International Journal of Research in Engineering, Science and Management
Volume-2, Issue-5, May-2019
www.ijresm.com | ISSN (Online): 2581-5792

Stock Market Prediction and Analysis using Machine Learning Algorithms

Ruchit Desai[1], Deep Shah[2], Manan Shah[3], Prakshal Shah[4], Suchitra Patil[5]
[1,2,3,4]*Student, Department of Information Technology, K. J. Somaiya College of Engineering, Mumbai, India*
[5]*Assistant Professor, Dept. of Information Technology, K. J. Somaiya College of Engineering, Mumbai, India*

***Abstract*: Stock market is one of the foremost sources of raising resources in the developing economies. The prediction of stock exchange trend could be terribly tough and highly difficult task because it is depends on various factors like economic conditions, Investors sentiments and political events. Because of the various factors and to achieve scoring profits it is mandatory to accurately predict the value of the stock. Along with the prediction of the stocks it is equally important to sell the stocks at the right time to maximize the profits and stay in the economy. The project focuses on allowing users to get an estimation of the stock prices in the near future by studying and analyzing the trends and fluctuations in the stock prices as per season and trend in the market. We will establish a relationship between fluctuating prices, volume and data mining techniques when applied to the stocks. Artificial intelligence forecasting models, such as ANN and SVM, they can automatically extract knowledge of economic movements from historical data; overcome many shortcomings and difficulties of traditional quantitative analysis method [5]. So in this project we will use Support vector Regression (SVR), Recurrent Neural Network (RNN) techniques to predict the stock market prices. To help the user for better visualization and analysis of the stock we will provide a dashboard with respect to the shares that they have shown interest in.**

***Keywords*: Support Vector Machines, Recurrent Neural Network, Classification, Regression**

1. Introduction

Forecasting of the financial market has been an attractive topic to the scholars and researchers of various fields. The field of financial forecasting is characterized by data intensity, noise, non-stationarity, unstructured nature, and hidden relationships. Predicting financial indicators is therefore a difficult task. However, forecasting is important in the sense that it provides concrete data for investment decisions [1]. Financial market is an abstract concept where financial commodities such as bonds, stocks and precious metals transactions happen between buyers and sellers. In the present scenarios of financial market world, especially in the stock market, forecasting the trends or the prices of the stocks using machine learning techniques and artificial neural network are the most attractive issue to be investigated. Stock market prediction is challenging and attractive topic to economics, history, finance, mathematics and computer science. The project which we are developing will try to provide solution for the users to buy and sell the stocks for greater profits by suggesting the best stocks for short term as well as long term. Whenever there is an unusual behavior in the shown stock, a popup will be shown to the user giving him suggestion to either to buy or sell the stock. Data mining refers to the extracting the knowledge from the large datasets. Some of the functionalities of data mining are class description, association and correlation, classification, prediction, clustering, outlier analysis and similarity analysis. Data classification can be done with many methods; one of the methods is Decision tree. It is a graphical representation of the all the possible outcomes and the paths by which they can be reached. We will be using two best prediction methods support vector machine and Recurrent neural network. Now Support vector machine is a useful and powerful machine learning techniques to predict the stock prices. It can produce good prediction result if the value of the important parameters can be determined properly. Data of the companies are collected from National Stock Exchange (NSE). Historical data set are collected and based on the historical data prediction is made and the prediction using SVM is compared with the actual prices of NSE to Evaluate the model performance. The prediction of the prices is also done using RNN. RNN have a set of interconnected nodes that simulate the individual neurons, and are organized into different layers based on the function. Recent study shows that RNN are quite accurate when the data have a large amount of variation. RNNs are models that try to minimize classification error within the training data, SVMs may make classification errors within the training data in order to minimize the overall error across the test data. A major advantage of SVMs is that it finds global optimum, whereas neural network may find a local optimum.

2. Methodology

A. Support vector classification

Support Vector Machines has proved itself appropriate in the case of classification. They create a hyper plane which creates a separation such that points belonging to a particular category fall on one side of the plane and points of other category get separated on the other side. Consider an n-dimensional feature vector

x=(X1,….,Xn). The linear boundary or (HyperPlane) is given as

β0+ βX1+…..+ βnXn = β0+n∑i=1 βiXi=0 (1)

So According to the formula elements in one category should be greater than 0 and In one category should be less than 0. With labeled examples β0+∑ni=1 βiXi=y, where y is the label. In our classification, y belongs {-1,1}. Below equation is the HyperPlane Equation with Linear products

Y = β0+∑(iyix (i) * x (2)

Here the operator * denotes the inner products . Note that the inner product is weighted by its label. The optimal hyper plane is the set of classifier with maximum distance for a given finite set of learning patterns. The maximum margin hyper plane (MMH) best splits the data. MMH may not create a perfect differentiation between the classes, but we can add error variables ∑1,…∑n and keep their sum below some Budget B. The crucial element is that only the points closest to the boundary matter for hyper plane selection; all others are irrelevant. These points are known support vectors, and the hyper plane is known Support Vector Classifier (SVC) since it classifies each support vector into one of the classes.

B. Support Vector Regression

Consider the problem of approximating the set of data,

D=n(x1,y1),….(x1,y1) o , xRn, yR, (3)

With the linear function,

F(x)= (⎾,x)+b (4)

The minimum of the Optimal Regression Function is given as,

⎾(⎾,⌊)= ||⎾||²+C∑(⌊⁻ᵢ , ⌊⁺ᵢ) (5)

Where C is pre-specified value, and ⌊- ,⌊+ are slack variables. These slack variables represents upper and lower constraints on the outputs of the system.

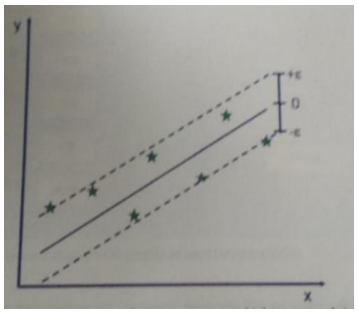

Fig. 1. Displays the epsilon parameters which are used in support vector regression while using Epsilon-sensitive loss function

The quality of estimation is measured by the loss function L(y). SVM regression uses a new type of loss function called ∑-insensitive loss function proposed by Vapnik:

0 for |f(x)y|<∑
L(y) = |f(x)-y|-∑ otherwise

SVM regression performs linear regression in the high-dimension feature space using ∑- insensitive loss and, at the same time, tries to reduce model complexity by minimizing ||⎾||².This can be described by introducing(non-negative) slack variables ⌊ᵢ,⌊*ᵢ i-1….n to measure the deviation of training samples outside ∑- insensitive zone. Thus SVM regression is formulated as minimization of the following functional:

Min ||⎾||²+C∑(⌊⁻ᵢ , ⌊⁺ᵢ)
Yᵢ – f(xᵢ, ⎾)<∑ +,⌊*ᵢ
f(xᵢ, ⎾)- Yᵢ <∑ +,⌊ᵢ
⌊ᵢ,⌊*ᵢ > 0, i = 1…..n (6)

C. Approach

In this project the stock price is obtained from NSE. Dataset obtained from, NSE preprocessed to make it suitable for the training model. The parameters taken for consideration are Date, High, Low, Open, Close, Adj close over the long period turnover. Based on this parameter the variance was calculated. The idea was that the stable companies will have a small variance as their prices do not fluctuate as much. Finally, the future prices of the stock are estimated using SVM regression algorithm. SVM regression is trained on the historical dataset of the companies which has input parameters as opening and closing prices, Lowest and Highest prices over the days and volume sold in those days. Dataset used had historic trends and patterns over the past years.

Table 1
Describes the dataset used for the implementation purpose. The dataset consists of High, Low, Open, Close, Adj Close and volume of the stocks traded

	A	B	C	D	E	F	G
1	Date	High	Low	Open	Close	Volume	Adj Close
2	########	24.95	24.48	24.94	24.53	17714100	21.98225
3	1/4/2010	25.19	24.66	24.66	24.85	26795000	22.26902
4	1/5/2010	24.85	24.35	24.72	24.82	28669900	22.24213
5	1/6/2010	24.92	24.38	24.77	24.46	24560700	21.91952
6	1/7/2010	24.61	24.08	24.46	24.38	30469700	21.84783
7	1/8/2010	24.75	24.25	24.28	24.68	23542400	22.11667
8	#########	24.8	24.37	24.69	24.69	19002400	22.12564
9	#########	24.61	24.31	24.56	24.56	26204600	22.00913
10	#########	24.99	24.55	24.63	24.8	24912300	22.22421
11	#########	25.58	25.05	25.13	25.34	44685200	22.754
12	#########	25.64	24.99	25.47	25.24	42296200	22.6642
13	#########	25.5	25.12	25.32	25.33	22732200	22.74502
14	#########	25.13	24.51	25.05	25.06	39971500	22.50257

User first selects the company from the list of the companies listed. Once the user selects the companies the results are displayed. The results of open close parameters are displayed for the company selected. Then the GUI displays the predicted open price of the next few days. User has option to view the

graph of the selected company.

The fig. 2, shows the graphical comparison between actual open price and predicted price. So above we have seen the SVM forecasting and next we will use RNN for prediction.

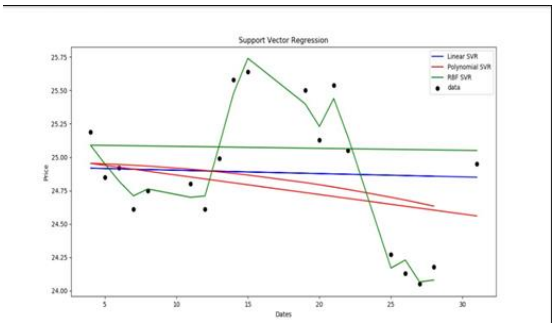

Fig. 2. Displays comparison between actual and predicted open price

D. Recurrent Neural Network(RNN)

As the financial indicators are changing very frequently fluctuating and stock market is highly violent. As there is the advancement in the technology there is the need to gain the steady fortune is needed and better prediction methods should be implied. So for steady prediction the Recurrent neural network has been used. RNN has been proved to best efficient method for processing of sequential data. Long Short term memory is successful architecture of RNN. LSTM introduces the memory cell a unit of computation that replaces traditional artificial neurons in the hidden layer of the network. With this memory cells, network is able to effectively associate memories and input remote in time, hence suit to grasp the structure of data dynamically over time with high prediction capacity.

Step 1. Raw Data

In this step first the training data set is imported from the csv file which is present in the desktop and this dataset is used for the stock prices prediction.

```
# Importing Training Set
ds = pd.read_csv('TICKER.csv')
dataset = ds.iloc[:, [2,5]].values
X = ds.iloc[:, 2].values
y = ds.iloc[:, 5].values
```

Step 2. Data preprocessing the preprocessing involves following steps a)Data discretization: It is important for numerical data it is nothing but the process of data reduction b)Data Transformation: Normalizing the data c) Data cleaning: if any missing values are present will be filled here d)Data Integration : after all the steps finally integration is done.

```
# Feature Scaling
scaler  = MinMaxScaler(feature_range=(0, 1))
dataset_scaled = scaler.fit_transform(dataset)
X = dataset_scaled[:, 0]
y = dataset_scaled[:, 1]
# Splitting the dataset into the Training set and Test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

```
# Sizes of dataset, train_ds, test_ds
dataset_sz = X.shape[0]
train_sz = X_train.shape[0]
test_sz = X_test.shape[0]
```

```
# reshape our data into 3 dimensions, [batch_size, timesteps, input_dim]
X_train = np.reshape(X_train, (train_sz, 1, 1))
y_train = np.reshape(y_train, (train_sz, 1))
```

Step3. Feature extraction: Feature extraction is nothing but selection the data which are actually required they are open , close high, low and volume.

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
from keras.models import load_model
```

Step 4. Training neural Network: Now in this stage we will train our data by submitting our data to neural network with assigning random biases and weights .There is a sequential input layer next 3 LSTM Layers and dense layer. The dense layer also has an activation function and last linear activation function in output layer.

```
# Initializing the RNN
regressor = Sequential()
```

```
# Adding fist LSTM layer and Drop out Regularization
regressor.add(LSTM(units=500,      return_sequences=True, input_shape = (X_train.shape[1], 1)))
regressor.add(Dropout(.2))
```

```
# Adding 2nd layer with some drop out regularization
regressor.add(LSTM(units=500, return_sequences=True))
regressor.add(Dropout(.2))
```

```
# Adding 3rd layer with some drop out regularization
regressor.add(LSTM(units=500, return_sequences=True))
regressor.add(Dropout(.2))
```

```
# Adding 4th layer with some drop out regularization
regressor.add(LSTM(units=500, return_sequences=False))
regressor.add(Dropout(.2))
```

Regularization

Regularizing is making sure that weights do not get too large and it do not just focus only one point, hence overfitting. So we should always have a large penalty for large weights (Now the large will totally depend on the regularizer used).

Dropouts

If some of the neurons suddenly stops working then

overfitting occurs To prevent this overfitting new method called Dropouts have been used. This forces the model to not be over dependent on specific group of neurons and consider all of them. So to predict the trend without focusing on any one neuron the robust Dropout method is used.

Step5. Prediction: The prediction of the value is calculated and the structure is rebuild based on the predicted result. Then the error is calculated using Mean squared error based on the actual and the predicted result.

```
real_stock_price = np.array(X_test)
inputs = real_stock_price
inputs = np.reshape(inputs, (test_sz, 1, 1))
predicted_stock_price = regressor.predict(inputs)

# rebuild the Structure
dataset_test_total = pd.DataFrame()
dataset_test_total['real'] = real_stock_price
dataset_test_total['predicted'] = predicted_stock_price

# real data price VS. predicted price
predicted_stock_price
= scaler.inverse_transform(dataset_test_total)

# count of Wrong predicted value after applying treshold
err_cnt=error_count(predicted_stock_price[:,0],
predicted_stock_price[:, 1], toler_treshold = 5.0)

# Calc difference between real data price and predicted price
diff_rate=calc_diff(predicted_stock_price[:,0],
predicted_stock_price[:, 1])

# MSE
mse=mean_squared_error(predicted_stock_price[:,0],
predicted_stock_price[:, 1])
```

Step 6. Visualization: For the easy and better visualization the graph for the user has been plotted between the actual and predicted result. As it will be easy for the user to get knowledge from it.

```
############ Visualizing the results ############
inputs = np.array(X)
inputs = np.reshape(inputs, (dataset_sz, 1, 1))

all_real_price = np.array(y)
all_predicted_price = regressor.predict(inputs)

# rebuild the Structure
dataset_pred_real = pd.DataFrame()
dataset_pred_real['real'] = all_real_price
dataset_pred_real['predicted'] = all_predicted_price

# real test data price VS. predicted price
all_prices = scaler.inverse_transform(dataset_pred_real)

## Visualising the results
plot(predicted=all_prices[:, 0])
plot(real=all_prices[:, 1])
plot(predicted=all_prices[:, 0], real=all_prices[:, 1])
```

Graph showing the actual and the predicted cost of the google stock:

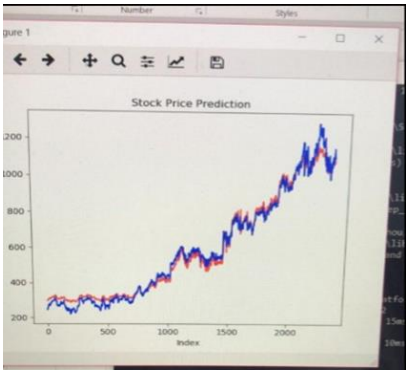

Fig. 3. Displays comparison between actual and predicted open price of google stocks

E. Accuracy

Now we have implemented both SVM and RNN and made prediction on the training data. But the key factor on selection between the two algorithms is Accuracy. The one with the higher accuracy can be used to precisely predicting the stock prices of future event. Below picture show the accuracy of the algorithm:

The below pictures shows the SVM accuracy of reliance and TCS company's prediction:

1) Reliance Company

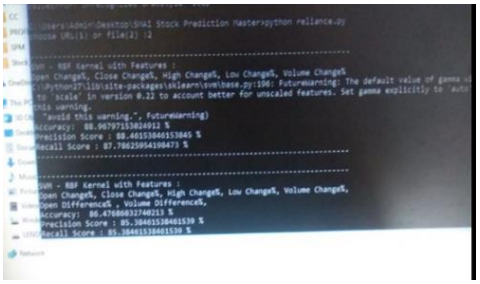

Fig. 4. Displays accuracy and prediction of reliance stock company

2) TCS company

Fig. 5. Displays accuracy and prediction of TCS stock company

3. Conclusion

The popularity of stock market is growing rapidly, which is encouraging researchers to find out new methods and techniques. The forecasting is not only giving benefit to researchers but also to the investors who are dealing with the stocks. In this Paper, we study the use of Support Vector Machine and Recurrent Neural Network for Stock market analysis. SVM is an appropriate and promising type of tool for financial forecasting. We Propose a combining model by integrating SVM classification and regression for complete analysis. But the accuracy of the prediction of RNN goes far above the SVM technologies. So RNN and LSTM provides a good knowledge of the future situation of the stock market. All these techniques were compared to find the best predicting model. The results showed that MLP performed best and predicted the market with accuracy of 77%. The results suggested that behavior of market can be predicted using machine learning techniques [10].

4. Future scope

A. Feature selection

Feature selection often affects the complete functioning of a project on a large scale by either expanding it to greater extent or bypassing some of the features. For enhancing the project we can add specified features for future work. Here we have looked at open and close price of the stocks. Future work would involve features related to the specific company parameters such as its Price/Earnings ratio, its market status, working capital ratio, cost-of-capital etc. Features related to broader factors include mutual funds, bonding, interest rates and other social and economic factors.

B. Adding granularity

One of the main drawbacks of our study is that we are focusing on the price trend of the stocks and are not considering the factors which affect the stock price such as global and political issues which have a great impact on stock markets. So for avoiding this drawback text feed analysis can be implemented in near future which will not only consider the stock prices but also economic factors affecting.

References

[1] Zhen Hu; Jie Zhu; Ken Tse “Stock market prediction using Support Vector Machine” 6th International Conference on Information Management, Innovation Management and Industrial Engineering, 2013.
[2] K. Nirmala Devi, V. Murali Bhaskaran, G. Prem Kumar, “Cuckoo Optimized SVM for Stock Marlket Prediction,” IEEE Sponsored 2nd International Conference on Innovations in Information, Embedded and Communication systems (ICJJECS), 2015.
[3] Dingxian Wang, Xiao Liu, Mengdi Wang, “A DT-SVM Strategy for Stock Futures Prediction with Big Data,” 2013 IEEE 16th International Conference on Computational Science and Engineering.
[4]
[5] Mehak Usmani; Syed Hasan Adil “Stock market prediction using machine learning techniques,” 3rd International Conference on Computer and Information Sciences (ICCOINS), 2016.
[6] Yunyun Zhang; Wei Shen “Stock Yield Forecast Based on LS-SVM in Bayesian Inference,’ ETP International Conference on Future Computer and Communication, 2009.
[7] W. A. Gruver, "Algorithmic trading systems," *2015 IEEE 13th International Symposium on Intelligent Systems and Informatics (SISY)*, Subotica, 2015, pp. 17-17.
[8] Roger Achkar; Fady Elias-Sleiman, “Comparison of BPA-MLP and LSTM-RNN for Stocks Prediction” 2018 6th International Symposium on Computational and Business Intelligence (ISCBI), 2018.
[9] E. W. Saad, D. V. Prokhorov and D. C. Wunsch, "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks," in *IEEE Transactions on Neural Networks*, vol. 9, no. 6, pp. 1456-1470, Nov. 1998.
[10] P. Tino, C. Schittenkopf and G. Dorffner, "Financial volatility trading using recurrent neural networks," in *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 865-874, July 2001.
[11] Kamran Raza, “Prediction of Stock Market performance by using machine learning techniques,” International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT), 2017.