Comparison of Predictive Algorithms: Backpropagation, SVM, LSTM and Kalman Filter for Stock Market

Divit Karmiani[1], Ruman Kazi[2], Ameya Nambisan[3], Aastha Shah[4], Vijaya Kamble[5]

[1,2,3,4,5]*Sardar Patel Institute of Technology, Mumbai*
[1]*divit.karmiani@spit.ac.in,* [2]*ruman.kazi@spit.ac.in,* [3]*ameya.nambisan@spit.ac.in,*
[4]*aastha.shah@spit.ac.in,* [5]*vijaya kamble@spit.ac.in*

Abstract: The variation and dependency on different parameters of stock market makes prediction a complex process. Artificial neural Networks have been proven to be useful in such cases to predict the stock values.The parameters involved and the commonly used algorithms are discussed and compared in this paper. In case of backpropagation algorithm, a feed forward network is present and weights are modified by back propagating the error. Similarly, significant modification is introduced in Sup-port Vector Machines Algorithm(SVMA) which results in higher accuracy rates. Presence of kernel and other parameters make it more flexible. Long Short-Term Memory(LSTM), another commonly used time series forecasting algorithm, is a special type of Recurrent Neural Network(RNN) that uses gradient descent algorithm. This paper provides a comparative analysis between these algorithms on the basis of accuracy, variation and time required for different number of epochs. The T-test hypothesis test was used for further analysis to test the reliability of each algorithm.

Keywords: Artificial Neural Network(ANN), Support Vec-tor Machines(SVM), Long Short-Term Memory(LSTM), Back-propgation, Recurrent Neural Netwrok(RNN), T-tests

I. INTRODUCTION

Prediction or forecast is essentially a calculated guess based on some previous evidence or facts. In todays growing economy prediction and analysis of stock market is of great importance as it reflects the economic conditions, but is a very complicated task. This is because of non-linear behavior of stock data. Many other factors which create hurdles in the way to predict stock include social, economic conditions, politics, rumours and media hype, traders behaviour, etc. Professional traders have tried to make forecasts and have developed a variety of analysis methods such as technical, fundamental, quantitative, and so on [1]. These methods use different sources ranging from news to price data, but they all have a common aim - predicting the companys future stock prices so they can make educated decisions on their trading options. Over the past 10 years, neural network - one of the most intelligent data mining techniques - has been used extensively. This has led to many researchers applying machine learning techniques to stock market as well some of which have produced quite promising results.

As every algorithm has its own advantage and disadvantage this research will study 3 different machine learning techniques commonly being in use for stock market prediction. A compar-ative approach will be followed to find out optimal technique for stock market prediction under different circumstances. Comparison will be made on the basis of their performance on the same input dataset and for the same number of epochs. As the main purpose of any predictive model is to minimize error (in case of stock market - minimize the risks), our given problem statement was addressed to achieve as much accuracy as possible by tweaking the algorithms.

II. LITERATURE SURVEY

Several studies have been done to analyze these algorithms to be used for stock market predictions. In most cases Arti-ficial Neural network(ANNs) were used which suffered from overfitting problem this was because of the large number of parameters to be fixed [2], and the little prior user knowledge about the relevance of the inputs in the analyzed problem. SVM [3] and Backpropagation algorithms were used for many problems centered around classification. They tried to over-come most of the limitations of ANN and on many instances reasonable accuracy was achieved. LSTM was used for many time series problems and similar approach is used to predict trend for stock by memorizing history data. Kalman filter is a two stage algorithm which employs linear regression within data and represents 'true value' of the market. It strikes a balance between data and predictions based on relative amount of noise [4].

Several other machine learning techniques are also being used for stock market prediction. There is no specification made by which we can choose the optimal solution for stock market prediction. This paper analyzes these algorithms using same amount of test and training data to maintain the consistency of testing environment. T-tests were performed for one-to-one comparison of these algorithms under similar conditions. These tests tell us the ability of these algorithms to provide consistent results with respect to each other.

III. THEORY

The three algorithms being focused in this paper are:

978-1-5386-9346-9/19/$31.00 ©2019 IEEE

A. Back Propagation

Back Propagation Algorithm is another supervised learning that is used to train a multi-layer feed forward network as it requires one or more fully interconnected layers. Like SVM, Backpropagation can also be used for both Classification and Regression problem. Stock market prediction can be treated as a classification problem. For each sample in the training dataset the output response id determined. The output is then compared with the given target and error is calculated using the Widrow-Hoff Delta learning rule or gradient descent rule [5]. As the term back propagation suggests, the weight updation and calculation of gradient happens backwards and using the least mean square error of the output response to the input sample.

The structure of the network consists of - the input layer which is connected to the hidden layer by interconnection weights and hidden layer which is further connected to the output layer by interconnection weights [5]. The increase in number of layers increases the computational complexity of the neural network which can cause the time taken for convergence and to minimize the error to be very high.

B. SVM

SVM (Support vector machine) is one popular algorithm used for many classification and regression problems. It is one of the supervised learning models that are based on the concept of hyperplanes as decision boundaries. In two dimensional space the hyperplane is a line. Based on the training samples, SVM classifies them into one of the different membership classes. While building the model, SVM training algorithm assigns new examples to one class or other class and assigns a hyperplane to output. This makes it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped such that the points in different categories are a a maximum distance from the decision boundary [6]. SVM uses five-fold cross-validation technique to estimate probability estimates instead of a direct method. This makes it more expensive. They are effective in higher dimensions, especially when the number of dimensions exceeds the number of samples. This is due to the presence of hyperparameters which include gamma, regularization parameter(C) and the choice of kernel available in the svm classifier. C decides the extent to which misclassification of data is allowed, gamma denotes how far influence of a training sample reaches and kernel which could be linear, poly or rbf which determines the learning of the hyperplane [7].

C. LSTM

LSTM or Long Short Term memory is a type of recurrent neural network (RNN)and uses an appropriate gradient descent algorithm. A RNN network feeds the output of a block as the input for the next iteration and uses its feedback connection to store recent events in the form of activation. Unlike multilayer perceptron, LSTM uses diverse processing elements or blocks to build a better recurrent network capable of learning long term dependencies.The inputs to a LSTM cell are the previous hidden and memory states and the current input while the outputs are the current hidden and memory state. The memory in LSTM can be seen as a gated cell, where three gates - input, forget and output gate control the ability to add or remove information or let it impact the output at the latest present(current) time step [8]. It uses a sigmoid neural net layer and weights to assign importance to information. This helps the gates protect and control the flow of information overcoming the vanishing gradient problem traditional RNN has. The stochastic gradient descent based algorithm used helps enforce constant error propagation(neither exploding or vanishing) through its internal units. LSTM can be used for time series predictions and are hence used in stock market [9].

IV. MODEL CREATION AND METHODOLOGY

Our approach in this analysis consists of three major steps:

- Data-set creation
- Implementation of algorithm
- Result and analysis

A. Data-set creation

For training and testing predictive algorithms, we compiled database for training and testing purpose [10]. The database should be having minimum deviation and should account normal behaviour. It was important to select a time frame where the stock market did not show any sudden jump or deep in prices. Often, there are times of economic collapse in certain sector, industry or country. The extent of these collapse might not be fathomable and are unpredictable.

For the training and testing of the algorithms and models, we chose time frame of January 2009 to October 2018. Since, during 2007-2008 stock market took a hit because of financial crisis. Yahoo Finance provides Historical Database which can be used for the training of the model. It provides historical database of a stock in the form of following parameters :

- Date: The date of when the following values collected
- Open: Opening value of the stock
- High: Highest value achieved by the stock on that date
- Low: Lowest value achieved by the stock on that date
- Close: Closing value of the stock for that date
- Adj Close: Closing value of the stock before market opens next day, adjusted by considering corporate actions and distributions
- Volume: Total number of shares that changed hands during that day

So, we collected stock price historical data from Yahoo Finances for the following 9 companies belonging to same sector of Technology:

- Apple
- Acer
- Amazon
- Google
- HP
- IBM
- Intel
- Microsoft
- Sony

We compiled our own database from these parameters. The key terms used in deriving the input parameters are:

- Momentum: It is the measure of detecting the flow of a parameter such as stock price in our case. It is defined as +1 if the price has increased and -1 if the price has been dropped from yesterday.
- Volatility: It determines the fluctuations in the same parameter at different times. For our case it will be determined by difference between yesterday's price and today's price divided by yesterday's price.

Hence for the input dataset we found out following parameters from the above parameters:

- Index Momentum: It determines the momentum of the market. For our case taken as 5 days of average.
- Index Volatility: This helps in determining the fluctuations in market. It is also calculated as average of last 5 days.
- Sector Momentum: This parameter considers other companies in the same sector and calculates the momentum of these companies for the last 5 days.
- Stock Momentum: This is average of the last 5 days of momentum of the respective company
- Stock Price Volatility: This is average of the last 5 days of stock price of respective company.

Data set consisted of these 6 dimensions for each of the stocks. I.E. each day has 9 records, one record for each company stock. The total number of rows for our input data set are around 22212.

B. Implementation of Algorithm

1) Support Vector Machine: This algorithm was implemented in python using the scikit-learn library for machine learning. The implementation was done in the following steps [11] -

- Importing the library and csv data files
- Creating a SVM classifier
- Defining the kernel and setting its parameters
- Predicting and comparison with target for accuracy calculation

In this case, since the stock market data is non linear we used the rbf kernel and kept the hyperparameters C and gamma at their default value of 1 and scale(version 0.22) respectively. Higher values of C and gamma,causes the decision boundary to become more curvy and variance to increase thereby causing higher chances of overfitting. The decision function shape was chosen to be ovo as it gave better results [12]. Accuracy can be improved by tuning the parameters using function gridSearchCV() in sklearn.model selection.This may take a lot of time depending on the number of training samples. Once the classifier is trained by mapping the given features and labels, prediction is made using the the test data which consists of approximately 20% of the total data (4,412 samples). This is the compared with the target data present in the output file and accuracy is ascertained.

2) Backpropagation: We followed 5 steps to implement this algorithm on our dataset [13]:

a) Initialize Network:

- Here we initialize weights and biases using the number of inputs, number of hidden neurons and number of outputs.
- Weights are selected randomly from the range 0 to 1.

b) Forward Propagation: In this stage we can get an output by propagatingour inputs through all the hidden layers and output layer. We use this technique further in the process for the prediction. Forward propagation involves:

- Activation of Neuron: We find the activation value using weighted sum of inputs.
- Transfer of Neuron: Using this activation value and sigmoid function we transfer this activation to get the output.
- Forward Propagation: It propagates inputs through input, hidden and output layers and stores the output value.

c) Backward Propagation of error: Here we calculate the slope of the output by differentiating the sigmoid function. Now, error signal is generated using the weighted error of each neuron in output layer.

d) Train Network: The error calculated for each neuron using backpropagation method can be used to update the weights. Similarly we also update bias weights. In this stage learning rate determines how much change should be there in updating the weights. Smaller learning rate over larger database and iterations give better set of weights. Since, we are updating weights for each iteration, this is called as online learning. Once the weights are

updated the step repeats depending on the number of iterations defined (number of epochs) and we train the network.

e) *Predict*: For prediction we are forward propagating the input set to get the output. In our case we are using the output itself as the probability of pattern belonging to each output class [13]. We turned this output into crisp class prediction by selecting the class value with larger probability.

Then we compared this predicted output with the target output to get the accuracy of the algorithm. The database was divided into 5 batches (5 folds) out of which 4 were used for training the network and 1 fold for testing the network. This gave us around 80% of data for training and 20% for testing.

3) *Long Short Term Memory*(LSTM): Implementation of LSTM was done using the following steps [14]-

a) *Dataset creation for LSTM:*

- Converting the time series into supervised learning problem:
  - We have data divided into input set and output set.
  - We defined a value for time step. Using this time step we shift the dataset and concatenate these two series to get the output set.
- Converting the time series into stationary data:
  - Stationary data is easier to model and is better for forecasting.
  - It is obtained by removing the trend from the data which is achieved by differencing the data.
- Observations are transformed to specific scale:
  - LSTM works on the data which is within the scale of activation function of the network.
  - Since the default activation function for LSTM is hyperbolic tangent (tanh) which has output range from -1 to +1 which is ideal for time series data.

b) *Model Development:*

- LSTM is a type of Recurrent Neural Network (RNN). This type of neural network is useful when remembering over the long sequence of data and it doesnt depend on the window lagged dataset as input.
- LSTM layer takes 3 inputs:
  - Samples: The rows of input dataset having independent observations.
  - Time steps: These are time steps of the parameters for the input dataset.
  - Features: The parameters considered and observed for input dataset to predict the output.

While compiling the network we need to mention loss function and also the optimization algorithm. So, we are using mae as loss function and ADAM as optimization algorithm. ADAM will select a suitable learning rate for the network. And then this model is fit to the training data and we predict the output. Then, we calculated the accuracy of the model.

V. RESULTS AND ANALYSIS

To make sure that one algorithms has a consistent performance and it performs better than other we tried to run same algorithm multiple times. Each algorithm was implemented 30 times and for each time prediction accuracy is calculated on test data. For each run different training and testing dataset is used. Find below accuracy results for each algorithm.

A. SVM Result:

For 10 runs of SVM algorithm we got approximately 66.9823 mean accuracy with 0.05256 standard deviation. This shows thatSVM performance is consistent for 10 runs this is due to nature of SVM algorithm. Algorithm will keep training until it can classify maximum testing data resulting in higher accuracy(Table I).

B. LSTM Result:

For 10 runs of LSTM algorithm we got approximately 68.51635 mean accuracy with standard deviation of 0.71779. But for larger number of epochs LSTM gives less variance and thus more reliable output (Table II).

TABLE I: RESULTS OBTAINED FOR SVM

No. of Epochs	Mean Accuracy (in percentage)	Standard Deviation	Variance	Time Required (in seconds)
10	66.9823	0.05255	0.00276	255.4932
30	67.0316	0.11561	0.01337	933.3335
50	67.0895	0.15970	0.02550	1438.2164
70	67.2219	0.18522	0.03430	2154.9157
100	67.1212	0.16695	0.02787	2258.1878

TABLE II: RESULTS OBTAINED FOR LSTM

No. of Epochs	Mean Accuracy (in percentage)	Standard Deviation	Variance	Time Required (in seconds)
10	68.51635	0.71779	0.51523	574.5679
30	68.98083	0.05528	0.00306	3504.9207
50	68.95468	0.04981	0.00248	5403.6272
70	68.96183	0.05109	0.00261	7448.0353
100	69.04171	0.05204	0.00271	10692.7814

C. Backpropagation Result:

For 10 runs of Backpropagation algorithm we got approximately 68.649 mean accuracy with standard deviation of

0.55375. As it can be observed Backpropagation performs similar to SVM but is faster. It has higher fluctuation in accuracy as compared to LSTM so that might cause issue when a steady accuracy is required(Table III).

TABLE III: RESULTS OBTAINED FOR BACKPROPAGATION

No. of Epochs	Mean Accuracy (in percentage)	Standard Deviation	Variance	Time Required (in seconds)
10	68.649	0.55375	0.30664	12.6310
30	68.433	0.54018	0.29180	52.8089
50	68.339	0.89537	0.80169	113.1676
70	68.249	0.69470	0.48260	143.1701
100	67.434	2.43245	5.91679	229.9802

D. Observations

From the above obtained results we also found out the T-test values for different combination of algorithms.T-tests are one of inferential statistical methods used to compare means. It tells us how many standard units the two means are apart. An assumption is made that the dependent variable is normally distributed and thus helps calculate the probability. The probability or amount of confidence(alpha level, level of significance) is set as 95%. The critical value for 95% confidence is found out for epoch = 10. And as we can see from the table IV, only the combination of Backpropagation-LSTM has T-stat less than critical value(CV). From table I and II we can see that LSTM has much less variance. So obtaining better results from LSTM has higher chances than Backpropagation.

TABLE IV: T-TEST VALUES FOR EPOCH = 10

	T-Stat	CV
SVM-Backpropagation	6.715	1.771
Backpropagation-LSTM	0.395	1.771
LSTM-SVM	7.666	1.734

VI. KALMAN FILTER AS ANOTHER PREDICTIVE ALGORITHM

Kalman algorithm is a recursive algorithm that uses time series data to eliminate inaccuracies obtained due to noise in measurement of different variables and produce estimates of variables more accurate than single measurements and hence a time series algorithm. Kalman deals with the uncertainties of the variables with weights higher to estimates with higher uncertainty [15]. Kalman algorithm works in the following steps. The first step involves estimates of the variables along with the noise involved in its measurement and other different noises. Once the outcome of the next measurement is obtained by the algorithm it updates its uncertainty matrix depending on the covariance between the predicted value and the measured value. Kalman filter hence develops a state transition model to follow the same flow as the data given and updates it at every step to find the most localized value and hence accurate prediction of the next state [16]. The stock market being a time series data and highly fluctuating becomes a good application to use Kalman filter which possesses real time tracking characteristics [17].

A. Algorithm and Flowchart

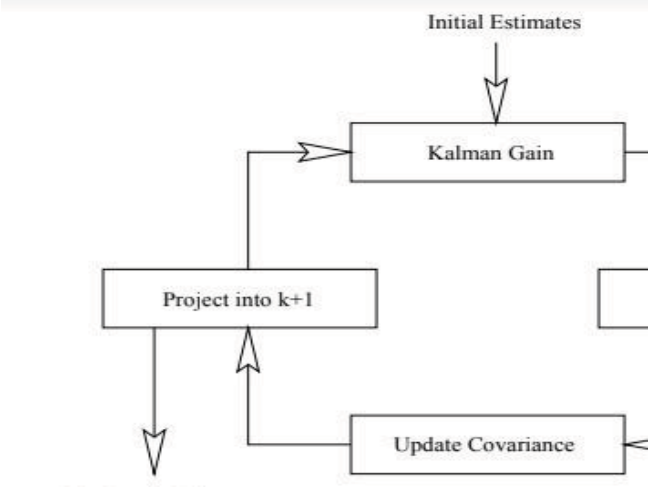

Fig. 1. Flowchart of Kalman Filter [18]

B. Data Set Creation

The data set used is the same as used for the other algorithms.However, the data set used in Kalman filtering is preprocessed as follows :

1) From the adj close parameter in the original dataset, an adjustment factor(adj factor) is calculated as adj factor = adj close/close.
2) The other parameters of the original dataset are multiplied/divided with the adj factor and we get the new values as :
   Open = Open * adj factor
   Close = Close * adj factor
   High = High * adj factor
   Low = Low * adj factor
   Volume = Volume / adj factor
3) Now, the adj close and open parameters are shifted by one row to enable to make comparison between the present and next days predicted value.
4) The difference between the various parameters is found by the following method :
   High diff = High - adj close shift
   Low diff = Low - adj close shift

Close diff = adj close - adj close shift
Open diff = Open shift - adj close shift

C. *Prediction and Comparison*

A prediction is made by using Kalman Filter on the various parameters of the data set and estimated stock prices of the next day are found out using the steps mentioned in the Kalman filter algorithm.The predicted and the actual values are compared by calculating the R2 score between them.

R2 is a statistic that will give some information about the goodness of fit of a model. In regression, the R2 coefficient of determination is a statistical measure of how well the regression predictions approximate the real data points. An R2 of 1 indicates that the regression predictions perfectly fit the data. An acceptable value is between 0 to 1.

D. *Experimental Results*

For the dataset of Apple,R2 score of 0.9856 was calculated which is within the accepted range of R2 score.The actual and predicted values are plot in a graph as follows.Refer fig. 2

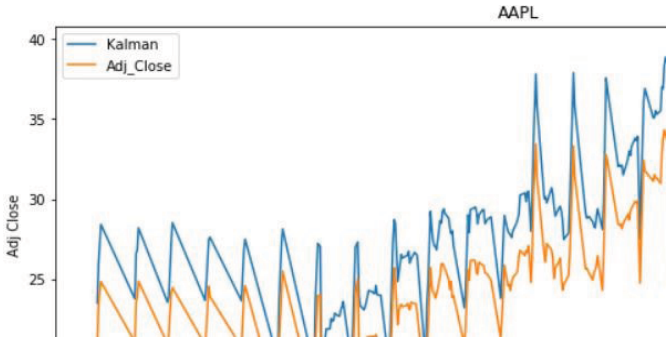

Fig. 2. Graph showing Actual and Predicted values for Apple

On using another data set, we observe that the variance between prices of two consecutive days is too high or the noise as perceived by the Kalman filter is very high. The predicted value is an indefinite value. As the noise decreases, the Kalman filter is able to predict the following day's closing price value accurately. This is shown in the graph below.Refer fig. 3

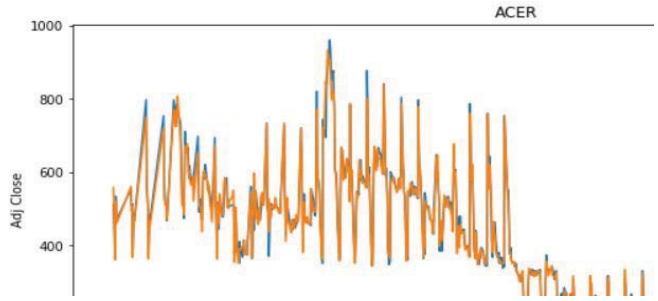

Fig. 3. Graph showing Actual and Predicted values for Acer

VII. CONCLUSION

Through this paper we have observed that we can use machine learning to predict and compare the stock market prices. The result shows how we can use historical data to predict stock movement with reasonable accuracy but the choice of algorithm depends on the requirement of parameters like time, variance and mean accuracy. If the requirement is high accuracy and low variance, LSTM would be a better choice but it is comparatively slower. If the requirement is high speed and accuracy then backpropagation is better. Also, from T-test result analysis we can conclude that LSTM is more reliable as compared to Backpropagation and SVM. For this implementation, we have incorporated 6 factors that affect stock performance. If a higher number of factors are used and after adequate preprocessing and filtering of data, it is used to train the network model, then a higher accuracy can be achieved.

REFERENCES

[1] A. Zheng and J. Jin, "Using ai to make predictions on stock market," Stanford University, Tech. Rep., 2017.

[2] O. Hegazy, O. S. Soliman, and M. Abdul Salam, "A machine learning model for stock market prediction," International Journal of Computer Science and Telecommunications, vol. 4, pp. 17–23, Dec 2013.

[3] Y. Xia, Y. Liu, and Z. Chen, "Support vector regression for prediction of stock trend," in Information Management, Innovation Management and Industrial Engineering (ICIII), 2013 6th International Conference on, vol. 2. IEEE, 2013, pp. 123–126.

[4] R. M. . N. Rhoads, "Predicting market data with a kalman filter," Technical Analysis of Stocks and Commodities, January 2010.

[5] Brilliant.org. (2018) Backpropagation. [Online]. Available: https://brilliant.org/wiki/backpropagation/

[6] S. Madge, "Predicting stock price direction using support vector ma-chines," Princeton University, Independent Work Report, 2015.

[7] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Sup-port vector machines," IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18–28, July 1998.

[8] C. Olah, "Understanding lstm networks," Aug. 2015. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs

[9] A.Moawad. (2018, Feb.) The magic of lstm neu-Ral networks. [Online]. Available: https://medium.com/datathings/the-magic-of-lstm-neural-networks-6775e8b540cd

[10] G. Bonde and R. Khaled, "Extracting the best features for predicting stock prices using machine learning," in Proceedings on the Inter-national Conference on Artificial Intelligence (ICAI). The Steering Committee of The World Congress in Computer Science, Computer , 2012, p. 1.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., "Scikit-learn: Machine learning in python," Journal of machine learning research, vol. 12, pp. 2825–2830, Oct 2011.

[12] A. J. Smola and B. Scholkopf,¨ "A tutorial on support vector regression," Statistics and computing, vol. 14, no. 3, pp. 199–222, 2004.

[13] J. Brownlee. (2016, Jul.) How to implement the backpropagation algorithm from scratch inpython. [Online]. Available: https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/

[14] J. Brownlee. (2016, Jul.) Time series prediction withlstm recurrent neural networks in python with keras. [Online]. Available: https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/

[15] C. Ku. (2017, May) Beating the naive model in the stock market. [Online]. Available: https://medium.com/@CalvinJKu/beating-the-naive-model-in-the-stock-market-62ec54436cf3

[16] J. Teow. (2017, May) Understanding kalman filters with python. [Online].Available:https://medium.com/@jaems33/understanding-kalman-filters-with-python-2310e87b8f48

[17] Y. Xu and G. Zhang, "Application of kalman filter in the prediction of stock price," in International Symposium on Knowledge Acquisition and Modeling (KAM). Atlantis press, 2015, pp. 197–198.

[18] T. Lacey, "Tutorial: The kalman filter," Computer Vision. [Online].

[19] Available:http://www.cc.gatech.edu/classes/cs7322298/spring/PS/kf1.p