CS6375.502 Final Project Report - S&P 500 Forecasting and Optimization of the Expiration period for Option Trading for Maximum gain Using Machine Learning

*

Soumil Mandal*, Md. Arafat Hossian Khan†, Priya Mehta‡ and Pritom Das Radheshyam§
The University of Texas at Dallas
*sxm180159@utdallas.edu, †mxk157830@utdallas.edu, ‡phm180000@utdallas.edu, §pxr180033@utdallas.edu

***Abstract*—For our CS6375.502 final project, we decided the problem statement to be weekly up/down forecasting of S&P 500 stock price movement using machine learning techniques. The reason for choosing SPY is its high liquidity on contracts. We want to predict the lower bound of the time span for choosing the best expiration date for a bullish call. As the underlying decision factor the price forecasting will be used as the contract price is directly related to the underlying stock price. Our aim is to create a probabilistic guarantee of return. In this report, we present a detailed description of our proposed systems, experiments conducted, and analyze the results as well. Our stacking based ensemble model which was created using support vector machine achieved an f1-score of 57% on the testing data.**

***Index Terms*—S&P 500 Forecasting, LSTM, Random Forest, SVM, News Analysis**

I. Introduction

During the last decade there is a great volume of stock trading, new emerging technologies are being adapted in every sub sector for the financial landscape. Stock markets are a powerful sign to global sustainability. The S&P 500 index monitors the stock performance of the largest publicly traded 500 companies of United States. Prediction of stock markets is extremely difficult because it is of highly dynamic in nature, non-linear, non-parametric system [4]. Predicting stock prices for every week is in fact a time series problem. There are various strategies that can tackle this problem by focusing on neural network model. Objective for any stock market forecasting is attaining greater heights of financial benefits and counter against market risks as much as possible.

The first half of the report implements LSTM approach. Our primary focus was implementing LSTM algorithms under the scope of Recurrent Neural Networks (RNN) since it tackles the vanishing gradient problem of RNNs and also have superior effectiveness in stock prediction while comparing with other neural algorithms [5]. We have also coupled LSTM along with implementation of the Attention Mechanism. This is very helpful in tackling very huge volumes of stock data as this kind of data keeps growing on the long run.The second half emphasizes on further improving market hypothesis by analysis of news headlines. Financial news articles can be categorised under non quantifiable data and prediction of future trend of stocks which may either rise or have an upturn with sentiment classification and analysis of data from news is very vital and also proven feasible by Wataru Souma [17]. The later part of the report reveals an interesting analysis on ideal return risk-return trade-off. Various experimentation results are provided to prove our claims.

II. Related Work

No wonder the stock market data might just be one of the most sought after and highest rated analyzed data of this decade. Successful forecasting of such prices is a crucial research by many academic professionals and business world. At earlier times, Fama [6] stated that it is impossible for any individual to beat the trading market because their stock prices are set up fairly. But contrary to his works, many attempts were done to obtain profits in predicting stock prices in advance. Clive [8] in his paper proved that although forecasting such prices seemed to be impossible but by careful and intensive analysis it can be predictable. There are manifold econometric techniques [10], [11] in getting short term future prices of stocks listed in stock exchanges.

But a better way to achieve higher scale or prediction is to implement computational power along with machine learning. It overcomes the assumptions and limitations of theoretical backgrounds and in contrast can also identify non linear relationships among the data which are unseen by many theoretical economists. Thus, many researches are being conducted to use ANN for stock market forecasting. Kim [12] implemented the theory of genetic algorithms and optimized thresholds for discretization of features and connection weights. This was done within the layers such that it can forecast prices stock index. The aftermath was very surprising as it was able to surpass the conventional neural networks. Wang [13] introduced the concept of back propagation with a wavelet denoising neural network in forecasting the Shanghai Composite Index. This model was successful in achieving better accuracy when compared with a conventional back propagation neural network.

LSTM methods were slowly proven to be effective as many researchers started to use them to solve time series problems. Gers [14] and Nelson [5] implemented LSTMs in prediction of future trends of stock market by using price of the stocks and various technical analysis indicators.Their experimental results proved that LSTM provided much higher accuracy of prediction than other famous machine learning models which were earlier utilised for stock predictions, for instance random forest, multilayer perceptron and models which are pseudo-random.

III. Data Formatting

We collected our S&P 500 data using the Yahoo Finance API [1]. The time range of our data set was from January 1993 to October 2019, with 6700 instances. The attributes were date, open, high, low, close and volume. A short description of each of these attributes taken from [9] is given below in Table 1.

TABLE I
ATTRIBUTE MEANINGS.

Date	The date of that trading day.
Open	The price of the stock at the very beginning of that trading day but the opening price does not need to be equal to the previous day's closing price.
High	The highest price the stock had during that trading day
Low	The lowest price the stock had during that trading day.
Close	The price of the stock at closing time of that trading day.
Volume	The number of stocks that were traded during that trading day.

For creating our models, we formatted the data in resolutions of five (business days in one week), i.e. took time steps of t(n) to t(n+4), and converted it to a single instance. To do this, we evaluated the attributes in the following manner open $\leftarrow$ open value of t(n), close $\leftarrow$ close value of t(n+4), high $\leftarrow$ max(high values of t(n) to t(n+4)), low $\leftarrow$ min(low values of t(n) to t(n+4)). We discarded the volume attribute. Then, we normalized each of the attributes and compressed them in the scale (0, 1). For training, we used 1200 such instances, and for testing, we used the next 100 instances. As our final goal was to predict a binary variable, i.e. whether the market will go up or go down instead of attribute values, we created the parallel up/down dataset as well by simply comparing close(t) with close(t-1), i.e. if close(t) $>$ close(t-1), market goes up, else, market goes down. The visualizations of the close values post normalization for both daily and weekly is shown below in Fig 1.

IV. Proposed Method

We propose three systems on the whole. The first one is a stacked LSTM [1] with attention [2] trained on S&P 500 values. For our second model, we train a Random Forest based

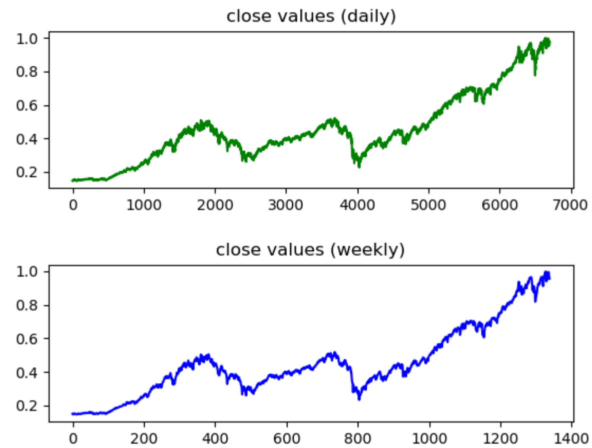

Fig. 1. Close values, where x-axis is time, and y-axis is the normalized value.

classifier on a news headlines data set. Finally, we create a third ensemble model 2, which takes the raw outputs from the LSTM and RF models and learns to map them to a 0/1, i.e. down/up output. This type of ensembling is also know as stacking. All the systems, along with the results are explained and analyzed in the following sections.

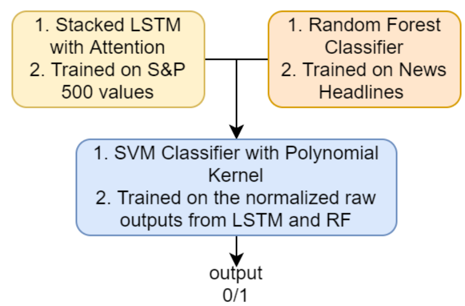

Fig. 2. Ensemble model.

V. LSTM with Attention

For implementing our LSTM model, we used the Keras library [2], while the attention layer was implemented from scratch. The network had a total of five hidden layers. A sequence from input layer to output layer was input, LSTM(128), attention, LSTM(64), LSTM(32), LSTM(16), output. Between each of the LSTM layers, we added a dropout [3] of 0.2 to prevent over-fitting. Here the number inside parenthesis represents the size of the layer. The concept of attention is very much inspired by the attention mechanism of humans, i.e. when interpreting a sequence, we tend to give more attention to certain parts, i.e. parts which are more important as compared to others. For example, while reading a paragraph, we give more importance to the semantic heavy words as compared to the stop words. Similarly, in

[1] https://finance.yahoo.com/quote/

[2] https://keras.io/

neural networks, attention helps the neural network identify the more important sub-sequences and try to learn them with higher priority. This is especially important when the time series data is quite long, which is the case in ours. For hyper-parameters, we set number of epochs to 25 and batch size to 32. The optimizer we used was Adam [7] (a stochastic optimizer) with a learning rate of 0.01. Overall, the number of trainable parameters in our neural net was 164,562.

In our model, we created the additive style attention, also commonly called as soft attention which was proposed by Bahdanau [2]. Essentially, it extracts a context vector con_i from the hidden states h_i of its previous layer, LSTM(128) in our case, and passes it to the next layer, for combining these results with its hidden states, i.e. hidden states of LSTM(32) in our case. The method of calculating con_i is by getting the mean of the previous hidden states, weighted using the attention scores $atten_i$. The equations used for calculating the same is given below.

$$con_i = \sum_j atten_{ij} \times h_j$$

$$atten_{\mathrm{i}} = softmax(f_{atten}(h_i, h_j))$$

Here, f_{atten} is the attention function which computes an alignment score (unnormalized) between the previous and current hidden states, i.e. h_j and h_i respectively. A feed forward network with a single hidden layer is used to calculate alignments for attention using the following equation.

$$f_{atten}(h_i, h_j) = {z_{\mathrm{a}}}^T tanh(Q_{\mathrm{a}}[h_{\mathrm{i}}, h_{\mathrm{j}}])$$

Here, z_{a} and Q_{a} are the parameters learned for attention. Similarly, it is also viable to use unique matrices, for example Q_1 and Q_2 to learn different transformations for h_i and h_j and then add them up.

$$f_{atten}(h_i, h_j) = {z_{\mathrm{a}}}^T tanh(Q_{\mathrm{a}}[h_{\mathrm{i}}, h_{\mathrm{j}}])$$

VI. Analyzing News Headlines

Along with numerical data, various studies also suggest that news data can have crucial impact and role in forecasting the stock prices. We decided to base our predictions for the market going up or down on the basis of the top headlines for a particular date. The model was trained on the Dow Jones News Sentiment Dataset (DJIA) [3] which consisted of the top 25 headlines for a particular date and a label which denoted the overall sentiment of the data-set. We modified the process a bit and instead of training on the sentiment labels, we trained the headlines on the market going up or down the next day. A Random Forest [15] model was trained for classifying the news headlines with a 100 estimators. The decision tree split was made on the basis of entropy. In order to feed the data to the Random Forest classifier, the text data was converted to a numeric representation using the TFIDF values (Term Frequency - Inverse Document Frequency). This vectorisation method reflects the relevance of a word in a data-set i.e. the words which actively contribute to the prediction for the market going up or down. A relevant word is not necessarily a word which occurs most frequently, but a word which contributes significantly to the prediction. The random forest model was trained on a dataset ranging from the dates 08/08/2008 to 01/07/2016. Preprocessing techniques like removing punctuation and numbers along with stemming were implemented to improve the accuracy. The model was tested on a dataset which consisted of headlines sorted date wise from the Reuters News Headlines dataset [4]. The labels for prediction were generated from the Adj. Close prices of consecutive days and labelling the market going up or down. The Adj. Close prices were scrapped from the Yahoo Finance API.

VII. Experiments & Results

A. Experiment 1 - Only S&P 500 Analysis

Here, we trained the neural network described in section V on the 5 resolution S&P 500 data that we have created. The train/test split was 1200, 99 where the all of them are in an ordered time steps manner, i.e. the testing data begins from the next time step of the last training data instance. As neural networks are initialized by random weights before training, we performed several runs, and the best performing model secured an f1-score of 54% on the test data. The complete evaluation metrics report, class wise, is shown below in Fig 3.

	precision	recall	f1-score	support
down	0.44	0.34	0.38	41
up	0.60	0.69	0.64	58
accuracy			0.55	99
macro avg	0.52	0.52	0.51	99
weighted avg	0.53	0.55	0.53	99
precision	0.54			
recall	0.54			
f1-score	0.54			

Fig. 3. Metrics report of neural network.

Given the volatility of S&P 500, we see that our model has performed quite well in forecasting the next 100 instances, i.e. 500 days. As a better experiment would be to see how the model performs in the initial days (as in a real life scenario its always possible to train a model till t(n) where t(n+1) is to be predicted), we created a graph which shows the change in f1-score with each increasing time step. The normalized S&P 500 values are plotted as well for reference.

From Fig 7 and Fig 8 we see that for the first initial time steps, the f1-scores is quite good, i.e. above 80%, and then steeply decreases and finally stabilizes at about 54%. This is quite satisfactory for real life scenarios where retraining of models occur very frequently with the recent time steps. In

[3]https://www.kaggle.com/aaron7sun/stocknews

[4]https://github.com/duynht/financial-news-dataset

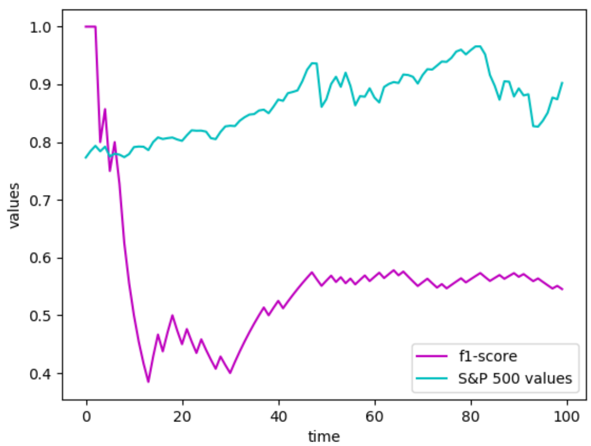

Fig. 4. Change in f1-scores with time.

Fig 8, we plot the correct (blue bars) and incorrect (orange bars) predictions by the neural net model.

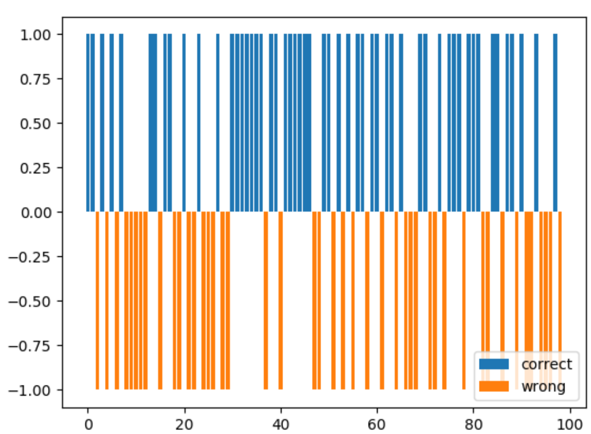

Fig. 5. Correct and wrong classifications by LSTM.

B. Experiment 2 - Only News Analysis

From Fig 6 we can see that along with several correct classifications, there are also several misclassifications. On careful analysis of the misclassifications and the possible reason, we came to a conclusion that many misclassifications were a result of the overall neutral sentiment of news headlines for that date. The complete evaluation report for RF is shown in Fig 7.

C. Experiment 3 - Ensemble Model

For creating an ensemble model, we decided to use the stacking method, also referred to as stacked generalization. In this method, a learner combines the predictions of several other learning algorithms to learn a mapping function. In our case, the two base models are deep neural net, and random forest. We combined the outputs of these two to create a feature vector of size two, and then passed it on to our final learning model, which was a support vector machine (SVM) [16] classifier. In this experiemtnt, both the NN and RF was trained on data with resolutions set to five. The evaluation results of the ensemble model is shown in Fig 8. We see an improvement over the individual models.

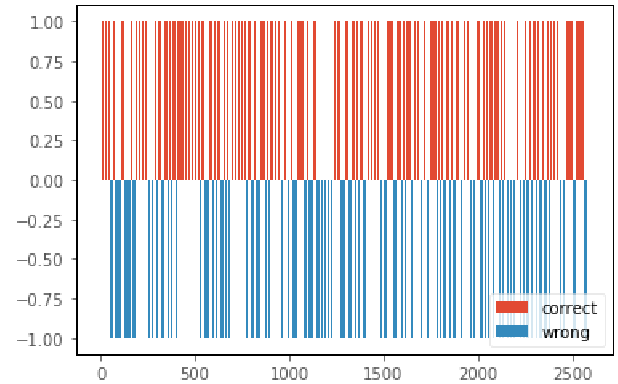

Fig. 6. Correct and wrong classifications by Random Forest.

	precision	recall	f1-score	support
down	0.64	0.69	0.66	1597
up	0.42	0.69	0.39	985
accuracy			0.57	2582
macro avg	0.53	0.53	0.53	2582
weighted avg	0.55	0.55	0.56	2582
precision	0.56			
recall	0.56			
f1-score	0.56			

Fig. 7. Metrics report of Random Forest.

	precision	recall	f1-score	support
down	0.52	0.30	0.38	43
up	0.59	0.78	0.67	55
accuracy			0.57	98
macro avg	0.55	0.54	0.53	98
weighted avg	0.56	0.57	0.54	98
precision	0.57			
recall	0.57			
f1-score	0.57			

Fig. 8. Metrics report of ensemble model.

VIII. Analysis

A. Experiment 4 - Option Trading Expiration Period

We know that the stock price will eventually go up with a high probability in the long run. But the money inflation may dominate the bullish stock price and eventually loosing the value of equity. But one can hold the shares for an

TABLE II
MODELS AND THEIR SETTINGS.

Experiment	Model	Parametes	Results
S&P 500 up/down prediction using OHLC data	LSTM + Attention (resolutions = 5) train : 92% test : 8%	five hidden layers of sizes 128-128-64-32-16 optimizer = Adam activation of attention = softmax learning rate = 0.01 batch size = 32 epochs = 25	label down (0): precision : 44% recall : 34% f1-score : 38% label up (1): precision: 60% recall: 69% f1-score: 64% overall precision: 54% recall: 54% f1-score: 54%
S&P 500 up/down prediction using Text Analysis	Random Forest (RF) (resolutions = 1) train : 75% test : 25%	n_estimators = 100 criterion = entropy	label down (0): precision : 64% recall : 69% f1-score : 66% label up (1): precision: 42% recall: 36% f1-score: 39% overall precision: 56% recall: 56% f1-score: 56%
S&P 500 up/down prediction using Ensemble Model	Stacking Ensemble SVM with outputs from LSTM and RF (resolutions = 5) train : 80% test : 20%	kernel = poly	label down (0): precision : 52% recall : 30% f1-score : 33% label up (1): precision: 59% recall: 78% f1-score: 67% overall precision: 57% recall: 57% f1-Score: 57%

unlimited amount of time having no risk of loosing them. On the other hand contact trading does a trade off between having risk of loosing the shares with much more profit. But again making trading highly volatile have risk of loosing. So, natural question to ask is what would be a good risk-return trade off. For that reason we want to have a short time contract for which we can bet of the stock price movement, which in other words called contract/option trading. This arises the question how much risk will return how much probable return with what probability. Figure 9 explains the answer. For simplicity, we guarantee the return but we do not mention the amount of return as that will depend on many other factors at the particular time. But approximately the longer the span till expiration is the less the return is. Figure 9 shows the graph where x-axis contains the f1-score of the prediction versus the candle width (1 day, 2, day, $\ldots$, etc.). The term resolution means the width of the candlestick. Let us define the following

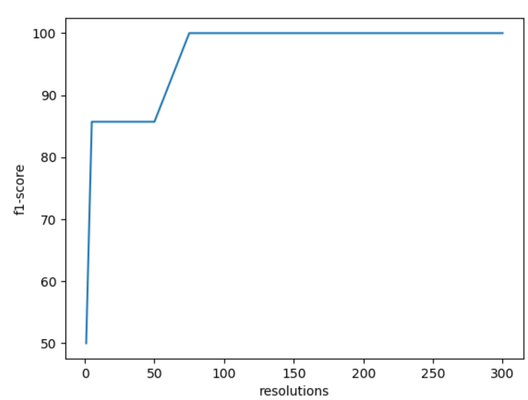

Fig. 9. Change in f1-score with resolutions. CDF of $P(G|X)$.

random variables,

$$G := 1 \text{ for gain, } 0 \text{ for loss}$$
$$X := \text{Expiration period} \in \mathbf{N}$$

Then, we claim,

$$\forall\ \alpha \in (0.5, 1)\ \exists\ x \in \mathbf{N}\ \ni\ P(G = 1|X = n) = \alpha$$

Figure 9 gives the cumulative distribution function. According to our claim of Equation (VIII-A), we say that for any probability of success α there exists a duration of n days for which the contact is to be made. From our analysis of the graph in Figure 9, we propose the following,

(a) High Risk - High Profit/Loss: For a probability of success around 55%, the value of time span for option call/put buy is around 5 days.
(b) Medium Risk - Medium Profit/Loss: For a probability of success of around 85%, the value of time span for option call/put buy is around 30 days.
(c) Low Risk - Medium Profit/Loss: For a probability of success of around 95%, the value of time span for option call/put buy is around 100 days.

B. Experiment 5 - Option Trading Based on This Analysis

We did some real life investment in option market based on this methodologies. Based on our experience of this method, we found the following,

1) Short term investment is extremely risky and highly prohibitive. Even if we have more win than loss we one loss at the end may cancel several consecutive wins happened before. We lost around 83% of the equity by short term investment. Figure 9 also shows a massive improvement of performance beyond the short term barrier of one week.
2) Medium term investment is best in stable economy. Our investment gained maximum 123% during last year with medium term investment as US economy was very stable.
3) Long term investment of about an year or more are the best to avoid loss. Our long term investment got 126% gain surviving a massive crash of stock market in December 2018.

C. Suggestions

Every investor and trader has their own psychology of trading. But based on our analysis and field experience, we suggest the following,

1) One should not do short term investment if interested in guaranteed return.
2) Medium and long term investment is to be done with good money management.

IX. Conclusion & Future Work

We can conclude that short term (daily or weekly) forecasting of S&P 500 still remains a very challenging task due to its highly volatile nature. Regression methods will not work in this case for the very reason that this kind if data is coming and derived from a tremendous dimensional abstract vector space and unfortunately the dimensions and bases are not known. So, instead of predictive approach a probabilistic approach (Markov process or advanced filtering methods) may produce better result. A finite dimensional Bernoulli process can be thought of as a candidate for mimicking the stock price movement. Moreover, the modeling for expected return with respect to day span is another challenge.

We are also working with candlestick pattern [5] identification from our dataset to further enhance the forecasting and prediction ability. Although candlestick works very well in a larger time frame, it can also produce satisfactory results for a small time frame. Candlestick is a type of chart that shows the high, close, open and low prices of the stocks or stock index for the selected time frame. It has been used for several years for the detection of certain patterns which can lead to understanding whether the stock will continue to rise or may fall down.

References

[1] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.
[2] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).
[3] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research 15.1 (2014): 1929-1958.
[4] Ahangar, Reza Gharoie, Mahmood Yahyazadehfar, and Hassan Pournaghshband. "The comparison of methods artificial neural network with linear regression using specific variables for prediction stock price in Tehran stock exchange." arXiv preprint arXiv:1003.1457 (2010).
[5] Nelson, David MQ, Adriano CM Pereira, and Renato A. de Oliveira. "Stock market's price movement prediction with LSTM neural networks." 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, 2017.
[6] Alfred Cowles 3rd. "Can Stock Market Forecasters Forecast?" Econometrica 1, no. 3 (1933): 309-24. doi:10.2307/1907042.
[7] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
[8] Granger, Clive WJ. "Forecasting stock market prices: Lessons for forecasters." International Journal of Forecasting 8.1 (1992): 3-13.
[9] Ahmed, War, and Mehrdad Bahador. "The accuracy of the LSTM model for predicting the S&P 500 index and the difference between prediction and backtesting." (2018).
[10] Keim, Donald B., and Robert F. Stambaugh. "Predicting returns in the stock and bond markets." Journal of financial Economics 17.2 (1986): 357-390.
[11] Fama, Eugene F., and Kenneth R. French. "The cross-section of expected stock returns." the Journal of Finance 47.2 (1992): 427-465.
[12] Kim, Kyoung-jae, and Ingoo Han. "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index." Expert systems with Applications 19.2 (2000): 125-132.
[13] Wang, Jian-Zhou, et al. "Forecasting stock indices with back propagation neural network." Expert Systems with Applications 38.11 (2011): 14346-14355.
[14] Gers, Felix A., Douglas Eck, and Jürgen Schmidhuber. "Applying LSTM to time series predictable through time-window approaches." Neural Nets WIRN Vietri-01. Springer, London, 2002. 193-200.
[15] Ho, Tin Kam. "Random decision forests." Proceedings of 3rd international conference on document analysis and recognition. Vol. 1. IEEE, 1995.
[16] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297.
[17] Souma, Wataru, Irena Vodenska, and Hideaki Aoyama. "Enhanced news sentiment analysis using deep learning methods." Journal of Computational Social Science 2.1 (2019): 33-46.

[5] https://en.wikipedia.org/wiki/Candlestick_pattern