Stock Market Prediction using Hidden Markov Model and Neural Network

Ali Sasani, Stephanie Tibado

***Abstract*— Luxury and comfort always comes after wealth. It is crucial and challenging task to completely understand the stock market and predict the future price. It has been proven that predicting future stock price with time series analysis is not reliable. It is known that Neural network has ability to extract valuable features for processing from data. In this paper, we applied some of machine learning techniques on stock market and tried to predict its trend and make profit based on that prediction. We applied multiple combination of feature extraction methods with NN and HMM. Among feature extraction methods we got the best results from DCT and PCA on raw data.**

I. INTRODUCTION

In the beginning of October 2008, World stock market was estimated to be around $36.6 trillion, US. Different techniques are being used in the trading community for prediction task [1]. Recently, the neural networks have emerged as one of the important techniques for price prediction [2]. The selection process of portfolio in electricity market is formulated as optimization problem and then is solved by Genetic Algorithm (GA) [3]-[7]. An Artificial Neural Network (ANN) is able to handle large sets of data at the same time it is able to work parallel with input variables [8]-[11]. Neural network mains strength is its capacity to identify patterns and anomalies and also detecting multi-dimensional non-linear connections in data [12].

Prediction of stock market trend using machine learning techniques is an interesting filed of research due to its potential financial gain [13]-[17]. In this paper, we compared multiple pattern recognition techniques on the stock market data and investigated the results. The paper is divided in six steps: 1- Data Collection, Pre-Processing, Indicator Extraction 2- Target calculation 3- feature extraction 4- processing the data using Neural Networks 5- processing using Hidden Markov Model (HMM) and 6- post-processing, which consists of a buy-sell strategy and testing of performance in terms of gains and losses. We will discuss each step in more detail in the following sections.

Ali Sasani, Computer Engineering Department, Islamic Azad University, Shiraz, IRAN.

Stephanie Tibado, Electrical and Computer Engineering Department, Florida Institute of Technology, Florida, USA.

II. DATA COLLECTION, PRE-PROCESSING, INDICATOR EXTRACTION

We obtained our raw input data from YAHOO's S&P 500 stock market index. Twenty years of data from January 1 of 1993 to February 1 of 2014 has been extracted. Additionally, the data was pre-processed. The obtained data which was in Excel format has been converted to a data matrix in MATLAB using xlsread.m. Then it was arranged in matrix from older to newer, rather than new to old as YAHOO files are. Also, the first three columns of each data record (row) are now the month, day, and year. Next, we need to extract the technical indicators. Technical Indicators, are any class of metrics whose value is derived from generic price activity in a stock or asset. Technical indicators look to predict the future price levels, or simply the general price direction, of a security by looking at past patterns. Past patterns include high and low, open, close, and volume raw stock data. All 27 technical indicators available from MATLAB's indicator.m function have been investigated. The indicator outputs are normalized so that the scale of the technical indicators ranges from approximately -1 to +1, as follows. First, the mean and standard deviation of the unscaled indicators are computed. Then, for each indicator for each day the mean is subtracted and divided by s 5*standard deviation. This linearly scales each indicator over short period of time for an approximate +-1 range. Time-varying long-term normalization was unnecessary since most indicator outputs were fairly similar in terms of range over the whole time span range of data.

The following three criteria were used when selecting indicators. First, the raw indicator function output vs. time were plotted. The plots with steady long-term graphs were the most highly regarded. Second, histograms of each raw indicator were plotted with 50 bins. Histograms that resemble a Gaussian distribution were selected. It was done on the assumption that the inputs for the stock market prediction system should be Gaussian. Third, another influence for selecting indicators was the information found on stockcharts.com. It had detailed information about each indicator, specifically, the commonly used ones that have steady long term results. Combining these three criteria, ten indicators were selected. They are: *momentum oscillators*: Rate of Change (ROC), Relative Strength Index (RSI), True Strength Index (TSI); *trends*: Moving Average Convergence-Divergence; *volume*: Chaikin Money Flow (CMF), Force Index (Force), Money Flow Index (MFI), Average Directional Index (ADX); volatility: Bollinger Bands, Volatility Ratio. As an example, Time and Histogram plots for Rate of Change is shown in figure 1.

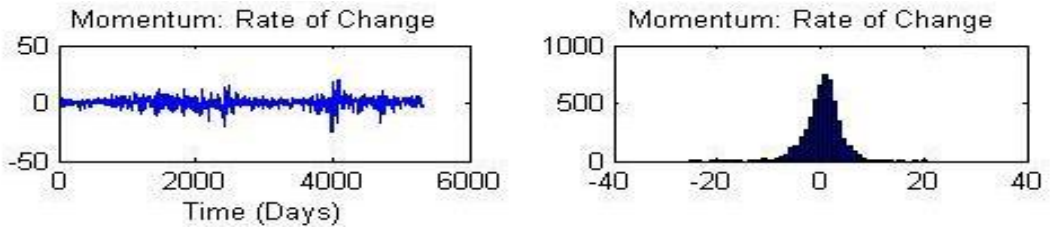

Figure 1. Time and Histogram plots for Rate of Change Indicaton

III. TARGET CALCULATION

The calculation of targets is based on twenty years of data. A window size of future 10 days is used and based on the highs and lows of the next 10 days, for each day in the 20 year data, targets ([percent high rise /100] and [percent low fall /100]) have been calculated.

%High Rise=(Max_value – Open_value)/open_value

% Low Fall =(Min_value – open_value)/open_value

Two options for calculating the targets are available in our code which based these options we can calculate these target in different way. By first option, we can use curve fitting method to determine Max_value. For each day, a best fit polynomial of the highs for the next 10 days and a best fit polynomial of the lows for the next 10 days are created. The maximum of the "highs" polynomial and the minimum of the "lows" polynomial is taken to determine the Max_value and Min_value variables. An example of the next 10 day highs, lows, and best fit polynomials for a particular day in 2011 is shown in Figure 2. By second option, for each day, the highest high and lowest low of the next 10 days is used to calculate the [percent high rise /100] and [percent low fall /100] based on the opening value of the current day. It should be mentioned that we used curve fitting method in our final target calculations.

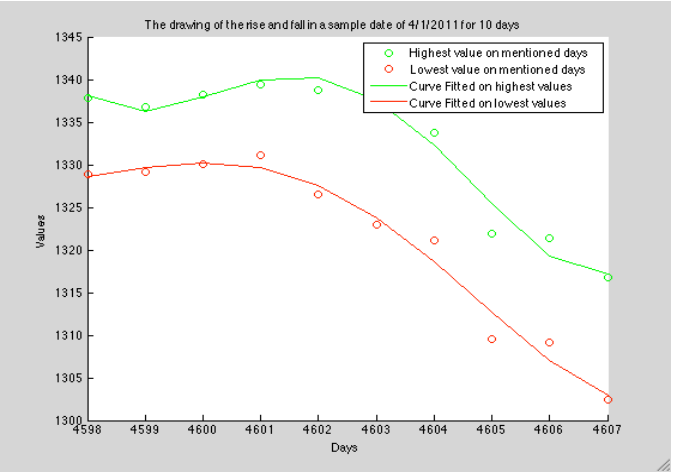

Figure 2. The next 10 day highs, lows, and best fit polynomials for a particular day in 2011

IV. FEATURE EXTRACTION

We have tested four feature extraction methods in this paper: Half Cosine Method, Discrete Cosine Transform, Pitch and Log Energy method and Principal Components Analysis. Any of the four methods can be used in future steps (NN or HMM).

Half cosine method

By using this method, each feature is obtained by multiplying the block of indicator data (1x50) by a discrete half cosine (50x1). Each of the five features are computed using formula (1), where *n* varies from 0 to 4.

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi x}{L}\right)$$
$$a_n = \frac{2}{L}\int_0^L f(x)\cos\left(\frac{n\pi x}{L}\right)dx. \quad (1)$$

This product results in a scalar value. In this fashion five features per indicator per block are obtained. Since, there are ten indicators 50 features are attained. Plot of half cosine method with five features for Relative Strength Index indicator are shown in Figures 3.

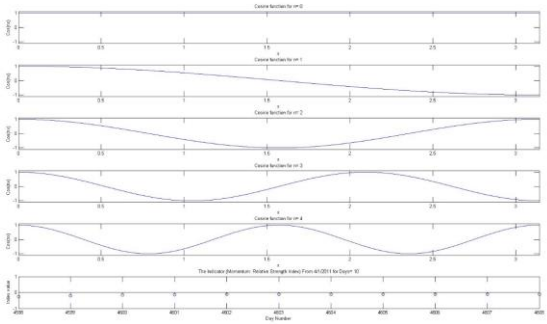

Figure 3. Output of half cosine method for the Relative Strength Index Indicator

Discrete Cosine Transform

Discrete cosine transform (DCT) transforms data into the frequency domain, and represents it by a set of coefficients. As a result, the energy of the data is concentrated in a few DCT components depending on the correlation in the data. It

expresses a signal in terms of a sum of cosine functions with different frequencies using the formula below.

$$w_k = c_k \sqrt{\frac{2}{n}} \sum_{t=0}^{n-1} a_t \cos\left[\frac{\pi}{n}\left(t+\frac{1}{2}\right)k\right], \qquad (2)$$

$$c_k = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 1, & k > 0 \end{cases}, \qquad k = 0, \ldots, n-1.$$

For k=0, the one dimensional DCT coefficient (the weight) w0 is the DC (direct current, zero frequency) coefficient. For k>0, the DCT coefficients are called alternative current (AC) coefficients. If data consists of correlated values, then most of the DCT coefficients will be zero or small numbers and only few large values. The largest values contain most of the important information about data. Therefore, only weights for k<5 are considered. It should also be taken into account that if the data is uncorrelated then most DCT coefficients will be large values. 50 DCT coefficients are calculated for 50 day block for each indicator value. In some experiments, MATLAB outputs less than 50 coefficients. This is due to data being correlated as discussed earlier. Out of 50 coefficients, absolute value of five highest are chosen as they contain most of the energy and most important data information. In Figure 4, the data and DCT coefficient outputs are plotted versus 50 days. There are 10 plots for each of the indicator chosen in step 1.

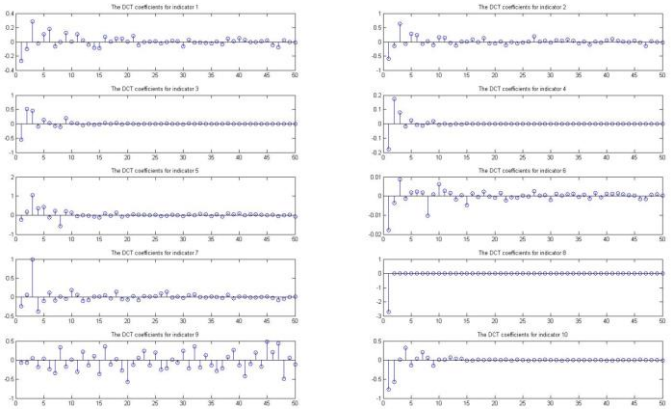

Figure 4. The DCT outputs for 50 days for each of the ten indicators

Pitch and log energy method

Pitch and log energy methods are combined to arrive at 5 features. This combination method is borrowed from the automatic speech recognition field. The first feature is obtained using log energy method. All column data is squared and summed and log is taken. This results in one value per day. The second step involves computing pitch energy features. This is done taking fast Fourier transform (fft) of the data. Followed by obtaining the absolute value of the fft, then log of the absolute value, followed by the inverse fast Fourier transform. This results in a Cepstrum. Within Cepstrum values, four highest values are chosen which contain most of the necessary information.

Principal Components Analysis

PCA seeks a new set of coordinates in the multivariate N-dimensional space for N features in such a way that the first N-1 new directions (principal components) explain as much as possible of all the original variables. In this paper, we have computed 5 PCAs for each indicator from each block of 50 days so we extracted 50 PCA type features. The results of this step will be used in future steps as input data for NN and HMM.

V. Neural Network

In this step, we are going to use the NN to predict the future of the stock market for some given days. In order to make the method general, a sliding window of days was used. This means one block of 250 days (one year) was used for training the NN and peaks and drops on prices over next 25 days are the outputs of NN. After all the computations, the data window is shifted by 10 days and all the NN computations are done again. Following is our NN setting:

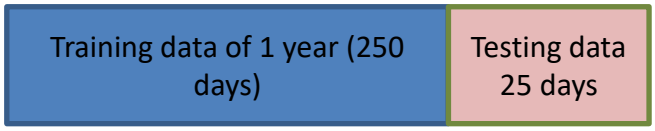

Figure 5. NN training and testing window

- Number of NNs: 2 networks
- Input data for NNs: extracted features of step 3
- Output data for NN1: %high rise
- Output data for NN2: %low fall
- Number of hidden layers: 4
- Internal layer function: '*tansig*'

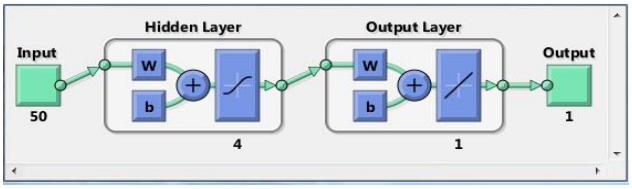

Figure 6. The designed NN

Different methods of feature extraction (Half Cosine, MATLAB DCT, Pitch and PCA) were used to train and test the NN. Since we have the actual values for each period of time, we can check the validity of our results. This shows in some cases that defined NN is a good tool for predicting the stock market but in some cases it can be less accurate. A detailed comparison is in Table 1. This table shows that among all these methods, half cosine and PCA gives more accurate results.

Table 1. Performance of different feature extraction methods using NN

Method	MSE of %high	MSE of %Low	Accuracy for 25 first days of 2012
Half cosine	0.0016	0.0011	%77.27
MATLAB DCT	0.0037	0.0044	%22.27
Pitch Method	0.0007	0.0025	%68.18
Using PCA	0.0006	0.0015	%86.36

VI. HMM

We have trained four HMM models based on following table:

Table 2- Market states used for creating HMM models

High-Low Condition	High Condition	Low Condition	Strategy
1	1	1	Placid Market
2	1	2	Good to sell
3	2	1	Good to buy
4	2	2	Risky Market

In order to improve the accuracy of HMM, we have created our HMM model as shown in figure 7. We have used a portion of our training data (20 days out of 250 days) as validation data. So we tested our model on validation data if the result was near to resulted states from actual data (above pre-defined threshold) then we keep the model and continue with the test data otherwise we re-run our HMM. We set our threshold to 60 percent in our experiment and defined an upper limit on the number of re-run which has been set to 5. The resulted accuracy is shown in figure 8. Mean value for the resulted accuracy is 48.2 percent.

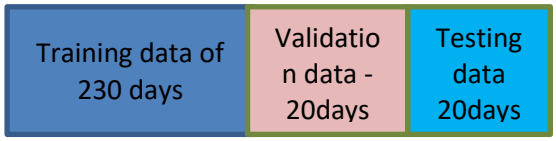

Figure 7. HMM model creation and testing strategy

These are other settings which have been used in HMM part:

- Number of States: 3
- Number of Mixtures: 4
- Number of Iteration: 3
- Input data: extracted features of step 3
- Number of features : 30

It is worth mentioning that in our experiment we have changed the setting several times and with the above reported numbers we got the best result.

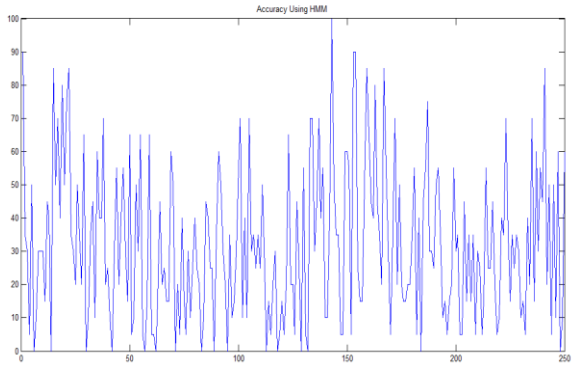

Figure 8. Resulted accuracy using HMM

Figure 9 shows one of our best results from HMM. It compares the predicated and actual states for one of test periods. As it is clear there were only two miss-prediction in this period.

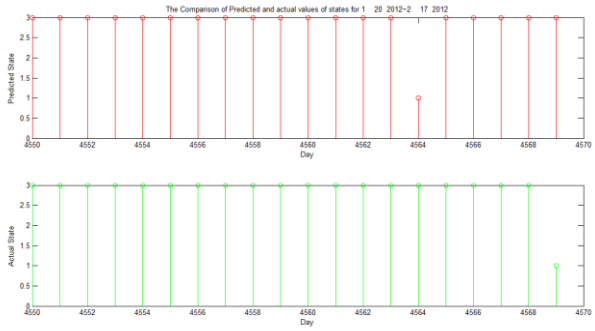

Figure 9. Comparison of predicted and actual states - one example of good result

Combination of NN and HMM

We thought in order to resolve the randomness issue which is inherent to NN and HMM, we can combine these together and obtain a better result. Motivating by this idea, we combined our HMM and NN in following way: first compare the outputs from HMM and NN and if the two outputs are similar (more than a pre-defined threshold) to each other then take a proper action according to buy-sell strategy otherwise re-run both HMM and NN. Like as before we defined an upper limit on the number of re-run which has been set to 5. Surprisingly we didn't get any improvement by this method and our resulted accuracy (Mean) dropped to 33.7 percent. So we didn't continue with this method in our final decision making process for buying and selling the stock.

VII. Post-processing and Buy-Sell strategy

In this section, we need to culminate this paper with making profit in the stock market by choosing a proper technique and an appropriate Buy/Sell Strategy. Several strategies can be applied and based on these strategies the resulted output would be different. We will test four different techniques in this part: NN, HMM, HMM_NN and HMM_PCA.

We have defined our buy-sell strategy as follows: We start with $10000 in the bank and on the first opening day we buy 20 shares based on the offered price of that day. In future, the number of shares to buy or sell is limited to one share per day. Figure 10 shows our bank balance after 20 years based on HMM prediction (red one) and compares it to the situation when we know the actual values ahead of time.

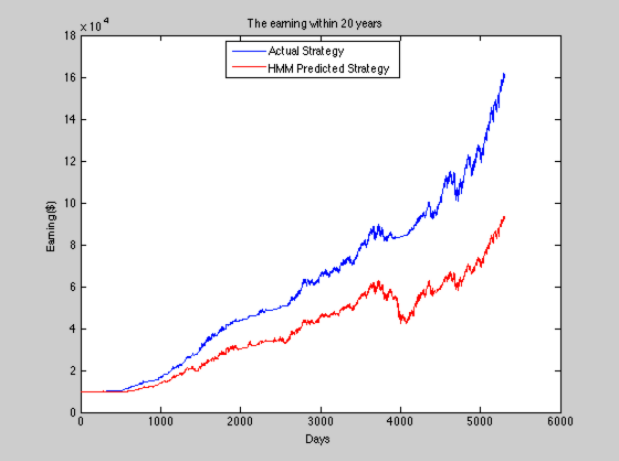

Figure 10. Bank Balance based on HMM prediction vs actual data

As it can be seen in figure 10 HMM could provide considerably satisfactory benefit to the investor. In addition, we could increase this amount by changing the number of shares that we buy or sell in each day.
Figure 11 shows how different methods are behaving based on our defined strategy. It can be seen that HMM by far has the best performance. All the other three methods almost performing similarly although NN and NN+HMM approximately follow the market pattern but they were not that efficient in our money making strategy.

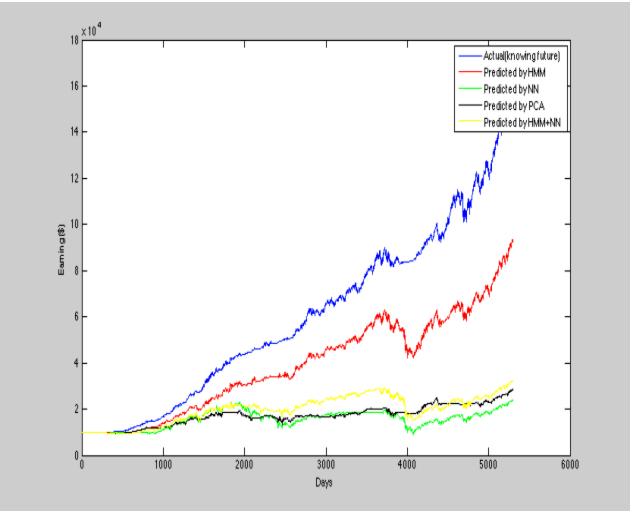

Figure 11. Bank Balance for All Methods

Figure 12 and 13 show the number of buy/sell in each year. The blue line shows how many shares we could buy in each year if we knew the actual data ahead of time and the orange line illustrates the number of shares sold by our strategy using HMM.

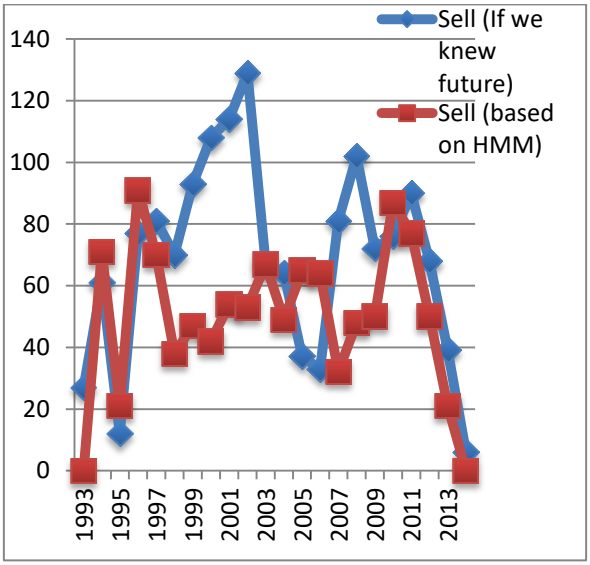

Figure 12. Number of Sell decision in each year

Similarly Figure 13 shows the number of shares bought by our method in each year and compares it with the number that we could buy if we knew the values ahead of time. In both cases our method is performing well although in some cases it could perform better.

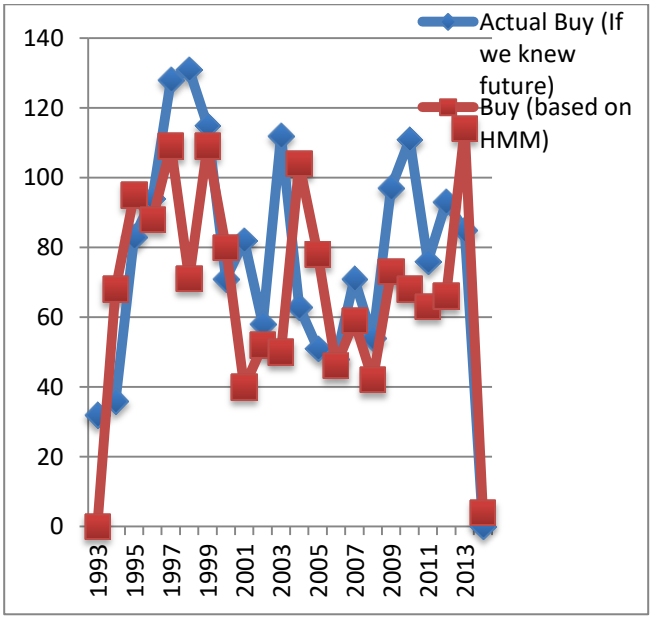

Figure 13. Number of Buy decision in each year

Since in this model, we are limited to sell or buy just one share per day, it may lead to money loss at some points. For example, consider the case that market trend is downward, the more reasonable choice would be to sell all the available stocks at the beginning of this trend as the prices will continue to decrease in future days but we are not allowed to do so. In other words, although we know the price is going to decrease further we could not withdraw our entire investment from stock so we will lose money. In order to address this problem, we removed the limitation of Buy/Sell one share per day and

investigated the result. Thus in the alternative model, we begin our investment by spending 10000$ in stock market and based on our buy/sell strategy we invest (buying as much shares as we can afford) or withdraw (selling all available shares) our money. Figure 14 shows the performance of this strategy. As it is clear in this figure there is no fall in our balance and it is cumulative when we know the actual value. However, when we are using HMM prediction, at some points we see some falls that means the behavior of stock market in the future was not predicted properly and we had some loss. So the performance of this strategy is better than the previous one as we can gain more profit when we remove the limitation. Table 3 clearly illustrates this. Figure 15 shows the number of shares based on this buy/sell strategy for HMM prediction and actual data.

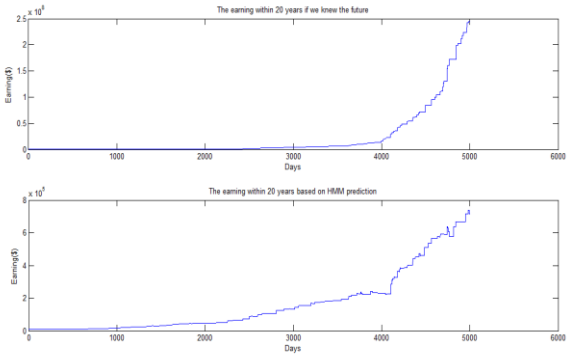

Figure 14. Bank Balance based on HMM prediction vs actual data

Table 3. Bank balance for HMM and actual value with/without limitation.

	HMM	Future is known
Bank_with_limitation	93050	163200
Bank_without_limitation	727300	244200000

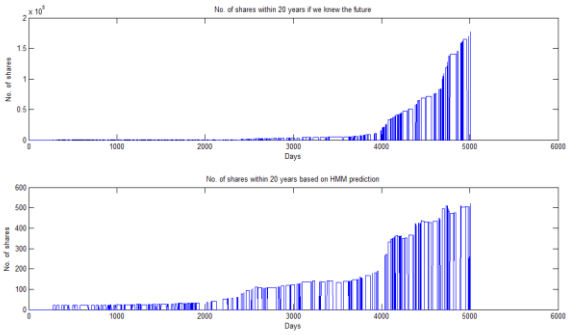

Figure 15. Number of available shares based on HMM vs actual data

Although it seems to be worthy to invest our entire budget in the market, it is extremely risky and the chance of loss is high because of some unpredicted situations. In contrast, when there is a limitation on buy/sell, the chance of loss would be low.

VIII. Conclusion

In this paper we applied some of machine learning techniques on stock market and tried to predict its trend and make profit based on that prediction. We have tested different combination of feature extraction methods with NN and HMM. Among feature extraction methods we got the best results from DCT and PCA on raw data. Our results in the last part show that the decisions we made based on the HMM are relatively similar to the decisions we would make if we knew the future. For profit making, choosing a proper strategy to Buy/Sell is very crucial so more effective strategies could lead to better performance in money making.

References

[1] S. Theodoridis, K. Koutroumbas, "Pattern Recognition", 4th Edition, Academic Press, 2009.

[2] L. Deng, D. O'Shaughnessy, "Speech Recognition: A dynamic and Optimization Oriented Approach", Marcel Dekker Press, 2003.

[3] M. Jabbar, and H. Ghorbaniparvar. "Determination of Volatile Components in Black Cardamom with Gas Chromatography-Mass Spectrometry and Chemometric Resolution ." I*nternational Journal of Engineering Research and Technology 3 (11), 1280-1286* 2014.

[4] M. Ghorbaniparvar and F. Ghorbaniparvar, "Portfolio optimization applied for wholesale electricity spot market (wesm) based on markowitz theory". *International Journal of Science and Modern Engineering*,. 2013.

[5] Dase, R.K. and Pawar, D.D., 2010. "Application of Artificial Neural Network for stock market predictions: A review of literature". *International Journal of Machine Intelligence, 2(2), pp.14-17.* 2008

[6] M. Ghorbaniparvar, X. Li, and N. Zhou, "Demand side management with a human behavior model for energy cost optimization in smart grids." In *Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on*, pp. 503-507. IEEE, 2015.

[7] "Technical Indicators and Overlays". Stockchart.com. Retrieved 18 February 2014. http://stockcharts.com/help/doku.php?id=chart_school:technical_indicators

[8] M. Jabbar, H. Ghorbaniparvar, F. Ghorbaniparvar, "Sensitivity of the General Linear Model to Assumptions" *International Journal of Innovative Science and Modern Engineering*, 2020

[9] Zekic, Marijana. "Neural network applications in stock market predictions-a methodology analysis." *In proceedings of the 9th International Conference on Information and Intelligent Systems, vol. 98, no. 1, pp. 255-263. Citeseer*, 1998.

[10] M. Ghorbaniparvar, "Survey on forced oscillations in power system." *Journal of Modern Power Systems and Clean Energy* 5, no. 5 (2017): 671-682.

[11] "Standard & Poor's 500 Index - S&P 500". Investopedia. Retrieved 18 February 2014.

[12] Brownstone, David. "Using percentage accuracy to measure neural network predictions in stock market movements." *Neurocomputing* 10, no. 3 (1996): 237-250.

[13] M. Ghorbaniparvar., and N. Zhou. "Bootstrap-based hypothesis test for detecting sustained oscillations." In *Power & Energy Society General Meeting, 2015 IEEE*, pp. 1-5. IEEE, 2015.

[14] M. Jabbar, and H. Ghorbaniparvar. "Use of GC-MS combined with resolution methods to characterize and to compare the essential oil components of green and bleached cardamom." *IJRCE* 5 (2014).

[15] Kim, Steven H., and Se Hak Chun. "Graded forecasting using an array of bipolar predictions: application of probabilistic neural networks to a stock market index." *International Journal of Forecasting* 14, no. 3 (1998): 323-337.

[16] M. Jabbar, and E. Konoz. "The combination of GC-MS with MCR Method to characterize and to compare the essential oil components of two types of cardamom" *20th Iranian Analytical Chemistry Conference*

[17] F. Ghorbaniparvar, and H. Sangrody. "PMU application for locating the source of forced oscillations in smart grids." In *IEEE Power and Energy Conference at Illinois (PECI)*, pp. 1-5. IEEE, 2018.