Empirical Evaluation of various Machine Learning algorithms for Stock Market Trend Prediction

Vyas Anirudh Akundy
University of Waterloo
vaakundy@uwaterloo.ca

***Abstract*— Predicting the trend of stock prices is a very interesting and tricky task in financial time series forecasting. There have been many attempts in the past to develop models to solve this problem using both classical machine learning and deep learning techniques. The unpredictability and dependency of the stock market on a large number of factors introduces complexity into the problem. In this project, I consider approaches like Support Vector Machines(SVM), Linear Regression as well as neural network models like Long Short-Term Memory Networks(LSTM) and simple feed-forward networks, and compare the performance of each technique on how well they predict the trend based on past historical data. The dataset used is the Dow Jones Industrial Average(DJIA) stock data. Each technique is evaluated based on how well it is able to emulate the actual stock trend, the time complexity and also on the mean squared error relative to the true prices.**

I. Introduction

The problem being addressed in this project is stock market trend prediction. The goal of the prediction is to estimate the future price, in this case the closing price, of a stock based on past historical data.

This is a very difficult problem to tackle because of the unpredictable movement of the prices. Generally there are two kinds of analysis - Fundamental and Technical analysis. Fundamental analysis involves trying to find out the true value of the stock to compare it with its cost/selling price. It looks at the industry, economy, company's health etc. to evaluate the profits of buying or selling a stock by comparing it with the true value. On the other hand, Technical analysis deals with predicting the stock solely based on statistical trends and by looking at charts.

This problem is important one both for investors and academicians. For investors, information about the trend of stock, along with other factors influencing the company, will help them make better decisions about their investments. For academicians, this is a very interesting and tricky problem as it is a non-stationary time series problem which does not follow any particular trend.

In this project, I will present the application of few machine learning algorithms to the trend prediction, compare each algorithm's performance on their time complexity and on how well each algorithm predicts the trend.

The remainder of this paper is organized into the following sections. Section II will cover a brief review of previous works concerning this problem. Section III will describe the algorithms being studied in detail and Section IV will present the empirical analysis which will provide insights into the performance of the various algorithms. Section V will conclude the evaluation with a brief summary.

II. Literature Survey

Over the past few years, there have been many studies regarding the prediction of the stock price movement and trend prediction utilizing a variety of tools and techniques. This section will cover a few of the related works.

Artificial Neural Networks(ANNs) and Support Vector Machines(SVM) have been shown to possess good predictive capability in financial time series modeling. [8] examined the application of ANN and SVM to predict the stock price movement, i.e they have defined a classification task to predict whether there will be a rise or fall in the price next day based on the previous 10 days data. They have used ten technical indicators(derived from the stock exchange data) as their input fea-

tures to a 3-layer feed-forward neural network. Their study suggests that these two models(ANN-75.74% and SVM(71.52%) have outperformed previous work in the literature.

[4] have presented a detailed study of various Neural Network architectures such as Convolutional Neural Networks(CNN) and Long Short-Term Memory(LSTM) Networks and Wavelet Neural Networks(WNN). They explored each architecture individually and also compiled them together to form an ensemble of neural networks which performed much better than any one individually. They employed various feature extraction/selection techniques to obtain appropriate feature to be sent as input to their models. In their study, they showed that using CNN with PCA(Principal Component Analysis) selected features has actually worsened their model because PCA loses the time-based structure of the data.

The above works relied purely on numerical data only, however [1] explored the use of text data, i.e new articles along with numerical data to predict the movement of the price. They have employed a Deep Neural Network consisting of simple dense layers and LSTMs which take in a combination numerical and textual data(which has been represented as Paragraph Vectors). Their study revealed that using distributed representations of textual information along with numerical information are better than the numerical-data-only methods.

Apart from Neural networks, there have been successes in implementing other techniques, one of which is a hybrid combination of SVM and the autoregressive integrated moving average (ARIMA) model. ARIMA is a popular forecasting model which is similar to linear regression. In [7], they have used the ARIMA model as a preprocessor to filter the linear patterns in the data. Then they fed the error terms to an SVM to reduce the error produced by the ARIMA model and finally make predictions. Their experiement showed that ARIMA and SVM model when used together, complemented each other and greatly improved the predictive performance than when taken individually.

III. Techniques implemented

The algorithms explored in this project are as follows:

- Neural Networks(NN)
- Support Vector Machines(SVM)
- Generalized Linear Regression

Each of the above technique is explained briefly in sub-sections A, B and C.

A. Neural Networks

Neural Networks are a family of models that are designed to mimic the human brain. NN are used to approximate functions that produce some kind of non-linear mapping from a given input variable **x** to an output variable **y** One kind of NN, known as Recurrent Neural Networks(RNN) are the most widely used for sequential data such as time series. In this project, I have implemented the Long Short-Term Memory Networks(LSTM)[6] and simple feed forward networks. LSTMs are specifically useful when we have a long time dependent data, i.e data which depends on the past values. This is the reason I chose to work with LSTMs because stock market data is a time series data and the essence of the time seires is captured by LSTMs. The output of each LSTM cell is obtained as follows:

$$i_t = \sigma(W^{ii}x_t + W^{hi}h_{t-1})$$
$$f_t = \sigma(W^{if}x_t + W^{hf}h_{t-1})$$
$$o_t = \sigma(W^{io}x_t + W^{ho}h_{t-1})$$
$$\tilde{c}_t = \tanh(W^{i\tilde{c}}x_t + W^{h\tilde{c}}h_{t-1})$$
$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$
$$y_t = o_t * \tanh(c_t)$$

where i_t is the input gate, f_t is forget gate, o_t is the output gate, c_t the cell update and y_t is the output

B. Support Vector Machines(SVM)

SVM [3] is a powerful machine learning algorithm which was very popular before the advent of neural networks. It can be used for both classification and regression tasks. The basic intuition behind SVM is to find a hyperplane that best separates the given data points into the appropriate classes(in case of classification). A kernel function is used to map the data to a higher dimension where it can perform linear regression or classification.

Date	Opening Price	High	Low	Closing Price	Volume
2006-01-03	211.47	218.05	209.32	217.83	13137450
2006-01-04	222.17	224.70	220.09	222.84	15292353
2006-01-05	223.22	226.00	220.97	225.85	10815661
2006-01-06	228.66	235.49	226.85	233.06	17759521
2006-01-09	233.44	236.94	230.70	233.68	12795837

TABLE I
SAMPLE DATA

According to [2] "SVM estimates the regression using a set of linear functions that are defined in a high-dimensional feature space" . I have used SVM as one of the algorithms since it has been shown to produce good results with low computational power in the past.

C. Generalized Linear Regression

Linear Regression is one of the most simplest algorithms in machine learning. This algorithm tries to find a linear function that best fits the given data based on simple linear algebra. The parameters learned by this algorithm are the coefficients of the linear function. Linear regression will work successfully if the data is linear or at least as linear as possible. However, stock market trend is highly unpredictable and non-linear.
Hence one way would be to map the given data to a higher dimension where it would be linear in the higher dimension. We could perform linear regression in the higher dimension thus doing non-linear regression in the original space. In this project, I have taken a polynomial mapping of the data and experimented with different degrees.
The problem of linear regression can be framed as follows:

$$y(x, w) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_M x_M$$

where y is the prediction and x is the input, with some loss function to optimize the weights/parameters **w**, given the target t_n

$$L_2(w) = \frac{1}{2}\sum_{n=1}^{N}(y(x_n, w) - t_n)^2$$

and in case of a generalized linear model we can replace x with $\phi(x)$, a non-linear mapping to a higher dimension. So the new set of equations will be given as

$$y(\phi(x), w) = w_0 + w_1\phi(x_1) + w_2\phi(x_2) + ... + w_M\phi(x_M)$$

$$L_2(w) = \frac{1}{2}\sum_{n=1}^{N}(y(\phi(x_n), w) - t_n)^2$$

IV. EMPIRICAL EVALUATION

This section will discuss the performance and complexity of each technique, experiments conducted and also briefly about the data being used.

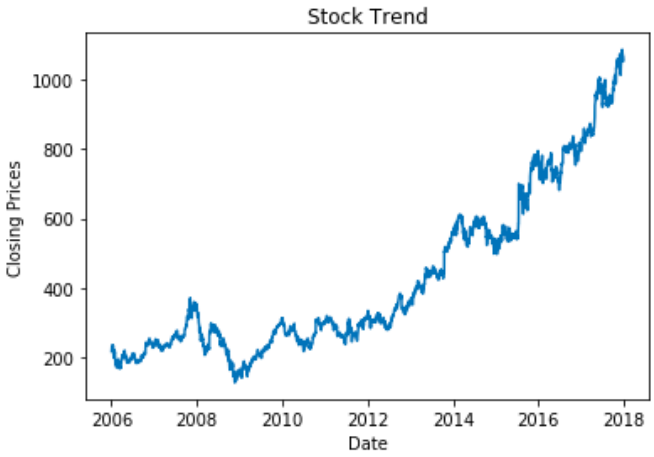

Fig. 1. Change in Close prices with time

A. Data

The dataset for this study has been obtained from the Kaggle [5] website, titled "DJIA 30 Stock Time Series". The dataset consists of historical stock prices of 30 companies in the Dow Jones Industrial Average(DJIA) dated from 2006-01-01 to 2018-01-01. Out of the 30 companies, I have selected Google's stock prices for the evaluation. The first five rows of the data is shown in Table I. A plot of the closing prices, which is the prediction target, is shown in Figure 1. The data was standardized to zero mean and unit standard deviation to bring all the features to a common scale as the larger values would have skewed the

models predictions.
For the purpose of the evaluation, the dataset was divided into a training set and test set. There are a total of 3019 data points in the complete dataset out of which 80% were used for training and 20% were used as a testing set.

B. Long Short-Term Memory Networks:

Figure 3 illustrates the architecture used for the study. It consists of two LSTM layers followed by a dense layer and an output layer. The input features are the opening price, highest price, lowest price, closing price and the volume. A single input consists of 10 data points which are the past 10 days prices. So the model is trained to look at the past 10-day data to predict the next day's closing price. For example one data point consisted of prices from 2006-01-03 to 2006-01-12, and the target is the closing price on 2006-01-13.
The parameters of the model were experimented with, and the best parameters were selected by using GridSearchCV provided by the sklearn package. Table 2 represents the different parameter values used to conduct the analysis. The best model was found for the following parameter setting : {Epochs: 25; Batch Size: 32; LSTM Layer 1 : 70 hidden units; LSTM Layer 2 : 100 hidden units; Dense Layer 1 : 50 hidden units}Computationally, it took more time as a large number of weights are involved . The model was built using the Keras framework.
After training the model on the training set, predictions were made on the testing set and they were compared with the actual labels. The comparison was done with the help of a plot of the actual stock price trend versus the predicted stock price trend as shown in Figure 2. From the graph we can see that the LSTM, though has predicted lower values for the prices, maintains the pattern of the trend when compared to the actual trend to a good extent.

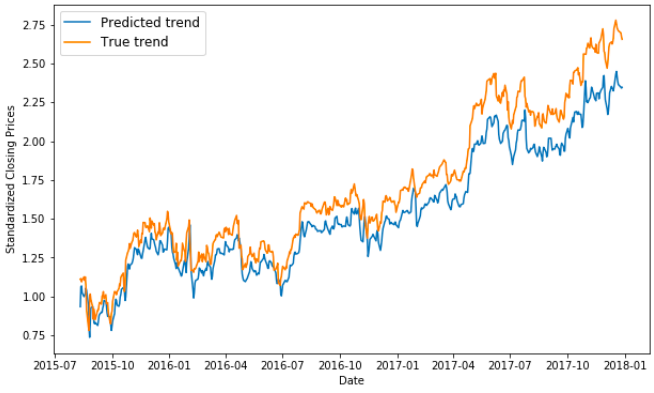

Fig. 2. Predictions for the LSTM model

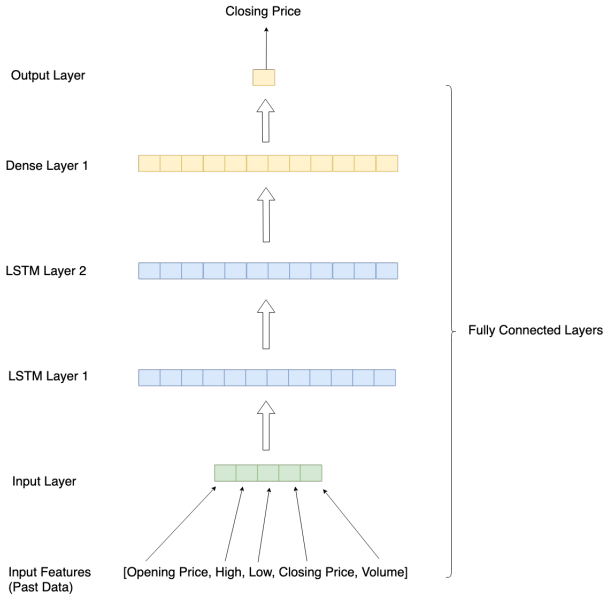

Fig. 3. Architecture for the Experiment

Parameter	Range of values
Epochs	10,15,20,25...100
Batch Size	32,64,128,256,512
Number of hidden Units	10,20,30...100

TABLE II
PARAMETER SEARCH FOR LSTM

C. Feed-Forward Neural Networks

The architecture of this neural network is similar to Figure 2 except that the LSTM layers are replaced by Dense layers or fully connected layers. These layers were also constructed with the Keras framework and the number of epochs, batch size and number of neurons were varied as per Table III.
The best model was found for the following parameter setting : {Epochs: 30; Batch Size: 64; Dense Layer 1 : 90 hidden units; Dense Layer 2 : 90 hidden units}
Computationally it took lesser time as compared to LSTMs because of less complexity involved in

the simple dense networks. Looking at Figure 4, we can see that the simpler dense neural network not only predicted almost similar prices, but also maintained the variation in the prices to a greater extent than the LSTM model.

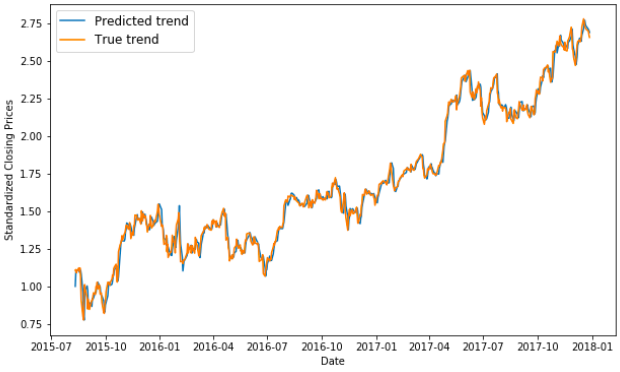

Fig. 4. Predictions for Feed-Forward Nerual Network

Parameter	Range of values
Epochs	10,20,30,...100
Batch Size	32,64,128,256,512
Number of hidden Units	10,20,30,...100

TABLE III

PARAMETER SEARCH FOR FEED FORWARD NEURAL NETWORK

D. Support Vector Machines(SVM)

Parameter	Values
Kernels	RBF, Linear, Polynomial(degree=1, 2, 3)
Gamma(γ)	0.1, 0.2, 0.3, ... 4
C	1, 10, 50, 100

TABLE IV

PARAMETER SEARCH FOR SVM

A similar experiment was conducted. The input features fed into the model were standardized values of opening price, high, low, volume and the closing price was chosen as the target value. The parameters that were explored were the kernel function, gamma value(γ) and regularization constant(C). The kernel functions considered were rbf, polynomial and linear kernel. Table IV summarizes the different parameter settings that were used to conduct the experiment.
For each of the kernel functions, the value of γ and regularization constant C were varied as in the table. As before, GridSearchCV was used to find the best parameters. The experiment revealed the following results.
Mean Squared Error(MSE) for **'Linear'** kernel with C = 1 is **MSE = 0.0068369** and Mean Squared Error(MSE) for **'Polynomial'** Kernel with C = 1 and gamma $\gamma = 3.1$ is **MSE = 0.0068109**. Hence, though very close in terms of the plot, the Polynomial kernel is marginally better than the linear kernel(both performing much superior to the RBF kernel). The plot of the actual versus predicted trend is shown in Figure 5

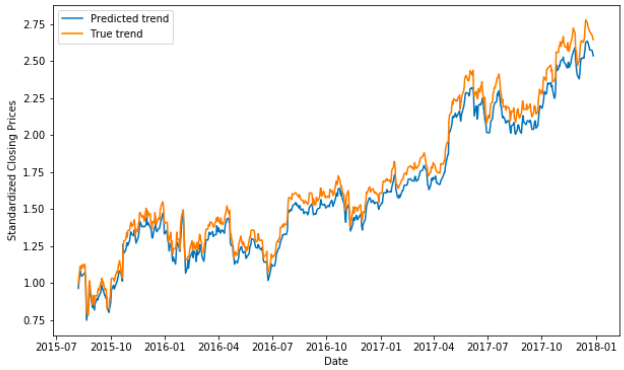

Fig. 5. Predictions for the SVM model

E. Generalized Linear Regression

The given 4 features - open, high, low, volume were mapped to a higher dimension using a monomial basis function. Different values of the degree of the monomial were experimented. For example if the there are 2 input features

$$[x_1, x_2]$$

then the degree 2 polynomial mapping is given as

$$[1, x_1, x_2, x_1^2, x_1x_2, x_2^2]$$

This is the new feature space which is used as the input features for the model. Thus we can still perform non-linear regression in the original space, but actually performing linear-regression in the higher dimension. The degree of the polynomial was varied from 1 to 5. Table V shows the values mean squared error as the degree was increased.

From the MSE values above, the polynomial mapping that best fits the data is with degree 1,

Degree of Polynomial mapping	MSE
1	**0.00023943**
2	0.00026975
3	0.00146371
4	0.02023719
5	3.28671376

TABLE V

MSE VALUES FOR DIFFERENT DEGREE POLYNOMIALS

Algorithm/Technique	Running Time(s)
LSTM	51.27
Feed-forward NN	6.43
SVM	0.01792
Generalized Linear Regression	0.00698

TABLE VII

COMPARISON OF RUNNING TIME

and the corresponding plot for the actual versus predicted trend is shown in Figure 6

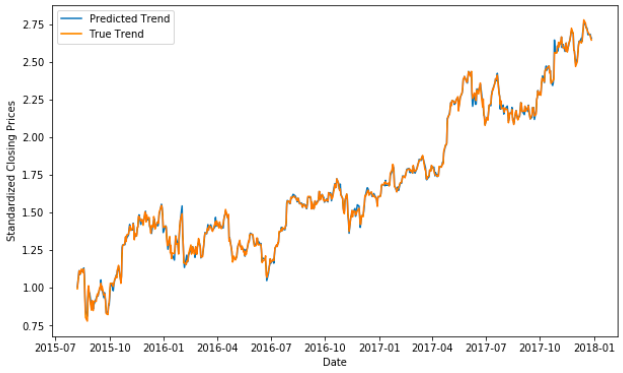

Fig. 6. Predictions for Linear Regression model

F. Comparison of different techniques

The aim of this study was to compare the different techniques based on their performance, i.e how well each method predicted the prices, and also on the complexity of each technique.

Algorithm/Technique	Training MSE	Testing MSE
LSTM	8.9787e-04	2.3723e-02
Feed-forward NN	9.2461e-04	2.0034e-03
SVM	1.11664e-03	6.8109e-03
Generalized Linear Regression	6.958e-05	2.3943e-04

TABLE VI

COMPARISON OF MSE VALUES

The tables VI and VII present the mean squared error(MSE) and running time values for the different approaches used in predicting the trend of the stock market closing prices. In terms of the number of parameters involved in the optimization, neural network models are very complex as there are a large number of weights involved based on the number of hidden units, hence having a larger running time as compared to the other models. It is interesting to note here that the simpler dense nerual network model is outperforming the LSTM model which can be observed by looking at the predictions and also at the MSE values. The simplest of all is the linear regression model, which has also outperformed all the other techniques. In this the number of parameters(weights) involved are proportional to the number of features being used as input.

Looking at Table VI, based on the training and testing MSE's, we can see that the SVM model is generalizing very well as compared to the other techniques due to the smaller drop in MSE from training to testing, indicating less overfitting than the other models , which is in accordance with theory.

V. CONCLUSION

This project presented an empirical evaluation of different machine learning techniques to address the problem of stock market trend prediction. The techniques used were Neural Networks, Support Vector machines and Generalized Linear Regression. The techniques were compared based on how closely they predicted or emulated the behaviour of the closing prices trend(mean squared error) and also on their time complexity.

Based on the evaluation, out of the three methods, linear regression surpassed all the other techniques both in terms of mean squared error values and the predicted trend of the prices, which was on par with the performance of the simple dense neural network.

However, we cannot conclude by saying that this problem is completely solved, since this study involved the use of only one particular dataset. In the future, more datasets with a much more complex trend will be evaluated to see whether

these models generalize as well as they are doing now. Additionally, more input features and a more comprehensive feature engineering can be considered than what has been done now, and textual information can be combined along with numerical information which may provide additional insights into solving this problem.

REFERENCES

[1] R. Akita et al. "Deep learning for stock prediction using numerical and textual information". In: *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. June 2016, pp. 1–6. DOI: 10.1109/ICIS.2016.7550882.

[2] L. J. Cao and F. E. H. Tay. "Support vector machine with adaptive parameters in financial time series forecasting". In: *IEEE Transactions on Neural Networks* 14.6 (Nov. 2003), pp. 1506–1518. ISSN: 1045-9227. DOI: 10.1109/TNN.2003.820556.

[3] C. Cortes and V Vapnik. "Support-Vector Network". In: *Machine Learning* 20 (1995), pp. 273–297.

[4] Luca Di Persio and O Honchar. "Artificial neural networks architectures for stock price prediction: Comparisons and applications". In: 10 (Jan. 2016), pp. 403–413.

[5] *DJIA 30 Stock Time Series*. URL: https://www.kaggle.com/szrlee/stock-time-series-20050101-to-20171231.

[6] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[7] Ping-Feng Pai and Chih-Sheng Lin. "A hybrid ARIMA and support vector machines model in stock price forecasting". In: *Omega* 33 (Dec. 2005), pp. 497–505. DOI: 10.1016/j.omega.2004.07.024.

[8] Omer Kaan Baykan Yakup Karaa Melek Acar Boyacioglub. "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange". In: *Elsevier* 38 (2011).