COMPARATIVE ANALYSIS OF LINEAR REGRESSION, MULTILAYER PERCEPTRON AND SUPPORT VECTOR MACHINES FOR ITS UTILIZATION IN STOCK PRICE PREDICTION

Deep Kotecha¹, Shivam Patel² and Smeet Patel³

¹,²,³ Department of Information Technology, G H Patel College of Engineering and Technology

Abstract—Stock price prediction is a momentous problem with no coherent solution. The stock market is a high risk and possibly high-profit entity and that is why predicting accurate stock prices could help address prevalent issues in the finance sector. Stock market being highly stochastic in nature, predicting stock prices accurately poses an exceptionally difficult challenge. Hence all known models to predict the stock prices should be meticulously analyzed to provide state-of-the-art forecasting for further explorations; hence we present a comparative analysis of three supervised learning methods: Regression Analysis, Multilayer Perceptron, and Support Vector Machines to determine the superlative amongst them. Stock data of S&P 500 index is used throughout the research. The average mean absolute error of 2.0145 is achieved using Linear Regression, 4.080 using Multilayer Perceptron and 4.5466 using Support Vector Machines. It was derived that Linear Regression outperforms Multilayer Perceptron and Support Vector Machines.

Keywords—Stock Price Prediction, Supervised Learning, Multilayer Perceptron, Support Vector Machines, Linear Regression

I. INTRODUCTION

A stock represents a share in the ownership of a company, hence establishing a symbiotic relation between investors and the company. The growth or fall of the company has a direct impact on profit gained or loss suffered by the investor. This makes the investor deeply interested in trends associated with the price of the given stock. Almost all the divisions of the society, be it an individual or an organization, deals in stocks. The number of people involved in stock trading makes stock markets a crucial aspect of economic stability of the country in consideration. Hence, it has a direct impact on the economy of the country and people, in general.

If an investor could predict the trend of a stock price, he could profoundly reap rewards from it. But the task of stock price prediction has remained an unsolved challenge even for the brightest of minds working in this field. [1] The Efficient Market Hypothesis (EMH) is a theory proposed by professor Eugene Fama that states it is impossible to beat the market since market prices should only react to new information or changes in discount rates. There have been attempts to predict stock prices accurately, but none have managed to achieve consistent results. The sheer enormity of the data generated and the fickle nature observed in stock prices makes it an arduous task.

With constant automation of almost every sector in the world, it is only logical to apply automation to the issue of stock price prediction. Applying soft computing methods with machine learning has the potential to answer the aforementioned need of independence of human involvement in price prediction.

Extensive research has been done to analyze stock market patterns and predict stock prices. [2] used a Multilayer Perceptron to predict the day ahead closing price of S&P 500 stock index. [3]
developed a neural network based on genetic algorithm for Singapore stock exchange. [4] proposes a
detailed study on development of neural network for modeling time series data. [5] predicts stock
market index movement using artificial neural networks as well as support vector machines. [7]
developed a prediction model using the Hadoop Distributed File System and Hadoop Map Reduce
Technique. [8] developed prediction model using Decision Trees for forecasting stock prices. [9] did
a comparative study of stock price tends using time delay, recurrent networks and probabilistic
neural networks.

This paper compares the performance of three techniques namely Linear regression,
Multilayer Perceptron and Support Vector Machines on the dataset of S&P 500 index. The rest of the
paper is organized as follows: Section 2 covers background information on aforementioned methods,
Section 3 discusses dataset in consideration and describes the proposed model for considered
techniques, Section 4 lists the results of these models, and Section 5 gives the conclusion.

II. MODELING TECHNIQUES

Many soft-computing methods have been tested for predicting the stock price value. For this
research, we decided to employ three major supervised learning techniques of machine learning
domain. And each of the techniques is introduced in subsequent paragraphs.

2.1. Linear Regression

Linear Regression has proved its worth in the field of statistics since a long time and more recently is
being used extensively in the field of machine learning. Linear Regression is one of the most basic
and common techniques used in the predictive analysis. It is a linear model which assumes a linear
relation between input variables and an output variable, i.e it tries to find a linear combination of
input variables to obtain the output variable. A generic look of linear regression model utilizing n
input variables to predict output y can be described as follows:

\[ y = A_0 + \sum_{i=1}^{n} A_i x_i \]  

where \( A_0, A_1, \ldots, A_n \) are the parameters of the linear model.

Due to the assumption of linearity between input and output variables, often transformations on data
are required to be done to fit the chosen linear model to the data.

2.2. Multilayer Perceptron

Multilayer Perceptron is a feed forward artificial neural network paradigm which consists of
computing nodes called neurons which are inspired by the human brain neurons which ensemble to
form a network operated on weights. The triggering of neurons is controlled by a mathematical
function called the activation function. The activation function is common to all neurons of the
artificial neural network.

Multilayer perceptron consists of three types of layers:
1. Input layer
2. Hidden layer
3. Output layer

Input layer consists of the nodes containing input attributes. Generally, number of neurons is equal to
the number of input attributes. The hidden layer acts as the processing unit between the input layer
and the output layer. The number of neurons in this layer is highly dependent on the application for which the model is being used. Whereas output layer consists of predicted output and generally consist of only one output neuron.

2.3. Support Vector Machines

A support vector machine is a distinguished technique formally defined by a separating hyperplane or a set of hyperplanes. Support vector machines are supervised learning models that analyze data and recognize patterns. The motive of support vector machines is to find an optimal hyperplane that gives the maximum margin on training data. The maximum margin hyperplane best splits the data. The optimal hyperplane maximizes the distance from the plane to any data point. Support vector machines are very resilient to overfitting. The boundary of the support vector machine only depends on a few data points from the training dataset called the support vectors.

The linear decision boundary cannot successfully separate data points in some cases but it still is possible to define a maximum margin hyperplane. Thus, Support Vector Machine can get more complex boundaries using the kernel trick. So using different kernels in support vector machine can help generate different non-linear boundaries which successfully divide the data points where linear boundaries generally fail. The effectiveness of support vector machines is highly dependent on the selection of kernel and selection of the kernel parameters. Some kernel choices available are Radial Basis Function (RBF) kernel, Hyperbolic Tangent kernel, Polynomial kernel. The best combination of kernel parameters is often selected using grid search.

III. PROPOSED MODELS

Our paper uses the historical dataset of S&P 500 index for training and testing previously mentioned methods. S&P 500 reflects trend associated with prices of 500 large-cap companies of United States of America. The features considered for training of all the three considered techniques are Open, High, Low, and Close index values.

All the three prediction methods - Linear Regression, Multilayer Perceptron, and Support Vector Machine have been trained with nine years of working datasets for each decade from 1950 to 2010. These classifiers are made to predict close index values for last year of each decade since 1950 till 2010 and prediction for the year 2016 were made by using a training set containing dataset of years 2007-2015. Thus in-sample data for any predicted year ranges nine years prior to that year and out-sample is data for the year that is forecasted. Configurational setup for proposed models is described subsequently.

3.1. Linear Regression

For generating an effective model, it is vital that parameters for fitting the data points are chosen wisely so as to make accurate predictions. The M5 method selects the optimum attributes for our dataset. The parameters shown in equation (1) must be optimized in order to get a model that gives the best forecast. The value of these parameters scales the whole real number line.

3.2. Multilayer Perceptron

Our model of Multilayer Perceptron constitutes of only three layers with two of them being input layer and output layer each and one being the hidden layer.
The activation function used for the nodes is the sigmoid function given by following equation:

\[ f(x) = \frac{1}{1 + e^{-x}} \]  \hspace{1cm} (2)

where \( x \) is the input to the activation function. And the value of \( x \) is the sum of the product of incoming activation levels \( S_j (j^{th} \text{ node}) \). The incoming sum (for node \( j \)) is computed as follows:

\[ S_j = \sum_{i=0}^{n} w_{ji} a_i \]  \hspace{1cm} (3)

in which \( w_{ji} \) is the incoming weight from unit \( i \), \( a_i \) is the activation value of unit \( i \) and \( n \) the number of units that send connections to unit \( j \) [6].

The proposed model consist of single hidden layer of neurons optimized to perform the task at hand. All the neurons in the hidden layers are controlled by the aforementioned sigmoid activation functions, hence they can also be called sigmoid nodes. Whereas the neurons in input and output layers are driven in a linear fashion, hence they can also be called linear nodes.

Apart from selecting a good activation function, in order to obtain the state-of-the-art accuracy, we need to control the learning rate for neurons which defines the step size in weight space in each iteration of weight update equation, set the momentum which specifies the degree to which the neural network can resist itself from converging to a local minimum, and number of epochs which is the number of times the model iterates over the entire dataset.

Table 1. Multilayer Perceptron Hyperparameter Configuration

| Hyperparameter       | Value |
|----------------------|-------|
| Learning Rate        | 0.3   |
| Momentum             | 0.2   |
| Number of Epochs     | 500   |

The weight function updation equation used in the multilayer perceptron is as follows:

\[ W_{\text{next}} = W + \Delta W \]  \hspace{1cm} (4)

\[ \Delta W = - \text{Learning Rate} \times \text{Gradient} + \text{Momentum} \times \Delta W_{\text{previous}} \]  \hspace{1cm} (5)

3.2. Support Vector Machines

Our support vector machine utilizes radial basis function kernel (RBF kernel). RBF kernel represents the similarity between vectors, new data vectors and support vectors, as a decaying function of the
distance between the vectors. The RBF kernel is defined as follows:

$$K_{RBF}(x,x_i) = \exp\left[-\gamma ||x - x_i||^2\right]$$  \hspace{1cm} (6)

where $x$ is the input vector, $x_i$ is the support vector, $\gamma$ (gamma) represents the “spread” of the kernel.

For optimum performance of our model, we had to ensure optimum value for parameter $\gamma$ and $C$ (a tuning parameter that defines the amount of violation of the margin that is allowed). Our model uses the values for gamma and C as mentioned in the following table:

**Table 2. Support Vector Machine Hyperparameter Configuration**

| Hyperparameter | Value |
|----------------|-------|
| $\gamma$       | 0.01  |
| $C$            | 1.0   |

**IV. RESULTS**

To better observe if our trained models stood the test of time or not, predictions for distinct years for each decade since 1950 to 2016 were made. Forecasting for any considered year is from 1st January to 31st December of that year. To evaluate the performance of trained models, we chose Mean Absolute Error as a metric of error measurement. Mean absolute error showcases the closeness of the predicted value to that of the actual value.

**Table 3. Results of opted methods for distinct years from 1950 – 2016**

| Forecast Year | Linear Regression | Multilayer Perceptron | Support Vector Machines |
|---------------|-------------------|-----------------------|-------------------------|
| 1959          | 0.0000            | 0.1436                | 0.1554                  |
| 1969          | 0.1988            | 0.2046                | 0.2922                  |
| 1979          | 0.2242            | 0.2331                | 0.2907                  |
| 1989          | 0.6725            | 1.0527                | 1.3297                  |
| 1999          | 5.0294            | 15.8511               | 9.7363                  |
| 2009          | 3.9415            | 6.3755                | 6.4058                  |
| 2016          | 4.2242            | 4.7053                | 6.6161                  |

**Table 4. Results of opted methods for distinct years from 1950 – 2016**

| Supervised Technique     | Average Mean Absolute Error |
|--------------------------|----------------------------|
| Linear Regression        | 2.0415                     |
| Multilayer Perceptron    | 4.080                      |
| Support Vector Machine   | 4.5466                     |

**V. CONCLUSION AND FUTURE WORK**

We aimed at evaluating the efficiency of three supervised machine learning techniques: linear
regression, multilayer perceptron, and support vector machines, and deciding the superior alternative amongst these three classifiers for the dataset in consideration and the span of time for which predictions were to be made. The intuition behind predicting close index values for distinct years of each decade from 1950-2016 was to ensure that these models work well enough on S&P 500 index datasets irrespective of changing times. Forecasting for one year is done for each trained model. The results show remarkable success for regression method. The worst performance of regression has a mean absolute error of 5.0294. Even in such condition, it outperformed more complex methods such as multilayer perceptron and support vector machines. It can be averred from the results that support vector machines are outperformed by multilayer perceptron in all instances but one.

It is interesting how linear regression can outperform multilayer perceptron and support vector machines because of the negligible overfitting in Linear Regression in our model. The fact that we predicted over the index price of the S&P 500 which tends to get linear, as this index price is calculated based on aggregation of performances of 500 selected companies, supports the use of linear regression. Thus, Linear regression comes out as a winner amongst these three methods for forecasting close index price of the S&P 500 dataset for a period of one year when the models were trained with nine years’ worth of historical dataset. Hence, our study concludes that linear regression performed better than multilayer perceptron and support vector machines in this case. To further test this conclusion, different datasets could be used and long term or short term predictions could be derived. Moreover, testing the proposed models on stock data of individual companies and in different market scenario might yield a different and more interesting outcome.

An opportunity for further research also emerges from applying and tweaking different hyper parameters of the forecasting models and using different linear and polynomial kernels for prediction. This would test the performance of the model under changed circumstances. Including other technical indicators and external factors such as news affecting the involved companies, as attributes for forecasting as well as using techniques other than the ones used in this paper might result in better outcomes.

REFERENCES

[1] Malkiel, B. G. and Fama, E. F., “Efficient Capital Markets: A Review of Theory and Empirical Work”, The Journal of Finance 25, pp. 383-417, May 1970
[2] Deep Kotecha, Shivam Patel, Smeet Patel, “Stock price prediction using artificial neural networks”, International Conference on Intelligent Systems and Signal Processing, 2017.
[3] Phua, P.K.H. Ming, D., Lin, W., “Neural Network With Genetic Algorithms For Stocks Prediction”, Fifth Conference of the Association of Asian-Pacific Operations Research Societies, Singapore, 5th–7th July, 2000.
[4] J.H.wang, J.Y.Leu, “Stock market trend prediction using ARIMA-based neural network,” Proc. of IEEE conference on neural networks, vol.4, pp.2160-2165, 1996.
[5] Y. Kara, O. K. Baykan, M. A. Bayacioglu, “Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange”, Expert Systems with Application. May 2011
[6] Mehrrotra K, Mohan CK, Ranka S., “Elements of Artificial Neural Networks”, MIT Press, Cambridge, MA, 1997.
[7] Lathika J Shetty, Shetty Mamatha Gopal, “Developing Prediction Models for Stock Exchange Dataset Using Hadoop Map Reduce Technique” International Journal of Engineering and Techniques Vol. 2 Issue 3, May - June 2016.
[8] Qasem A. Al-Radaideh, Adel Abu Assaf, Eman Alnagi, “Predicting Stock Prices Using Data Mining Techniques”, The International Arab Conference on Information Technology, 2013.
[9] E. Saad, D. Prokhorov, D. Wunsch, “Comparative Study of Stock Trend Prediction Using Time Delay, Recurrent and Probabilistic Neural Networks”, IEEE Transactions on Neural Network, Vol.9, No.6, pp.1456-1470, November, 1998.