Convolutional neural network for stock price prediction using transfer learning

Yohei Komori
WorldQuant University
Email: hurumori@gmail.com

ABSTRACT

The goal of this paper is to build a trading algorithm by applying image recognition neural network - Convolutional Neural Network(CNN) - to the 2D technical candle stick charts. First, this paper shows a research survey of the previous paper. Second, this paper explains the basic theory of CNN model and how it can works on chart images. Next, this project performs an experimental study of CNN on S&P 500 index from January 1, 1985 to June 30, 2020. The CNN model structure used in this paper is transferred from inception v3 with three additional layers, and the technical indicators used in the input chart image are simple moving average (25 days). The label data used in the model are categorical - either up, flat, or down. The model has 50% accuracy on the test set when conducting three-days ahead forecast, which is higher than the simple momentum strategy and contrarian strategy, indicating its high alpha generating potential. One-day ahead forecast and five-days ahead forecast have lower accuracy than the three-days forecast. This means you might have the best performance when you close your position at T + 3.

1 Introduction

There are many technical indicators for mean-reverting trading strategy including Simple Moving Average (SMA), Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), and bollinger bands. Most technical indicators are primarily shown as graph, and human arbitrary traders often decide whether to buy or sell by the shape of these graph. Previous studies, however, convert the technical indicators to simple numerical values and use these numerical values as signals. By compressing technical indicator charts to numerical values, it loses much information. The shape of the technical indicator charts or relative positions among several different technical indicators could provide more information than simple numerical values.

One solution to this problem might be the use of (multivariate) time series analysis or recurrent neural network. The other solution to this problem is to applying image recognition neural network - convolutional neural network - to the technical chart itself. This approach is resembling the human arbitrary traders. There are several research papers using this latter approach including Omer Berat Sezer, Ahmet Murat Ozbayoglu (2018) (2019) and Hyun Sik Sim, Hae In Kim, Jae Joon Ahn (2018) among others.However, much remain to be studied. There are many possible CNN models and many technical indicators which can be converted to images

The goal of this paper is to build a trading algorithm by applying image recognition neural network - Convolutional Neural Network (CNN) - to the 2D technical charts. First, this paper conducts a research survey of the previous papers. Second, this paper explains the basic theory of CNN model and how it can works on chart images. Next, this project executes an experimental study of CNN on SP 500 index from January 1, 1985 to June30, 2020.

Electronic copy available at: https://ssrn.com/abstract=3756702

2 Literature Review

Technical analysis is widely used among traders and investors but remained controversial among researchers. It assumes that the Efficient Market Hypothesis by Fama (1970) does not hold and the historical price information on a particular asset provides insights for the future performance of the asset. There are many indicators used in technical analyses and the selection of the right indicators is one of the most important topics in technical analysis. The two most popular technical trading indicators among researchers are Moving Average Convergence-Divergence (MACD) and Relative Strengths Index (RSI) and the most common way of applying these technical indicators is to use the numerical output signals from the indicators.

Another new approach to predict the asset price based on technical indicators is the machine learning. Support Vector Machine or Recurrent Neural Network (RNN) with a Long Short-Term Memory(LSTM) are the most popular machine learning model in predicting asset price. They take financial time series as inputs and predict the future price. Convolutional Neural Network (CNN) can take technical indicators' 2-D images as inputs and predict the stock price. In contrast to the traditional approach, the advantage of the CNN is that it can use not only the recent numerical signals but the whole history of the indicators which are drawn on the 2-D chart. It can also manage many indicators. Two early studies report that technical analysis does not yield profits. Jensen and Benington (1970) investigates the US stock market and conclude that past price information does not provide insights for the future performance. Allen and Karjalainen (1995) test genetic algorithm on technical trading rules for the SP 500 and reports that it does not earn excess return over buy-and-hold strategy.

In contrast, there are many recent empirical studies which support the trading rules based on technical indicators. Chong and Ng (2008) finds that MACD and RSI are profitable in FT30 index. Chong, Ng and Liew (2014) also shows that MACD and RSI generate positive returns in SP 500 or Dow Jones. Nor and Wickremasinghe (2017) shows that although MACD has poor performance, but RSI can make profit in Australian stock market. Rodriguez, Martel and Rivero (2005) shows that technical trading rules is superior to buy-and-hold strategies in Madrid Stock Market. Mckenzie (2007) and Yu Nartea Gan and Yao (2013) both investigate the emerging markets and shows the strong predictive power of technical trading rules.

Hao and Gao (2020) tests the performance of several machine learning models, and shows that Support Vector Machine (SVM) and LSTM have higher performance than the simple moving average model. Liu, Zhang, Wang, and Feng (2020) compares the performance of a CNN-LSTM hybrid model which take either financial time series as input or sentiment text data as input. Both show higher performance than the simple buy-and-hold strategy, but the latter sentiment model works better.

Sezer and Ozbayoglu (2018) tests the CNN model using 15 technical indicators as inputs on US stocks and shows it outperforms not only the buy-and-hold strategies but also the simple technical indicators rules including RSI and SMA. Sim, Kim, and Ahn (2019) uses 9 technical indicators as inputs for CNN and reports profit potential for the SP 500. Sezer and Ozbayoglu (2019) uses only the bar chart images of the Dow Jones 30 stocks prices. Although it underperforms the buy-and-hold strategy in the period of bull market, it reports that CNN model outperforms the buy-and-hold strategy in the period of bear market.

Although some old report shows that technical analysis does not yield positive returns, most recent studies report that the technical analysis outperforms the buy-and-hold strategy in stock markets. LSTM taking financial time series as inputs or CNN taking the 2-D stock price chart images as inputs are the new approach of adapting technical analysis and it also shows that technical analysis can outperforms the buy-and-hold strategy.

3 Convolutional Neural Network Overview

3.1 Basic Theory

CNN is a class of deep neural networks whose applications are focusing on imagine analytic. In detail, most CNNs are made of several convolution layers and pooling layers. Normally, convolution layers are responsible for extracting high-level features in the images through padding operation. This will pack information in a small field into a single feature rather than a grid of features. Pooling layer can help decrease the computational power required to process through pooling operation, which divides the entire graph into several sub-segments, and extra a feature for each segment. In this way, dimensionality reduction can be achieved. Meanwhile, pooling layer can also help extract dominant features. Figure 1 shows a simple structure of CNN. The CNN model used in our project is deeper than this simple image.

Electronic copy available at: https://ssrn.com/abstract=3756702

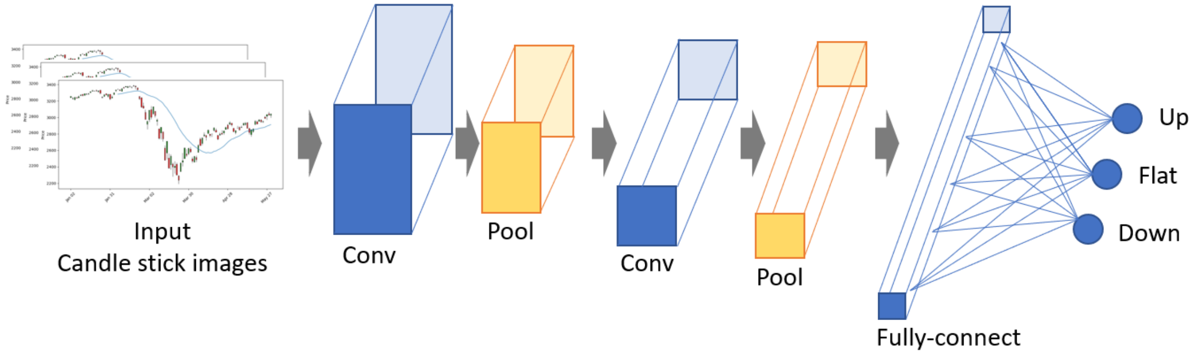

Figure 1: CNN structure

3.2 CNN model used in this project

There are two approaches when training a CNN model. The first approach is to build a model and train the model from scratch. The second approach is to use a pre-trained model. The pre-trained model is a model that is trained on a different training set and learned the features of the training set. In a deep CNN, the earlier layers detect a very basic features of images like horizontal / vertical lines or a curving lines. The middle layers detect general shapes or pattern - i.e., circle, square, or stripe patterns - and the latter layers distinguish specific images - i.e., dog pr cat. This means that the layers up to the middle of a trained CNN model can be used in other images. There are many pre-trained CNN models including AlexNet, ResNet, Inception and VGG among others. Instead of the train the CNN from scratch, using a pre-trained CNN model is the best practice in image recognitions (transfer learning). According to Sezer andOzbayo-glu (2019), we should avoid using a very deep model because it may cause overfitting and reduce the accuracy of the test set.

This study uses Inception v3 CNN model which has mid-depth layers created by google and trained on ImageNet. According to the official website, it has 78.1% accuracy on the ImageNet dataset. It has more than 70 layers including many convolutional layers, maxpooling layes, average pooling layers and fully-connected dense layers. ImageNet (image-net.org) is a very large database of images. It has 14 million images and they are labeled by more than 20,000 categories. We add three additional layers in the last end of Inception V3 and only trained those layers. The original learned parameters of inception V3 layers are not changed. The first layer of the three additional layers is a flattening layer, which flatten the data into one dimensional. The second layer is the fully connected layer with 1,024 hidden units, relu activation with 20% dropout rate. Dropout is a technique to avoid over-fitting. The final layers is the softmax layer for classification. In total, the model has 47,514,531 parameters with 38,539,267 trainable parameters (for the the last three layers) and 8,975,264 non-trainable parameters.

4 Methodology

4.1 Key considerations behind each step

The entire trading system can be divided into the following steps:

1. Collect market data. In this step, market data are collected from providers with formatted data, e.g. Yahoo finance. Note that we need to consider the accuracy of data as well the effectiveness of data. Meanwhile, we need to consider how we fill missing data. If the data provider has already filled the missing data, we need to know what interpolation method is used, since this will have direct effect on the effectiveness of market patter we found.

2. Process market data into a standard format. In detail, we will consider computing the standard information (open, close, max, min, etc.) for unit time interval. We will also compute different technical indicators. The CNN is a process of extracting key information from the images. However, many technical indicators are already key information we use based on our past experience. Training CNN with the help of technical indicators, we may expect to have higher accuracy.

Electronic copy available at: https://ssrn.com/abstract=3756702

3. Generate images. Instead of plotting all possible indicators on a single images, we should carefully select the indicators. Putting too much information will make lots of noise to trend, which will make it harder for model to grasp the main trend. For example, we can generate generate candle plots with simple moving averages (Figure 1). Further, we need to be careful with the selection of the window length.

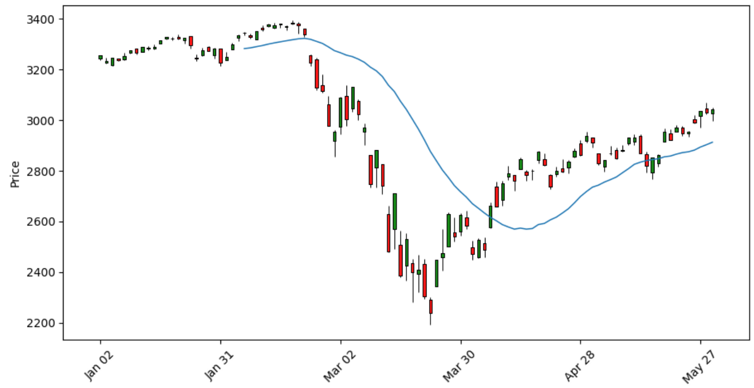

Figure 2: Candle stick plot with simple moving average

4. Labeling images. We have two basic choices when dealing with the stock market date, labeling the data either as a regression problem or as a categorical problem.

5. Train the CNN model using generated images. In addition to the selection of the CNN model, which is described in the previous section, the key of this step of this stage is to carefully separate the training set with the test set.

6. Test the CNN model on the test set and compare the result with other strategies. We have to the select the best evaluation method and compare the performance of the CNN model with other traditional base case methods. If the CNN model cannot perform better than the base cases, the CNN model might be useless.

4.2 Methodology in detail

This project collected the daily price data of S&P 500 from January 1, 1985 to June 30, 2020 from Yahoo finance. After downloading the data, it was split into the training set and the test set. The last four years were used for test set and the remainder are used for training. To avoid the spillover effect of training set to test set, I did not split the data randomly. The training data and test data were divided into 30 days terms by 10 days sliding. 20 days are shared in adjoining terms. The data has 791 terms for the training and 98 terms for the test. After splitting the data, candle stick images with simple moving average (SMA 25 days) were created.

The duration of each window length is important. If we choose a short window, we could have many data for training. However, the information contained in the short-term window is relatively small compared to a longer window. Sezer and Ozbayo-glu (2019) selects 30 days.

The next step is the labeling of the data. There are two ways to label the data. One way is to predict the price or return of the asset (regression problem) and the other is to predict the direction of the asset (categorical problem). This project chose categorical labeling. Sim, Kim and Ahn (2019) uses two categories (either up or down), while Sezer and Ozbayoglu (2018) labeled the data into three categories (buy, hold, or sell). There are also several options for the forecasting term, for example, one-term ahead forecast, one-week ahead forecast, or one-month ahead forecast.

Electronic copy available at: https://ssrn.com/abstract=3756702

This work labeled the data into three categories namely ”up”, ”flat” and ”down”. When the forecasting price is within the range of plus or minus 0.5% the data set was categorised as ”flat”. For the forecasting price, this work tested three cases and compared the accuracy between them. The three cases are one-day ahead forecast, three-days ahead forecast and five-days ahead forecast. The one-day ahead forecast compares the closing price of the last day in each candle stick image with the next day closing price. When the next day is up / down more than 0.5% , the image was categorized as ”up” / ”down”. The three-days / five-days ahead forecast compares closing price of the last day in the candle stick with the closing price of three / five days ahead.

After labeling the data, the Inception v3 model with three additional layers were trained on the training set. RMSprop is used as the optimizer and categorical cross-entropy loss as a loss function. The training are executed 100 times (100 epochs). The programming language used for this project is python. The core library used for building and training the CNN is tensorflow (tensorflow.org) with keras API (kereas.io). The python code used in this project is on the github: (https://github.com/YoheiKo/CNN-candle-stick).

Accuracy might be the main evaluation method. A simple benchmark accuracy might be 33% (in three categories case) or always predicting one direction movement. This research set two additional base cases; namely momentum strategy and contrarian strategy. A simple momentum strategy in this paper assumes that it predicts ”up” if (1) the most recent daily prices were up for two consecutive days or (2) the most recent daily closing price was the highest in the last 30 days or (3) the latest price breaks the 25 day moving average price. It predicts ”down” in the opposite case and ”flat” when neither. The simple contrarian strategy takes the opposite ”buy / up” and ”sell / down” position of the momentum strategy when conditions (1) or (2) satisfies, although it takes the same ”hold / flat” position in the case of ”hold / flat”. Because moving average is an indicator for momentum strategy, condition (3) is not applied to the contrarian strategy. If the model performs better than the base cases, it has profit potential.

5 Results

5.1 Results of the model

Figure 1-3 are the plots of the training accuracy and test set accuracy (validation accuracy) during the 100 epochs. Because I stopped the training when the training accuracy reaches 85%, the horizontal length of the three result images are slightly different. The red lines in three images - training accuracy - are climbing consistently which is expected, while the blue lines - test accuracy - are fluctuating but almost flat. This simply means that we do need 100 epochs for training.

The blue line in figure two (three-days ahead forecast) is hovering at the highest accuracy range while the blue line in figure three (five-days ahead forecast) is clearly lower among the three. In fact, the test set accuracy reaches 0.449 at epoch 7 in one-day ahead forecast, 0.5 at epoch 21 in three-days ahead forecast, and 0.4286 at epoch 35 in five-days ahead forecast.

The base cases accuracy are shown in table 1. The performace of the CNN model is far above the base cases accuracy which support the effectiveness of CNN model applied to S&P 500 price forecast. When compared the test set accuracy between the three cases, the three-days ahead forecast has the highest accuracy of 50%. The second highest is the the one-day ahead forecast of 44.9%. The five-days ahead ahead forecast has the lowest accuracy rate of 42.9%.

The result indicates that the accuracy of the five-days ahead forecast is the lowest because it is too far from now. Although the first day is closest to the historical data, the result indicates that it could be a noise movement. The three-days ahead forecast is the best. It is not too far from the history and the noise effect are smoothed. This conclusion could also be applied to arbitrary swing trading strategy based on candle sticks because the CNN model in this project uses the same candle sticks images and simple moving averages as inputs. You might have the best performance when you close your position at T + 3.

Electronic copy available at: https://ssrn.com/abstract=3756702

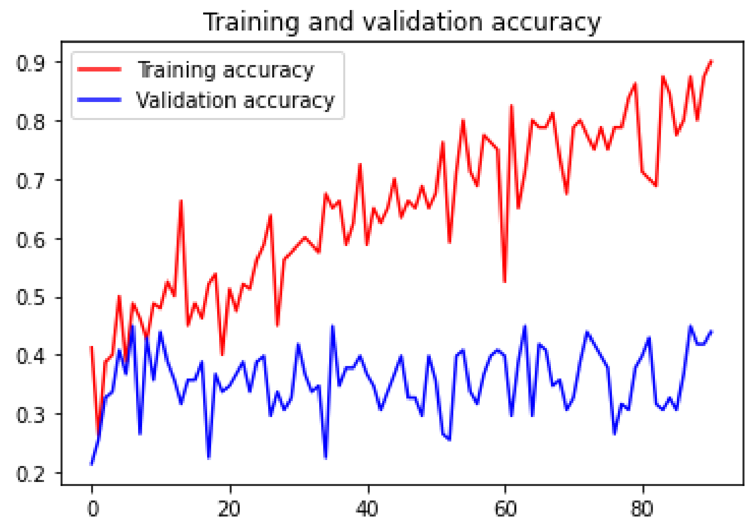

Figure 3: Training and validation accuracy for one-day ahead forecast

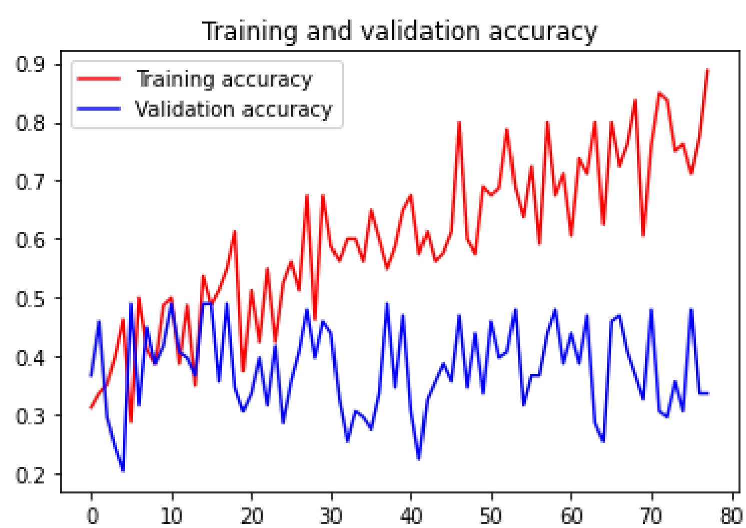

Figure 4: Training and validation accuracy for three-days ahead forecast

Electronic copy available at: https://ssrn.com/abstract=3756702

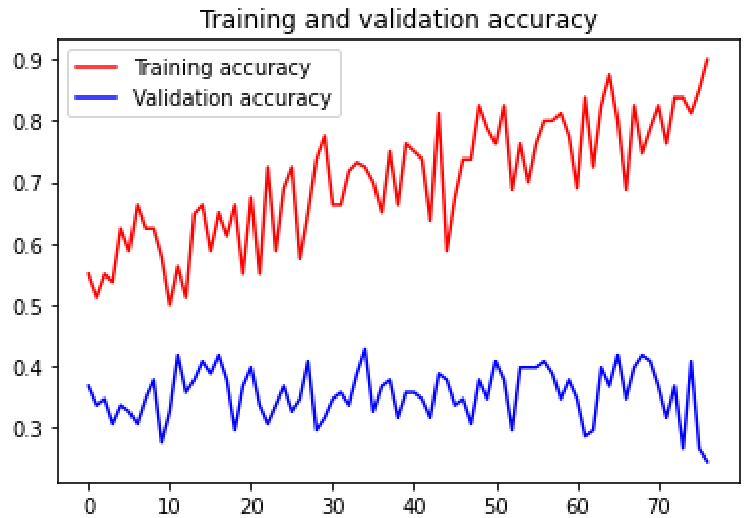

Figure 5: Training and validation accuracy for five-days ahead forecast

5.2 Comparison to the base cases

Table 1 compares the accuracy of the CNN-Inception V3 model with the other base case strategies. CNN Inception V3 model performs highly better than the other strategies and base cases indicating the alpha potential of the model. Other findings from the tables are: (1) Momentum strategy is better than the contrarian strategy. (2) 3-days ahead forecast yields the best also in the case of momentum strategy and contrarian strategy. (3) Momentum strategy and contrarian strategy does not generate sufficient alpha compared to the always ”buy” strategy.

Accuracy Table

Model / Strategy	1 day ahead forecast	3 days ahead forecast	5 days ahead forecast
CNN Inception V3 model	44.9%	50.0%	42.9%
Momentum strategy	28.1%	36.4%	29.6%
Contrarian strategy	25.8%	33.3%	29.6%
Always ”buy” bull strategy	37.5%	34.3%	43.9%
Always ”flat” strategy	15.6%	44.4%	22.4%
Always ”sell” bear strategy	46.9%	21.2%	33.7%

Table 1: Test set accuracy of CNN Inception V3 model and other base cases strategies

6 References

[1] Fama, E.F. (1970). Efficient capital market, a review of theory and empirical work. J.Financ. pp.383-417
[2] George A. Benington, Michael C. Jensen. (1970). Random Walks and Technical Theories: Some Additional Evidence. Journal of Finance, vol. 25. pp. 469-482.
[3] Franklin Allen, Risto Karjalainen. (1995). Using genetic algorithms to find technical trading rules. Journal of Financial Economics, vol. 51. pp. 245-271.
[4] Terence Tai Leung Chong, Wing-Kam Ng. (2008). Technical analysis and the London stock exchange: Testing the MACD and RSI rules using the FT30. Applied Economic Letters, vol. 15. pp. 1111 – 1114.
[5] Terence Tai Leung Chong, Wing-Kam Ng, Venus Khim-Sen Liew. (2014). Revisiting the Performance of MACD and RSI Oscillators. Journal of Risk and Financial Management, vol. 7. pp. 1-12.
[6] Safwan Mohd Nor, Guneratne Wickremasinghe. (2017). Market efficiency and technical analysis during different market phases: further evidence from Malaysia. Investment Management and Financial Innovations, vol. 14.
[7] Fernando Fernandez-Rodriguez, Christian Gonzalez-Martel, Simon Sosvilla-Rivero. (2005). Optimization of technical rules by genetic algorithms: evidence from the Madrid stock market. Applied Financial Economics, Taylor Francis Journals,

Electronic copy available at: https://ssrn.com/abstract=3756702

vol. 15, pp. 773-775.
[8] Michael D. McKenzie. (2007). Technical Trading Rules in Emerging Markets and the 1997 Asian Currency Crises. Emerging Markets Finance and Trade, vol. 43. pp. 46-73.
[9] Hao Yu, Gilbert V Nartea, Christopher Gan, Lee J. Yao, (2013). Predictive ability and profitability of simple trading rules: Recent evidence from Southeast Asian stock markets. International Review of Economics Finance, Elsevier. vol. 25, pp. 356-371.
[10] Omer Berat Sezer, Ahmet Murat Ozbayoglu. (2018). Algorithmic Financial Trading with Deep Convolutional Neural Networks: Time Series to Image Conversion Approach. Elsevier Applied Soft Computing Journal. Vol. 70, pp. 525–538.
[11] Hyun Sik Sim ,1 Hae In Kim,2 and Jae Joon Ahn (2019). Is Deep Learning for Image Recognition Applicable to Stock Market Prediction? Hindawi, vol 2019
[12] Omer Berat Sezer, Ahmet Murat Ozbayoglu. (2019). Financial Trading Model with Stock Bar Chart Image Time Series with Deep Convolutional Neural Networks. Intelligent Automation and Soft Computing journal
[13] Gudelek, M. U., Boluk, S. A., Ozbayoglu, A. M. (2017, November). A deep learning based stock trading model with 2-D CNN trend detection. In 2017 IEEE Symposium Series on Computational Intelligence (SSCI) pp. 1-8. IEEE.
[14] Sezer, O. B., Ozbayoglu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. Applied Soft Computing, 70, pp. 525-538.
[15] Yaping Hao, Qiang Gao. (2020). Predicting the Trend of Stock Market Index Using the Hybrid Neural Network Based on Multiple Time Scale Feature Learning. Applied Sciences, Vol. 10(11), 3961.
[15] Liu, S., Zhang, X., Wang, Y., Feng, G. (2020). Recurrent convolutional neural kernel model for stock price movement prediction. PLoS One, Vol. 15(6)

Electronic copy available at: https://ssrn.com/abstract=3756702