1} + \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o), \\ h_t &= o_t \tanh(c_t), \end{aligned} \tag{7}$$

where σ is the logistic sigmoid function; x is the current input of the network; and i, f, o, and c are the *input gate*, *forget gate*, *output gate*, *cell and cell input* activation vectors, respectively, which have the same size as the hidden vector h. The weight matrix subscripts have an obvious meaning; for example, W_{hi} is the hidden-input gate matrix and W_{xo} is the input–output gate matrix. Because the weight matrices from the cell vector to the gate vector (e.g., W_{ci}) are diagonal, element m in each gate vector only receives input from element m of the cell vector.

The original LSTM model in Hochreiter and Schmidhuber (1997) applies a custom-designed approximate gradient calculation that allows the weights to be updated after

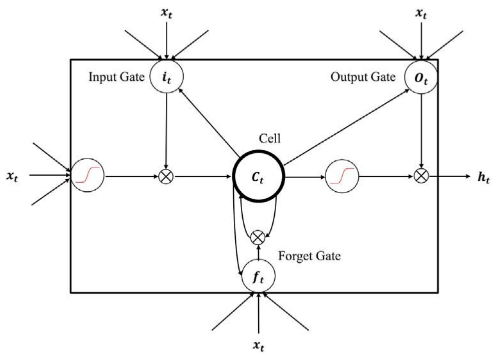


Fig. 5. Architecture of memory cells of the long short-term memory (LSTM) model.
Source: Graves (2013).

every timestep. Graves (2013) adopts the full gradient calculated with backpropagation through time instead. He also points out that one difficulty when training LSTM with the full gradient is that the derivatives may become excessively large, which creates numerical problems. Graves (2013) modifies the derivative of the loss with respect to the network inputs to the LSTM layers (before the sigmoid and tanh functions are applied) to lie within a predefined range. We follow Graves's (2013) method in this study.

4.2. Specification of the LSTM model and empirical design

To investigate the predictive power of investor sentiment in more detail, we construct several LSTM models to examine different issues, such as whether daily investor sentiment indicators can predict future open prices, whether daily investor sentiment indicators can predict future closing prices, and whether hourly investor sentiment indicators can improve the forecasting performance about future open or closing prices. The forecasting objects of this paper are the directions of log returns of open prices and closing prices; that is, up direction when $R_t > 0$ or down direction when $R_t < 0$. We are concerned with two types of horizons: daily and hourly. We test all three measures of investor sentiment indicators in this study. Furthermore, we investigate whether the LSTM model can improve the forecasting performance of investor sentiment indicators; so, the results can change the conclusion about whether investor sentiment has some power to predict future prices. We compare the results of the LSTM models with those of other benchmark models, such as the SVM, logistic regression, and Naïve Bayes models. The classification threshold of the models is simply set to 0.5. The forecasting performance of each trained model based on the training sample is evaluated based on the accuracy of the out-of-sample forecasts of log price changes direction (up or down) in the test sample in terms of precision, recall, and F-score.

The LSTM model in this study is specified as follows: Using investor sentiment indicators and market historical transaction data as input time series, two fully connected layers are included in our LSTM model—a rectified linear unit (ReLU) layer and a SoftMax linear layer—which are commonly used activation functions in neural networks. We choose to generate two outputs, which correspond to the possibility of a rise or fall in price. The output dimensionalities of the ReLU layer and SoftMax layer are denoted as (U_R, U_S), respectively. All coefficients of the model are obtained by training with the python deep learning library of Keras.[5] Categorical cross-entropy is set as the objective loss, and the validation fraction is 5%. To accelerate the convergence, we choose the root mean square propagation algorithm (Tieleman & Hinton, 2012), which is a popular neural network optimizer in deep learning frameworks. Moreover, data points are shuffled during training, and the learning rate is set to 0.02 in our work; all initial weights are set as small positive constant terms, which is similar to the normalized initialization given in Glorot and Bengio (2010). To prevent overfitting, the dropout has been set to 20%, and we add an L2 regularization constraint with the parameter 0.01 in the open price predictions and that with the parameter 0.001 in the closing price predictions.

For the input time series of the LSTM model, we choose a maximum of 10 latest consecutive observations in the model. To investigate whether investor sentiment has predictive power, we compare different input time series for LSTM models and different measures of investor sentiment. Specifically, we introduce different investor sentiment indicators in the input time series, compare the results of LSTM models with and without investor sentiment indicators, and compare the difference in the results of the LSTM models using various sentiment measures. In the baseline LSTM model that is utilized for comparison with other models, the input time series only includes the previous 10 days' latest historical transaction data of the market index, such as closing prices, open prices, and trading volume. We record this baseline input as $X_{0,t-1}$,

$$X_{0,t-1} = \left\{CLOSE_{t-10,9,\ldots,1}, OPEN_{t-10,9,\ldots,1}, VOLUME_{t-10,9,\ldots,1}\right\},$$

where $\left\{CLOSE_{t-10,9,\ldots,1}, OPEN_{t-10,9,\ldots,1}, VOLUME_{t-10,9,\ldots,1}\right\}$ represent the historical time series of closing prices, open prices, and trading volume of the CSI 300 index at day $t-10, t-9, \ldots, t-1$, respectively. We incorporate different indicators of investor sentiment extracted from online messages.

$$X_{1,t-1} = \left\{CLOSE_{t-10,9,\ldots,1}, OPEN_{t-10,9,\ldots,1}, VOLUME_{t-10,9,\ldots,1}, B_{t-10,9,\ldots,1}\right\},$$
$$X_{2,t-1} = \left\{CLOSE_{t-10,9,\ldots,1}, OPEN_{t-10,9,\ldots,1}, VOLUME_{t-10,9,\ldots,1}, B^{*}_{t-10,9,\ldots,1}\right\},$$
$$X_{3,t-1} = \left\{CLOSE_{t-10,9,\ldots,1}, OPEN_{t-10,9,\ldots,1}, VOLUME_{t-10,9,\ldots,1}, B^{Att}_{t-10,9,\ldots,1}\right\},$$

where B_{t-i}, B^{*}_{t-i}, and B^{Att}_{t-i} are the investor sentiment indicators discussed in Section 3. According to the aggregation method that is employed to construct the investor sentiment indicators discussed in Section 3, the daily investor

[5] https://github/fchollet/keras, GitHub Repository.

sentiment for day t is extracted according to the posted messages from 15:00 of the previous trading day to 15:00 of the current trading day. We note that the latest information as of 15:00 of the previous day is employed in the daily models to forecast the open prices and closing prices.

To improve the timeliness of information, we attempt to add more information near the time of the predicted price. Thus, we add hourly investor sentiment indicators for the most recent day (from the first hour at 16:00 of the previous day after the market closes to at least one hour before the time of the predicted prices) to the models based on the daily data models. Moreover, we examine how many hours in advance investor sentiment can predict the prices' change directions. Therefore, we construct a series of models using the information ending at different hours. The input time series, including hourly data, is recorded as $X_{i,t-1,t_h-h-1}$,

$$X_{i,t-1,t_h-h-1} = \Big\{CLOSE_{t-10,9,\ldots,1}, OPEN_{t-10,9,\ldots,1}, VOLUME_{t-10,9,\ldots,1}, Sentiment_{i,t-10,9,\ldots,1}, Sentiment_{i,t_h-h-10,9,\ldots,1}\Big\},$$

where $Sentiment_{i,t-10,9,\ldots,1}$ is the set of daily sentiment indicators of the most recent 10 days. $Sentiment_{i,t} = \{B_t, B_t^*, B_t^{Att}\}$ represents three investor sentiment indicators. $Sentiment_{i,t_h-h-10,9,\ldots,1}$ represents the 10 nearest observations of hourly investor sentiment indicators at least $h+1$ hours before t_h to forecast the price at time t_h. With the changing set of h, we obtain different input data groups for hourly investor sentiment.

4.3. Forecasting results of daily investor sentiment

We separate our whole sample collected from January 1, 2009, to October 31, 2014, into two subsamples: training sample and test sample. The training sample entails the period from January 1, 2009, to March 21, 2014, for a total of 1273 trading days. The test sample entails the period from March 24, 2014, to October 31, 2014, for a period of 140 trading days. We train all the models based on the training sample and apply the trained models to obtain out-of-sample forecasts. As a kind of deep neural network method, LSTM introduces randomness from a commonly used dropout algorithm in deep learning, which is proposed to avoid overfitting by randomly omitting part of the feature detectors in each training case (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012; Krizhevsky, Sutskever, & Hinton, 2012). The randomness makes networks and their testing results slightly different, even for the unchanging training sample. To reduce the impact of randomness, we perform the trained LSTM model 50 times for the test sample and obtain the average of the forecasts as the forecasting result of the LSTM model. Other benchmark models have no randomness problem. Because the forecasting objects are the directions of returns, we adopt the accuracy measures of precision, recall, and F-score of accuracy to evaluate the out-of-sample forecasting performance.

As the results of the two types of message aggregation methods are similar, we only report the forecasting results of the sentiment indicators constructed using a non-repeated message aggregation method with the three-class message classification method, to compare with the forecasting results of sentiment indicators constructed using the all message aggregation method and three-class classification method. To conserve space, we do not provide other results related to the sentiment constructed using the non-repeated message aggregation method. Specifically, the main parameters in the LSTM models are obtained as follows: For forecasting open prices in the LSTM model, the parameters for different message classification methods (three-class classification and binary classification and voting) and different message aggregation methods (total messages or only non-repeated messages) are
$(U_R, U_S)^{open}_{three-class,totalmessages} = (10, 30)$,
$(U_R, U_S)^{open}_{three-class,nonrepeated} = (10, 30)$, and
$(U_R, U_S)^{open}_{binary,totalmessages} = (10, 10)$, respectively; for forecasting the closing prices in the LSTM model, the parameters are $(U_R, U_S)^{close}_{three-class,totalmessages} = (6, 5)$,
$(U_R, U_S)^{close}_{three-class,nonrepeated} = (6, 5)$, and
$(U_R, U_S)^{close}_{binary,totalmessages} = (30, 5)$, respectively. To compare the forecasting performance, we report the forecasting accuracy measures of the LSTM models and those of the benchmark models, including the SVM, logistic regression, and Naïve Bayes classification models in the tables. Tables 6 and 7 show the forecasting results for open prices using daily investor sentiment. Table 6 reports the results of the models using daily sentiment indicators constructed by the three-class Naïve Bayes message classification, and Table 7 reports the results of the models using daily investor sentiment indicators constructed by the multiple binary classifications and voting algorithm. Tables 8 and 9 show the forecasting results for closing prices using daily investor sentiment indicators.

Based on the results of forecasting open prices, as reported in Tables 6 and 7, the forecasting performances of different models are all acceptable, with the lowest accuracy of 55%–57.86% for the SVM model. LSTM can improve the forecasting accuracy by 6.33%–23.47% with the inputs of only daily prices and trading volumes. Table 6 shows that the difference in the aggregation methods of the sentiment indicators is very small. The best forecasting performance, with an accuracy reaching 80.20%, is achieved by the LSTM model using B_t^* constructed by the three-class Naïve Bayes message classification algorithm and aggregating all the messages in each category, as shown in Panel A of Table 6. Panel B of Table 6 presents similar results; that is, the best forecasting model is the LSTM model using B_t^{Att} constructed by the three-class Naïve Bayes message classification algorithm and aggregating only non-repeated messages in each category. According

[6] To conserve space, we do not report the forecasting performance of the models with the daily sentiment obtained based on multiple binary classification and voting algorithm and constructed using non-repeated aggregation in forecasting open prices of the market index, because the results are similar to that of the daily sentiment obtained based on multiple binary classification and voting algorithm and constructed using aggregating total messages. The results that are not reported here are available upon request.

Table 6
Performance of models with the daily sentiment obtained based on three-class message classification algorithm in forecasting open prices of the market index.

Panel A: Sentiment constructed by using three-class classification strategy and aggregating total messages							
Method & Input	Price change direction: Up			Price change direction: Down			Total
	p	*r*	*f*	*p*	*r*	*f*	Accuracy
LSTM+X_0	84.64%	77.22%	77.88%	80.74%	79.79%	77.21%	78.47%
LSTM+X_1	85.62%***	77.81%	78.85%	81.36%	81.38%***	78.66%***	79.54%
LSTM+X_2	83.87%	81.56%***	80.20%***	83.91%***	78.76%	78.60%	80.20%***
LSTM+X_3	83.62%	77.64%	76.90%	81.55%	78.06%	76.44%	77.84%
SVM+X_0	60.00%	37.50%	46.15%	52.63%	73.53%	61.35%	55.00%
SVM+X_1	62.50%	41.67%	50.00%	54.35%	73.53%	62.50%	57.14%
SVM+X_2	63.83%*	41.67%	50.42%*	54.84%*	75.00%*	63.35%*	57.86%*
SVM+X_3	60.42%	40.28%	48.33%	53.26%	72.06%	61.25%	55.71%
LR+X_0	72.60%	73.61%*	73.10%*	71.64%*	70.59%	71.11%*	72.14%*
LR+X_1	71.83%	70.83%	71.33%	69.57%	70.59%	70.07%	70.71%
LR+X_2	71.43%	69.44%	70.42%	68.57%	70.59%	69.57%	70.00%
LR+X_3	72.86%*	70.83%	71.83%	70.00%	72.06%*	71.01%	71.43%
NB+X_0	70.59%	66.67%	68.57%	66.67%	70.59%*	68.57%	68.57%
NB+X_1	71.23%	72.22%*	71.72%	70.15%	69.12%	69.63%	70.71%
NB+X_2	71.23%	72.22%*	71.72%	70.15%	69.12%	69.63%	70.71%
NB+X_3	72.22%*	72.22%*	72.22%*	70.59%*	70.59%*	70.59%*	71.43%*
Panel B: Sentiment constructed by using three-class classification strategy and aggregating non-repeated messages							
Method & Input	Price change direction: Up			Price change direction: Down			Total
	p	*r*	*f*	*p*	*r*	*f*	Accuracy
LSTM+X_0	84.64%***	77.22%	77.88%	80.74%***	79.79%***	77.21%	78.47%
LSTM+X_1	83.90%	78.25%	78.12%	81.59%	78.76%	76.97%***	78.50%
LSTM+X_2	82.71%	80.22%	78.86%	82.71%	77.41%	77.17%	78.86%
LSTM+X_3	80.74%	86.69%***	82.03%***	86.85%	73.53%	76.68%	80.30%***
SVM+X_0	60.00%	37.50%	46.15%	52.63%	73.53%*	61.35%	55.00%
SVM+X_1	62.00%	43.06%*	50.82%	54.44%	72.06%	62.03%	57.14%
SVM+X_2	63.27%*	43.06%*	51.24%*	54.95%*	73.53%*	62.89%*	57.86%*
SVM+X_3	62.00%	43.06%*	50.82%	54.44%	72.06%	62.03%	57.14%
LR+X_0	72.60%*	73.61%*	73.10%*	71.64%*	70.59%*	71.11%*	72.14%*
LR+X_1	71.43%	69.44%	70.42%	68.57%	70.59%*	69.57%	70.00%
LR+X_2	70.42%	69.44%	69.93%	68.12%	69.12%	68.61%	69.29%
LR+X_3	70.27%	72.22%	71.23%	69.70%	67.65%	68.66%	70.00%
NB+X_0	70.59%*	66.67%	68.57%	66.67%	70.59%*	68.57%	68.57%
NB+X_1	70.42%	69.44%*	69.93%*	68.12%*	69.12%	68.61%*	69.29%*
NB+X_2	70.42%	69.44%*	69.93%*	68.12%*	69.12%	68.61%*	69.29%*
NB+X_3	70.42%	69.44%*	69.93%*	68.12%*	69.12%	68.61%*	69.29%*

Notes: a. SVM, LR, and NB denote the support vector machine, logistic regression, and Naïve Bayes classification algorithms, respectively.
b. X_0, X_1, X_2, and X_3 are the different combination of daily inputs.
c. The precision, recall, and F-score are applied to evaluate the direction forecasting performance, which are recorded as *p, r*, and *f*, respectively, where $f = 2pr/(p+r)$. The overall model accuracy is also reported.
d. * represents the best forecasting model among the same method with different inputs, where *** represents the best forecasting model among all methods.

to the results of Panel A of Table 6, the LSTM model, which incorporates daily investor sentiment indicators, can improve the forecasting accuracy by approximately 8.06%–22.34% compared with the best benchmark models. Most importantly, incorporating daily investor sentiment indicators into different models, except the logistic regression model, can improve the accuracy by 1.73%–2.86%.

The results of investor sentiment indicators constructed by the three-class Naïve Bayes message classification algorithm are better than those of the sentiment indicators constructed by the multiple binary classifications and voting algorithm. We discover that adding investor sentiment indicators constructed by the multiple binary classification algorithm can only improve models with low forecasting accuracy, such as the SVM and Naïve Bayes models, with an improvement in accuracy of 0.71%–1.43%. However, the addition of indicators cannot improve models with excellent forecasting accuracy that are based on the input data of daily prices and trading volume, such as the LSTM and logistic regression models, as shown in Table 7. Compared with the results in Table 6, the results in Table 7 imply that the investor sentiment indicators constructed by the multiple binary classifications and voting algorithm provide less predictable information about future returns than the sentiment indicators constructed by the three-class classification algorithm. This outcome may be because of the disadvantage of the voting strategy. As reported in Table 2, although the multiple binary classifications and voting algorithm can substantially improve the recall of the neutral category and improve the precision of the negative category, this message classification algorithm considerably decreases the recall of the negative category and precision of the neutral category and decreases all the accuracy measures of the positive category. These results mean that more messages, including bullish or bearish expectations, are incorrectly classified into the neutral category. Although the binary classification and voting algorithm make the

Table 7
Performance of the models with daily sentiment obtained based on multiple binary classifications and voting algorithm in forecasting open prices of the market index.[6]

Sentiment indicators constructed by multiple binary classifications and aggregating total messages							
Method & Input	Price change direction: Up			Price change direction: Down			Total
	p	r	f	p	r	f	Accuracy
LSTM+X_0	84.64%***	77.22%***	77.88%***	80.74%***	79.79%***	77.21%***	78.47%***
LSTM+X_1	76.69%	75.44%	75.05%	75.11%	73.40%	72.94%	74.45%
LSTM+X_2	76.40%	76.75%	75.11%	76.28%	71.93%	72.53%	74.41%
LSTM+X_3	76.14%	73.71%	72.99%	73.86%	71.75%	70.96%	72.76%
SVM+X_0	60.00%*	37.50%	46.15%	52.63%	73.53%*	61.35%*	55.00%
SVM+X_1	57.81%	51.39%*	54.41%*	53.95%*	60.29%	56.94%	55.71%*
SVM+X_2	57.81%	51.39%*	54.41%*	53.95%*	60.29%	56.94%	55.71%*
SVM+X_3	57.81%	51.39%*	54.41%*	53.95%*	60.29%	56.94%	55.71%*
LR+X_0	72.60%*	73.61%*	73.10%*	71.64%*	70.59%*	71.11%*	72.14%*
LR+X_1	71.62%	73.61%	72.60%	71.21%	69.12%	70.15%	71.43%
LR+X_2	71.23%	72.22%	71.72%	70.15%	69.12%	69.63%	70.71%
LR+X_3	70.27%	72.22%	71.23%	69.70%	67.65%	68.66%	70.00%
NB+X_0	70.59%	66.67%	68.57%	66.67%	70.59%*	68.57%	68.57%
NB+X_1	70.83%*	70.83%*	70.83%*	69.12%*	69.12%	69.12%*	70.00%*
NB+X_2	70.83%*	70.83%*	70.83%*	69.12%*	69.12%	69.12%*	70.00%*
NB+X_3	70.83%*	70.83%*	70.83%*	69.12%*	69.12%	69.12%*	70.00%*

Note: Similar to the note of Table 6.

Table 8
Performance of the daily sentiment obtained based on three-class message classification algorithm in forecasting closing prices of the market index.

Panel A: Sentiment constructed by using three-class classification strategy and aggregating total messages							
Method & Input	Price change direction: Up			Price change direction: Down			Total
	p	r	f	p	r	f	Accuracy
LSTM+X_0	51.18%	55.39%	52.89%*	46.44%	44.18%	45.22%	49.94%
LSTM+X_1	51.66%	51.89%	51.74%	48.82%	48.59%	48.66%	50.29%
LSTM+X_2	51.77%*	53.44%*	52.42%	47.97%	47.26%	47.58%	50.44%*
LSTM+X_3	51.21%	52.28%	51.70%	48.33%	47.26%	47.75%	49.84%
SVM+X_0	51.40%*	76.39%	61.45%	48.48%	23.53%	31.68%	50.71%*
SVM+X_1	51.35%	79.17%***	62.30%***	48.28%	20.59%	28.87%	50.71%*
SVM+X_2	50.89%	79.17%***	61.96%	46.43%	19.12%	27.08%	50.00%
SVM+X_3	50.45%	77.78%	61.20%	44.83%	19.12%	26.80%	49.29%
LR+X_0	51.02%	69.44%	58.82%	47.62%	29.41%	36.36%	50.00%
LR+X_1	53.19%	69.44%	60.24%	52.17%	35.29%	42.11%	52.86%
LR+X_2	53.68%***	70.83%*	61.08%*	53.33%***	35.29%	42.48%	53.57%***
LR+X_3	52.63%	69.44%	59.88%	51.11%	33.82%	40.71%	52.14%
NB+X_0	48.05%	51.39%	49.66%	44.44%	41.18%	42.75%	46.43%
NB+X_1	48.72%	52.78%*	50.67%*	45.16%	41.18%	43.08%	47.14%
NB+X_2	48.72%	52.78%*	50.67%*	45.16%	41.18%	43.08%	47.14%
NB+X_3	47.44%	51.39%	49.33%	43.55%	39.71%	41.54%	45.71%
Panel B: Sentiment constructed by using three-class classification strategy and aggregating non-repeated messages							
Method & Input	Price change direction: Up			Price change direction: Down			Total
	p	r	f	p	r	f	Accuracy
LSTM+X_0	51.18%	55.39%	52.89%	46.44%	44.18%	45.22%	49.94%
LSTM+X_1	51.08%	68.33%	57.35%	34.95%	30.69%	32.49%	50.05%
LSTM+X_2	51.46%*	73.18%*	59.06%*	33.09%	26.56%	28.69%	50.54%*
LSTM+X_3	51.09%	68.25%	57.65%	38.29%	30.91%	33.95%	50.11%
SVM+X_0	51.40%*	76.39%***	61.45%***	48.48%	23.53%	31.68%	50.71%*
SVM+X_1	50.46%	76.39%	60.77%	45.16%	20.59%	28.28%	49.29%
SVM+X_2	50.46%	76.39%	60.77%	45.16%	20.59%	28.28%	49.29%
SVM+X_3	50.46%	76.39%	60.77%	45.16%	20.59%	28.28%	49.29%
LR+X_0	51.02%	69.44%*	58.82%*	47.62%	29.41%	36.36%	50.00%
LR+X_1	53.41%***	65.28%	58.75%	51.92%***	39.71%	45.00%	52.86%***
LR+X_2	53.41%*	65.28%	58.75%	51.92%***	39.71%	45.00%	52.86%***
LR+X_3	52.17%	66.67%	58.54%	50.00%	35.29%	41.38%	51.43%
NB+X_0	48.05%	51.39%	49.66%	44.44%	41.18%	42.75%	46.43%
NB+X_1	47.44%	51.39%	49.33%	43.55%	39.71%	41.54%	45.71%
NB+X_2	47.44%	51.39%	49.33%	43.55%	39.71%	41.54%	45.71%
NB+X_3	47.44%	51.39%	49.33%	43.55%	39.71%	41.54%	45.71%

Note: Similar to the note of Table 6.

Table 9
Performance of the daily sentiment obtained based on multiple binary classification and voting algorithm in forecasting closing prices of the market index. [7]

Sentiment indicators constructed by multiple binary classification and aggregating total messages							
Method & Input	Price change direction: Up			Price change direction: Down			Total
	p	*r*	*f*	*p*	*r*	*f*	Accuracy
LSTM+X_0	51.18%*	55.39%	52.89%	46.44%	44.18%	45.22%	49.94%
LSTM+X_1	50.95%	72.31%	59.42%	45.48%	26.85%	33.00%	50.23%*
LSTM+X_2	50.94%	75.32%*	60.51%*	42.98%	23.48%	29.89%	50.14%
LSTM+X_3	50.45%	69.97%	58.24%	44.59%	27.91%	33.59%	49.54%
SVM+X_0	51.40%	76.39%	61.45%	48.48%	23.53%	31.68%	50.71%
SVM+X_1	50.93%	76.39%	61.11%	46.88%	22.06%	30.00%	50.00%
SVM+X_2	51.85%***	77.78%***	62.22%***	50.00%	23.53%	32.00%	51.43%***
SVM+X_3	51.38%	77.78%	61.88%	48.39%	22.06%	30.30%	50.71%
LR+X_0	51.02%*	69.44%*	58.82%*	47.62%	29.41%	36.36%	50.00%
LR+X_1	50.52%	68.06%	57.99%	46.51%	29.41%	36.04%	49.29%
LR+X_2	50.52%	68.06%	57.99%	46.51%	29.41%	36.04%	49.29%
LR+X_3	50.52%	68.06%	57.99%	46.51%	29.41%	36.04%	49.29%
NB+X_0	48.05%	51.39%	49.66%	44.44%	41.18%	42.75%	46.43%
NB+X_1	48.68%	51.39%	50.00%	45.31%	42.65%	43.94%	47.14%
NB+X_2	48.05%	51.39%	49.66%	44.44%	41.18%	42.75%	46.43%
NB+X_3	48.68%	51.39%	50.00%	45.31%	42.65%	43.94%	47.14%

Note: Similar to the note of Table 6.

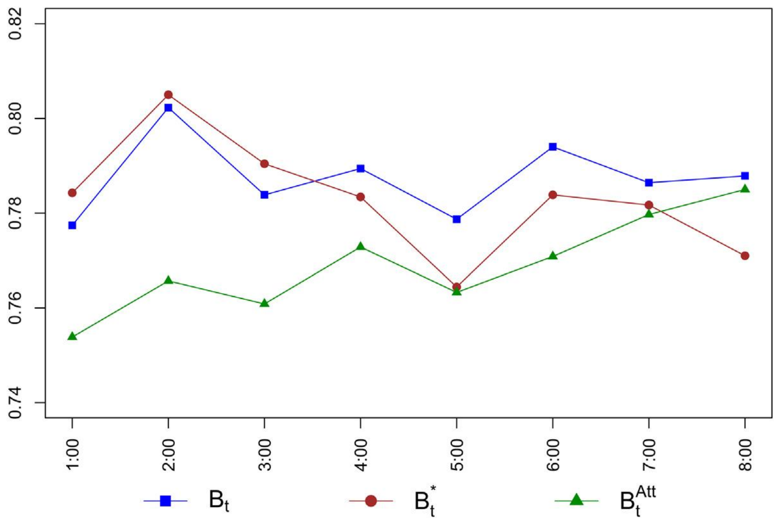
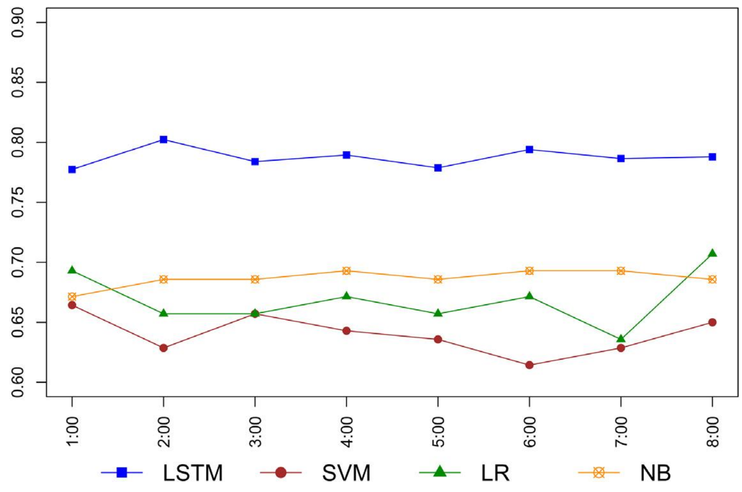


Fig. 6. Accuracy of long short-term memory (LSTM) models with different hourly investor sentiment indicators in forecasting open prices of the market index.

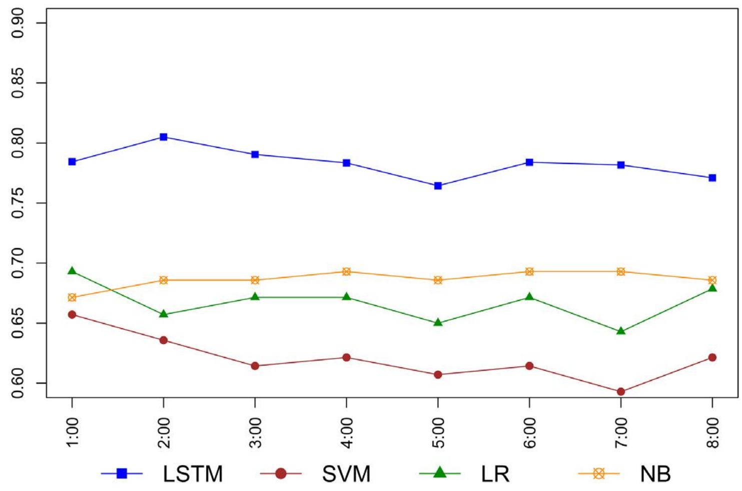
classification accuracy more balanced in each category, the amount of information in the positive and negative categories is reduced. The results in Tables 6 and 7 reveal that the three-class classification algorithm enables the classified messages in the positive and negative categories to contain more information, which is more useful for discussing the predictive power of investor sentiment. Therefore, we prefer to choose the investor sentiment indicators constructed using the three-class Naïve Bayes messages classification algorithm.

Based on the closing price forecasting results in Tables 8 and 9, the accuracy of none of the models is acceptable; the accuracy of most models is below 50%, and the accuracy for some models is only slightly above 50%. Consistent with the results of open price forecasting, the performance of the sentiment indicator constructed by the three-class Naïve Bayes message classification algorithm is better than the sentiment constructed by the multiple binary Naïve Bayes message classifications and voting algorithm. The results of the sentiment indicators constructed using all messages and only non-repeated messages are similar. From Panel A of Table 8, we determine that the best forecasting model is the logistic

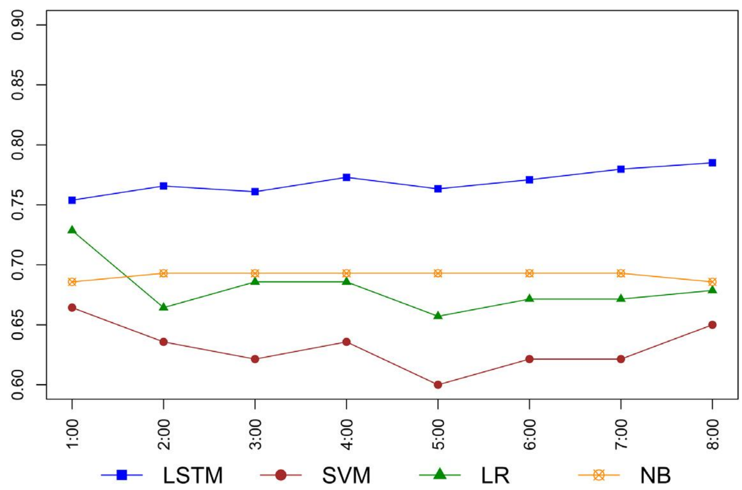
7 To conserve space, we do not report the forecasting performance of models with the daily sentiment obtained based on multiple binary classification and voting algorithm and constructed using non-repeated aggregation in forecasting closing prices of the market index, because the results are similar to that of the daily sentiment obtained based on multiple binary classification and voting algorithm and constructed using aggregating total messages. The results that are not reported here are available upon request.



(A) Using hourly investor sentiment indicator B_t



(B) Using hourly investor sentiment indicator B_t^*



(C) Using hourly investor sentiment indicator B_t^{Att}

Fig. 7. Accuracy of different models with hourly investor sentiment indicators in forecasting open prices of the market index.

Table 10
Performance of the long short-term memory (LSTM) model with hourly investor sentiment in forecasting open prices of the market index.

Panel A: Using hourly investor sentiment indicator B_t								
$t_h = 9{:}30$		Price change direction: Up			Price change direction: Down			Total
$h+1$	t_h-h-1	p	r	f	p	r	f	Accuracy
8.5	1:00	74.79%	91.06%*	80.97%	90.29%*	63.65%	71.67%	77.74%
7.5	2:00	80.82%	85.25%	81.38%*	85.92%	74.91%	77.49%	80.23%*
6.5	3:00	78.55%	86.33%	80.48%	86.86%	69.97%	73.59%	78.39%
5.5	4:00	77.84%	87.78%	81.03%	87.82%	69.59%	74.77%	78.94%
4.5	5:00	75.31%	89.81%	80.71%	89.00%	65.24%	72.54%	77.87%
3.5	6:00	77.14%	89.17%	81.56%	88.70%	69.06%	75.38%	79.40%
2.5	7:00	76.14%	89.44%	81.23%	88.42%	67.21%	73.93%	78.64%
1.5	8:00	77.14%	89.19%	81.28%	88.86%	67.76%	73.71%	78.79%
Benchmark	LSTM+X_1	85.62%*	77.81%	78.85%	81.36%	81.38%*	78.66%*	79.54%
Panel B: Using hourly investor sentiment indicator B_t^*								
$t_h = 9{:}30$		Price change direction: Up			Price change direction: Down			Total
$h+1$	t_h-h-1	p	r	f	p	r	f	Accuracy
8.5	1:00	76.86%	89.06%	81.09%	88.94%	67.18%	73.02%	78.43%
7.5	2:00	77.91%	90.06%	82.73%*	89.08%	70.38%	76.62%	80.50%*
6.5	3:00	76.75%	89.89%	81.58%	89.74%	67.56%	74.31%	79.04%
5.5	4:00	77.60%	87.78%	80.79%	87.94%	68.35%	73.26%	78.34%
4.5	5:00	73.24%	90.89%*	80.01%	89.98%	61.15%	69.60%	76.44%
3.5	6:00	75.72%	90.58%	81.29%	90.02%*	65.47%	72.76%	78.39%
2.5	7:00	76.49%	88.47%	80.54%	88.16%	67.26%	73.35%	78.17%
1.5	8:00	74.85%	89.78%	80.35%	89.02%	63.68%	70.73%	77.10%
Benchmark	LSTM+X_2	83.87%*	81.56%	80.20%	83.91%	78.76%*	78.60%*	80.20%
Panel C: Using hourly investor sentiment indicator B_t^{Att}								
$t_h = 9{:}30$		Price change direction: Up			Price change direction: Down			Total
$h+1$	t_h-h-1	p	r	f	p	r	f	Accuracy
8.5	1:00	73.83%	86.94%	78.39%	86.88%	63.15%	68.72%	75.39%
7.5	2:00	75.03%	86.75%	79.19%	86.58%	65.79%	71.09%	76.57%
6.5	3:00	73.69%	89.14%	79.55%	88.25%	62.26%	68.83%	76.09%
5.5	4:00	75.25%	88.92%	80.32%	88.17%	64.97%	71.10%	77.29%
4.5	5:00	73.20%	90.67%	79.95%	89.47%	61.15%	69.12%	76.33%
3.5	6:00	73.66%	91.69%*	80.74%*	90.38%*	61.62%	70.07%	77.09%
2.5	7:00	78.97%	84.36%	79.58%	85.18%	71.21%	73.94%	77.97%
1.5	8:00	77.62%	87.50%	80.59%	87.48%	68.97%	74.05%	78.50%*
Benchmark	LSTM+X_3	83.62%*	77.64%	76.90%	81.55%	78.06%*	76.44%*	77.84%

Notes: a. t_h is the time at which the price is to be forecasted.
b. $h+1$ represents the hours in advance that hourly investor sentiment is employed. t_h-h-1 represents the time at which the latest hourly investor sentiment is in the inputs of the model.
c. The precision, recall, and F-score are applied to evaluate the direction forecasting performance, and are recorded as p, r, and f, respectively, where $f = 2pr/(p+r)$. The overall model accuracy is also reported.
d. * represents the best forecasting model for each investor sentiment indicator.

Table 11
Accuracy of other benchmark models with hourly investor sentiment in forecasting open prices of the market index.

$t_h = 9{:}30$	B_t			B_t^*			B_t^{Att}		
t_h-h-1	SVM	LR	NB	SVM	LR	NB	SVM	LR	NB
1:00	66.43%	69.29%	67.14%	65.71%	69.29%	67.14%	66.43%	72.86%	68.57%
2:00	62.86%	65.71%	68.57%	63.57%	65.71%	68.57%	63.57%	66.43%	69.29%
3:00	65.71%	65.71%	68.57%	61.43%	67.14%	68.57%	62.14%	68.57%	69.29%
4:00	64.29%	67.14%	69.29%	62.14%	67.14%	69.29%	63.57%	68.57%	69.29%
5:00	63.57%	65.71%	68.57%	60.71%	65.00%	68.57%	60.00%	65.71%	69.29%
6:00	61.43%	67.14%	69.29%	61.43%	67.14%	69.29%	62.14%	67.14%	69.29%
7:00	62.86%	63.57%	69.29%	59.29%	64.29%	69.29%	62.14%	67.14%	69.29%
8:00	65.00%	70.71%	68.57%	62.14%	67.86%	68.57%	65.00%	67.86%	68.57%

Note: "LR" represents the logistic regression model, and "NB" represents the Naïve Bayes classification models.

regression model using B_t^* constructed by the three-class Naïve Bayes message classification algorithm, with an accuracy of 53.57%. Compared with different models, we discover that the LSTM model with inputs of daily prices and trading volume does not improve the performance in forecasting closing prices. Even incorporating the inputs of the daily sentiment indicator in the LSTM model does not improve the forecasting performance much in comparison to the best benchmark models. Moreover, the accuracy improvement in the forecasting models by

Table 12
Performance of the long short-term memory (LSTM) model with hourly investor sentiment in forecasting closing prices of the market index.

Panel A: Using hourly investor sentiment indicator B_t

$t_h = 15:00$		Price change direction: Up			Price change direction: Down			Total
$h+1$	$t_h - h - 1$	p	r	f	p	r	f	Accuracy
14	1:00	50.35%	47.00%	48.50%	47.73%	51.12%*	49.27%*	49.00%
13	2:00	50.69%	48.56%	49.31%	47.00%	50.12%	48.42%	49.31%
12	3:00	51.40%	54.92%	52.96%	48.61%	45.09%	46.60%	50.14%
11	4:00	50.79%	63.11%	56.17%	47.25%	35.18%	40.11%	49.54%
10	5:00	51.14%	59.06%	54.69%	48.04%	40.24%	43.57%	49.91%
9	6:00	50.53%	61.97%	55.59%	46.87%	35.68%	40.33%	49.20%
8	7:00	51.59%	64.67%	57.34%	48.74%	35.71%	41.09%	50.60%
7	8:00	51.75%	64.78%	57.47%	48.79%	35.91%	41.21%	50.76%
6	9:00	50.24%	62.47%	55.63%	46.18%	34.38%	39.26%	48.83%
5	10:00	50.85%	62.69%	56.01%	47.21%	35.76%	40.30%	49.61%
4	11:00	51.77%	64.64%	57.39%	48.71%	36.03%	41.16%	50.74%
3	12:00	50.97%	62.06%	55.81%	46.88%	36.85%	41.11%	49.81%
2	13:00	55.91%	76.67%*	64.53%*	59.77%*	35.91%	44.35%	56.87%*
1	14:00	54.97%*	70.78%	61.69%	56.56%	38.76%	45.35%	55.23%
Benchmark	LSTM+X_1	51.66%	51.89%	51.74%	48.82%	48.59%	48.66%	50.29%

Panel B: Using hourly investor sentiment indicator B_t^*

$t_h = 15:00$		Price change direction: Up			Price change direction: Down			Total
$h+1$	$t_h - h - 1$	p	r	f	p	r	f	Accuracy
14	1:00	49.54%	42.69%	45.41%	46.34%	54.29%*	49.86%	48.33%
13	2:00	51.11%	45.72%	48.02%	48.60%	54.00%	50.93%*	49.74%
12	3:00	51.83%	54.97%	53.18%	49.10%	45.94%	47.25%	50.59%
11	4:00	51.55%	65.36%	57.57%	48.72%	34.91%	40.51%	50.57%
10	5:00	51.03%	64.42%	56.75%	47.72%	34.53%	39.60%	49.90%
9	6:00	50.58%	68.83%	58.18%	45.62%	28.82%	35.06%	49.40%
8	7:00	51.28%	67.39%	58.16%	48.14%	32.18%	38.36%	50.29%
7	8:00	51.81%	67.44%	58.54%	49.18%	33.50%	39.69%	50.96%
6	9:00	50.64%	66.06%	57.23%	46.65%	31.74%	37.50%	49.39%
5	10:00	51.34%	65.14%	57.32%	48.30%	34.59%	40.03%	50.30%
4	11:00	52.15%	66.67%	58.43%	49.67%	35.09%	40.89%	51.33%
3	12:00	52.94%	69.22%	59.92%	51.79%	34.88%	41.45%	52.54%
2	13:00	58.46%*	86.92%	69.80%*	72.25%*	34.29%	45.76%	61.36%*
1	14:00	57.69%	87.33%*	69.38%	72.22%	32.18%	43.87%	60.54%
Benchmark	LSTM+X_2	51.77%	53.44%	52.42%	47.97%	47.26%	47.58%	50.44%

Panel C: Using hourly investor sentiment indicator B_t^{Att}

$t_h = 15:00$		Price change direction: Up			Price change direction: Down			Total
$h+1$	$t_h - h - 1$	p	r	f	p	r	f	Accuracy
14	1:00	50.14%	45.17%	46.93%	45.69%	52.68%*	48.84%	48.81%
13	2:00	52.92%	53.03%	52.60%	50.53%	50.29%	49.98%*	51.70%
12	3:00	52.50%	60.36%	55.99%	50.16%	42.15%	45.53%	51.51%
11	4:00	52.75%	64.42%	57.90%	50.81%	38.88%	43.82%	52.01%
10	5:00	50.33%	64.61%	56.46%	46.25%	32.41%	37.80%	48.97%
9	6:00	51.32%	69.19%	58.81%	48.42%	30.56%	37.08%	50.43%
8	7:00	50.96%	66.92%	57.80%	47.73%	31.88%	38.06%	49.90%
7	8:00	52.44%	68.86%	59.48%	50.75%	33.88%	40.46%	51.87%
6	9:00	51.91%	67.78%	58.69%	49.28%	33.38%	39.53%	51.07%
5	10:00	54.09%	70.56%	61.16%	53.81%	36.44%	43.23%	53.99%
4	11:00	53.81%	68.03%	59.99%	52.77%	38.03%	43.93%	53.46%
3	12:00	54.84%	71.86%	62.06%	55.90%	37.29%	44.25%	55.07%
2	13:00	60.34%*	89.28%	71.88%*	77.95%*	37.44%	49.74%	64.10%*
1	14:00	58.62%	89.97%*	70.87%	77.49%	32.56%	44.96%	62.09%
Benchmark	LSTM+X_3	51.21%	52.28%	51.70%	48.33%	47.26%	47.75%	49.84%

Notes: Similar to the note of Table 10.

incorporating the sentiment indicator is 0.5%–3.57% for the LSTM and logistic regression models. The results show that the text-extracted sentiment does have some slight predictive power for future closing prices. However, the low accuracies of the models imply that the information of the historical daily prices, trading volume, and sentiment has become stale for forecasting the closing price.

4.4. Forecasting results of models by incorporating hourly investor sentiment

As discussed in Section 4.2, we incorporate hourly investor sentiment indicators in the models to explore how many hours in advance investor sentiment indicators can provide forecasts. We compare the new models that incorporate different inputs of hourly investor sentiment indicators with the model that employs daily data of the

Table 13
Accuracy of other benchmark models with hourly investor sentiment in forecasting closing prices of the market index.

$t_h = 15{:}00$	B_t			B_t^*			B_t^{Att}		
$t_h - h - 1$	SVM	LR	NB	SVM	LR	NB	SVM	LR	NB
1:00	50.00%	51.43%	47.14%	50.00%	51.43%	47.14%	49.29%	52.14%	45.00%
2:00	50.00%	50.71%	46.43%	49.29%	51.43%	46.43%	48.57%	50.00%	45.00%
3:00	50.71%	48.57%	47.14%	50.00%	50.00%	47.14%	50.71%	48.57%	45.00%
4:00	49.29%	49.29%	47.86%	50.00%	47.86%	47.86%	51.43%	50.71%	46.43%
5:00	50.00%	49.29%	47.14%	48.57%	48.57%	47.14%	50.71%	50.71%	47.14%
6:00	50.00%	49.29%	47.14%	44.29%	47.86%	47.14%	45.00%	49.29%	47.15%
7:00	49.29%	48.57%	47.14%	45.00%	48.57%	47.14%	44.29%	48.57%	47.14%
8:00	50.00%	50.71%	45.71%	47.14%	47.86%	45.71%	45.00%	50.00%	47.14%
9:00	50.00%	49.29%	45.71%	46.43%	47.14%	45.71%	46.43%	49.29%	46.43%
10:00	49.29%	50.00%	45.71%	46.43%	54.29%	45.71%	50.71%	52.14%	46.43%
11:00	48.57%	51.43%	46.43%	47.14%	51.43%	46.43%	51.43%	50.00%	46.43%
12:00	48.57%	50.00%	45.00%	51.43%	53.57%	45.00%	54.29%	54.29%	45.00%
13:00	51.43%	61.43%	47.86%	57.86%	62.86%	47.86%	62.14%	61.43%	47.86%
14:00	51.43%	62.14%	47.86%	58.57%	65.00%	47.86%	62.86%	62.86%	47.86%

Note: “LR” represents the logistic regression model, and “NB” represents the Naïve Bayes classification model.

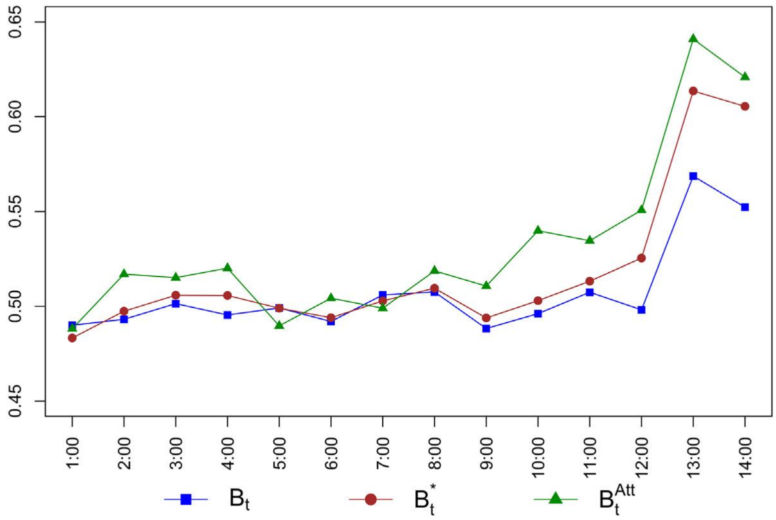
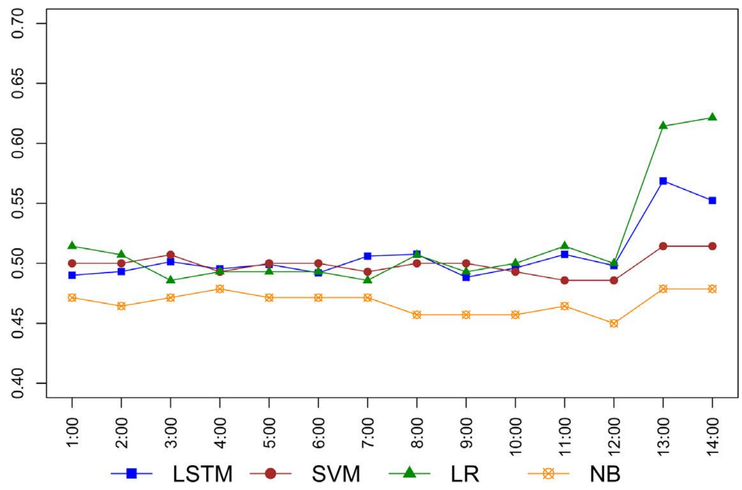


Fig. 8. Accuracy of the long short-term memory (LSTM) models with different hourly investor sentiment indicators in forecasting closing prices of the market index.

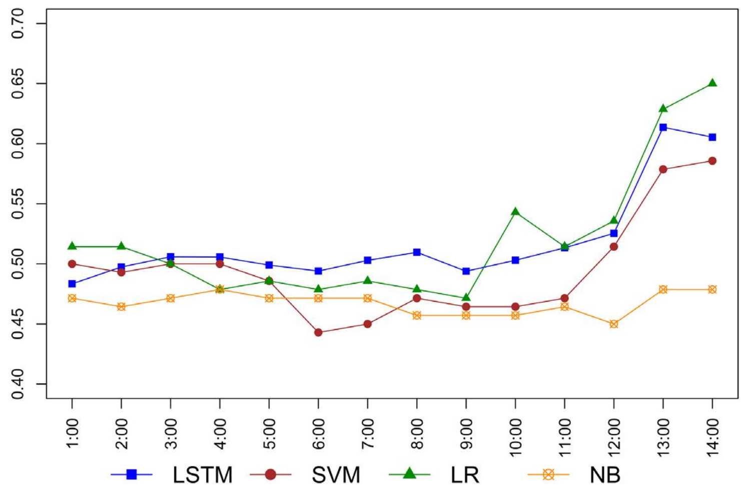
same investor sentiment indicator and other daily inputs. As illustrated in Section 4.3, the use of the three-class message classification algorithm and aggregation of all messages approach ensures that the constructed investor sentiment indicator contains more information of expectations and beliefs about future stock returns. Therefore, we adopt the investor sentiment indicators constructed by using this method for the empirical analysis on hourly sentiment.

We compare the results of the LSTM models that incorporate hourly investor sentiment with the results of the LSTM models that employ only historical daily inputs to forecast the open prices of the CSI 300 index, as reported in Table 10. We also provide the hourly performance of forecasting open prices of other benchmark models in Table 11. The results of forecasting open prices in Table 10 show that when we add hourly investor sentiment indicators several hours in advance, the recall and F-score of the “up direction,” precision of the “down direction,” and total accuracy are improved, regardless of which investor sentiment indicator is employed. The improvement of incorporating hourly sentiment indicators into the LSTM model is small—only 0.3%–0.69%—which may be due to the high accuracy of the LSTM models with only daily inputs. A comparison of the forecasting accuracy of the LSTM with other benchmark models reported in Table 11 indicates that the LSTM models provide the best forecasts for open prices, because the benchmark models’ accuracy is below 70%. The difference among various investor sentiment indicators for forecasting open prices is small, as shown in Figs. 6 and 7. As shown in Fig. 6, B_t^* provides the best accuracy for open prices at 2:00 AM, with the total accuracy reaching 80.50%. For forecasting open prices, the accuracy of the LSTM model with B_t^{Att} shows a fluctuating increasing trend over time, while the accuracy of models with B_t and B_t^* shows a similar pattern without an obvious trend over time and even with slight decrease between 2:00 AM and 5:00 AM. These results imply that the information in investor sentiment after the market closes does not substantially change.

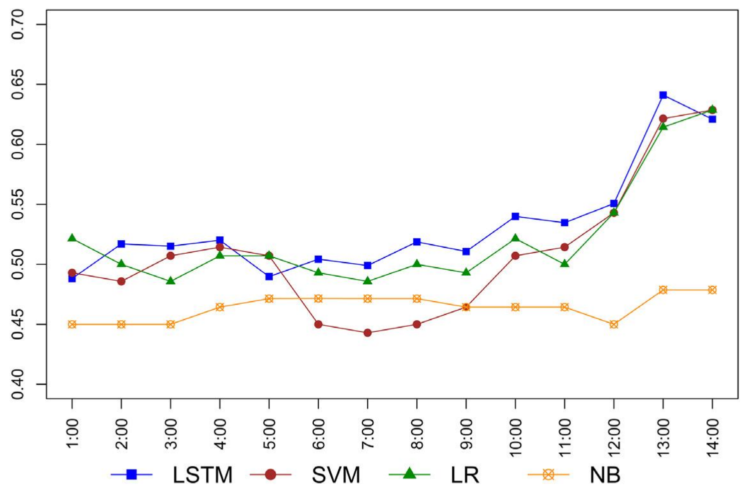
We compare the results of the LSTM models that incorporate hourly investor sentiment with the results of LSTM



(A) Using hourly investor sentiment indicator B_t



(B) Using hourly investor sentiment indicator B_t^*



(C) Using hourly investor sentiment indicator B_t^{Att}

Fig. 9. Accuracy of different models with hourly investor sentiment in forecasting closing prices of the market index.

models using only historical daily inputs in forecasting the closing prices of the CSI 300 index, as reported in Table 12. We also provide the hourly performance of forecasting closing prices of other benchmark models in Table 13. For closing price forecasts, the LSTM models that incorporate hourly investor sentiment indicators can improve all the performance measures and substantially improve the forecasting accuracy, as shown in Table 12. The LSTM model that incorporates two hours in advance sentiment can outperform the daily LSTM model, with the highest accuracy reaching 64.10% at 13:00 by using B_t^{Att}. For two hours in advance, the LSTM models that incorporate hourly sentiment can improve the forecasting accuracy by 6.58%–14.26% compared with daily LSTM models. The difference in the sentiment indicators in LSTM is very small, as shown in Fig. 8. As illustrated in Fig. 9, different models with different sentiment indicators yield different improvement effects. The results of B_t^* and B_t^{Att} are similar. The LSTM, SVM, and logistic regression models, which incorporate hourly sentiment indicator B_t^* or B_t^{Att}, can dramatically improve the accuracy of forecasting two hours in advance, while the improvement in the Naïve Bayes forecasting model is not obvious. The differences among the forecasting models with sentiment indicator B_t two hours in advance are more obvious than the other two sentiment indicators. Thus, for closing price forecasting, we prefer sentiment indicators B_t^* and B_t^{Att}. Moreover, the results show that investors incorporate new information and update expectations during trading hours, which enables the more recent investor sentiment in trading hours to have more predictive power. In Fig. 8 and Panel B and C of Fig. 9, we observe a slight increasing trend of the accuracies of the models after 9:00 and a dramatic increase at 13:00. This finding reveals that more predictive information during trading hours is incorporated in the hourly sentiment.

5. Conclusion

This study examines many aspects that could affect the price predictability of text-extracted investor sentiment in the Chinese stock market, including measurements of investor sentiment directly calculated through messages from internet stock message boards, different text mining methods for sentiment extraction, and different forecasting methods with varied forecasting horizons, model specifications, and information update schemes. Following the definition of investor sentiment in the finance field, we discover that the direct three-class Naïve Bayes messages classification algorithm performs better than the multiple binary classifications and voting algorithm in extracting expectations about future returns from user generated content, using the online posted messages from the stock message board of *Easymoney.com* in the Chinese market. Moreover, we determine that the repeated posting issue has minimal impact on our results, and the impact due to different indicator construction methods of investor sentiment is small. These results imply that the text mining algorithm for text-extracted investor sentiment has an important role in examining the predictive power of investor sentiment.

To accurately address the issue of the text-extracted investor sentiment's predictive power, we present a discussion from a deep learning perspective and compare the effectiveness of different forecasting models. Furthermore, we examine the potential influences of model specifications, forecasting horizons, and information update mechanisms. This paper obtains the following results: First, our results reveal that incorporating the text-extracted investor sentiment indicator into models can improve the forecasting accuracy according to the results of daily and hourly sentiment indicators. Second, our results uncover the differences among models and provide evidence of the superiority of the LSTM model to SVM, Naïve Bayes, and logistic regression, especially when the model inputs contain predictive information. Third, our results reveal that the information content of historical daily inputs is different for open and closing prices forecasting. We discover that the information in historical daily prices, trading volumes, and sentiment tends to be stale for forecasting the closing prices, but adequate information is contained in historical daily data for open price forecasting. Fourth, our results show that the text-extracted investor sentiment can help predict the open price in the Chinese stock market several hours earlier to some extent, and the predictive power of hourly text-extracted investor sentiment for closing price forecasting does exist but is only valid two hours earlier. These results imply that the information in investor sentiment after market close does not change substantially, whereas investors actively pay attention to the market, incorporate new information, and update their expectations during trading hours. This action leads to the greater predictive power of the more recent investor sentiment in trading hours for closing prices.

This paper reveals that the direct measurement of investor sentiment constructed by leveraging user generated messages and text mining methods has some predictive power in the Chinese stock market. By a careful comparison of different models and specifications, with the purpose of controlling the effects of the chosen forecasting method, forecasting horizon, and information update scheme, we can accurately address the predictive power of text-extracted investor sentiment. If the predictive power of investor sentiment is strong, we can expect that all models can reveal correct results despite their varied predictive abilities; however, if it is weak, some models may not be able to report correct results. When none of the models can work, investor sentiment has no predictive power, or the predictive power is too weak to be applied in practice. We do reveal the superiority of the LSTM model in detecting the predictive power of investor sentiment, which implies that deep learning methods are useful for examining the predictability of investor sentiment. This study can contribute to the literature on investor sentiment. We note that this kind of direct measurement of investor sentiment can be easily extended for individual stocks and in different frequencies. At the individual stock level, we can investigate the cross-sectional examination and link it to asset pricing issues via the direct measurement of investor sentiment, which we intend to explore in future work.

Acknowledgments

This work was supported by Prof. Wu's grants from National Key R&D Program of China [grant number 2019YFB2101804], Prof. Bu's grants from the National Natural Science Foundation of China [grant numbers 71671012, 91846108], and Prof. Wu's grants from National Natural Science Foundation of China [grant numbers 71531001, 71725002, U1636210, 71490723]. We thank the Special Issue Editors Professors Michael Clements and Ulrich Fritsche, and the anonymous referees for their valuable comments. This paper was presented and discussed at the 14th International Conference on Services Systems and Services Management (ICSSSM 2017). We thank the conference participants for their helpful comments on this paper.

References

Antweiler, W., & Frank, M. Z. (2004). Is all that talk just noise? The information content of internet stock message boards. *The Journal of Finance*, *59*(3), 1259–1294.

Baker, M., & Wurgler, J. (2006). Investor sentiment and the cross-section of stock returns. *The Journal of Finance*, *61*(4), 1645–1680.

Baker, M., & Wurgler, J. (2007). Investor sentiment in the stock market. *Journal of Economic Perspectives*, *21*(2), 129–152.

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS One*, *12*(7), Article e0180944.

Bengio, Y., Simard, P. Y., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166.

Berger, D., & Turtle, H. J. (2012). Cross-sectional performance and investor sentiment in a multiple risk factor model. *Journal of Banking & Finance*, *36*(4), 1107–1121.

Black, F. (1986). Noise. *The Journal of Finance*, *41*(3), 528–543.

Brown, G. W., & Cliff, M. T. (2004). Investor sentiment and the near-term stock market. *Journal of Empirical Finance*, *11*(1), 1–27.

Bu, H., & Pi, L. (2014). Does investor sentiment predict stock returns? The evidence from chinese stock market. *Journal of Systems Science and Complexity*, *27*(1), 130–143.

Bu, H., Xie, Z., Li, J. H., & Wu, J. J. (2018). Investor sentiment extracted from internet stock message boards and its effect on chinese stock market. *Journal of Management Science in China*, *21*(4), 86–101, (in Chinese).

Chen, H., De, P., Hu, Y. J., & Hwang, B. H. (2014). Wisdom of crowds: The value of stock opinions transmitted through social media. *Review of Financial Studies*, *27*(5), 1367–1403.

Da, Z., Engelberg, J., & Gao, P. (2011). In search of attention. *The Journal of Finance*, *66*(5), 1461–1499.

Das, S. R., & Chen, M. Y. (2007). Yahoo! for Amazon: Sentiment extraction from small talk on the web. *Management Science*, *53*(9), 1375–1388.

De Long, J. B., Shleifer, A., Summers, L. H., & Waldmann, R. J. (1990). Noise trader risk in financial markets. *Journal of Political Economy*, *98*(4), 703–738.

Ding, X., Zhang, Y., Liu, T., & Duan, J. (2015). Deep learning for event-driven stock prediction. In *IJCAI* (pp. 2327–2333).

Dixon, M., Klabjan, D., & Bang, J. H. (2016). Classification-based financial markets prediction using deep neural networks. *Algorithmic Finance*, 1–11, (Preprint).

Feuerriegel, S., & Fehrer, R. (2016). Improving decision analytics with deep learning: The case of financial disclosures. In *ECIS* (pp. Research-in).

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256).

Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.

Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (Icassp), 2013 Ieee international conference on* (pp. 6645–6649). IEEE.

Hamid, S. A., & Iqbal, Z. (2004). Using neural networks for forecasting volatility of S & P 500 index futures prices. *Journal of Business Research*, *57*(10), 1116–1125.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

Hochreiter, S. (1991). *Untersuchungen Zu Dynamischen Neuronalen Netzen* Diploma, Technische Universität München, 91(1).

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

Hribar, P., & McInnis, J. (2012). Investor sentiment and analysts' earnings forecast errors. *Management Science*, *58*(2), 293–307.

Joseph, K., Wintoki, M. B., & Zhang, Z. (2011). Forecasting abnormal stock returns and trading volume using investor sentiment: Evidence from online search. *International Journal of Forecasting*, *27*(4), 1116–1127.

Kamijo, K. I., & Tanigawa, T. (1990). Stock price pattern recognition-a recurrent neural network approach. In *Neural networks, 1990. 1990 IJCNN international joint conference on* (pp. 215–221). IEEE.

Kim, S. H., & Kim, D. (2014). Investor sentiment from internet message postings and the predictability of stock returns. *Journal of Economic Behaviour and Organization*, *107*, 708–729.

Kohzadi, N., Boyd, M. S., Kermanshahi, B., & Kaastra, I. (1996). A comparison of artificial neural network and time series models for forecasting commodity prices. *Neurocomputing*, *10*(2), 169–181.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Li, X. R., Shang, W., & Wang, S. Y. (2018). Text-based crude oil price forecasting: A deep learning approach. *International Journal of Forecasting*, preprint.

Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining text data* (pp. 415–463). Boston, MA: Springer.

Shen, F., Chao, J., & Zhao, J. (2015). Forecasting exchange rate using deep belief networks and conjugate gradient method. *Neurocomputing*, *167*, 243–253.

Shleifer, A., & Summers, L. H. (1990). The noise trader approach to finance. *Journal of Economic Perspectives*, *4*(2), 19–33.

Sirignano, J. 2016. *Deep learning for limit order books. SSRN working paper*.

Sundermeyer, M., Schlüter, R., & Ney, H. LSTM neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

Takeuchi, L., & Lee, Y. Y. A. (2013). *Applying deep learning to enhance momentum trading strategies in stocks*: *Technical report*, Stanford University.

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, *62*(3), 1139–1168.

Tetlock, P. C., Saar-Tsechansky, M., & Macskassy, S. (2008). More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, *63*(3), 1437–1467.

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, *4*(2), 26–31.

Tino, P., Schittenkopf, C., & Dorffner, G. (2001). Financial volatility trading using recurrent neural networks. *IEEE Transactions on Neural Networks*, *12*(4), 865–874.

Wang, J., Zhang, Y., Tang, K., Wu, J., & Xiong, Z. (2019). AlphaStock: A buying-winners-and- selling-losers investment strategy using interpretable deep reinforcement attention networks. The *25th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1900–1908).

Xiong, R., Nichols, E. P., & Shen, Y. (2015). Deep learning stock volatility with google domestic trends. arXiv preprint arXiv:1512.04916.

Zhao, J., Dong, L., Wu, J., & Xu, K. (2012). Moodlens: an emoticon-based sentiment analysis system for chinese tweets. In *Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1528–1531). ACM.

Zhao, Y. Y., Qin, B., & Liu, T. (2010). Sentiment analysis. *Journal of Software*, *21*(8), 1834–1848.