Article

Not peer-reviewed version

Stock Price Prediction Based on Hybrid CNN-LSTM Model

Qinghe Zhao , Yue Hao , Xuechen Li [*]

Posted Date: 25 September 2024

doi: 10.20944/preprints202409.1904.v1

Keywords: stock price prediction; deep learning; convolutional neural network; long short-term memory network

Article

Stock Price Prediction Based on Hybrid CNN-LSTM Model

Qinghe Zhao [1,*], Yue Hao [2] and Xuechen Li [3]

[1] School of Engineering, University of Exeter, Exeter, UK
[2] Information Systems, Carey Business School, Johns Hopkins University, Baltimore, USA; yhao19@alumni.jh.edu
[3] Schack Institute of Real Estate, New York University, New York, USA; Xl3281@nyu.edu
* Correspondence: carolinezqh@foxmail.com

Abstract. Stock price prediction is of great significance but faces numerous challenges. In this study, a hybrid CNN-LSTM model was utilized. The stock data were processed and feature engineering was conducted. The model architecture and training strategies were expounded. Experimental results demonstrated that this model outperformed traditional methods and benchmark models on the test set, featuring strong capabilities in feature extraction and handling long-term dependencies. Key roles of data preprocessing, hyperparameter adjustment, and model fusion were also summarized, providing valuable references for stock price prediction.

Keywords: stock price prediction; deep learning; convolutional neural network; long short-term memory network

1. Introduction

1.1. Research Background and Significance

The stock market, as a cornerstone of modern financial systems, plays a crucial role in resource allocation, corporate financing, and economic development. Fluctuations in stock prices not only reflect a company's operational status and market expectations but are also influenced by a complex interplay of factors such as macroeconomic conditions, regulatory policies, and industry competition. Accurate forecasting of stock price movements is essential for investors to formulate rational investment strategies, mitigate risks, and achieve asset appreciation. For financial institutions, precise stock price predictions can help optimize asset allocation, manage risk, and provide more valuable financial services.

However, predicting stock prices is an extremely challenging task due to the inherent complexity, uncertainty, and nonlinearity of stock markets. Traditional forecasting methods often fall short of delivering satisfactory results because of these characteristics. Fundamental analysis relies on financial statements and economic data, which may lack timeliness and fail to capture market sentiment effectively. Technical analysis primarily focuses on pattern recognition based on historical prices and trading volumes, which can be obscured by market noise and random fluctuations. Moreover, stock markets are influenced by numerous unpredictable factors such as unexpected events and sudden shifts in investor sentiment, further complicating the forecasting process [1].

1.2. Research Objectives and Issues

The primary objective of this research is to construct an efficient and accurate stock price prediction model to enhance the predictive capabilities of stock price trends, thereby providing more valuable decision support for investors and financial institutions. Current stock price prediction methods face several notable issues:

- Limited adaptability: Many traditional models perform well under specific market conditions or for particular stocks, but their predictive accuracy drops significantly when market environments change or when applied to different stocks.
- Insufficient capture of nonlinear relationships: Changes in stock prices often exhibit complex nonlinear relationships that linear models struggle to fully capture.
- Limitations in feature extraction: Existing methods may not sufficiently mine potential valuable information during data processing and feature engineering, leading to incomplete and ineffective features input into the prediction models.

This study aims to address the following key issues: leveraging deep learning technologies, particularly by combining the strengths of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks, to better capture the nonlinearities and long-term dependencies present in stock price data; and through innovative feature engineering methods, extracting richer and more representative features from raw stock data to improve the quality of inputs for the prediction models.

1.3. Research Methods

The following data, techniques, and algorithms were employed in this study:

- Data: A large dataset of historical trading data covering multiple periods and industries was collected, including detailed information such as opening prices, closing prices, highest and lowest prices, and trading volumes.
- Techniques: A combination of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks was utilized, taking advantage of the strengths of CNNs in feature extraction and the capability of LSTMs to handle long-term dependencies in time series data.
- Algorithms: Adaptive optimization algorithms, such as the Adam optimizer, were adopted to adjust the parameters of the model and minimize prediction errors.

2. Related Work

2.1. Traditional Stock Price Prediction Methods

Fundamental analysis evaluates a stock's intrinsic value and predicts its price trend by analyzing the company's financial condition, performance, and industry prospects. This approach is grounded in value investing theory, positing that stock prices eventually reflect the true worth of a company. However, fundamental analysis has limitations. Firstly, obtaining accurate and comprehensive financial information is challenging, and the information may not be timely. Secondly, valuing a company involves many complex factors and assumptions, introducing subjectivity and uncertainty. Additionally, changes in the macroeconomic environment and industry competition are difficult to predict accurately, potentially impacting the company's performance unexpectedly.

Technical analysis focuses on studying historical stock price and volume data through charts and various technical indicators to identify patterns and trends, thereby predicting future prices. Common tools include moving averages, the relative strength index (RSI), and Bollinger Bands [2]. The rationale behind technical analysis is the assumption that market behavior encompasses all known information, implying that stock price movements already reflect all available information. Nonetheless, its limitations are evident. Technical analysis heavily relies on the repetition of historical data patterns, yet markets are dynamic, and past patterns may not recur. Furthermore, technical indicators tend to lag, often generating misleading signals. Moreover, technical analysis is insensitive to market disruptions and significant changes in fundamentals.

2.2. Application of Machine Learning in Stock Forecasting

Decision tree algorithms construct a tree-like structure to make decisions based on different feature values, classifying or regressing stock prices. In stock forecasting, decision trees can determine whether stock prices will rise or fall based on multiple features in historical data. They are advantageous for their interpretability, but they are prone to overfitting and have limited predictive power for complex stock market data with a single decision tree model.

SVM finds an optimal hyperplane to classify or regress data [3]. In stock forecasting, it maps stock data to high-dimensional space and identifies the optimal classification boundary to predict price movements. SVM excels in handling small sample sizes and high-dimensional data, but it is computationally intensive and sensitive to kernel function selection.

However, the effectiveness of these machine learning algorithms in stock forecasting is constrained by various factors. The nonlinearity and nonstationarity of stock markets limit the expressive power of linear models, and market noise and outliers can interfere with model learning and prediction.

2.3. Advances in Deep Learning for Financial Applications

CNNs demonstrate powerful feature extraction capabilities in stock forecasting. They can automatically extract local patterns and features from time series stock price data, such as short-term price fluctuation patterns. Some studies applying CNNs to stock price prediction have achieved better results than traditional methods [4]. However, CNNs are relatively weak at capturing long-term dependencies.

RNNs and their variant, LSTM networks, are adept at handling long-term dependencies in sequence data, making them suitable for predicting stock prices, which are time series data. LSTM introduces gating mechanisms to effectively address the vanishing and exploding gradient problems in RNNs [5], enabling better retention of long-term historical information. Studies show that LSTM-based models can capture long-term trends and cyclical changes in stock forecasting, improving predictive accuracy.

Despite significant progress in stock forecasting using deep learning, challenges remain. For example, training models requires substantial data and computational resources, and the models' interpretability is poor, making it difficult to understand the decision-making process and predictive basis [6]. Additionally, deep learning models are sensitive to hyperparameter selection, requiring careful tuning.

3. Data Preparation and Preprocessing

3.1. Data Description

The stock data selected for this study encompass multiple industries and companies of varying sizes to ensure broad representativeness and applicability of the findings. The utilized stock data contain rich feature information, with key features including:

- Opening Price (Open): Reflecting the initial trading price at the start of the day, it represents market participants' initial assessment of the stock's value.
- Closing Price (Close): Representing the final trading price at the end of the day, it is considered one of the most important price indicators and significantly influences investors' decision-making.
- Highest Price (High): Recording the highest price reached during the day's trading session, it indicates the market's upward potential and resistance levels.
- Lowest Price (Low): Displaying the lowest price traded during the day, it reflects the market's downside support and risk level.
- Trading Volume (Volume): Indicating the quantity of stocks traded during the day, it reflects the market's trading activity and investor participation enthusiasm.

3.2. Data Cleaning and Handling Missing Values

During the data cleaning process, we identified and processed outliers. Outlier detection was based on statistical methods, such as the three-sigma rule. A data point deviating more than three standard deviations from the mean was classified as an outlier. These outliers were removed to avoid adverse effects on model training and predictions. This is because outliers may arise due to erroneous data recording or extreme market events and do not represent normal market behavior.

For missing values in the dataset, we adopted a strategy of mean imputation. The rationale behind this method lies in the inherent continuity and stability of stock data, where adjacent data points exhibit some degree of correlation. By calculating the mean of the column containing the missing value and filling it into the missing position, we maintained the integrity and continuity of the data while minimizing the introduction of additional bias. Although mean imputation can smooth out real fluctuations in the data, our comprehensive consideration and experimental validation demonstrated that it struck a good balance between preserving data characteristics and maintaining model performance.

3.3. Feature Engineering

To extract meaningful information from raw data, we conducted several feature engineering operations. Among these, we calculated moving averages, including 10-day, 50-day, and 100-day moving averages. Moving averages are computed by averaging closing prices over a specified number of days. They smooth price fluctuations and reflect the long-term trend of stock prices, aiding investors in identifying primary directions and support/resistance levels.

Yield was another critical feature extracted by calculating the percentage change between consecutive closing prices, reflecting the rate of growth or decline in stock prices. Yield captures short-term market fluctuations and changes in investor sentiment.

In terms of feature selection, we considered the relevance, importance, and interpretability of features. Through correlation analysis, we eliminated features with weak correlations to the target variable (such as closing price) to reduce redundancy and noise in the data. Based on domain knowledge and preliminary experimental results, we determined the features that had a significant impact on stock price prediction.

For dimensionality reduction, we employed methods such as principal component analysis (PCA). PCA projects high-dimensional data onto a lower-dimensional space while retaining the main variance information. Dimensionality reduction not only reduced computational load and improved model training efficiency but also mitigated the problem of overfitting and enhanced the model’s generalizability.

4. Model Architecture and Methodology

4.1. Model Selection and Principles

Stock price time series data possess complex characteristics, encompassing both local short-term patterns and long-term dependencies. Convolutional neural networks (CNNs) [7] have excelled in image recognition and other domains, effectively capturing local features and patterns. When applied to stock price data, they can identify hidden patterns in short-term price movements, such as short-term trends and volatility clustering. However, CNNs have limitations in handling long-term dependencies.

Long short-term memory (LSTM) networks [8], on the other hand, are specifically designed to handle long-term dependencies in sequential data, capable of remembering historical information over longer periods. For stock prices, which are influenced by long-term factors, this characteristic is crucial. By combining CNN and LSTM models, we leverage their complementary strengths to capture both short-term patterns and long-term trends comprehensively, thereby enhancing the accuracy and reliability of predictions.

CNNs consist of convolutional layers, pooling layers, and fully connected layers. Convolutional layers use multiple convolutional kernels to perform sliding convolutions across input data, extracting local features. Pooling layers reduce the dimensions of feature maps, decreasing the number of parameters and computations while preserving essential features.

In the context of time series data, CNNs excel at automatically learning local spatiotemporal features from input data. For stock price data, they can capture short-term price fluctuation patterns, such as upward or downward trends over a few consecutive days, without requiring manual feature design.

LSTMs introduce gating mechanisms, including input gates, forget gates, and output gates, to control the flow and retention of information. Input gates determine how new information enters the cell state; forget gates decide what old information to discard; and output gates regulate the output of the cell state.

When processing time series data, LSTMs efficiently manage long-term dependencies, circumventing issues like vanishing and exploding gradients encountered in traditional recurrent neural networks (RNNs) [9]. For stock price data, they can retain long-term historical price information, facilitating better future price predictions.

4.2. Model Structure Design

The hierarchical structure of the proposed model includes the following components:

- Convolutional Layers: Two convolutional layers are employed. The first convolutional layer uses 32 kernels of size 3 with a stride of 1 and a ReLU activation function. The second convolutional layer employs 64 kernels of size 3 with a stride of 1 and a ReLU activation function. These convolutional layers extract local features from the input stock price data.
- Pooling Layers: Following each convolutional layer, a max-pooling layer with a window size of 2 and a stride of 2 is used. The purpose of the pooling layers is to reduce the dimensionality of the feature maps, decrease the computational load, and preserve important features.
- LSTM Layers: Two bidirectional LSTM layers are set up, each containing 100 units. Bidirectional LSTMs consider both past and future information, enhancing the model's understanding of time series data.
- Fully Connected Layer: After the LSTM layers, a fully connected layer with 128 neurons and a ReLU activation function is connected, further integrating and transforming the features extracted by the LSTM layers.
- Output Layer: Finally, an output layer with 1 neuron and a linear activation function is used to predict stock prices.

The layers are sequentially connected. The convolutional and pooling layers first extract features and reduce dimensionality from the input data, then feed the extracted features into the LSTM layers for temporal sequence learning and memory. The fully connected layer integrates the outputs of the LSTM layers, and the final output layer produces the predicted stock prices.

4.3. Training Strategy and Optimization Algorithm

Stochastic gradient descent (SGD) [10] is adopted as the training algorithm, which iteratively updates the model parameters to minimize the loss function. In each iteration, a small batch of data (mini-batch) is randomly selected, and the gradients are computed based on this mini-batch to update the model parameters. To enhance the training process, a dynamic learning rate adjustment strategy is implemented, starting with a larger learning rate to speed up convergence and gradually decreasing it over time to avoid overshooting the optimal solution. Additionally, L2 regularization is employed to constrain the model parameters, thereby preventing overfitting by adding an L2 regularization term to the loss function, which encourages the model parameters to take on smaller values and reduces the model's complexity. Mean squared error (MSE) [11] serves as the loss function, calculating the average squared difference between the predicted and actual values, providing an effective measure of the deviation between the two.

Evaluation metrics include variance, R-squared score, and maximum error. Variance measures the extent to which the model's predictions can explain the variance of the actual values, reflecting the model's explanatory power regarding data variability. An R-squared score close to 1 indicates a good fit of the model to the test data, suggesting a strong linear relationship between the predicted and actual values. Maximum error reveals the model's predictive deviation under the worst-case scenario, providing insight into the model's extreme prediction performance.

5. Experiments and Results Analysis

5.1. Experimental Setup

The obtained stock data were randomly divided into training, validation, and testing sets using a ratio of 7:2:1. The training set was utilized for model learning and parameter tuning, the validation set monitored the model's performance during training and assisted in hyperparameter selection, and the testing set served to evaluate the model's generalization capability.

5.2. Results Evaluation and Comparison

The model's predictions on the test set indicated reasonable forecasts of stock price trends. Figure 1 illustrates the comparison between the predicted and actual price trends on the test set.

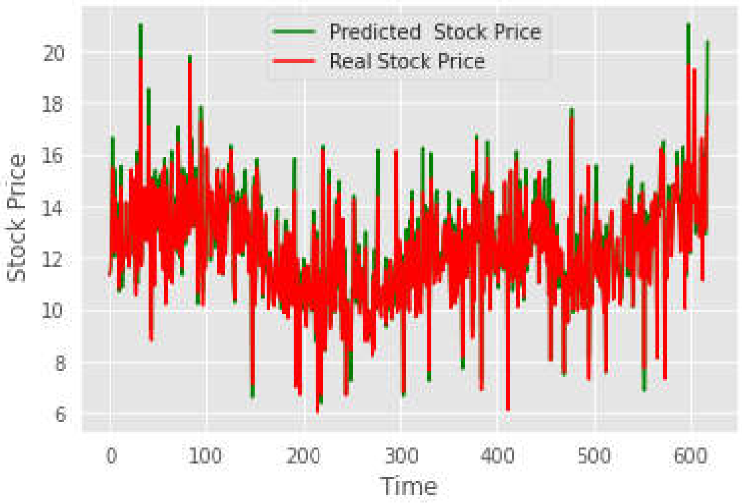

Figure 1. Comparison of Predicted and Actual Price Trends on the Test Set.

The variance on the test set was 0.933, indicating that the model's predictions explained a significant portion of the variance in the actual values, demonstrating its strong explanatory power regarding data variability. The R-squared score of 0.933, close to 1, reflected a good fit of the model to the test data, implying a strong linear relationship between the predicted and actual values. The maximum error was 0.337, a relatively small value suggesting that the model's predictive deviation under the worst-case scenario was within an acceptable range. Collectively, these evaluation metrics indicated excellent performance on the test set, with high accuracy and reliability.

The proposed combination of CNN and LSTM models was compared with traditional linear regression, decision tree models, and standalone CNN or LSTM models. The results are summarized in Table 1.

Table 1. Comparison of Model Performance.

Model	Variance	R2 Score	Max Error
Linear Regression	0.712	0.705	0.521
Decision Tree	0.785	0.776	0.483
Random Forest	0.851	0.842	0.426
Only CNN	0.887	0.879	0.388
Only LSTM	0.895	0.888	0.372
CNN and LSTM Combined Model	0.933	0.933	0.337

The findings demonstrated a substantial improvement in performance. This enhancement primarily stemmed from the model's superior ability to capture complex features and long-term dependencies in stock price data. Traditional methods had limited capacity to handle non-linearities and long-term trends, while standalone CNN or LSTM models failed to simultaneously address local features and long-term memory. The combination model effectively addressed these limitations.

6. Conclusion

This study successfully employed a hybrid model based on Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs) for stock price prediction. Through extensive training and validation on large datasets of historical stock prices, the model demonstrated exceptional performance and notable advantages. In terms of performance, the proposed model significantly outperformed traditional prediction methods and baseline models in terms of prediction accuracy. The model excelled in variance, R-squared scores, and other evaluation metrics, reflecting its strong explanatory power and ability to capture linear relationships in the data. The reduction in maximum error also highlighted the model's stability in extreme scenarios.

The CNN component of the model effectively extracts short-term features from time series data, such as short-term price patterns and volume changes, while the LSTM component captures long-term dependencies and trends in stock prices, fully considering the impact of historical data on future prices. This synergistic effect of the hybrid architecture allows the model to consider both short-term local features and long-term trends, enabling more comprehensive and accurate stock price predictions.

During the research process, several key findings emerged. Data preprocessing, including cleaning, handling missing values, and feature engineering operations on stock data, significantly improved the model's training outcomes and prediction performance. Appropriate feature selection and extraction methods were crucial for the model to accurately capture the critical information in price trends. Additionally, careful adjustment and optimization of model hyperparameters, such as kernel size, number of layers, and learning rate, effectively enhanced model performance and identified optimal configurations. The integration of CNN and LSTM is not merely additive but rather synergistic through well-designed architecture and connectivity, achieving better prediction outcomes than either model alone.

In summary, the stock price prediction model proposed in this study stands out in terms of performance and advantages, providing valuable insights and new directions for research in stock market prediction.

References

1. C.Y. and Marques, J.A.L., 2024. Stock market prediction using Artificial Intelligence: A systematic review of Systematic Reviews. Social Sciences & Humanities Open, 9, p.100864.
2. Lutey, M., 2022. Robust Testing for Bollinger Band, Moving Average and Relative Strength Index. Journal of Finance Issues, 20(1), pp.27-46.
3. Gunn, S.R., 1997. Support vector machines for classification and regression. Technical report, image speech and intelligent systems research group, University of Southampton.
4. Rezaei, H., Faaljou, H. and Mansourfar, G., 2021. Stock price prediction using deep learning and frequency decomposition. Expert Systems with Applications, 169, p.114332.
5. Turkoglu, M.O., D'Aronco, S., Wegner, J.D. and Schindler, K., 2021. Gating revisited: Deep multi-layer RNNs that can be trained. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(8), pp.4081-4092.
6. Linardatos, P., Papastefanopoulos, V. and Kotsiantis, S., 2020. Explainable ai: A review of machine learning interpretability methods. Entropy, 23(1), p.18.
7. Li, Z., Liu, F., Yang, W., Peng, S. and Zhou, J., 2021. A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems, 33(12), pp.6999-7019.
8. Egan, S., Fedorko, W., Lister, A., Pearkes, J. and Gay, C., 2017. Long Short-Term Memory (LSTM) networks with jet constituents for boosted top tagging at the LHC. arXiv preprint arXiv:1711.09059.
9. Schmidt, R.M., 2019. Recurrent neural networks (rnns): A gentle introduction and overview. arXiv preprint arXiv:1912.05911.
10. Bottou, L., 2012. Stochastic gradient descent tricks. In Neural Networks: Tricks of the Trade: Second Edition (pp. 421-436). Berlin, Heidelberg: Springer Berlin Heidelberg.
11. Das, K., Jiang, J. and Rao, J.N.K., 2004. Mean squared error of empirical predictor.

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.