mathematics

MDPI

Article

Incorporating Multi-Source Market Sentiment and Price Data for Stock Price Prediction

Kui Fu and Yanbin Zhang *

School of Economics, Wuhan University of Technology, Wuhan 430070, China; fukui@whut.edu.cn
* Correspondence: 285070@whut.edu.cn

Abstract: The problem of stock price prediction has been a hot research issue. Stock price is influenced by various factors at the same time, and market sentiment is one of the most critical factors. Financial texts such as news and investor comments reflect investor sentiment in the stock market and influence market movements. Previous research models have struggled to accurately mine multiple sources of market sentiment information originating from the Internet and traditional sentiment analysis models are challenging to quantify and combine indicator data from market data and multi-source sentiment data. Therefore, we propose a BERT-LLA stock price prediction model incorporating multi-source market sentiment and technical analysis. In the sentiment analysis module, we propose a semantic similarity and sector heat-based model to screen for related sectors and use fine-tuned BERT models to calculate the text sentiment index, transforming the text data into sentiment index time series data. In the technical indicator calculation module, technical indicator time series are calculated using market data. Finally, in the prediction module, we combine the sentiment index time series and technical indicator time series and employ a two-layer LSTM network prediction model with an integrated attention mechanism to predict stock close price. Our experiment results show that the BERT-LLA model can accurately capture market sentiment and has a strong practicality and forecasting ability in analyzing market sentiment and stock price prediction.

Keywords: BERT; sentiment analysis; long short-term memory; attention mechanism; stock price prediction

MSC: 68T50; 68T07; 91B84

check for updates

Citation: Fu, K.; Zhang, Y. Incorporating Multi-Source Market Sentiment and Price Data for Stock Price Prediction. *Mathematics* **2024**, *12*, 1572. https://doi.org/10.3390/math12101572

Academic Editor: Liping Liu

Received: 13 April 2024
Revised: 11 May 2024
Accepted: 13 May 2024
Published: 17 May 2024

1. Introduction

According to the efficient market hypothesis [1], stock prices already incorporate all available valuable information, which means that analyzing stock prices using historical price data is not feasible, and this view suggests that using fundamental analysis or technical indicators to predict stock prices may not be effective in an efficient market. However, many pieces of evidence also suggest the opposite view. For example, Pedersen's study shows that those who are better at processing information have an edge in stock market investment [2]. However, due to the random volatility of financial time series, it is difficult for researchers to comprehensively analyze their characteristics to make accurate forecasts, and how to comprehensively analyze the market information to make more accurate forecasts has become an ongoing issue in the field of stock price forecasting.

In the early stages of research, scholars predominantly employed conventional statistical models, including autoregressive moving average (ARMA), autoregressive integrated moving average model (ARIMA), autoregressive conditional heteroskedasticity model (GARCH), etc. These classical statistical models maintain substantial relevance in contemporary predictive research. For example, Rounaghi and Nassir Zadeh applied the ARMA model to forecast monthly and yearly stock return time series in the S&P 500 and London Stock Exchange [3]. Herwartz employed the GARCH model to predict stock returns and obtained useful information for signaling one-step-ahead directions of stock price changes through independence testing [4].

However, with the evolution and diversification of financial markets, the complexity of financial time series has increased, rendering traditional econometric models seemingly inadequate for contemporary research. In order to adapt to higher data precision and complexity, machine learning models have been employed in the research of predicting financial time series. Traditional works often use models such as support vector machines (SVM), artificial neural networks (ANN), random forest (RF), and extreme gradient boosting (XGBoost). For instance, Qiu et al. adapted an artificial neural network to predict the return of the Japanese Nikkei 225 index and the result outperformed the traditional BP training algorithm [5]. Zhou et al. proposed a novel approach that integrates complete ensemble empirical mode decomposition with adaptive noise and XGBoost to forecast crude oil prices [6].

With the continuous breakthroughs in computing power and data capacity, an increasing number of studies are employing deep learning models for the prediction of financial time series [7,8]. Many research indicates that deep neural networks can better handle financial time series, especially the long short-term memory (LSTM) network introduced by Hochreiter and Schmidhuber in 1997 [9]. The application of the LSTM in stock price prediction and financial forecasting research has further elevated the study of deep learning in the financial domain. For example, Wu et al. applied LSTM and its variant models to predict Bitcoin prices [10], and Kim and Won proposed a combined LSTM model to predict the volatility of financial markets [11]. Up until the present moment, LSTM remains one of the most extensively utilized technologies in the field of time series prediction, and it continues to harbor significant untapped potential.

Applying deep learning to the field of financial time series prediction, the selection of model input features is one of the most crucial issues. The choice of input features is directly related to the model's ability to better learn the inherent correlations between time series. In previous works, the input features of the models typically included stock volume and price data. For instance, Barua and Sharma introduced technical indicators based on market data and used a CNN-BiLSTM model to predict index close prices [12]. Wang, W.Y. et al. constructed multiple input features using price data and selected the optimal combination of input features for prediction [13].

Recent studies aim to improve and diversify the selection of input features. Especially with the development of natural language processing technologies, data collection and processing methods are becoming increasingly diversified. Researchers are no longer limited to analyzing stock fundamental information and technical indicators. The study of market sentiment is receiving increasingly more attention. Researchers are beginning to collect text information, especially finance market news, to analyze the stock market. Many studies have shown the effectiveness of this approach for predicting stock trends (e.g., [14,15]). Results of previous works (e.g., [15–18]) suggest that using both market data and news-based information is helpful for the market prediction problem.

Researchers are not only confined to the sentiment of news, the analysis of retail investors' sentiment derived from social media has also become a focal point. For example, Poongodi et al. developed a tweet node algorithm to construct a network of tweet nodes, aiming to extract potential associations in Twitter data for stock market prediction [16]. Poongodi et al. analyzed the typical trends in the online communities and social media platforms to understand and extract insights that could be used to predict the cryptocurrency price movement trends [17]. However, there is still room for improvement in enriching the data sources for sentiment analysis and refining and standardizing market sentiment analysis methods.

Regarding technical applications, previous sentiment analysis in financial markets relied more on manually annotated dictionaries to analyze the sentiment of financial texts [18,19]. With the development of deep learning, many deep learning models have been applied to text analysis and achieved significant results. For example, Daudert introduced an adaptive feedforward neural network that utilizes recorded text and contextual information for fine-grained sentiment analysis [20]. Jing et al. used a CNN-based sentiment analysis model for sentiment analysis of financial texts [21].

The transformer model in particular, due to its capability in capturing long-range dependencies and thus analyzing semantics more effectively, has significantly propelled the development of natural language processing technologies. In recent research, transformer-based natural language processing methods have shown promising results in financial text data analysis. Particularly Google's BERT model [22], as a transformer-based pre-trained model, made remarkable progress in natural language processing and was applied to sentiment analysis of financial texts in many studies (e.g., [23,24]). For instance, Hiew et al.'s study shows that a BERT-based sentiment analysis approach is superior to models such as FastText or a multichannel Convolutional Neural Network (CNN) [25]. However, there is still significant room for research and exploration of BERT's application in the financial market.

Regarding analysis methods, previous works on market sentiment mainly focused on sentiment classification. Based on existing techniques for sentiment polarity analysis, text sentiment is classified into positive, neutral, or negative categories, and the number of texts with different sentiment tendencies is used to calculate sentiment scores as model inputs for stock price prediction [26].

Although there have been many attempts to apply sentiment analysis to price prediction, current research still has several shortcomings. Previous works on market sentiment mainly focused on sentiment polarity (positive/negative/neutral expression), much research has expanded on this foundation. For example, Chou split news headlines into words and then analyzed the sentiment polarity of each word to calculate sentiment scores for stock price prediction [27]. Cristescu et al. analyzed the sentiment polarity of news headlines and used a regression model to predict prices [28]. These methods resulted in an inevitable loss of data accuracy and have a significant limitation. Moreover, most existing research focused more on the market sentiment of the target stock and ignored the sentiment impact of its related sectors. For example, Fazlija and Harder only used news related to an underlying asset to construct sentiment indicators for stock price trend prediction [29]. Deng et al. only used investor sentiment related to an underlying asset for prediction [30]. In addition, previous research mainly used single news or post data sources, which are relatively limited (e.g., [26,31]). Furthermore, retail investors account for a large percentage of the stock market, and existing research has largely ignored the impact of this group on market sentiment. How to extract market sentiment information more accurately and comprehensively and make more accurate stock price predictions based on sentiment information is an essential issue in current research.

To address these issues, we propose the BERT-LLA model, which combines sentiment analysis with technical indicators. Following Li, Q. et al. [32], Nassirtoussi et al. [33], and Wang, H. et al. [34], we combine news and investor reviews for market sentiment analysis, while using financial texts from upstream and downstream industries to form multi-channel data. We also propose a comprehensive sentiment index calculation method for combining news and investor comments. We leverage the BERT model for sentiment analysis and calculate the sentiment index series and the technical indicator time series for model prediction. The main contributions of this research are:

- We propose a prediction model called BERT-LLA that leverages a pre-trained model for financial sentiment analysis and outperforms the baselines in test sets.
- We propose a comprehensive sentiment index calculation method for combining news and investor comments to standardize the use of these two types of text information.
- We consider the impact of market sentiment in the company's upstream and downstream sectors and propose combining the sentiment of the upstream and downstream sectors for stock price prediction, which can solve the problem of a relatively limited source of text data.

- We propose a related sector selection method based on semantic similarity and sector heat, which can help us screen related sectors for stock price prediction more intelligently and effectively.
- We analyze the impact of the weight of investor comments and news on prediction accuracy and confirm our experience that news has a more substantial effect on market sentiment. We also obtain the relatively optimized values of the weight, which have enlightening significance for subsequent research on the synergistic effect of investor sentiment and news on market sentiment.

The rest of the paper is organized as follows: Section 2 describes our proposed model and corresponding details. Section 3 describes the experimental design and presents the experimental results and discussions. Section 4 summarizes our work and points out future directions for research.

2. Methodology

Our work primarily focuses on leveraging market sentiment and price data for financial time series prediction. As such, we first define the concept of sentiment index time series and the technical indicator time series as follows.

Definition 1. *We define the sentiment index time series* $\{S_t\}$ *as a* $2N$*-dimensional vector,* $(S_{1,t|pos}, S_{1,t|neg}, \ldots S_{N,t|pos}, S_{N,t|neg})'$*. The sentiment index time series* S_t *is constructed based on the sentiment scores of news and investor comment texts, which will be explained in Section 3.1. The value of* N *depends on the number of upstream and downstream sectors we select.*

Definition 2. *We define the technical indicator time series* $\{Q_t\}$ *as a* M*-dimensional vector,* $Q_t = (Q_{1,t}, Q_{2,t}, Q_{3,t}, \ldots, Q_{M,t})', Q_M \in Q$*, where* Q *represents the important technical indicators calculated from the original market data.*

The framework of our model is shown in Figure 1. Our model consists of three parts, and the details of each part are explained in the following subsections.

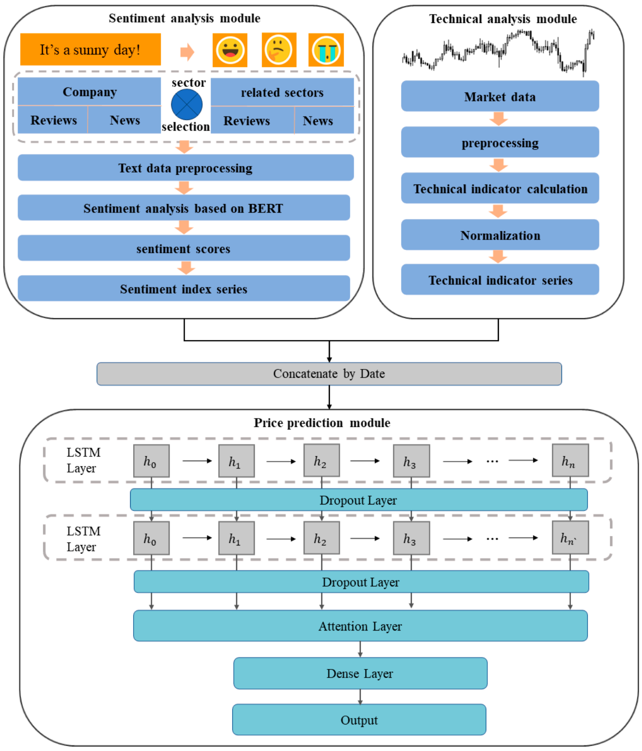


Figure 1. The structure of our stock price prediction model.

2.1. *Sentiment Analysis Module*

2.1.1. Sentiment Analysis Based on BERT

The BERT (Bidirectional Encoder Representation from Transformers) model was proposed by Google AI Research in October 2018 [22]. It is a pre-training model that has achieved significant milestones in natural language processing (NLP). Compared to traditional context-free embedding models, the BERT model is based on contextual embedding, allowing it to understand text data in the context better and extract key contextual relationships.

As shown in Figure 2, the sentiment analysis model based on BERT consists of multiple transformer blocks, which are used to extract features from the input vectors. For the BERT model, the model input was obtained by combining three parts: token embeddings, segment embeddings, and position embeddings. After receiving the word vectors of the input text and feeding them into BERT, the model performed two pre-training tasks: masked language modeling (MLM) and next sentence prediction (NSP). After pre-training, the BERT model can be fine-tuned for downstream tasks based on the specific requirements of the task.

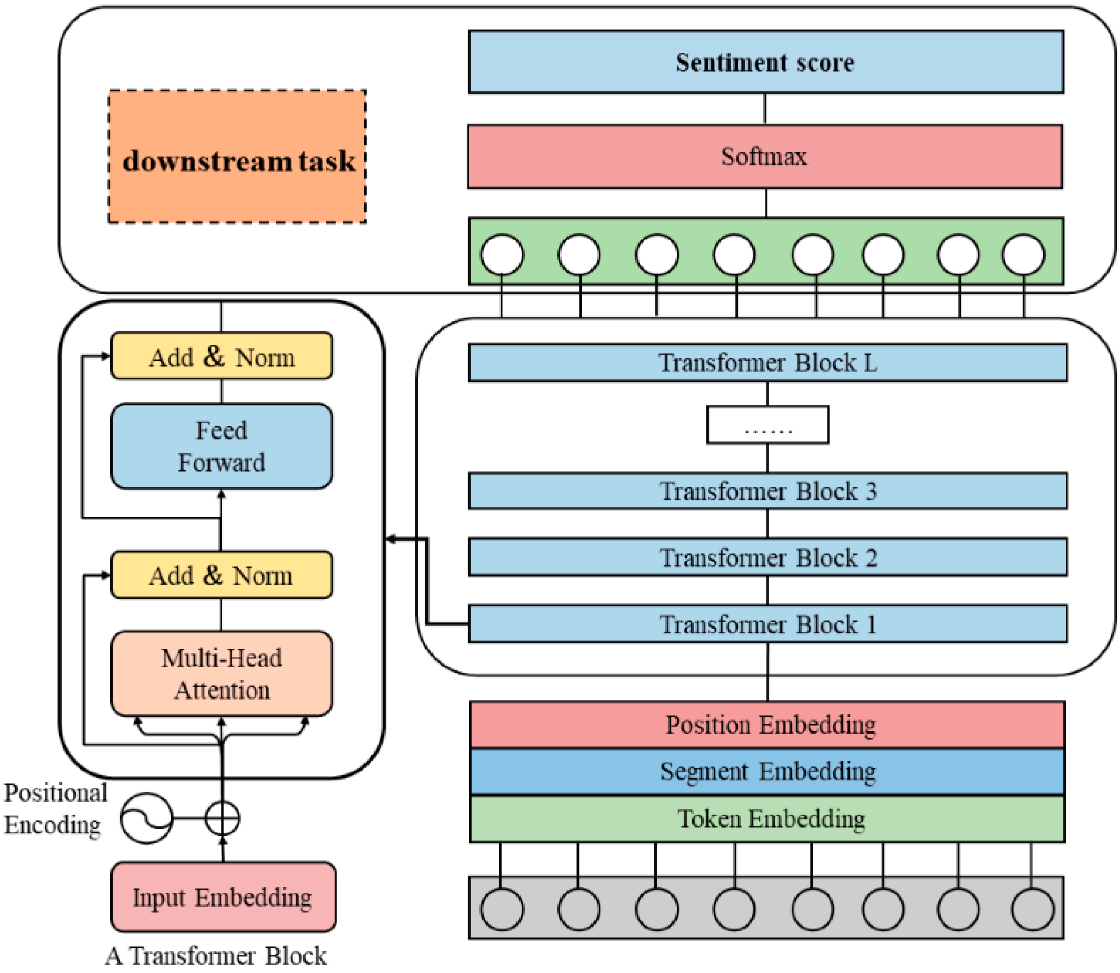


Figure 2. Sentiment analysis based on BERT.

The downstream task of this paper was to calculate sentiment scores based on unstructured financial text data. Based on this, we propose a method to fit the feature vectors output by BERT into the interval [0,1] using the softmax function to represent sentiment polarity. This sentiment polarity represents the probability of the text being positive or negative, and these probabilities were used as sentiment scores for news and investor comments to calculate the sentiment index further. In order to achieve better results, we used three BERT models: original BERT, RoBERTa, and FinBERT, and selected the best model based on prediction performance.

The core mechanism of the BERT model is the multi-headed self-attention mechanism, which serves to select the information that is more critical to the current task goal from a large amount of information. The multi-headed attention mechanism is calculated as follows:

- The input text is transformed into an embedding vector, and the embedding vector is multiplied by three matrices: W^Q, W^K, and W^V to obtain the word-embedding representations of query, key, and value, denoted as Q, K, and V.
- To calculate the attention value, the formula is as follows:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{1}$$

where d_k denotes the vector dimension and $1/\sqrt{d_k}$ is a scaling factor to prevent the point multiplication result from being too large to affect the back propagation of the gradient.

2.1.2. Sentiment Index Calculation

The calculation of the sentiment index in this paper involves the calculation of the sentiment index for both the company itself and its upstream and downstream sectors. The sentiment of the upstream and downstream sectors also impacts the company's stock price. Therefore, we introduced the sentiment index of the upstream and downstream sectors as parameters into the model to participate in stock price prediction.

The sentiment index was calculated based on the sentiment scores of news and investor comments. From experience, the sentiment expressed in news and the sentiment expressed in investor comments have different strengths of impact on the stock market. Therefore, we assigned different weights, w_1 and w_2, to the sentiment scores of news and investor comments, respectively.

In the experimental section, we discuss the impact of the values of w_1 and w_2 on the final prediction results. The initial values were set as $w_1 = w_2 = 0.5$ for ease of calculation.

The formula for calculating the sentiment index is as follows:

$$S_t = w_1 \times News\_score_t + w_2 \times Review\_score_t \tag{2}$$

The sentiment score of news is as follows:

$$News\_score_t = \sum\nolimits_{i=1}^{n} news\_score_{i,t}, \tag{3}$$

where n is the number of news articles on the t-th trading day.

The sentiment score of comments is as follows:

$$Review\_score_t = \sum\nolimits_{i=1}^{m} review\_score_{i,t}, \tag{4}$$

where m is the number of comments on the t-th trading day.

2.1.3. Selection Method of Related Sectors

Since we use NIO and Tesla as our experimental dataset, we drew a schematic diagram of the industry chain of the new energy automobile industry, as shown in Figure 3, considering that we introduce the sentiment of the associated industries into the prediction model. It can be seen that the new energy vehicle industry chain can be divided into four parts: upstream, midstream, downstream, and post-market, and our experimental dataset belongs to the downstream industry of new car-making forces.

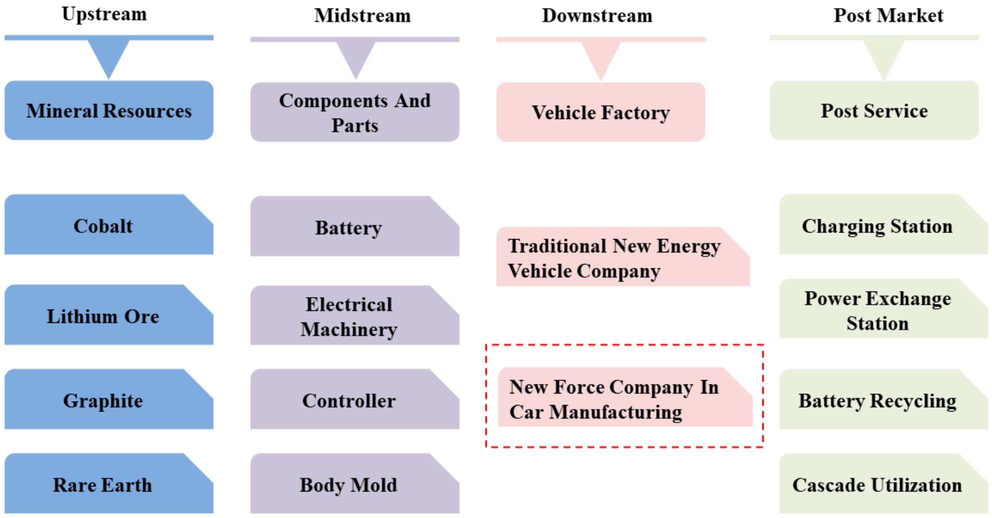


Figure 3. The industry chain of the new energy automobile industry. The red dashed line represents the industry of the experimental target.

At the same time, it is evident that there are many upstream- and downstream-related sectors. To select the most related industry sectors to assist us in extracting market sentiment more accurately for stock price prediction, we propose a correlation calculation method based on semantic similarity and sector heat factor, as shown in Figure 4 below.

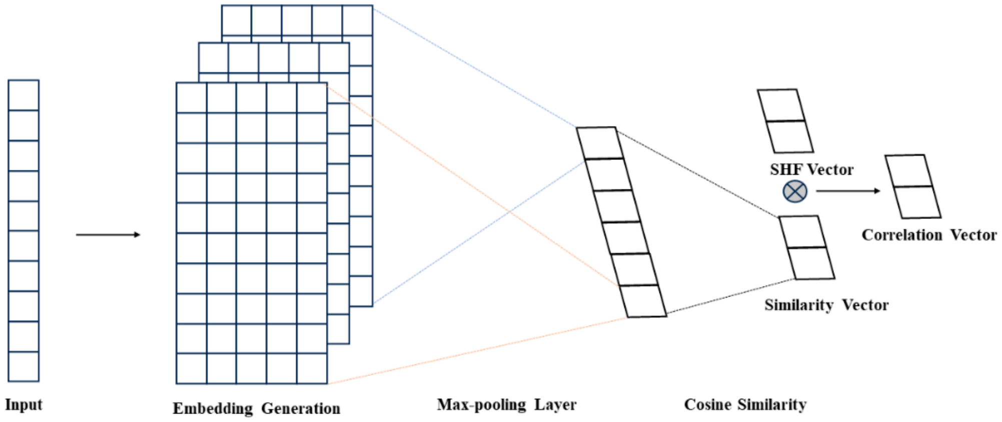


Figure 4. Semantic similarity-based correlation coefficient calculation process.

The process is as follows:

Input: The model's inputs are the texts of the financial sectors involved in the industry chain.

Embedding Generation: The embedding generation procedure used a BERT-based semantic vector generation approach, where we used a Sentence-BERT model that had already been trained on a large semantically similar dataset for vector generation, ensuring that the loss of semantic information in the generated vectors was minimized.

Max-pooling Layer: A max-pooling layer addresses the most essential features by pooling over every feature map bearing a close resemblance to the feature selection process.

Cosine Similarity: We computed the cosine similarity based on the pooled semantic feature vectors to obtain the semantic similarity vector. The cosine similarity between them can be calculated using Equation (5).

$$\cos(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1| \times v_2} \tag{5}$$

At the same time, we considered that superficial semantic similarity was not enough to represent the correlation relationship in the stock market, and the market heat between different sectors in different periods affects the sector correlation relationship to a greater extent, and sectors with higher heat tend to have a more significant impact on the sector correlation. Based on this, we propose Sector Heat Factor (SHF) to quantify sector heat, which is calculated using the number of news and investor comments collected during the same period, using the following formula:

$$SHF = Gaussian\left(\frac{num(News + Review) - minNum}{maxNum - minNum}\right), \tag{6}$$

where *minNum* is the minimum value of the total number of news and investor comments in the same period board, and *maxNum* is the maximum value. Considering the discontinuities in the data, we fit data with a Gaussian function, which can make the data fall within a reasonable graphical interval. The specific formula is as follows:

$$Gaussian = ae^{-(x-b)^2/2c^2}, \tag{7}$$

where a, b, and c are constants and $a > 0$, the parameter a refers to the peak of the Gaussian curve, b is its corresponding horizontal coordinate, and c is the standard deviation. In this paper, we set $a = 0.8,\ b = 1,\ c = 0.6$.

2.2. Technical Indicators Calculation Module

2.2.1. Selection and Calculation of Technical Indicators

In stock trading, the most popular technical indicators include trend indicators, momentum indicators, and volume indicators [35]. These technical indicators are transformed into specific features through different parameter settings. We select a total of 12 technical indicators from these three categories. These indicators, with varying configurations of parameters such as time length, reflect the trends and fluctuations of stocks from different aspects, which can provide rich market signals for the prediction model. The specific selection of technical features is shown in Table 1. Due to different time length configurations, different technical indicators start at different time points. Therefore, we retain the time with all technical indicators while omitting the time with missing values to obtain the technical indicator time series $\{Q_t\}$.

Table 1. Technical indicators for stock prices.

Type	Technical Indicators	Abbreviation
Trend indicators	Moving average (10)	MA (10)
	Moving average (20)	MA (20)
	Exponential moving average (10)	EMA (10)
	Exponential moving average (20)	EMA (20)
	Moving Average Convergence/Divergence (6,15,6)	MACD (6,15,6)
	Moving Average Convergence/Divergence (12,26,9)	MACD (12,26,9)
Momentum indicators	Relative strength index (6)	RSI (6)
	Relative strength index (12)	RSI (12)
	William's %R (14)	WILLR (14)
	Momentum index (14)	MOM (14)
Volume indicators	On Balance Volume	OBV
	Chaikin A/D Oscillator (3,10)	ADOSC (3,10)

2.2.2. Data Processing

The data we use is complete and standardized. We only need to normalize the technical indicators to balance the differences caused by different scales and units among the features. This ensures all features conform to the same data distribution during model training.

We use the min–max normalization method [36] to map the feature data to the range $[0,1]$, the specific formula is as follows:

$$Q^* = \frac{Q - Q_{min}}{Q_{max} - Q_{min}}, \tag{8}$$

where Q is the original data value, Q^* is the normalized value, Q_{min} is the minimum value of the original data, and Q_{max} is the maximum value of the original data.

De-normalization:

$$P = P^*(P_{max} - P_{min}) + P_{min}, \tag{9}$$

where P is the de-normalized predicted value of the model, P^* is the model's prediction value, P_{min} is the minimum value of the original data, and P_{max} is the maximum value of the original data.

2.3. Stock Price Prediction Module

2.3.1. Input Feature Preparation

In Algorithm 1, we introduced the preparation process of input features for the prediction model. In the previous section, we have described the concept of technical indicator time series $\{Q_t\}$ and sentiment index time series $\{S_t\}$. In this part, we describe the specific calculation process of the input features. $P(N_j) \leftarrow softmax(W * N_j + b), P(R_j) \leftarrow softmax(W * R_j + b)$ are the predicted probabilities of news text and investor comment text under the positive and negative categories, respectively. W is the weight vector of the BERT model, and b is the deviation value. We used a text dataset with sentiment labels to fine-tune the BERT model by minimizing the loss function. Finally, the two types of indicators were merged and combined into a composite matrix, and then we fed the matrix into a prediction model for model training.

Algorithm 1 Feature Preparation in Our Prediction Model

Input: N_j is news sequence, and R_j is investors' review sequence during h trading days. $Q_{(t-h,t)} \in Q^{M*h}$ is technical indicator data during h trading days from M technical indicators of space Q.
Output: Matrix $X_{(t-h,t)}$ from concatenation of technical indicator and sentiment index time series.

```
for $i \in [1, h]$ do:
    $News\_score_{t_i},\ Review\_score_{t_i} \leftarrow 0$
    for $j \in [0, N-1]$ do:
        while $t_i < t(N_j), t(R_j) < t_{i+1}$ do
            $P(N_j) \leftarrow softmax(W * N_j + b)$
            $P(R_j) \leftarrow softmax(W * R_j + b)$
            $News\_score_{j,t_i|pos,neg} \leftarrow News_{score\, j,t_i|pos,neg} + P_{pos,neg}(N_j)$
            $Review\_score_{j,t_i|pos,neg} \leftarrow Review_{score\, j,t_i|pos,neg} + P_{pos,neg}(R_j)$
            $S_{j,t_i|pos} \leftarrow w_1\text{News\_score}_{j,t_i|pos} + w_2\text{Review\_score}_{j,t_i|pos}$
            $S_{j,t_i|neg} \leftarrow w_1\text{News\_score}_{j,t_i|neg} + w_2\text{Review\_score}_{j,t_i|neg}$
            $j \leftarrow j+1$
        end while
    end for
    $X_{t_i} \leftarrow \left[Q_{t_i}, S_{1,t_i|pos}, S_{1,t_i|neg}, \ldots S_{N,t_i|pos}, S_{N,t_i|neg}\right]$
    $i \leftarrow i+1$
end for
return $X_{(t-h,t)} = \{X_{t_i} | i = 1, \ldots h\}$
```

2.3.2. Long Short-Term Memory

RNNs are good at handling time series data but suffer from issues such as vanishing or exploding gradients when dealing with long sequences. LSTM, a variant of RNN, is specifically designed to learn long-term dependencies. Its gate structure allows for controlling the information flow, effectively avoiding the problems of vanishing or exploding gradients in long sequences.

LSTM consists of an input, output, and hidden layer. The main features of LSTM are contained in the hidden layer called memory cells. Each cell has a structure of three gates to maintain and adjust its cell state (C_t). The structure of a memory cell is shown in Figure 5.

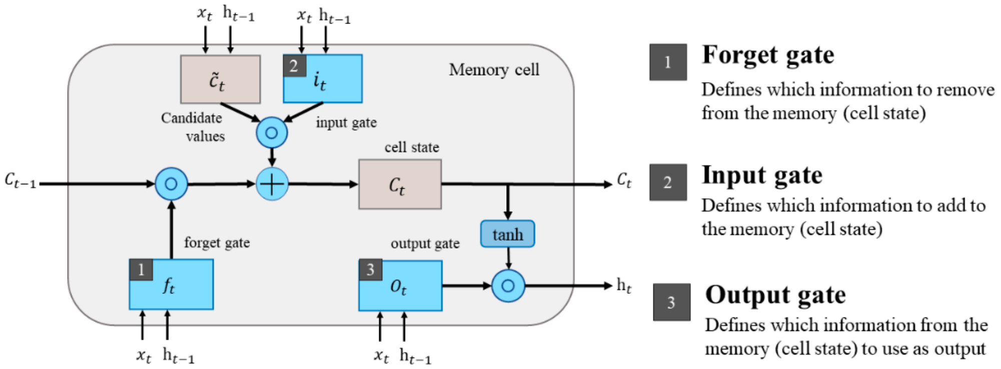


Figure 5. The cell structure of LSTM.

The learning process of each LSTM unit can be represented by the following procedure.

In the first step, the forget gate decides which information will be discarded from the previous cell state. The output vector is calculated based on the input vector x_t at the current moment, the output h_{t-1} of the memory cell at moment $t-1$, and the bias vector b_f.

$$f_t = sigmoid\left(W_{f,x}x_t + W_{f,h}h_{t-1} + b_f\right) \tag{10}$$

In the second step, the LSTM decides which information should be stored in the cell state C_t. This step consists of two operations:

- Calculating the candidate value $\widetilde{c}_t$, which may potentially be added to the cell state C_t.
- Calculating the activation value i_t of the input gate.

$$\widetilde{C}_t = \tanh\left(W_{\widetilde{c},x}x_t + W_{\widetilde{c},h}h_{t-1} + b_{\widetilde{c}}\right) \tag{11}$$

$$i_t = sigmoid(W_{i,x}x_t + W_{i,h}h_{t-1} + b_i) \tag{12}$$

In the third step, the new cell state C_t is calculated based on the results of the previous two steps.

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{13}$$

In the last step, the output of the memory cells is calculated according to the following two formulas:

$$o_t = sigmoid(W_{o,x}x_t + W_{o,h}h_{t-1} + b_o) \tag{14}$$

$$h_t = o_t * \tanh(C_t) \tag{15}$$

where $W_{f,x}, W_{f,h}, W_{\widetilde{c},x}, W_{\widetilde{c},h}, W_{i,x}$, $W_{i,h}, W_{o,x}$ and $W_{o,h}$ are weight vectors, b_f, $b_{\widetilde{c}}$ and b_o are bias vectors, f_t, i_t and o_t are the activation values of the respective gates.

2.3.3. Dropout

Overfitting is an essential issue in neural network training. In machine learning models, if a model has a large number of parameters but a small amount of sample data, it is prone to overfitting. This leads to the model having a low loss function and high prediction accuracy on the training dataset, while it is the opposite in validation or test phases.

Dropout can effectively alleviate the occurrence of overfitting for neural networks. Dropout was first proposed by Hinton et al. to address the problem of overfitting [37]. Its core mechanism is to randomly deactivate neurons in the network with a probability of p to reduce the local dependence of the neural network and thereby improve the model's generalization ability.

We add a dropout layer after each LSTM layer, as shown in Figure 1. Except for the first layer of LSTM, the input to other LSTM layers at time t is calculated as follows:

$$Dropout(h_t^{l-1}) = Bernoullip * h_t^{l-1} \tag{16}$$

where $Dropout(\cdot)$ represents the dropout operation that sets the output hidden state to zero by a specified probability $1 - p$. l is the layer number in the neural network, and $Bernoullip$ is a random variable's discrete probability distribution which takes a value of 1 with the probability p and a value of 0 with the probability $1 - p$.

2.3.4. Attention

As share price movements are related to specific time points, such as new government policies or significant R&D developments in the company, we introduced the attention mechanism into the two-layer LSTM model. The attention mechanism can give different weights according to the state of the time series at different moments so that it can extract the essential information of the trading day at the critical time.

$$u_t = \tanh(Wh_t) \tag{17}$$

$$u_t = \tanh(Wh_t)\alpha_{t,i} = \frac{\exp(u_t^T u)}{\sum_{t=1}^T \exp(u_t^T u)} \tag{18}$$

$$\hat{y} = \sum_{t=1}^T \alpha_t h_t \tag{19}$$

In the equations above, u is a trainable parameter matrix used to represent context information and $\alpha_{t,i}$ is the allocation coefficient of input states, both of which are randomly initialized and optimized during the training procedure.

2.3.5. Prediction Model

In summary, the input data for our prediction model is a 3D feature tensor incorporating multi-source sentiment and technical indicators. Figure 6 shows the schema of how data are represented in our model. From this we can extract the corresponding data for model training.

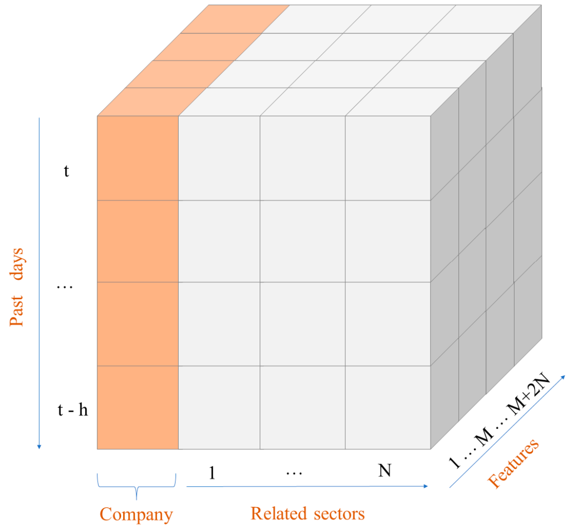


Figure 6. 3D input tensor for prediction model.

The architecture of our prediction model is shown in Figure 1 above. In order to fully utilize temporal information and extract key time points of stock price trends, we adopted a combination architecture of dual-layer LSTM and attention for stock price prediction. Dropout layers were added after each LSTM network to improve the model's generalization ability. An attention mechanism was introduced to extract critical information and enhance the accuracy of the model's predictions.

It is important to note that bidirectional LSTM networks could not be used in this paper because stock market data follows a strict time series format, which means that future data cannot be used to predict present stock prices.

3. Experiment

3.1. Datasets

3.1.1. Price Data

We used market data from two stocks, NIO (stock code: NIO) and Tesla (stock code: TSLA), to examine the effectiveness of our model. The time span was from September 2020 to September 2022. We used the Python language (version 3.8.1) and the yfinance package (version 0.1.45) to download the required stock price data. Based on the market's price, we calculated corresponding technical indicators to build the time series of technical indicators for each stock. The normalized data for stock price indicators are shown in Table 2.

Table 2. Normalized indicator data.

Period	**MA(10)**	**MA(20)**	**EMA(10)**	**EMA(20)**	**…**
29 September 2020	0.10087	0.07586	0.09679	0.07185	…
30 September 2020	0.10526	0.07816	0.10575	0.07711	…
1 October 2020	0.10964	0.08046	0.11528	0.08307	…
2 October 2020	0.11403	0.08275	0.12071	0.08717	…
5 October 2020	0.12061	0.08735	0.12683	0.09180	…
6 October 2020	0.12280	0.09195	0.12809	0.09393	…
7 October 2020	0.13157	0.09655	0.13274	0.09785	…
8 October 2020	0.13815	0.09885	0.13606	0.10112	…
…	0.14693	0.10344	0.13889	0.10415	…

As the new energy vehicle sector has been a rapidly growing emerging industry in recent years, this paper selected these two stocks as they are leading companies in this industry. The reason for choosing these two stocks as research objects is twofold. Firstly, as leading companies in the new energy vehicle sector, these two stocks are representative and have significant price fluctuations, thus having great potential for analysis and research value. Secondly, these companies are well-known and highly discussed with active stock trading. This generates a larger volume of high-quality data from news reports and investor discussions, which can be used for experimental model training.

3.1.2. Text Data

Corresponding to the price data, the text data used in this paper included news headlines and investor comments from the same period for the two stocks. Additionally, we picked the three most relevant sectors as text data sources for the model. Similarly, news headlines and investor comments from the same period were selected for text data sources.

We used a Python-based web crawler to collect investor comment text data from stock forums on the Eastmoney website, while the news headline texts were collected from the industry news section of the Choice financial terminal app.

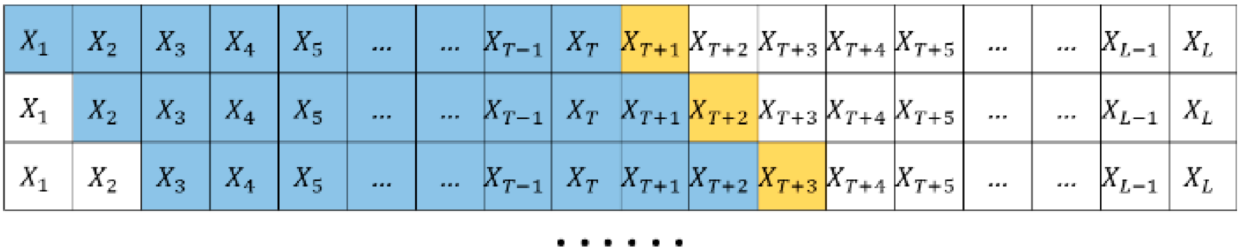
After text data cleaning, we fed them into the sentiment analysis module, which had been fine-tuned. The sentiment scores of the three BERT models were calculated, as shown in Table 3.

Table 3. Text sentiment score samples.

News Text	Translation	BERT-Positive	BERT-Negative	RoBERTa-Positive	RoBERTa-Negative	FinBERT-Positive	FinBERT-Negative
什么原因？“蔚小理”变成“理小蔚”，理想汽车8月交付量暴增，蔚来连续“败北”	What is the reason why “Xiaoli Wei” became “Li Xiaowei”, and Li Auto’s delivery volume skyrocketed in August, while NIO has been continuously “defeated”?	0.120	0.870	0.001	0.999	0.001	0.999
华为无人驾驶引发A股放量大涨	Huawei’s unmanned driving technology triggers a significant surge in A-shares trading volume	0.880	0.120	0.999	0.001	0.999	0.001
Review Text	**Translation**	**BERT-Positive**	**BERT-Negative**	**RoBERTa-Positive**	**RoBERTa-Negative**	**FinBERT-Positive**	**FinBERT-Negative**
你降价，我“画饼” 蔚来不可期	“You lower the price, I paint a rosy picture—NIO is unexpected.”	0.470	0.530	0.001	0.999	0.250	0.750
飞流直下三千尺	Flying waters descending straight three thousand feet	0.395	0.605	0.001	0.999	0.001	0.999

3.1.3. Generation of Training and Testing Sets

We divide data into a training set and a testing set. The training set consisted of 75% of the data, which was used to train the model parameters. The remaining 25% of the data served as the testing set to evaluate the model's performance. Considering the short timeliness and limited dataset in the emerging new energy vehicle industry, we employed the sliding window method to train the model. Within each training set, it was further divided into a training–testing set, referred to as a "study period". In this paper, a "study period" was set as 100 trading days, where the first 75 data points were used as the training set, and the last 25 data points were used as the testing set. The generation process of the training–testing settings is illustrated in Figure 7.



X_1	X_2	X_3	X_4	X_5	...	...	X_{T-1}	X_T	X_{T+1}	X_{T+2}	X_{T+3}	X_{T+4}	X_{T+5}	...	...	X_{L-1}	X_L
X_1	X_2	X_3	X_4	X_5	...	...	X_{T-1}	X_T	X_{T+1}	X_{T+2}	X_{T+3}	X_{T+4}	X_{T+5}	...	...	X_{L-1}	X_L
X_1	X_2	X_3	X_4	X_5	...	...	X_{T-1}	X_T	X_{T+1}	X_{T+2}	X_{T+3}	X_{T+4}	X_{T+5}	...	...	X_{L-1}	X_L

Figure 7. Sliding window training method.

3.2. Experiment Setup

3.2.1. Hyper-Parameters Setting

In Bert, we set $2 \times e^{-5}$ as the learning rate and 100 as the number of epochs. We used CrossEntropyLoss as the loss function, and fine-tuned Bert by using financial text data with sentiment labels. Figure 8 illustrates the training and validation accuracy for BERT. Our model does not appear to be overfitting. Similar results are shown in Figures 9 and 10 for RoBERTa and FinBert but with some differences.

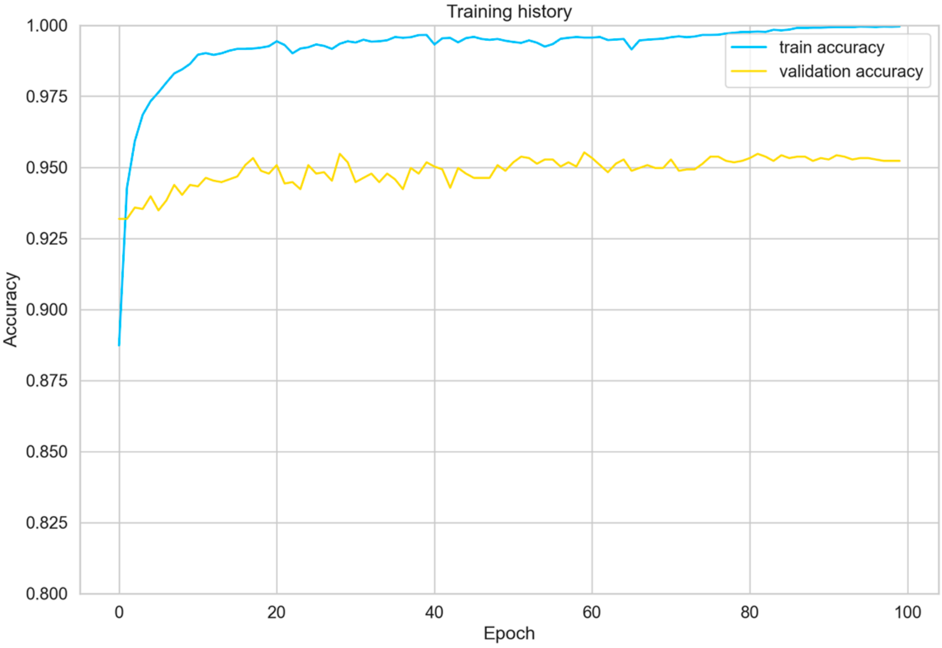


Figure 8. Training and validation accuracy for BERT.

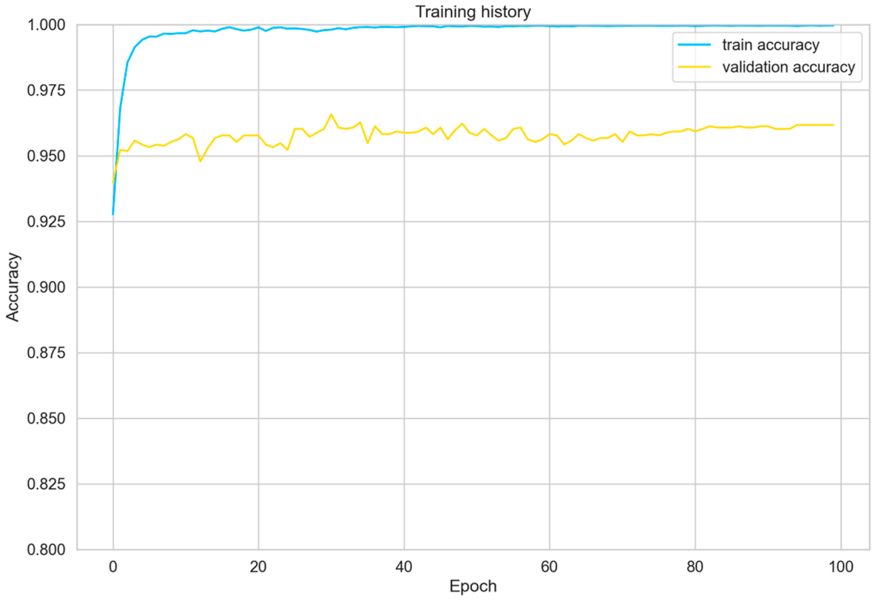


Figure 9. Training and validation accuracy for RoBERTa.

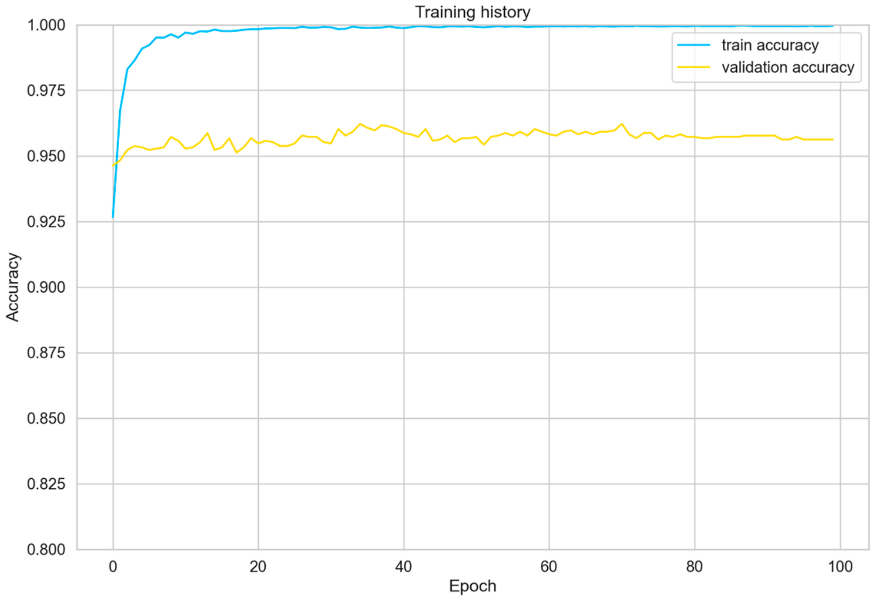


Figure 10. Training and validation accuracy for FinBERT.

In our prediction model, the first layer was a sequential LSTM with 64 units. The input was the concatenation vector of the market data and sentiment analysis features. The second layer was a non-sequential LSTM that further extracted critical information from the sequence with 32 units.

We used the MSE loss function to minimize the loss values of all samples in the training set. The batch size was set as 32, and the Adam optimizer with an initial learning rate of 0.001 was used to train the model. Additionally, the number of epochs was set as 1000 to guarantee the convergence of the training process. We set the dropout rate as 0.1 to avoid overfitting.

As shown in Figures 11 and 12, the loss functions based on different BERT models exhibited significant downward trends and eventually converged within 1000 epochs.

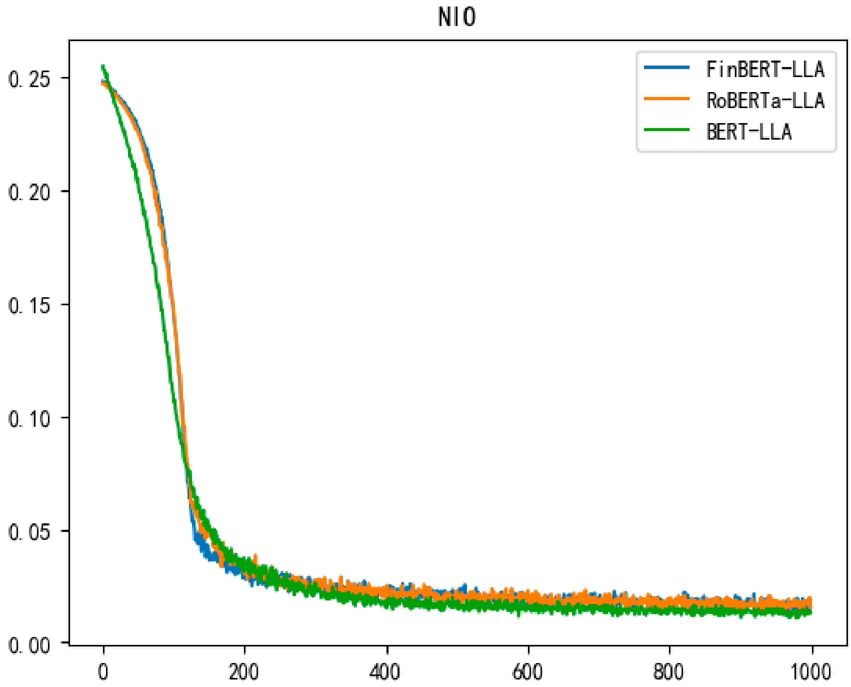


Figure 11. Loss function line graph for NIO.

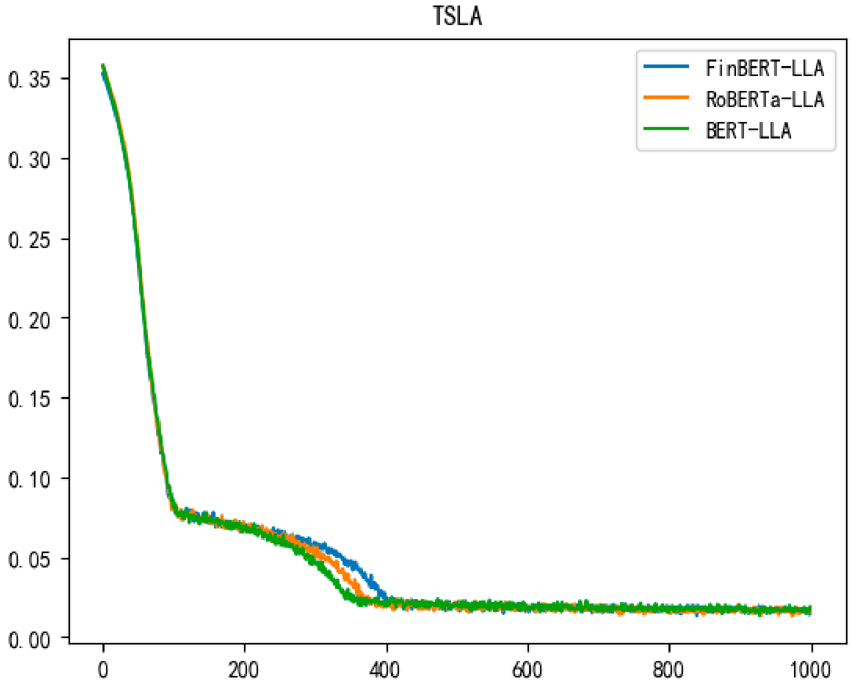


Figure 12. Loss function line graph for TSLA.

3.2.2. Baselines

In the experiment, we adopted six models as baselines: LSTM-attention, LSTM, RNN, CNN, RF, and SVM.

Furthermore, to verify the impact of incorporating multi-channel data sources in text sentiment analysis on prediction accuracy, different input datasets were used in the BERT-LLA model to evaluate the influence of input features on prediction accuracy. The experiment scheme is shown in Table 4, where a value of one represents that the data are used for model input, and zero is the opposite.

Table 4. Model input feature scheme.

Scheme	Sentiment Index Time Series	Related Sector Sentiment Index Time Series	Technical Indicator Time Series
(1)	1	1	1
(2)	1	0	1
(3)	0	0	1

3.2.3. Evaluation Metrics

We used three metrics to evaluate the performance of each model:

Mean Square Error (MSE):

$$\text{MSE} = \frac{1}{m}\sum\nolimits_{i=1}^{m}(y_i - \hat{y}_i)^2 \tag{20}$$

Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{m}} \tag{21}$$

Mean Absolute Percentage Error (MAE):

$$\text{MAE} = \frac{1}{m}\sum\nolimits_{i=1}^{m}|y_i - \hat{y}_i| \tag{22}$$

where m is the sample size, y_i is the true value of the i-th sample, $\hat{y}_i$ is the predicted value of the i-th sample.

MSE and RMSE are effective metrics for measuring the errors between observed values and predicted values. They provide a quantitative measure of how close the predictions are to the actual values. MAE is a metric that measures the accuracy of the prediction method. It calculates the average absolute percentage difference between the predicted and true values, indicating the overall accuracy of the model's predictions. It is clear that the smaller the MSE, RMSE, and MAE values, the smaller the model's prediction error.

3.2.4. Quantitative Strategy Setting

To verify whether the model's prediction results can assist trading to some extent, this paper proposes a trading strategy that integrates prediction results. The strategy utilizes fast- and slow-moving averages, along with price prediction results to generate trading signals. The fast-moving average was set as a 5-day simple moving average, while the slow-moving average was set as a 10-day simple moving average. The strategy comprises two buy signals and two sell signals:

- Buy Signal 1: The predicted price exceeds the 5-day moving average price by a 5% threshold.
- Buy Signal 2: The 5-day moving average crosses above the 10-day moving average.
- Sell Signal 1: The predicted price falls below the 5-day moving average by a 5% threshold.
- Sell Signal 2: The 5-day moving average crosses below the 10-day moving average.

As a comparison, we used a strategy that excluded the Buy Signal 1 and Sell Signal 1, referred to as DMA (double-moving-average strategy). The effectiveness of the price prediction results for trading applications was assessed by comparing the performance of these two strategies in backtesting.

3.3. Experiment Results and Discussion

3.3.1. Related Stock Sector Selection

We screen 10 related financial sectors in the new energy vehicle industry chain on the Eastmoney website, collected news and investor review data in the same period, and calculated the sector heat factor, the cosine similarity, a correlation coefficient according to Section 3.1.3, and we ranked the correlation coefficients in descending order. The results are shown in Table 5. We selected the three sectors with the most significant correlation coefficients, namely new energy, autonomous driving, and lithium battery, and fed the sentiment data of these three sectors into the prediction model to assist in stock price prediction.

Table 5. Related data of sector correlation calculations.

Sector	News Number	Review Number	SHF	Cosine Similarity	Correlation Coefficient
New Energy	23,957	30,492	1.67248	0.768	1.28446
Autonomous Driving	2830	4732	1.24432	0.637	0.79263
Lithium Battery	3693	7998	1.27698	0.612	0.78151
Automotive Parts	81,354	825	1.79971	0.452	0.75767
Charging Post	2338	1118	1.21416	0.549	0.66657
Complete Vehicles	5975	457	1.23578	0.496	0.61294
Fuel Cell	2704	1431	1.21898	0.427	0.52050
Automotive Chip	1305	353	1.20172	0.416	0.49991
Semiconductor	12,469	6686	1.20172	0.387	0.46506
Power Battery Recycling	1128	199	1.19948	0.370	0.44380

3.3.2. Prediction Performance and Comparison with the Baseline

After training the model and de-normalizing the predicted results, we obtained the fitted curves of the BERT-LLA model, as shown in Figures 13 and 14. It was observed that despite the significant volatility in the stock prices of NIO and Tesla, the model's predicted values were quite close to the actual values in both the training and prediction sets.

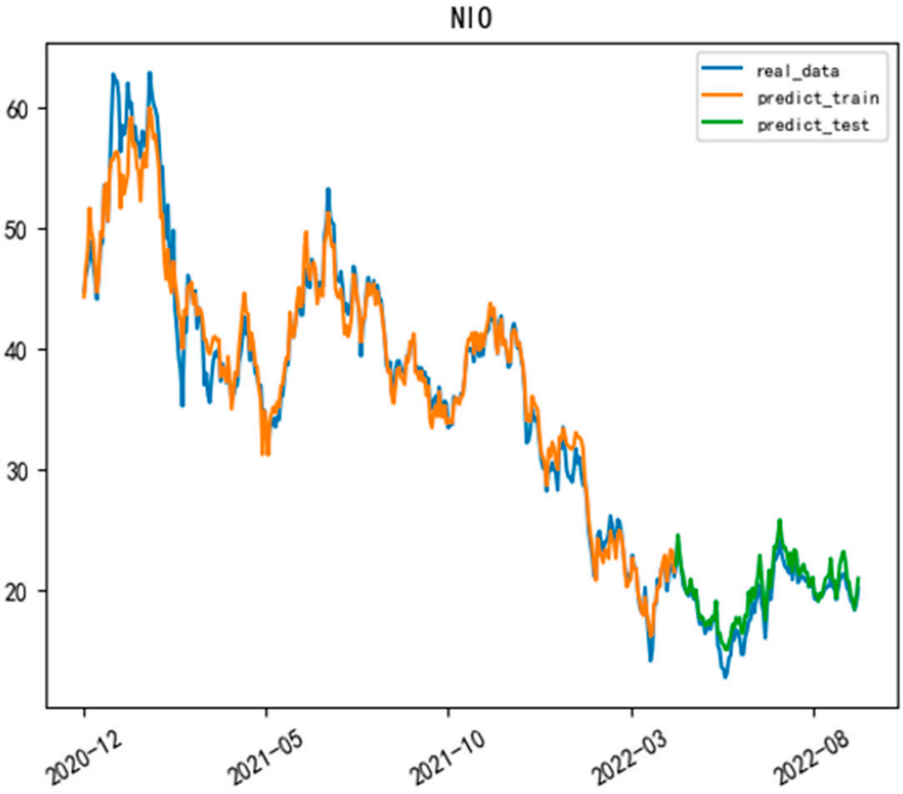


Figure 13. NIO prediction results based on BERT-LLA.

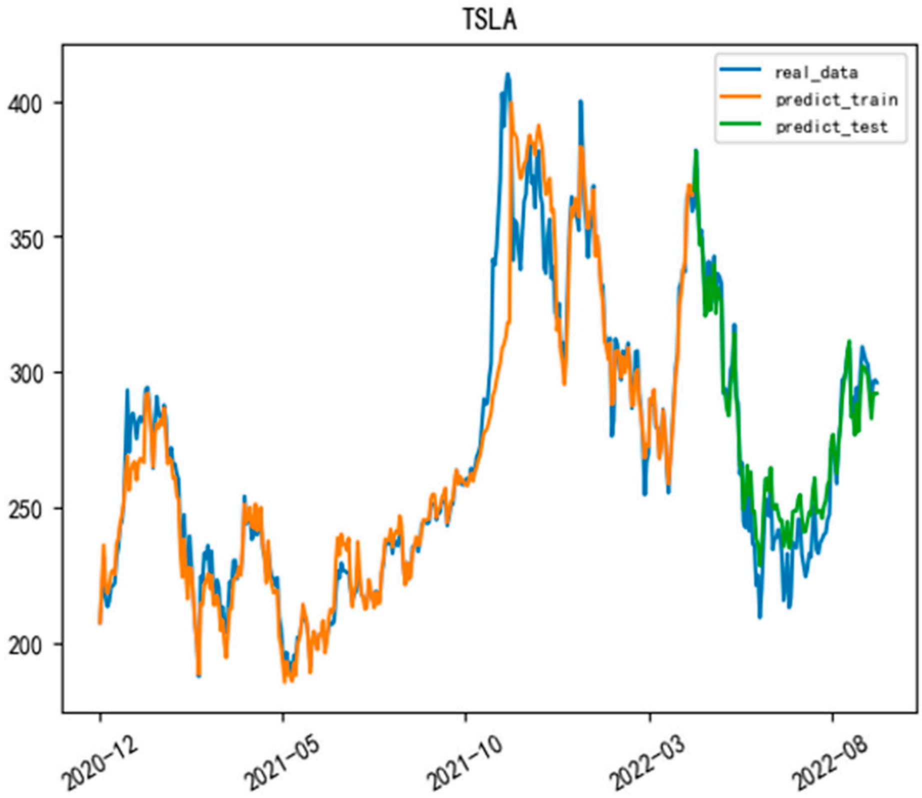


Figure 14. TSLA prediction results based on BERT-LLA.

We compared the best-performing BERT model with six benchmark models: LSTM-attention, LSTM, RNN, CNN, RF, and SVM. Their prediction performance is shown in Table 6.

Table 6. The evaluation metrics in two datasets.

Company	Model	MSE	RMSE	MAE
NIO	BERT-LLA	0.00046	0.02147	0.01771
	LSTM-attention	0.00053	0.02376	0.01841
	LSTM	0.00054	0.02419	0.01882
	RNN	0.00058	0.02578	0.01876
	CNN	0.00078	0.02810	0.02140
	RF	0.02471	0.15720	0.15446
	SVM	0.02717	0.16486	0.16235
TSLA	BERT-LLA	0.00095	0.03091	0.02560
	LSTM-attention	0.00103	0.03304	0.02647
	LSTM	0.00108	0.03395	0.02694
	RNN	0.00112	0.03484	0.02776
	CNN	0.00126	0.03952	0.03144
	RF	0.03542	0.18821	0.16501
	SVM	0.03062	0.17498	0.15313

The results indicated that the SVM and RF models performed relatively poorly, highlighting their limitations in handling time series problems. The CNN network showed better predictive performance compared with the SVM and RF models. The RNN, LSTM, and LSTM-attention networks excelled in handling time series problems, exhibiting relatively good predictive performance. The proposed BERT-LLA model had the smallest evaluation metrics and the best predictive performance, presenting that the model accurately captured critical information between the time series nodes through the dual-layer LSTM network and attention mechanism, significantly enhancing the accuracy of the predictions.

3.3.3. The Effect Comparison of Different Input Features

The prediction performance from different input feature schemes are shown in Table 7. It is evident that the prediction performance of input feature scheme (1) is superior to the other two schemes.

Table 7. Prediction performance of different input data schemes.

Company	Input Feature Scheme	MSE	RMSE	MAE
NIO	(1)	0.00046	0.02147	0.01771
	(2)	0.00074	0.02724	0.02458
	(3)	0.00101	0.03175	0.02653
TSLA	(1)	0.00095	0.03091	0.02560
	(2)	0.00119	0.03451	0.03014
	(3)	0.00128	0.03643	0.03085

The data indicate that the proposed stock price prediction method, which combines sentiment analysis, significantly improves the accuracy of the model's predictions. Additionally, the prediction method incorporating the sentiment data of related sectors is superior to solely relying on technical analysis for prediction. This confirms the feasibility of our sentiment analysis method in stock price prediction research.

We plot the de-normalized prediction results of different input feature schemes in the training and testing sets, as shown in Figures 15 and 16.

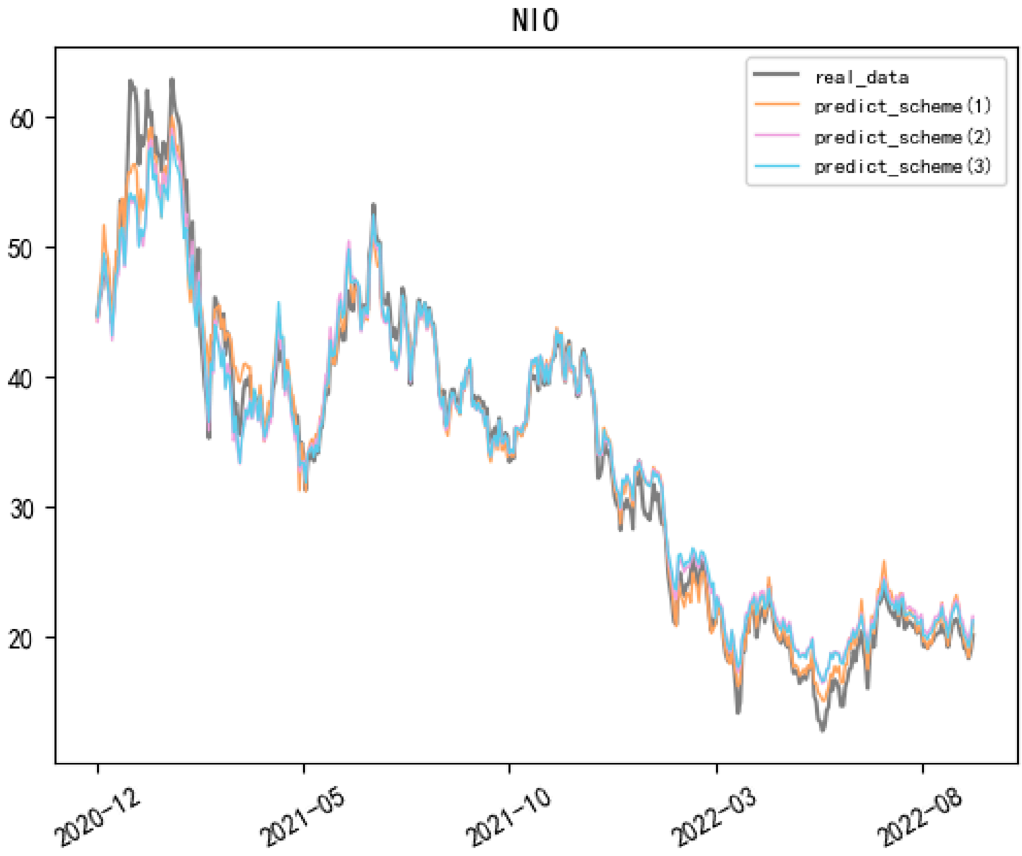


Figure 15. Prediction results of different input schemes for NIO.

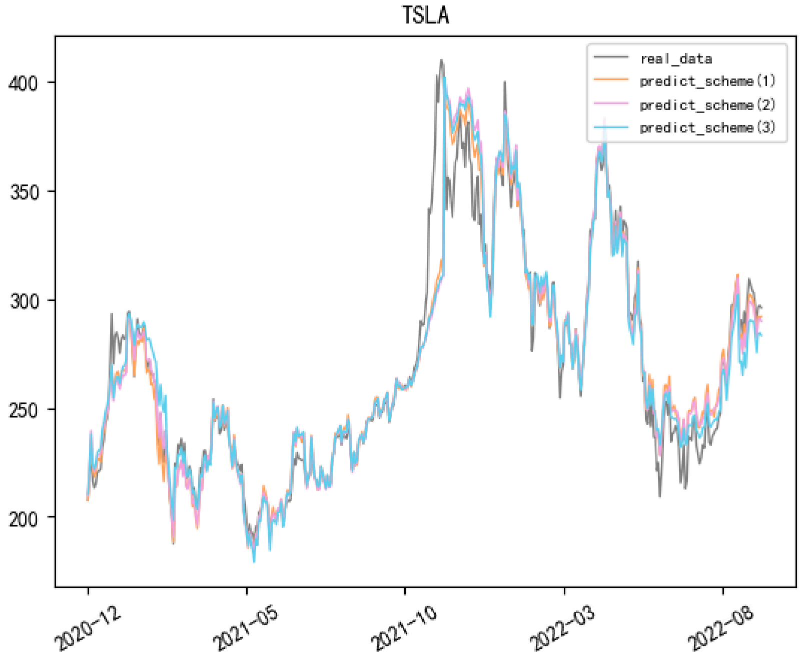


Figure 16. Prediction results of different input schemes forTSLA.

3.3.4. Comparison of the Weights of News and Investor Reviews

In order to investigate the impact of time series sentiment indices corresponding to investor comment text data and news text data on the model's prediction accuracy, we took NIO as an example and calculated the sentiment index under different weights. Then, we fed these sentiment indices into our model to obtain the MSE indicator. The heatmap was plotted based on the MSE indicator data, as shown in Figure 17.

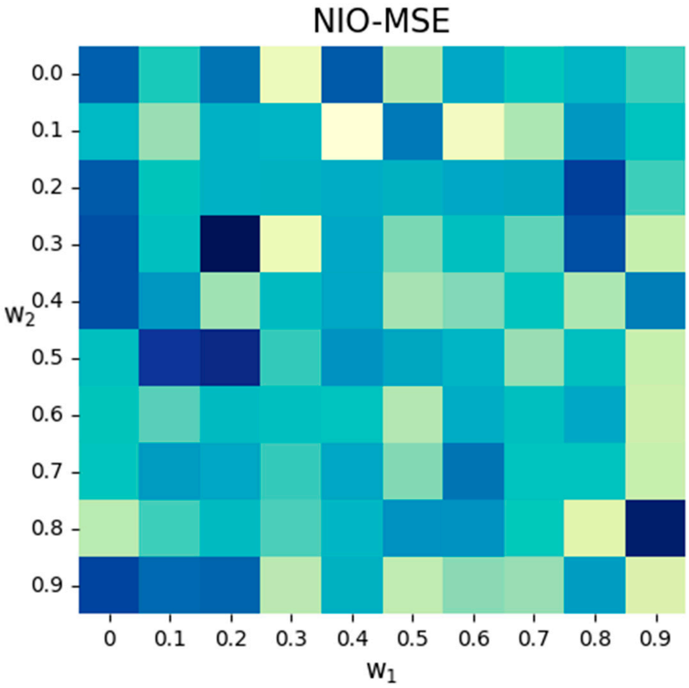


Figure 17. Model prediction accuracy heat map for NIO. The picture shows the distribution of the MSE under different weights w_1 and w_2. The color intensity represents the value magnitude, with darker colors indicating larger values.

From the heatmap, we can intuitively see that when $w_1 = 0.4$, $w_2 = 0.1$, the color of the heatmap is the lightest, indicating relatively optimal prediction accuracy. This is consistent with common sense, as stock investors tend to exhibit a certain degree of bias in expressing negative sentiment on online platforms. On the other hand, market sentiment represented by news information is more objective and has a more significant impact on market trends. At the same time, when $w_1 = 0$, which means not using the news sentiment index for prediction, the model's prediction accuracy is relatively biased. This indicates that solely relying on investor sentiment is difficult to represent the overall market sentiment. These results demonstrate the importance of the proposed market sentiment analysis method and the weight analysis of investor comment and news text in stock price prediction research.

3.3.5. Application of Prediction Results in Trading Strategy

Based on the quantitative strategy proposed above, we chose Backtrader, an open-source quantitative framework based on Python, with the initial capital set at 1 million. The backtesting period spanned from 1 January 2022 to 30 August 2022. The underlying asset for the strategy was NIO, and the transaction commission rate was set to 0.1% for a single trade of 10,000 shares. The backtesting results for the two strategies are shown in Table 8. We also plot the net worth curves for both strategies, as shown in Figure 18.

Table 8. Backtesting results for both strategies.

	DMA + Prediction	**DMA**
Sharpe ratio	0.955	−1.045
Maximum drawdown (%)	2.158	9.246
Total return rate (%)	8.958	−8.778
Total assets	1,089,586.92	912,221.70

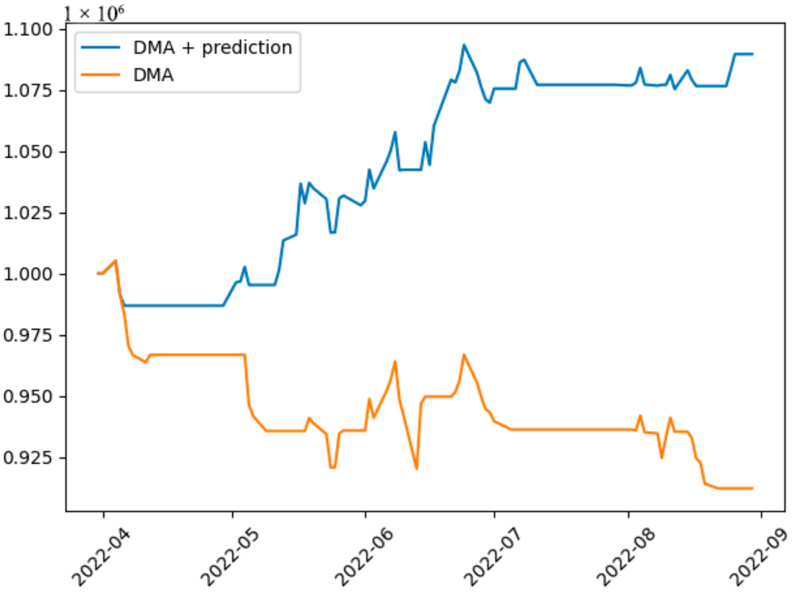


Figure 18. Net worth curves for both strategies.

It is evident that DMA + Prediction strategy, which incorporates the price prediction results, is able to achieve a more stable return with less volatility and exhibits superior performance during the backtesting period compared to the simple double-moving-average strategy. This reflects the practical application value of our model in financial trading.

4. Conclusions

We propose a stock price prediction model that combines market sentiment and price data from multiple sources. Based on BERT model, market sentiment is captured from textual data, and sentiment index time series is constructed using the sentiment analysis method proposed above. At the same time, we combine the technical indicator time series to construct the prediction model through LSTM with the attention mechanism. Experimental results show that the integration of market sentiment improves prediction accuracy to a large extent. Our model demonstrates robustness and generalization in experiment datasets. Our work provides a reference for utilizing unstructured textual data in stock price prediction, particularly in the context of the hot sector of new energy vehicles, validating the practical application value of the proposed approach in predicting stock prices of popular sectors. We also propose a method that can quickly and efficiently screen for related industry sectors promptly and efficiently. Furthermore, our study explores the impact of the weights of two types of textual data on the model's prediction results, which expands the construction approach of sentiment indexes based on financial texts.

There are also some limitations in the paper:

- Lack of analysis on long texts such as industry research reports and lengthy stock reviews that significantly impact stock price trends.
- No specific research on sudden market hot topics or scenarios. Based on experience, sudden major news events have a significant impact on stock price trends.
- A large amount of financial text data are used in this paper, but we have not analyzed the validity of this financial text data, which may lead to some of the data being ineffective and thus affecting the results of the experiment.

In future research in related fields, there are directions that can continue to be explored:

- We know that K-line patterns contain a wealth of market information, and how to introduce K-line pattern features into a model is an attractive research topic.
- Additionally, research in related fields should not be limited to daily data. Exploring how to capture market sentiment changes in short periods using text data to assist high-frequency trading is also an important area for future in-depth research.
- With the development of multimodal technology, exploring the intrinsic mechanism of stock prices should not be limited to price and text data, but also analyzing and applying multimodal data information such as video, audio, and image information in the financial field will be an important development direction.

Author Contributions: Conceptualization, K.F. and Y.Z.; methodology, K.F. and Y.Z.; software, Y.Z.; validation, K.F. and Y.Z.; formal analysis, Y.Z.; investigation, Y.Z.; resources, K.F.; data curation, Y.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, Y.Z. and K.F.; visualization, Y.Z.; supervision, K.F.; project administration, K.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Fundamental Research Funds for the Central Universities (104972024KFYqk0009).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Fama, E.F. The behavior of stock-market prices. *J. Bus.* **1965**, *38*, 34–105. [CrossRef]
2. Pedersen, L.H. *Efficiently Inefficient: How Smart Money Invests and Market Prices Are Determined*; Princeton University Press: Princeton, NJ, USA, 2019.
3. Rounaghi, M.M.; Zadeh, F.N. Investigation of market efficiency and Financial Stability between S&P 500 and London Stock Exchange: Monthly and yearly Forecasting of Time Series Stock Returns using ARMA model. *Phys. A Stat. Mech. Its Appl.* **2016**, *456*, 10–21. [CrossRef]
4. Herwartz, H. Stock return prediction under GARCH—An empirical assessment. *Int. J. Forecast.* **2017**, *33*, 569–580. [CrossRef]

5. Qiu, M.; Song, Y.; Akagi, F. Application of artificial neural network for the prediction of stock market returns: The case of the Japanese stock market. *Chaos Solitons Fractals* **2016**, *85*, 1–7. [CrossRef]
6. Zhou, Y.; Li, T.; Shi, J.; Qian, Z. A CEEMDAN and XGBOOST-Based Approach to Forecast Crude Oil Prices. *Complexity* **2019**, *2019*, 1–15. [CrossRef]
7. Jiang, W. Applications of deep learning in stock market prediction: Recent progress. *Expert Syst. Appl.* **2021**, *184*, 115537. [CrossRef]
8. Thakkar, A.; Chaudhari, K. A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. *Expert Syst. Appl.* **2021**, *177*, 114800. [CrossRef]
9. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
10. Wu, C.-H.; Lu, C.-C.; Ma, Y.-F.; Lu, R.-S. A New Forecasting Framework for Bitcoin Price with LSTM. In Proceedings of the 18th IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018.
11. Kim, H.Y.; Won, C.H. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Syst. Appl.* **2018**, *103*, 25–37. [CrossRef]
12. Barua, R.; Sharma, A.K. Dynamic Black Litterman portfolios with views derived via CNN-BiLSTM predictions. *Financ. Res. Lett.* **2022**, *49*, 103111. [CrossRef]
13. Wang, W.; Li, W.; Zhang, N.; Liu, K. Portfolio formation with preselection using deep learning from long-term financial data. *Expert Syst. Appl.* **2019**, *143*, 113042. [CrossRef]
14. Bollen, J.; Mao, H.; Zeng, X. Twitter mood predicts the stock market. *J. Comput. Sci.* **2011**, *2*, 1–8. [CrossRef]
15. Lee, H.; Surdeanu, M.; MacCartney, B.; Jurafsky, D. On the Importance of Text Analysis for Stock Price Prediction. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, 26–31 May 2014.
16. Poongodi, M.; Nguyen, T.N.; Hamdi, M.; Cengiz, K. Global cryptocurrency trend prediction using social media. *Inf. Process. Manag.* **2021**, *58*, 102708. [CrossRef]
17. Ni, H.; Wang, S.; Cheng, P. A hybrid approach for stock trend prediction based on tweets embedding and historical prices. *World Wide Web-Internet Web Inf. Syst.* **2021**, *24*, 849–868. [CrossRef]
18. Dragut, E.C.; Wang, H.; Sistla, P.; Yu, C.; Meng, W. Polarity Consistency Checking for Domain Independent Sentiment Dictionaries. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 838–851. [CrossRef]
19. Skuza, M.; Romanowski, A. Sentiment Analysis of Twitter Data within Big Data Distributed Environment for Stock Prediction. In Proceedings of the 2015 Federated Conference on Computer Science and Information Systems, Lodz, Poland, 13–16 September 2015; pp. 1349–1354.
20. Daudert, T. Exploiting textual and relationship information for fine-grained financial sentiment analysis. *Knowl.-Based Syst.* **2021**, *230*, 107389. [CrossRef]
21. Jing, N.; Wu, Z.; Wang, H. A Hybrid Model Integrating Deep Learning with Investor Sentiment Analysis for Stock Price Prediction. *Expert Syst. Appl.* **2021**, *178*, 115019. [CrossRef]
22. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
23. Colasanto, F.; Grilli, L.; Santoro, D.; Villani, G. AlBERTino for stock price prediction: A Gibbs sampling approach. *Inf. Sci.* **2022**, *597*, 341–357. [CrossRef]
24. Pornwattanavichai, A.; Maneeroj, S.; Boonsiri, S. BERTFOREX: Cascading Model for Forex Market Forecasting Using Fundamental and Technical Indicator Data Based on BERT. *IEEE Access* **2022**, *10*, 23425–23437. [CrossRef]
25. Hiew, J.Z.G.; Huang, X.; Mou, H.; Li, D.; Wu, Q.; Xu, Y. BERT-based Financial Sentiment Index and LSTM-based Stock Return Predictability. *arXiv* **2019**, arXiv:1906.09024v2.
26. Sun, T.; Wang, J.; Zhang, P.; Cao, Y.; Liu, B.; Wang, D. Predicting Stock Price Returns Using Microblog Sentiment for Chinese Stock Market. In Proceedings of the 2017 3rd International Conference on Big Data Computing and Communications (BIGCOM), Chengdu, China, 10–11 August 2017; pp. 87–96.
27. Chou, H.-C.; Ramachandran, K.M. Combination of Time Series Analysis and Sentiment Analysis for Stock Market Forecasting. Ph.D. Thesis, University of South Florida, Tampa, FL, USA, 2021.
28. Cristescu, M.P.; Nerisanu, R.A.; Mara, D.A.; Oprea, S.-V. Using market news sentiment analysis for stock market prediction. *Mathematics* **2022**, *10*, 4255. [CrossRef]
29. Fazlija, B.; Harder, P. Using financial news sentiment for stock price direction prediction. *Mathematics* **2022**, *10*, 2156. [CrossRef]
30. Deng, S.; Zhu, Y.; Yu, Y.; Huang, X. An integrated approach of ensemble learning methods for stock index prediction using investor sentiments. *Expert Syst. Appl.* **2024**, *238*, 121710. [CrossRef]
31. Mohan, S.; Mullapudi, S.; Sammeta, S.; Vijayvergia, P.; Anastasiu, D.C. Stock Price Prediction Using News Sentiment Analysis. In Proceedings of the 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService), Newark, CA, USA, 4–9 April 2019.
32. Li, Q.; Chen, Y.; Jiang, L.L.; Li, P.; Chen, H. A tensor-based information framework for predicting the stock market. *ACM Trans. Inf. Syst.* **2016**, *34*, 1–30. [CrossRef]
33. Nassirtoussi, A.K.; Aghabozorgi, S.; Wah, T.Y.; Ngo, D.C.L. Text mining of news-headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment. *Expert Syst. Appl.* **2015**, *42*, 306–324. [CrossRef]

34. Wang, H.; Lu, S.; Zhao, J. Aggregating multiple types of complex data in stock market prediction: A model-independent framework. *Knowl.-Based Syst.* **2019**, *164*, 193–204. [CrossRef]
35. Hu, Y.; Liu, K.; Zhang, X.; Su, L.; Ngai, E.; Liu, M. Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review. *Appl. Soft Comput.* **2015**, *36*, 534–551. [CrossRef]
36. Patro, S.G.K.; Sahu, K.K. Normalization: A preprocessing stage. *arXiv* **2015**, arXiv:1503.06462. [CrossRef]
37. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580, 212–223.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.