CNN-BASED STOCK PRICE FORECASTING BY STOCK CHART IMAGES

Jeongseok Bang[1]

Doojin Ryu[1,*]

Abstract

We exploit the recent development in deep learning technology to forecast stock price. Focusing on image-type big data, this study analyzes the possibility of predicting future stock prices using a convolutional neural network (CNN) model trained by visual representations of stock price data and technical indicators. We find that including technical indicators partially increases accuracy. The model with an input range of five days is the most accurate but is likely to be not appropriately learned, considering the recall, precision, and test datasets. On the contrary, the difference between the precision and label means of the test dataset is greatest when we train the model using images of the past 20 days along with technical indicators.

Keywords: Convolution neural networks; Stock chart image; Stock price forecasting; Technical indicators

JEL Classification: G11, G12, G17

1. Introduction

One of the most important questions in financial research is whether the stock market can be systematically predicted. According to Fama's efficient market hypothesis, the stock market efficiently reflects the information in prices. In other words, the more efficient the market is, the faster it reaches equilibrium owing to instant arbitrage, and stock prices follow a random process. If the semi-strong efficient market hypothesis is correct, it is impossible to achieve systemic arbitrage using information publicly obtained from the market. However, following the work of Fama and French (1992), studies find increasing evidence of anomalies in the stock market. Jegadeesh and Titman (1993) propose the concept of stock price momentum, suggesting that long-term price trends can generate excess returns that existing asset pricing models cannot explain. This idea has led to the establishment of effective arbitrage strategies ranging from days to weeks (De Groot, Huij, and Zhou, 2012; Novy-Marx and Velikov, 2016).

Consequently, numerous studies have investigated whether the stock market can be predicted based on cross-sectional analyses. Studies explore stock market forecasting by autoregressive integrated moving average, vector autoregressive, and autoregressive conditionally heteroskedastic. However, due to the inherent complexity of stock markets, recent research has increasingly focused on non-linear approaches, particularly deep learning, to predict asset prices (Chen et al., 2018; Nakano, Takahashi, and Takahashi, 2018; Shen and Shafiq, 2020). According to Bustos and Pomares-Quimbaya (2020), studies use support vector machines, tree-ensemble models, and multiple perceptron models to make stock market predictions. Furthermore, as the recurrent neural network models evolve, researchers develop stock market prediction models with long short-term memory and gated recurrent units using stock prices, technical indicators, and macroeconomic data as inputs (Jiang et al., 2020). Meanwhile, as many investors predict future stock prices through stock price charts, studies also attempt to predict short-term price changes using chart images instead of numerical data (Chen et al., 2021). If a significant number of investors recognize stock price movements as images and trade stocks, stock prices forecasting by stock price images may be effectual (Jiang, Kelly, and Xiu, forthcoming). If meaningful prediction patterns originated from price and numerous technical indicators exist, then they can be extracted using an image-based deep learning model.

1 Department of Economics, Sungkyunkwan University, Seoul, Republic of Korea

In a related study, Chen et al. (2021) propose a graph convolutional feature-based convolutional neural network (GC-CNN) model that considers both individual stocks and the correlations between stocks. They find that GC-CNN-based strategies perform better than other models do. They analyze stock price charts and abstract images of technical indicators using a Daul-convolutional neural network (CNN) model. Jiang, Kelly, and Xiu (forthcoming) present a strategy to train CNNs with stock chart images and construct long-short portfolios based on the probability that stock prices will rise. They find that this strategy has a relatively high Sharpe ratio compared with other short-term momentum strategies and is robust to transaction costs.

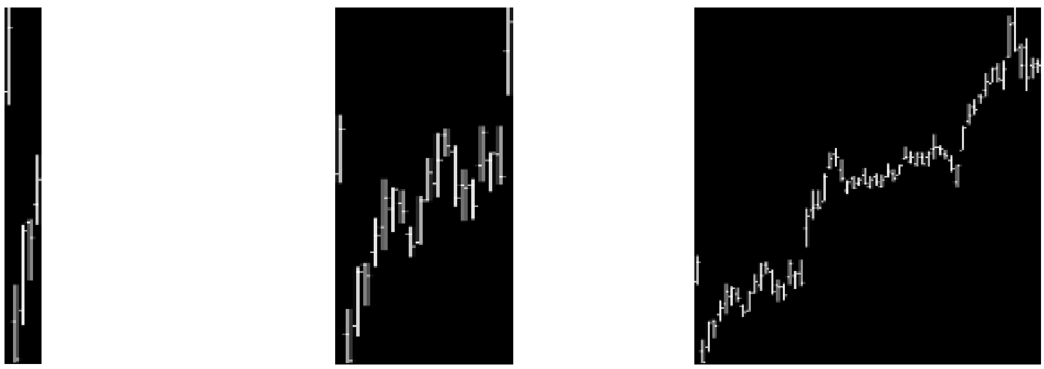
This study expands on Jiang, Kelly, and Xiu's (forthcoming) research to investigate potential stock price patterns. The patterns can be captured by a CNN model which is learned by stock price images. Thus, we consider common technical indicators, such as the Bollinger band and the relative strength index (RSI), in addition to the moving averages used in previous studies to examine whether adding technical indicators widely used in the industry improves the model's predictability. We train our CNN model using opening, high, low, and closing prices; trading volumes; moving averages; Bollinger bands; and the RSI in bar chart images with input windows of 5, 20, and 60 days. We find that forecasting the next five days using the prior five days performs better than the other input windows do. In addition, the trading volumes, moving averages, and partial Bollinger bands provide more accurate predictions when they are expressed as images. However, when we analyze the recall, precision, and label mean of the test dataset, the high accuracy of five days is only due to the label mean of the test data, suggesting that learning the model with five days input window may be spurious. Instead, using 20- and 60-day windows likely leads to more meaningful learning despite their relatively low accuracy.

2. Data and Methodology

2.1 Stock Price Image Transformation

We use the Yahoo Finance API to obtain stock price data from the top 100 NASDAQ-listed companies by market capitalization as of June 2023. The data cover the period from January 1, 2000, to December 31, 2022. We convert the numeric data, including opening, high, low, and closing prices and trading volumes, into bar chart images. To this end, following Jiang, Kelly, and Xiu's (forthcoming) methodology, we convert numerical price data into a matrix between [0, 255]. We use 5-, 20-, and 60-day input windows and 5-day output windows. Through this process, we derive the images shown in Figure 1. Each image has a size of (256, input window size × 3). Of the three pixels assigned to each date, the first pixel represents the opening price, the second pixel shows a line connecting the low and high prices, and the third pixel represents the closing price. The price information is simplified, as it is mapped to a single integer out of the 256 possible values. The highest and lowest values in the given price series correspond to the top and bottom of the image, respectively.

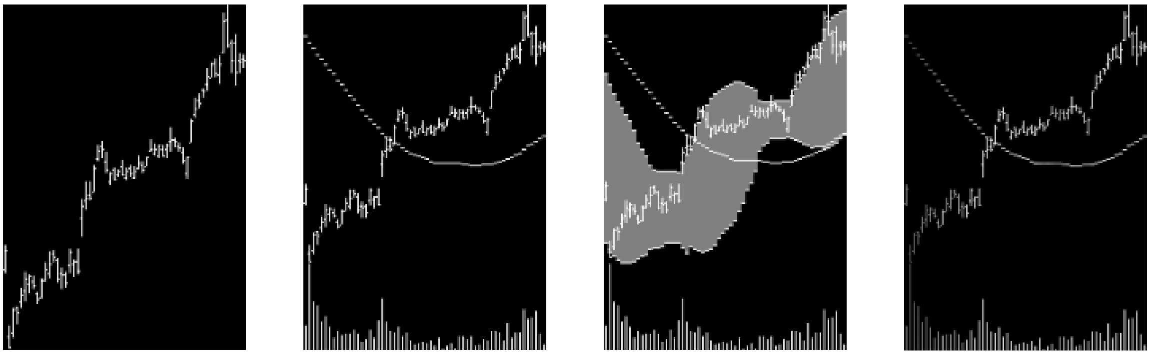
Figure 1. Input images for 5-, 20-, and 60-day input windows



(a) 5-day window (b) 20-day window (c) 60-day window

The images used to train the CNN model are constructed in four versions, shown in Figure 2. Version 1 represents the opening, high, low, and closing prices in bar chart formats. Version 2 depicts the moving average lines of the closing prices corresponding to each input window in the top 75% of the image and the trading volume data in the bottom 25% of the image. Version 3 adds the upper and lower Bollinger bands (20-day windows) based on closing prices. The area between the upper and lower bands is shaded at 128, the midpoint between zero and 255. Version 4 follows the same approach as version 2, but each column is multiplied by the RSI (calculated on a 14-day basis), and the products are expressed as brightness. Versions 3 and 4 allow us to incorporate additional technical indicators, such as Bollinger bands and the RSI, into the images.

Figure 2. Input image data using various indicators (Versions 1, 2, 3, and 4)



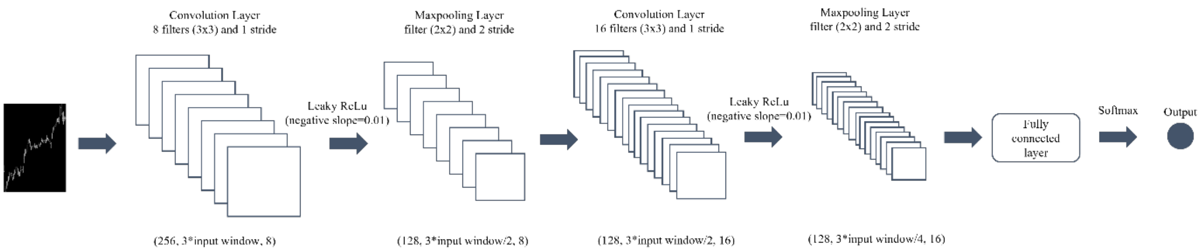
(a) Version 1 (b) Version 2 (c) Version 3 (d) Version 4

We construct the output data as follows. The output equals one if the stock price increases over the next five days and zero otherwise. The training data span the period from January 1, 2000, to December 31, 2018, and the test data cover the period from January 1, 2019, to December 31, 2022.

2.2 CNN Architecture

In this study, we use a CNN model because the data take the form of images. The structure of the CNN used in this study consists of two layers composed of convolution, activation functions, and pooling for each, as depicted in Figure 3. In the first layer, we use the convolutional layer, a set of eight filters with a 3 × 3 shape, to train the model on the stock price chart images. This process captures the spatial patterns in the images. To preserve the original image shapes, we apply a padding of one. We also use a leaky ReLU with a gradient of 0.01 for the activation function. A max-pooling layer with a 2 × 2 filter and a stride of two is adapted to reduce the height and width of the image by half. We use the Adam optimizer with a batch size of 50, 10 training epochs, and a learning rate of 0.001.

Figure 3. CNN architecture



3. Results

Table 1 represents the numbers of image samples from the training and test data used in the analysis. Because the entire stock price time series for each firm is split by the given input window, the number of observations is largest when the input window is set to five days and is smallest when the input window is 60 days.

Table 1. Numbers of image samples in the training and testing data

	Version 1	Version 2	Version 3	Version 4
5 days	(83094, 19246)	(83830, 19444)	(83545, 19435)	(83756, 19444)
20 days	(20934, 4861)	(20840, 4858)	(20840, 4858)	(20840, 4858)

60 days	(6948, 1556)	(6853, 1553)	(6853, 1553)	(6853, 1553)

Table 2 presents the accuracy, precision, recall, and F1 scores of the trained CNN model using the input windows and the four image versions. The values in bold are the highest scores for each metric. Overall, the 5-day input window provides the highest accuracy, and version 3, which includes Bollinger bands, is the most accurate version for this window. For both the 5-day and 60-day input windows, versions 2-4, which include additional technical indicators, are more accurate than version 1. In addition, for the 60-day input window, version 2 is the most accurate.

Table 2. Results of CNN models

		Accuracy	Precision	Recall	F1 Score
5 days	Version 1	0.5142	0.5482	0.9199	0.6870
	Version 2	0.5377	0.5468	0.8476	0.6648
	Version 3	0.5417	0.5483	0.6646	0.6008
	Version 4	0.5304	0.5485	0.8145	0.6555
20 days	Version 1	0.5069	0.4958	0.5932	0.5402
	Version 2	0.5010	0.5053	0.5510	0.5272
	Version 3	0.5037	0.5077	0.4669	0.4864
	Version 4	0.5023	0.4893	0.5902	0.5350
60 days	Version 1	0.4968	0.4932	0.5756	0.5312
	Version 2	0.5119	0.5016	0.6071	0.5493
	Version 3	0.5061	0.4833	0.5900	0.5314
	Version 4	0.5029	0.4977	0.5756	0.5338

Version 4 with a 5-day window achieves the highest value of precision that indicates how well the predicted classes align with the real classes. For the 20-day input window, versions 2-3 with additional technical indicators provide more precision than version 1 does. For the 60-day input window, version 2, which includes moving averages and trading volumes, exhibits the highest precision. Recall measures how accurately the predicted classes match the actual classes for a specific class. The recall values for the 5-day window are close to one. However, for the 20-day and 60-day input windows, the recall values are around 0.4-0.5. Finally, the F1 score, which is the harmonic mean of precision and recall, is generally higher for the 5-day input window, indicating a balance of precision and recall.

Considering the overall accuracy, one may conclude that the CNN model performs best when the input window is five days. However, the extremely high recall values in the binary classification indicate that the model may have learned to select only one class instead of achieving well-fitted training. To validate the accuracy of the classes predicted using the trained CNN model, we compare them with the true ratio of label 1 in the test dataset. Table 3 presents the precision, the true ratio, and the differences between them. Versions 1-4 have relatively high precision for the 5-day input window, with relatively small differences ranging from 0.0007 to 0.0024. This result supports that the high accuracy observed in the model trained with 5-day images results from the statistical characteristics of the test data rather than proper model training. However, the differences in precision and true ratio for the 20-day and 60-day windows are relatively high, reaching a maximum of 0.0135. This result suggests that model training is not entirely meaningless despite its low accuracy. Version 3 with a 20-day image window, which has an accuracy of 0.5037, has the largest difference. Thus, despite being less accurate than the model using a 5-day input window, the models with longer image windows are likely to have been trained more meaningfully. These issues seem to stem from the low predictive power of the models and can potentially be resolved when the accuracy, precision, and recall are all high.

Table 3. Differences between the precision and the true ratio

		Precision	True ratio	Difference
5 days	Version 1	0.5482	0.5466	0.0016
	Version 2	0.5468	0.5461	0.0007
	Version 3	0.5483	0.5461	0.0022
	Version 4	0.5485	0.5461	0.0024
20 days	Version 1	0.4958	0.4956	0.0002

	Version 2	0.5053	0.4942	0.0111
	Version 3	0.5077	0.4942	0.0135
	Version 4	0.4893	0.4942	-0.0049
60 days	Version 1	0.4932	0.4891	0.0041
	Version 2	0.5016	0.4900	0.0116
	Version 3	0.4833	0.4900	0.0067
	Version 4	0.4977	0.4900	0.0077

4. Conclusion

Many studies propose stock price forecasting with machine learning and deep learning techniques to effectively predict the stock market. Some of these approaches attempt to process stock price data as images and train deep learning models. This study represents stock movements as images and analyzes whether image-based deep-learning models can predict future stock prices. We convert the stock price and technical indicator data into images for input windows of 5, 20, and 60 days and train CNN models using these images. The results show that the highest accuracy is achieved when the input window is five days. However, the scrutiny based on the precision and the true ratio of testing data shows that learning over a shorter window may not extract potential patterns in stock price flows. Instead, we observe that more meaningful learning occurs when we use 20- and 60-day windows. Including technical indicators, such as trading volumes and moving averages, seems to improve accuracy, although the effects of Bollinger bands are partially evident. Because it exploits a cutting-edge method to uncover hidden patterns in stock price movements that have not been easily captured due to the dynamic nature of financial markets.

Reference

Bustos, O., and Pomares-Quimbaya, A. (2020). Stock market movement forecast: A systematic review. *Expert Systems with Applications*, 156, 113464.

Chen, L., Qiao, Z., Wang, M., Wang, C., Du, R., and Stanley, H. E. (2018). Which artificial intelligence algorithm better predicts the Chinese stock market?. *IEEE Access*, 6, pp. 48625-48633.

Chen, W., Jiang, M., Zhang, W. G., and Chen, Z. (2021). A novel graph convolutional feature based convolutional neural network for stock trend prediction. *Information Sciences*, 556, pp. 67-94.

De Groot, W., Huij, J., and Zhou, W. (2012). Another look at trading costs and short-term reversal profits. *Journal of Banking and Finance*, 36(2), pp. 371-382.

Fama, E. F., and French, K. R. (1992). The cross-section of expected stock returns. *Journal of Finance*, 47(2), pp. 427-465.

Jegadeesh, N., and Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *Journal of Finance*, 48(1), pp. 65-91.

Jiang, J., Kelly, B. T., and Xiu, D. (Re-) Imag (in) ing price trends. *Journal of Finance*, Forthcoming.

Jiang, M., Liu, J., Zhang, L., and Liu, C. (2020). An improved Stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms. *Physica A: Statistical Mechanics and its Applications*, 541, 122272.

Nakano, M., Takahashi, A., and Takahashi, S. (2018). Bitcoin technical trading with artificial neural network. *Physica A: Statistical Mechanics and its Applications*, 510, pp. 587-609.

Novy-Marx, R., and Velikov, M. (2016). A taxonomy of anomalies and their trading costs. *Review of Financial Studies*, 29(1), pp. 104-147.

Shen, J., and Shafiq, M. O. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data*, 7(1), pp. 1-33.