Using Machine Learning Classifiers to Predict Stock Exchange Index

Mustansar Ali Ghazanfar, Saad Ali Alahmari, Yasmeen Fahad Aldhafiri, Anam Mustaqueem, Muazzam Maqsood, and Muhammad Awais Azam

Abstract—Predicting stock exchange index is an attractive research topic in the field of machine learning. Numerous studies have been conducted using various techniques to predict stock market volume. This paper presents first detailed study on data of Karachi Stock Exchange (KSE) and Saudi Stock Exchange (SSE) to predict the stock market volume of ten different companies. In this study, we have applied and compared salient machine learning algorithms to predict stock exchange volume. The performance of these algorithms have been compared using accuracy metrics on the dataset, collected over the period of six months, by crawling the KSE and SSE website.

Index Terms—Stock exchange prediction, machine learning, SVM, neural networks, Bayesian network, Ada-boost.

I. INTRODUCTION

Forecasting of stock market has been a vital topic in different fields of computational sciences because of its possible monetary profit. Stock market is a place where high capital is invested and companies do trading of their shares. Stock market forecasting poses the challenge of disproving the Efficient Market Hypothesis, which states that the market is efficient and cannot be predicted. Researchers have worked hard to prove the fact that financial markets are predictable. With the advancement and availability of technology, stock markets are now more accessible to investors. Various models have been proposed, both in industry and academia, for stock market prediction ranging from machine learning, to data mining, to statistical models.

In this paper, we have predicted the stock exchange volume of Karachi stock exchange (www.kse.com.pk) by crawling the real time data of ten different companies (of different sectors) by using salient machine learning classifiers like SVM, KNN, Ada-boost, Naïve Bayes, Bayesian Networks, Multilayer Perceptron and RBF. The performance of these classifiers has been compared using different matrices that include mean absolute error, root mean square error and accuracy.

II. LITERATURE REVIEW

Stock rate is predicted by R.K. and Pawar D. D. in [1], with the help neural networks, which can extract useful information from a larger set of data. Halbert reported some results of research using neural network techniques to model the asset price movements [2]. A step by step model based on artificial neural network for classification and prediction is proposed by Jing Tao Yao and Chew Lim in [3].


Abdulsalam sulaiman et al. in [7] extracted values of variables from database by using the moving average [MA] method. These values are used to predict the future values of other variables through the use of time series data. Kuang Yu Huang and Chuen-Juan Jane in [8] proposed a hybrid model to predict stock market that use autoregressive exogenous (ARX) prediction model, grey system and rough set theory to predict stock market. Md. Rafiul Hassan et al. in [9] predicted behavior of financial market by using a mixed model of Hidden Markov Model, Artificial Neural Network and Genetic Algorithms.


Johan Bollen et al. in [12] used public sentiment to improve the prediction accuracy of prediction algorithms by analyzing twitter posts with tools like GPOMS and Opinion finder. Shunrong Shen and Tongda Zhang in [13] proposed a new prediction algorithm that used the notion of temporal correlation among global markets and various important products to predict trend of next day stock. SVM was used as a classifier in this study. Osman Hegazy et al. in [14] [15] proposed a model to predict the stock market prices by using Particle Swarm Optimization (PSO) and Least Square Support Vector Machine.

III. CLASSIFICATION APPROACHES EMPLOYED FOR PREDICTING KSE AND SAUDI STOCK DATA

In this section, we briefly describe various machine-learning algorithms used for forecasting. For their detailed implementation (and parameter setup), refer to our previous work [16].
A. Support Vector Machine (SVM)

SVM find the optimal separating hyper-plane between two classes by solving the linearly constrained quadratic optimization problem and the solution is relatively globally optimal. Researchers have claimed that SVM offer superior performance than other approaches [17]. As the varying nature of stock market data declares it dynamically as a non-linear data; therefore, it is assumed that optimal performance will be achieved by using a non-linear kernel. We have applied polynomial kernel and used 1v1 (1 verses 1) approach for multiclass classification.

B. Naïve Bayes Classifier

Naïve Bayes classifier is a statistical classifiers, which predict class membership based on probabilities. Naïve Bayes classifiers make use of class conditional independence, which makes it computationally faster. Class conditional independence means every attribute in the given class is independent of other attributes. Naïve Bayes makes class conditional independence assumption. Mathematically:

\[
P(Y|X) = \prod_{i=1}^{n} P(Y_i|X_i).\]

where \(Y_i\) represents attribute \(A_k\) for the given sample.

C. K-Nearest Neighbors

K-Nearest Neighbor (KNN) is a simple to implement machine learning classifier. Classification using similarity approach can map the problem of stock prediction. The training stock data and test data is stored into a set of vectors. Each stock feature is represented by an N dimension vector. The KNN classifier works as follows:

1) Compute k number of nearest neighbors.
2) Determine the distance between the test samples and the training samples by using metric such as Euclidean distance.
3) Perform sorting on all the training data is on the basis of distances
4) Decide class labels of k nearest neighbors on the basis of majority vote and assign it as a prediction value of the query record.

D. Ada-Boost

The ada boost algorithm works as follows:

1) Start with weights \(w_i = 1/n, i = 1, \ldots, n\)
2) For \(m = 1, \ldots, M\) do:

a) Fit the machine learning algorithm \(\phi_m(\cdot)\) using weights \(w_i\) on the training data.
b) Compute \(\text{error}_m = E_m[1_{y \neq \phi_m(\cdot)}]\) and \(\epsilon_m = \log ((1/\text{error}_m)/\text{error}_m)\)
c) Update \(w_i \leftarrow w_i \exp(\epsilon_m \cdot I(y \neq \phi_m(\cdot)))\), \(i = 1, \ldots, n\) and renormalize so that \(\sum w_i = 1\).

3) Output \(\phi(\cdot) = \text{sign} [\sum m \epsilon_m \phi_m(\cdot)]\).

E. Multilayer Perceptron

The approach used to train the multilayer perceptron is explained in below:

The open, high, low, current and change in volume are inputs to the network. The number of inputs to the network is 4. The output is the prediction of volume of each company. An architecture of Multi-Layer Perceptron is shown in Fig. 1.

The inputs are multiplied by their weights and summed. By using transfer function these input functions are mapped to output. Mostly, Hyperbolic and Sigmoid function are used as transfer function. In Multi-layered Perceptron, the weighted input \(X_j\) is computed as follows:

\[
X_j = \sum_{i} W_{ij} z_i,
\]

where, \(W_{ij}\) denotes the weight between the \(i\)th and \(j\)th unit and \(z_i\) denotes the activity level of the \(j\)th unit in the previous layer. Typically, the activity \(z_j\) is calculated using the sigmoid function as follows:

\[
z_j = 1/(1+e^{-x_j})
\]

Mostly, Back-Propagation algorithm is used that reduce the errors to adjust the weights. The error \(E\) is computed as:

\[
E = 1/2 \sum_{i} (z_i - d_i)^2
\]

where \(z_j\) represents activity level of the \(j\)th unit in the top layer and \(d_j\) represents the desired output of the \(j\)th unit respectively.

F. Bayesian Network

It is graphical model based on probability. Direct acyclic graph is used in this model to represent a set of random variables and their dependencies.

Let us suppose that there are two events S and R which can cause another event G. also S is directly affected by R. A Bayesian network can model such situation easily as can be seen in Figure 2. All three events are represented by these three variables. These variables can be either true or false.
The joint probability function is:

$$P(G, S, R) = P(G|S, R) P(S|R) P(R)$$

The model can tell the probability of $R$, given the $G$ by using the conditional probability formula:

$$P(R=T|G=T) = \frac{\sum SE(T,F)}{P(G=T)}$$

$$P(R=T|G=F) = \frac{\sum SE(T,F)}{P(G=F)}$$

G. Radial basis Function (RBF)

RBF is a feed forward neural network. Its structure is based on three layers, namely; input, hidden, and output layer. The input layer comprises of units of signal source. The problem requirements determines the number of units associated to hidden layer. The hidden layer retorts to the input model and yields the corresponding output. The hidden layer units’ activation function is RBF that can be demonstrated by Fig. 3.

In Fig. 3, $X = (x_1, x_2... x_m)$ denotes a m-dimensional input vector whereas and $W = (w_1, w_2, \ldots, w_n)$ denotes the weight of output layer. The Activation function is represented as $g_i(X)$, which is a Gaussian function. Where, $i = 1, 2, \ldots, n$. $n$ shows the number of neurons in hidden layer. In RBF, the output shown by $i$th neuron of hidden layer is expressed as:

$$q_i = g_i(||X-C_i||) = \exp(-||X-C_i||^2/2\sigma^2)$$

where $C_i$ is the center of $i$th activation function, and $||\cdot||$ is Euclidian norm. The term denoted by $o_i$ is the width of the respective field. The linear combination of units on the hidden layer defines the activation of the output layer, as shown in Equation 9:

$$y = \sum_{i=1}^{n} w_i g_i$$

where $w_i$ denotes the weights from hidden to output layer.

Fig. 3. Structure of RBF.

IV. EXPERIMENTAL SETUP

We conducted experiments on historical dataset, collected over the period of 6 months (Apr 2013 to Sep 2013) for ten different well-known companies of Pakistan, by crawling the Karachi stock exchange website (www.kse.com.pk). We have also tested our algorithms on Suadi Stock Dataset. The selected data has in total 5 input features including open, High, Low, Current and Change. As the aim is to predict the target volume values concerning upcoming n-days prices; therefore, the selected input data contain information of stock market data for historical period of 6 months.

We have labeled the data according to the variation in value in “change” feature. As, we are considering data for 6 months so very small fluctuations (which might be considered as noise instead of real signal) in these values can be ignored. Hence, we have assigned three different labels—A, B and C based on the “change” value. As we have dataset for a period of 6 months (which is quite a massive data), hence we have taken samples for all 10 companies after every 4 hours because prediction will not be affected by small changes, occur in the span of 4 hours. After taking samples after every 4 hours we have 24, 000 samples on average for all companies.

For labeling the data, a threshold has been set on the input feature “change”. The column of “change” ranges from -3 to 3 in values showing small fluctuations in data. Therefore, the threshold has been set to separate positive values from those of negative values. The values exactly matching to “0” are given class label “C”; values below “0” are assigned class “B” while those of above “0” are assigned class label “A”. The possible selected features and their labels are shown in Table I.

<table>
<thead>
<tr>
<th>Input Features Along- With Assigned Label</th>
<th>Open</th>
<th>High</th>
<th>Low</th>
<th>Current</th>
<th>Change</th>
<th>Volume</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>51.55</td>
<td>52.25</td>
<td>51.6</td>
<td>51.7</td>
<td>0.15</td>
<td>51652</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>53.99</td>
<td>52.2</td>
<td>52.46</td>
<td>0.46</td>
<td>329560</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>37.6</td>
<td>38.98</td>
<td>36.91</td>
<td>37.6</td>
<td>0</td>
<td>2</td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>

We sorted the dataset based on the time feature, where historical data is used for training and the most recent data is used for testing. Specifically, we divided the sorted dataset into 80% training set and 20% testing sets. We conducted 2-fold cross validation over 80% training set for learning the optimal parameters.

Three different indices are used as measures of prediction accuracy, which are Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Accuracy. They have been used in [16], [18], [19]. Mathematically, they are defined as:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |d_i - z_i|$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (d_i - z_i)^2}$$

$$Accuracy = \frac{N_c}{N}$$

where $N$ denotes the total number of samples forecasted, $di$ denotes the actual value of a sample, $zi$ denotes the forecasting value of a sample, and $N_c$ denotes the total number of correctly classified samples.
V. RESULTS AND DISCUSSIONS

The experimental results have been shown in Figure 4. We have repeated the experiment for 11 different companies including Saudi company by applying various algorithms and have calculated MAE (and RMSE) against each algorithm. The MAE is plotted against horizontal axis while the classifiers are shown on the vertical axis. The lowest MAE in each figure is representing the indication of the best prediction algorithm for that specific company. Figure 4(a,b,c) depicts the RMSA, MAEs and Accuracies for best three classifiers (Adaboost, Multilayer Perceptron and Bayes Net) out of ten classifiers we have tested for this research. From the figures it is obvious that these three classifiers Multilayer Perceptron, Adaboost and Bayes Net shows very good results and gives different results for different datasets of companies.

It is very difficult to predict the best classifiers out of these three classifiers but overall ADABOOST gives better results for both KSA as compared to other two classifiers. For both the datasets ADABOOST gives better accuracies and lower MAEs. The detailed results for these three classifiers are given in Appendix A.

VI. CONCLUSION AND FUTURE WORK

Stock market prediction is an area of potential research and of monetary benefit as its total market capitalization is massive. This paper attempts to become a fore bringer in prediction of Karachi Stock Exchange (KSE) and we have also tested our algorithm on Saudi Stock dataset for TASI company. In this study we crawled data of ten renowned companies taken from KSE over the period of six months. Different machine learning classifiers have been employed to predict the future volume of these companies. Ada-boost, Multilayer Perceptron and Bayesian Network have shown good results.
As a future work, we focus on using social media analysis, for instance, using Tweets’ sentiments for specific stock in addition to historical data for stocks' trend prediction. We reckon the resultant hybrid framework of these two approaches will further improve the results. A cross platform mobile application of this work is also in progress. We can also use more Saudi stock data to develop a model, which can run on both Pakistani and Saudi stock datasets.

### REFERENCES


### TABLE A1: A COMPARISON OF THE BEST THREE PERFORMING CLASSIFIERS IN TERMS OF ERROR AND ACCURACY FOR DIFFERENT SECTORS

<table>
<thead>
<tr>
<th>ADABOOST</th>
<th>MAE</th>
<th>RMSE</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electricity</td>
<td>0.5776</td>
<td>0.798</td>
<td>16.67</td>
</tr>
<tr>
<td>Industrial Engineering</td>
<td>0.4964</td>
<td>0.6657</td>
<td>37.25</td>
</tr>
<tr>
<td>Automobile and Parts</td>
<td>0.2902</td>
<td>0.5302</td>
<td>55.13</td>
</tr>
<tr>
<td>Fixed Line Telecommunication</td>
<td>0.0151</td>
<td>0.0211</td>
<td>100</td>
</tr>
<tr>
<td>Chemicals</td>
<td>0.1357</td>
<td>0.6379</td>
<td>78.88</td>
</tr>
<tr>
<td>Financial Services</td>
<td>0.1127</td>
<td>0.2821</td>
<td>89.58</td>
</tr>
<tr>
<td>Oil and Gas</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Banks</td>
<td>0.0088</td>
<td>0.939</td>
<td>98.67</td>
</tr>
<tr>
<td>Pharma and Bio Tech</td>
<td>0.5951</td>
<td>0.7487</td>
<td>21.11</td>
</tr>
<tr>
<td>General Industrials</td>
<td>0.6422</td>
<td>0.7916</td>
<td>16.66</td>
</tr>
<tr>
<td>TASI</td>
<td>0.3545</td>
<td>0.2545</td>
<td>54</td>
</tr>
</tbody>
</table>

**Multi-Layer Perceptron**

| Electricity    | 0.4988| 0.6021| 16.67        |
| Industrial Engineering | 0.4345| 0.6419| 37.25        |
| Automobile and Parts | 0.2162| 0.4259| 59.23        |
| Fixed Line Telecommunication | 0.3116| 0.4114| 58.25        |
| Chemicals      | 0.1142| 0.22  | 86.96        |
| Financial Services | 0.2538| 0.4895| 62.81        |
| Oil and Gas    | 0.0904| 0.2402| 91.48        |
| Banks          | 0.0762| 0.1913| 89.4         |
| Pharma and Bio Tech | 0.5756| 0.7219| 21.11        |
| General Industrials | 0.6638| 0.7994| 0            |
| TASI           | 0.2845| 0.6545| 75.46        |

**Bayesian Network**

| Electricity    | 0.5731| 0.7353| 16.67        |
| Industrial Engineering | 0.5858| 0.7011| 0            |
| Automobile and Parts | 0.5301| 0.6722| 16.67        |
| Fixed Line Telecommunication | 0.0814| 0.2553| 88.08        |
| Chemicals      | 0.0342| 0.1173| 97.52        |
| Financial Services | 0.259 | 0.4961| 61.8         |
| Oil and Gas    | 0.058 | 0.204 | 100          |
| Banks          | 0.0931| 0.2974| 86.09        |
| Pharma and Bio Tech | 0.4119| 0.6137| 42.77        |
| General Industrials | NA    | NA    | NA           |
| TASI           | 0.3545| 0.745 | 67           |


Mustansar Ali Ghazanfar holds a BSc in Software engineering (Gold Medalist) from UET-Taxila, Pakistan; MSc in software engineering and PhD in machine learning from University of Southampton UK. His area of research include recommender systems, prediction, stock market, and socio-economical and healthcare modelling.

Saad Ali Alahmari gets PhD in artificial intelligence and semantic web from University of Southampton UK. His areas of research include recommender systems, Web services, Semantic Web, Big data, and data mining.

Yasmeen Fahad Aldhafiri holds a MSc from University of Illinois at Urbana-Champaign. Her area of research include classification and prediction, stock exchange prediction and data mining.

Anam Mustaqueem is a PhD Scholar in the Department of Software Engineering at UET-Taxila. Her areas of interest are Machine learning, Medical Imaging, Software Quality Assurance, Wireless Networks and Adhoc Networks.

Awais holds a BSc in computer engineering (Gold Medalist) from UET-Taxila, Pakistan; a MSc in computer engineering and PhD in machine.

She is learning from University of Queen Mary UK. His areas of research include recommender systems, Ubiquitous computing, and Internet of things.

Muazzam Maqsood is a PhD scholar in Software Engineering Department, UET TAXILA. His area of research include speech classification and prediction.

Saad Ali Alahmari gets PhD in artificial intelligence and semantic web from University of Southampton UK. His areas of research include recommender systems, Web services, Semantic Web, Big data, and data mining.