FINANCIAL TIME SERIES CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK AND CANDLESTICK CHARTS

REPORT

Yicheng Wang
Department of Computer Science
University College London
yicheng.wang.21@ucl.edu

March 30, 2022

ABSTRACT

Financial time series prediction has been considered a very challenging task since there are numerous factors affecting the financial market. In this work, I present a Deep Learning approach to predict the market state on S&P 500 index, by using Convolutional Neural Network(CNN) and candlestick chart representation. From the stock historical intraday data, I converted the time series for each trading day into a candlestick chart image, then fed the candlestick chart images as input into a CNN network for predicting the closing market state. A market state is defined by a binary classification which compares the daily close price with the price change in the first hours of the day. The performance of my method is evaluated in stock market prediction with a promising result of 73.4% accuracy for S&P 500. Finally, I have shown by constructing two simple trading strategies, the model could be a support for traders to make trading decisions.

1 Introduction

Stock market trend prediction is always a very challenging task due to its inherent complex, random and dynamic nature. The price of a certain stock doesn't only reflect the value of a certain company or its future potential earnings, but also the aggregation of millions of investors' sentiment. Also, there are a large number of uncertainties and a plethora of factors involved that may affect the market value such as politics, economic conditions, unexpected events etc.

For decades, there has been numerous research from various sources attempting to extract predictions on stock price trends from available information to perform profitable trades. With the modern technology advances, machine learning techniques and deep learning neural networks have shown potential in making such predictions.

In 2011, Kara [6] proposed a method using Artificial neural networks(ANN) and support vector machines (SVM) for predicting the direction of movement in the daily Istanbul Stock Exchange (ISE) National 100 Index. The study shows that by passing 10 technical indicators as the feature input, ANN model performance in prediction is significantly better than SVM. Later in 2016, Di Persio [3] presented a novel Artificial Neural Network approach to predict stock market indices. The author compared the following models' performance on predicting the trend movements in S&P500 index : the Multi-layer Perceptron (MLP), the Convolutional Neural Net-works (CNN), and the Long Short-Term Memory (LSTM) recurrent neural networks techniques. He then proposed a novel approach based on a combination of wavelets and CNN, which outperformed the basic neural network ones. In 2019, Kusuma and Ou [7] came up with an interesting study by using candlestick chart representation as the feature input. In the paper, they performed a CNN neural network on stock price trend prediction and showed a 'promising' result with 92.2% and 92.1 % accuracy for Taiwan and Indonesian stock market dataset respectively. I failed to achieve the same result while testing it out on the same dataset with the same model configuration parameters. The evaluation metric of their project is suspiciously high, however, it gave me the inspiration for this report.

The objective of this project is to study the applicability of machine learning techniques, in particular the CNN Convolutional Neural Network, on the problem of stock market prices trend forecasting. In this study, I convert the intraday time series data to candlestick chart representation as feature engineering. Then, using the dataset as input, I trained the CNN model through various parameters to find the optimum model.

To examine the predictive performance of the CNN model, I compare it with the traditional machine learning models KNN, as well as the random prediction model. In addition, by using exactly the same experiment set-up, I compare the result of candlestick-chart-CNN with the wavelet-transform-CNN proposed by Du and Fernandez-Reyes. [4]

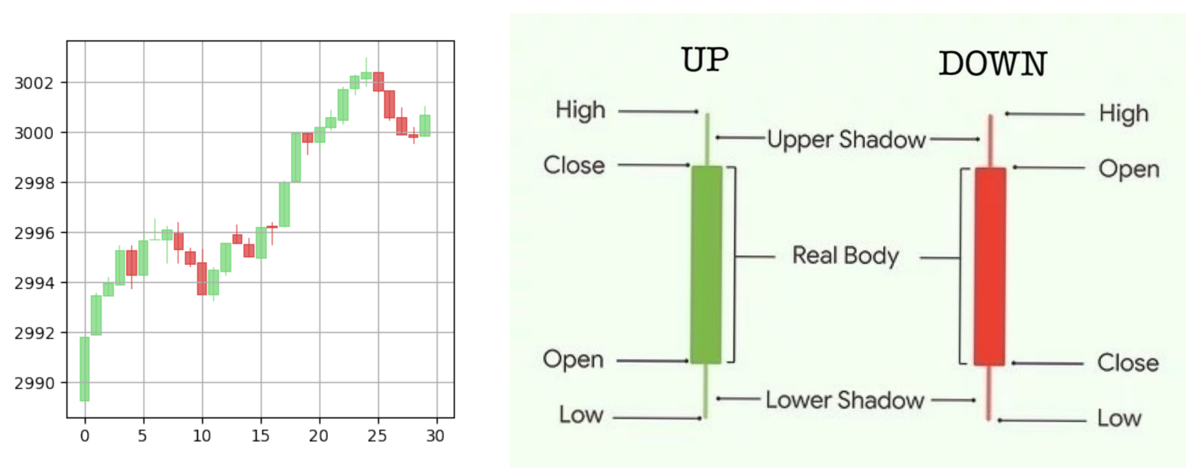
2 Methodology

2.1 Candlestick charts

Candlestick charts, so called Japanese candlestick chart, are thought to have been developed in the 18th century by Munehisa Homma, a Japanese rice trader. [9] Candlestick chart is currently one of the most popular plots for technical analysis in the financial market. Figure 1a shows an example of a 1-minute candlestick chart over 30 minutes of a trading day. A candlestick chart is made up of multiple sticks that each stick represents four pieces of information for a typical trading period: Open, Close, Highest, Lowest Price.

Figure 1b gives an example of two candlesticks. The opening and closing prices are indicated in the 'real body'. The highest and lowest prices are depicted by the upper and lower shadows. In the US stock market, a green real body means that the closing price is higher than the opening price over a certain time period. Otherwise, a red real body means the closing price is lower than the opening price. Candlestick provides a clear visual representation for traders to understand the relationship between High and low as well as open and close, and the trend movement over different time scales. Across the years, identifying candlestick patterns has been proven to be a successful auxiliary tool for traders in decision making. [8]

Due to the reason that candlestick provide a visual representation of time series and its shown that human eyes can extract useful information from it, in this study, I choose candlestick chart images as the features input to investigate the power of machine learning on extracting the features from images and its predictive performance on time series.



(a) 1-minute Candlestick chart over 30 minutes

(b) Two types of Candlesticks

2.2 Convolutional neural network

Convolutional neural network (CNN) was introduced in the 1980s by Yann LeCun. [1] In this study, I chose Convolution Neural Network (CNN) over other neural networks for the candlestick image prediction because It has proven very effective in areas of image recognition, processing, and classification. CNN is a feed-forward artificial neural network which each input image passes through a series of convolutional kernels: convolutional layers, pooling layers, fully connected layers. The idea behind CNN is to extract key features by reducing the image size into a form that is easier for processing without losing features which are critical for getting a good prediction. The purposed CNN structure used in this project is shown in figure 2.

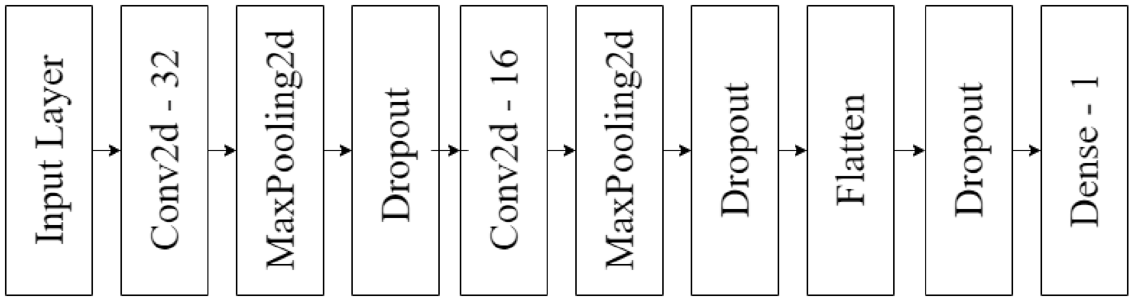


Figure 2: The Architecture of Used Convolutional Neural Network

2.2.1 Convolutional layer

The convolutional layer is a layer for applying the convolution operation on the input data, passing the result to the next layer. Convolution is the processing of adding up all the elements in its receptive field into a single value, by a weighted kernel. CNNs are made up of convolutional layers that have an input map I, a set of filters K, and biases b.

In the case of images, the input size usually comes up with height H, width W and Channels C, where for a colored image, channels = 3 (red, blue, green). The output for a convolution operation is:

$$(I * K)_{i,j} = \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} \sum_{c=1}^{C} K_{m,n,c} \cdot I_{i+m,j+n,c} + b \tag{1}$$

For a 1-channel image (grayscale image), where C = 1, the equation 1 can be transformed to:

$$(I * K)_{i,j} = \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} K_{m,n} \cdot I_{i+m,j+n} + b \tag{2}$$

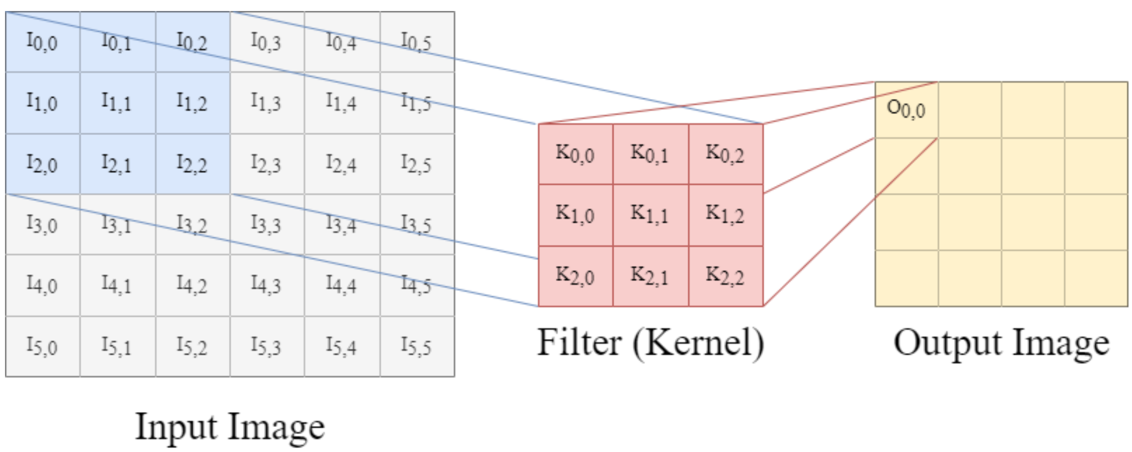


Figure 3: An Example of Convolution operation

As shown in figure 3 , the elements of an 6x6 input image (matrix form) in the blue window are element-wise multiplied with a 3x3 kernel weights in the red box to make a single value. Where from equation 2 , by ignoring the bias b, the first element pixel value of output image could be written as:

$$\begin{aligned}(I * K)_{0,0} = O_{0,0} &= \sum_{m=0}^{2} \sum_{n=0}^{2} K_{m,n} \cdot I_{m,n} \\ &= K_{0,0} \cdot I_{0,0} + K_{0,1} \cdot I_{0,1} + K_{0,2} \cdot I_{0,2} + ... + K_{2,2} \cdot I_{2,2}\end{aligned} \tag{3}$$

The objective of the convolutional layer is to extract high-level features from a given image. In addition to the convolutional layer, the output is usually passed through an activation function such as the Relu layer, before entering the next layer. Relu (Eq.4) is the most commonly used non-linear activation function in CNN.

$$f(x) = max(0, x) \tag{4}$$

2.2.2 Pooling layer

The pooling layer is responsible for reducing the spatial size of the data representation, which helps decrease the amount of parameters and computation, in other words, makes it more efficient for training the network. Also, it is a way for controlling the overfitting problem, where overfitting happens when a model learns the detail and noise in the training data too well so that it cannot generalize on new data. There are several types of pooling functions such as average-pooling, max-pooling, L2 norm pooling etc. The most common choice is max-pooling, which returns the maximum output from a certain window. In this project, I choose max-pooling as the pooling layer, because max pooling selects the brighter pixels from the image, which is helpful when the background of the image is dark.

2.2.3 Fully connected layer

The fully connected layer is a traditional Multi Layer Perceptron(MLP) that uses an activation function in the output layer. Every neuron in the previous layer is connected to each neuron in the next layer. Given an input image, the purpose of convolutional layer and pooling layer is to extract features and reduce dimensionality. The fully connected layer is used to combine the extracted features to classify the images. For binary classification, the last classification layer has only one neuron with the sigmoid activation function, where the sigmoid function, so called as Logistic Activation function, defined by $S(x) = 1/(1 + e^x)$, takes any real value as input and outputs values in the range 0 to 1.

2.3 Feature engineering

2.3.1 Data collection

In this project, I used the same dataset provided in week 3 wavefin project, which contains totally 2778 days of historical minutes data of S&P 500 index. This dataset is already processed that doesn't contain NA/NaN value. And each day has 390 data points where each data point has five attributes: 'Open', 'Close', 'High', 'Low' prices and 'Volume'. Considering that most of the data don't include 'Volume', I would not include it as a feature in my report.

2.3.2 Data splitting

In machine learning, training set and test set splitting is usually performed by shuffling the full dataset first. However, in financial time series related tasks, we cannot shuffle the total dataset before splitting since this process leads to a risk of partially fitting to testset or validation set, which gets very good metrics during training but poor results when applying to real-time data. [2] On the other hand, in the real-world situation, it is impossible to use future's data to train the current model, because we don't have the future information yet. In this project, I perform the splitting using the first ten years as the training set and last year as the test set. This splitting setup is identical to the wavelet-transform-CNN project[4], making it easier for performance comparison.

2.4 Experiment design

For better comparison, this study uses the same labeling method with wavelet-transform-CNN project. The time series stock forecasting task is considered as a binary classification problem with the output label 0 or 1 corresponding to 'Up' or 'Down'. On a typical trading day, the US stock market trades lasting 390 minutes. In the experiment I use the first 360 minutes in a trading day as the input for feeding the models to predict the closing market state.

The architecture of the proposed method is shown in Figure 4. The first 360 minutes of proceed time series data is converted to a 3-channel candlestick image, and each candlestick image is corresponding to one label: 0 or 1. The labels 'Up' and 'Down' are calculated by comparing the average price over 360 minutes with the stock market's closing price:

$$Label_y = \begin{cases} 1, \ \sum_{i=1}^{360} C_i/360 < C_{390} \\ 0, \ \sum_{i=1}^{360} C_i/360 >= C_{390} \end{cases} \tag{5}$$

Where C_i is the closing price of i-th minutes, i.e C_{390} is the closing price of 390th minute which is the closing price of that trading day.

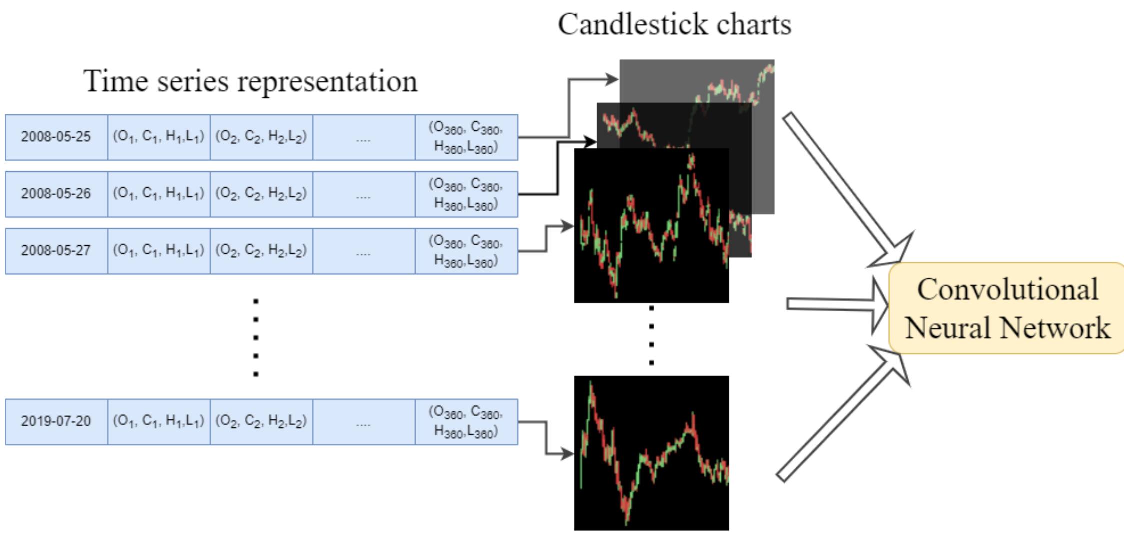


Figure 4: The Purposed Methodology Design.

The authors of the wavelet-transform-CNN project mentioned that the reason for choosing this labeling method is it yields a wider distribution of log-return between average closing price of first 360 minutes and closing price, which generates a larger margin that makes the convolutional Neural Network easier to learn.[4]

2.5 Baseline models

It's always helpful to understand the data by comparing the result with some baseline models. In this study, I compare the performance of the proposed method with the following models:

- **Random Null model**: a null model that randomly predicts the label 0 or 1
- **K-Nearest-Neighbour (KNN)**: a classical supervised machine learning algorithm which calculates the similarity between the input sample and the k training examples to determine the class to which the object is most likely to be allocated.
- **Wavelet-Transform-CNN** [4]: The raw time series are converted into feature images by calculating the log return of stock prices and then applying the wavelet transform. The resulting denoised wavelet representation of the data passed through a convolutional neural network with certain structure for time series binary classification.

2.6 Evaluation metrics

For evaluating the model performance, evaluation metrics around the algorithm performance are needed to compare results of my method and other methods including baseline models and other work. The metrics in this report are accuracy, precision, sensitivity (recall), specificity and F1- score. The formula to calculate these metrics are given below:

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \tag{6}$$

$$Precision = \frac{TP}{TP+FP} \tag{7}$$

$$Sensitivity(Recall) = \frac{TP}{TP+FN} \tag{8}$$

$$Specificity = \frac{TN}{TN+FP} \tag{9}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{10}$$

Where TP = Number of true positive values

TN = Number of true negative values

FP = Number of false positive values

FN = Number of false negative values

3 Results

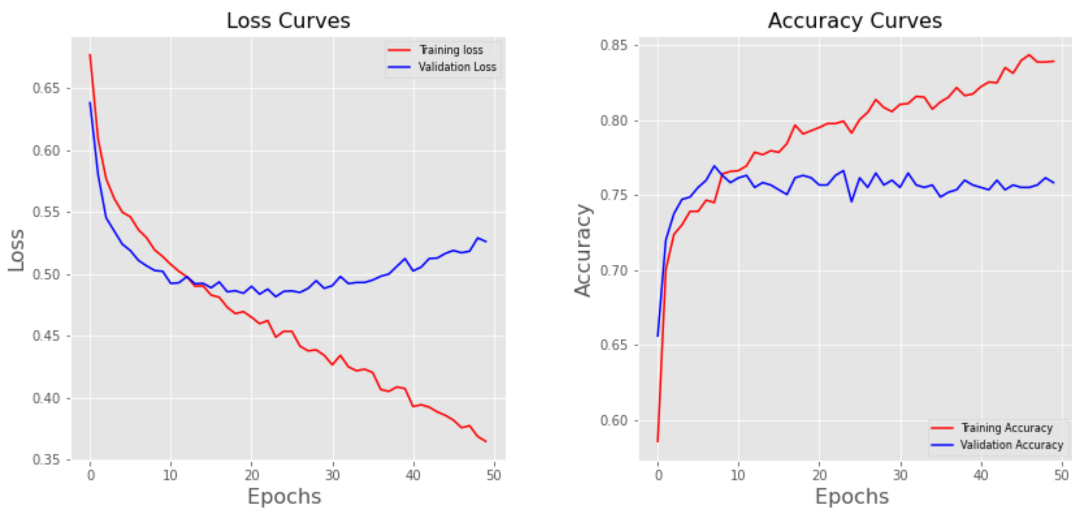


Figure 5: training loss/accuracy versus validation loss/accuracy

In this section, I perform classification based on the candlestick charts and CNN model, and then evaluate the model performance on S&P 500 test set by comparing with three baseline methods: Random Null, KNN and Wavelet-CNN.

Figure 5 indicates the loss curve and accuracy curve of the training set and validation set during the training process while the number of epochs is set to 50. We can see that the loss of the validation set starts to rise up after the 20th epoch and the accuracy of the validation set converges to around 0.75, which is the consequence of overfitting. By resolving this, I use a training strategy called early stopping[5]. I use a patience of 5, which means the model stop training if there is no improvement of validation loss during 5 consecutive epochs. In addition of patience, by testing various set of parameters, I achieve the relatively stable and accurate result by setting epoch = 30, learning rate = 1e-4.

Detailed results of the described methods and baselines are reported in table 1. We can see that all the methods have a higher evaluation performance compared to the random prediction. Among these four methods, the method of this project candlestick-CNN achieved the highest accuracy and f1 score, which outperforms Wavelet-CNN method by improving the accuracy and f1 score from 57.72%, 65.40% to 73.4%, 76.8% respectively. Furthermore, one of the most basic machine learning techniques KNN yields 63.3% of accuracy, which is slightly higher than Wavelet-CNN. This proves that candlestick chart representation does a better job on feature engineering in the financial time series classification task. Also, by comparing KNN and CNN methods, where the accuracy is improved by 10%, it underlines that CNN is more suitable and effective for capturing the key features from an image by convolution operation. From the promising results, we can conclude that the model performance on S&P 500 is competitive.

4 Discussion

From the result3, it's shown that the candlestick-CNN model provides a more accurate prediction than the wavelet-CNN model on S&P 500. In this section, I also implemented two simple trading strategies based on the predictions from

Model	Accuracy	Precision	Recall	Specificity	F1 Score
Random Null	0.463	0.537	0.395	0.553	0.455
KNN	0.633	0.621	0.750	0.621	0.679
Wavelet-CNN	0.577	0.608	0.707	0.401	0.654
Candlestick-CNN	**0.734**	**0.854**	0.699	**0.699**	**0.767**

Table 1: Results of models on S&P 500

candlestick-CNN for testing whether the work can help traders make investment decisions. The strategies are defined as following:

- **Long-only strategy**: For the prediction of S&P 500 on a given day, if prediction is 1, I buy the stock at the 360th closing price and sell the stock at the closing price of that day. No action is taken when the prediction is 0.
- **Long-short strategy**: For the prediction of S&P 500 on a given day, if prediction is 1, I buy the stock at the 360th closing price and sell the stock at the closing price of that day. Otherwise, if prediction is 1, I short the stock at the 360th closing price and close out the position at the closing price of that day.

As shown in figure 6, I compare the return of two strategies with S&P 500 index return of given period and random prediction. We can see that both strategies achieve a positive return at the end of date, where the long-short strategy has a 2.95% higher return than the S&P index return and the long-only strategy has a 1.43% higher return than the S&P 500 index return. Moreover, the graph indicates that there is a major drawdown in October 2018. Both strategies can effectively avoid drawdown and long-short strategy can even make profit by its short selling mechanism. Although this naive trading strategy simply assumes the trading could be successfully executed at every minute, which cannot be put straight in the real market, the model and the strategy can still be provided as a tool for help traders' decision making.

In conclusion, The results demonstrate that not only the candlestick-CNN model achieves a higher accuracy in financial time series trend prediction problems than wavelet-CNN, the model also has potential investment value in the financial market. For future work, it is essential to test if this model can carry out similar model performance on different dataset. For example, the next step is to apply the model on different stock indexes such as Dow Jones Index, Nasdaq Index, FTSE, etc. Also, it's important to test the model on different time periods. In this project, due to the difficulty of collecting minutes data, I use a fixed dataset throughout the project for better comparison. In financial time series tasks, due to its random, volatile, dynamic nature, testing on different dataset and different time periods is always necessary for achieving an accurate, robust model.

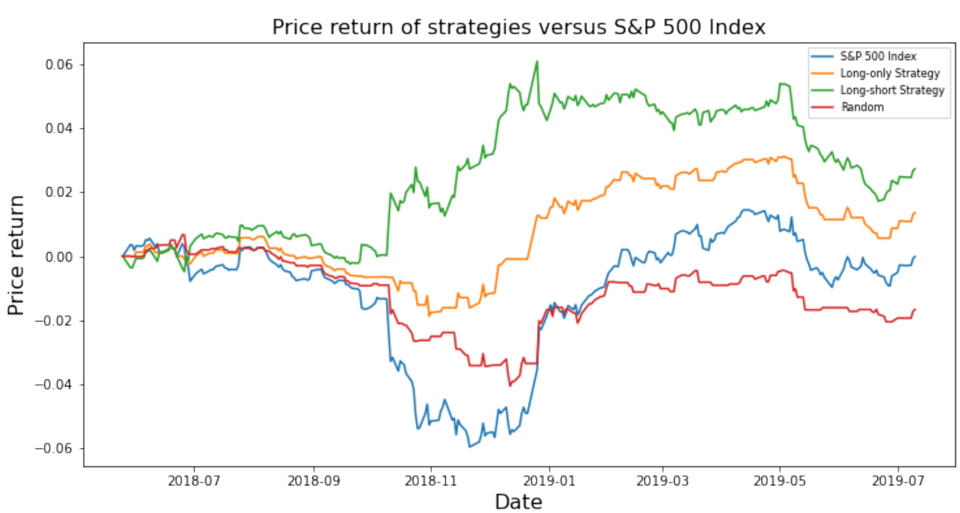


Figure 6: Comparison of Strategy price return and S&P Index

References

[1] Y. Bengio and Y. Lecun. Convolutional networks for images, speech, and time-series. 11 1997.

[2] F. Daniel. Financial time series data processing for machine learning, 2019.

[3] L. Di Persio and O. Honchar. Artificial neural networks architectures for stock price prediction: Comparisons and applications. 10:403–413, 01 2016.

[4] B. Du, D. Fernandez-Reyes, and P. Barucca. Image processing tools for financial time series classification, 2020.

[5] S. Jain. An overview of regularization techniques in deep learning. `https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/`, April 2018.

[6] Y. Kara, M. Acar Boyacioglu, and Ömer Kaan Baykan. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange. *Expert Systems with Applications*, 38(5):5311–5319, 2011.

[7] R. M. I. Kusuma, T.-T. Ho, W.-C. Kao, Y.-Y. Ou, and K.-L. Hua. Using deep learning neural networks and candlestick chart representation to predict stock market, 2019.

[8] T.-H. Lu, Y.-M. Shiu, and T.-C. Liu. Profitable candlestick trading strategies—the evidence from a new perspective. *Review of Financial Economics*, 21(2):63–68, 2012.

[9] G. L. Morris and R. Litchfield. *Candlestick charting explained: Timeless Techniques for trading stocks and futures*. McGraw-Hill, 2006.