Using Machine Learning Classifiers to Predict Stock Exchange Index

Mustansar Ali Ghazanfar, Saad Ali Alahmari, Yasmeen Fahad Aldhafiri, Anam Mustaqeem, 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].

Stock index (Taiwan) is predicted by Tiffany Hui-Kuang and Kun-Huang Huarng in [4] by using a novel fuzzy time series model. Stock market price (Nigeria) is forecasted by Akinwale et al in [5] by implementing error back propagation and regression analysis. Predictive relationship of numerous economic and financial variables is evaluated by David Enke and Suraphan Thawornwong in [6].

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-Jiuan 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.

Yi-Fan Wang et al. in [10] improved accuracy in prediction of stack indexes by using Markov chain concepts into fuzzy stochastic prediction. Hsien-Lun Wong et al. in [11] claimed that for short term period, time fuzzy time series performs better for prediction. They used fuzzy time series employing ARIMA and vector ARMA model for prediction.

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].

Manuscript received February 15, 2017, revised March 23, 2017.

Mustansar Ali Ghazanfar is with University of Engineering and Technology Taxila, Pakistan (e-mail:mustansar.ali@uettaxila.edu.pk.)

Saad Ali Alahmari is with Department of Computer Science, Shaqra University, Saudi Arabia

Yasmeen Fahad Aldhafiri is with Department of Business Administration, Jubail University College, Saudi Arabia

Anam Mustaqeem, Muazzam Maqsood, and Muhammad Awais Azam are with University of Engineering and Technology Taxila, Pakistan.

doi: 10.18178/ijmlc.2017.7.2.614

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. Naive 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. Naive Bayes classifier works as follows:

Let us suppose T represents a training set of samples. Let there are k classes, so class labels would be C_1, C_2... C_k. Each record is represented by an n-dimensional vector, $X = \{X1, X2,\ldots, Xn\}$. It represents n measured values of the n attributes $A1$, $A2$,..., An respectively. Classifier will predict the class of X based on highest a-posteriori probability. Thus we find the class that maximizes $(Ci|X)$. By Bayes Theorem, we have k:

$$P\ (Ci/\ X) = P\ (X|Ci)\ P\ (Ci)/\ P\ (X). \quad (1)$$

As $P(X)$ has same value for all classes, we can ignore it. Naïve Bayes makes class conditional independence assumption. Mathematically:

$$P\ (X|C_i) = \prod_{k=1}^{n} P(x_k|C_i). \quad (2)$$

The probabilities $(x1|C_i)$, $(x2|C_i)$,..., $(xn|C_i)$ are computed from the training set. In Equation 2, the term x_k denotes the value of 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. Decision is taken on the basis of similarity parameter such as Euclidean distance. 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 $wi = 1/n$, $i = 1, \ldots, n$
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 $err_m = E_w[I_{(y \neq \phi_m(\cdot))}]$ and $c_m = \log((1-err_m)/err_m)$
   c) Update $w_i \leftarrow w_i \exp[c_m \cdot I_{(yi \neq \phi m(\cdot))}]$, $i = 1,\ldots,n$ and renormalize so that $\sum w_i = 1$.
3) Output $\phi(\cdot) = \text{sign}[\sum_{m=1}^{M} cm\ \phi m(\cdot)]$.

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.

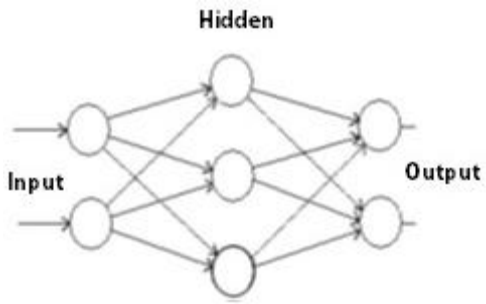

Figure 1: Structure of multilayer perceptron.

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 Xj is computed as follows:

$$Xj = \sum_i Wij\ zi, \quad (3)$$

where, Wij denotes the weight between the ith and the jth unit and zi denotes the activity level of the jth unit in the previous layer. Typically, the activity zj is calculated using the sigmoid function as follows:

$$zj = 1/1\text{-}e\text{-}xj \quad (4)$$

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

$$E = 1/2 \sum_i (zi\text{-}di)\ 2 \quad (5)$$

where zj represents activity level of the jth unit in the top layer and dj represents the desired output of the jth 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.

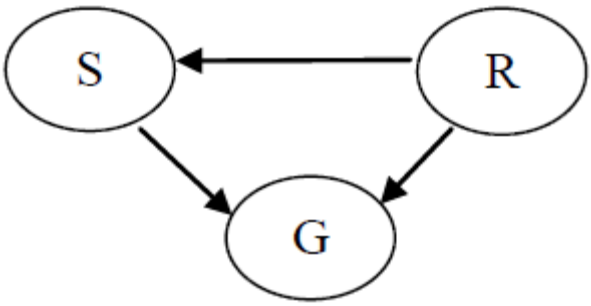

Fig. 2. Example of Bayesian network.

The joint probability function is:

$$P(G, S, R) = P(G|S, R)\, P(S|R)\, P(R) \quad (6)$$

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

$$P(R=T|\, G=T) = P(G=T,R=T) / P(G=T) = \sum SE(T,F)$$
$$P(G=T,S,R=T) / = \sum SE(T,F)\, P(G=T,S.R) \quad (7)$$

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, $X = (x1, x2... xm)$ denotes a m-dimensional input vector whereas and $W= (w1, w2, \ldots, wn)$ denotes the weight of output layer. The Activation function is represented as $gi(X)$, 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 ith neuron of hidden layer is expressed as:

$$qi = gi\,(\|X-Ci\|) = \exp(-\|X-Ci\|2/2\sigma i2). \quad (8)$$

where Ci is the center of ith activation function, and $\|*\|$ is Euclidian norm. The term denoted by σi 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:

$$y = \sum_{i=1}^{n} w_i q_i \,. \quad (9)$$

where wi denotes the weights from hidden to output layer.

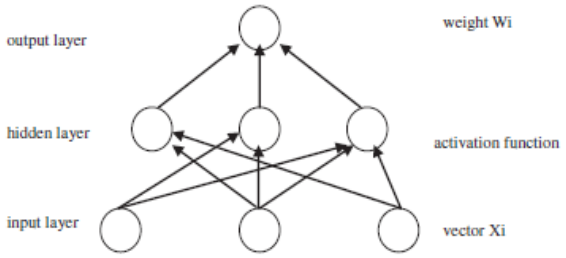

Fig. 3. Structure of RBF.

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 I: Input Features Along-Withassigned Label

Open	High	Low	Current	Change	Volume	Label
51.55	52.25	51.6	51.7	0.15	51652	A
52	53.99	52.2	52.46	0.46	329560	A
37.6	38.98	36.91	37.6	0	2	C

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:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |d_i - z_i| \quad (11)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - z_i)^2} \quad (12)$$

$$Accuracy = {}^{N_c}/_{N} \quad (13)$$

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.

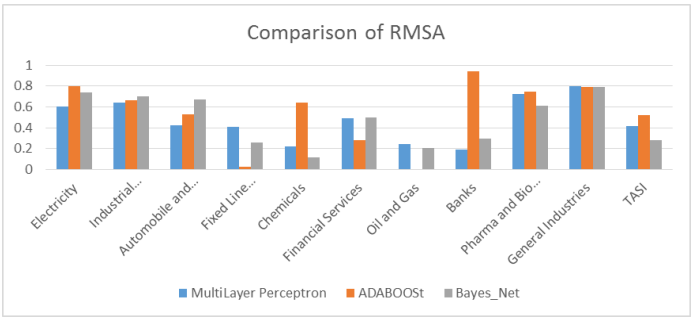

Fig. 4(A): Comparison of RMSA.

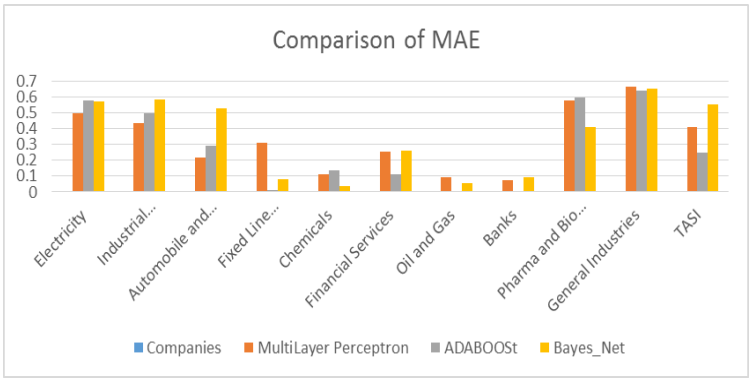

Fig. 4(B): Comparison of RMSA.

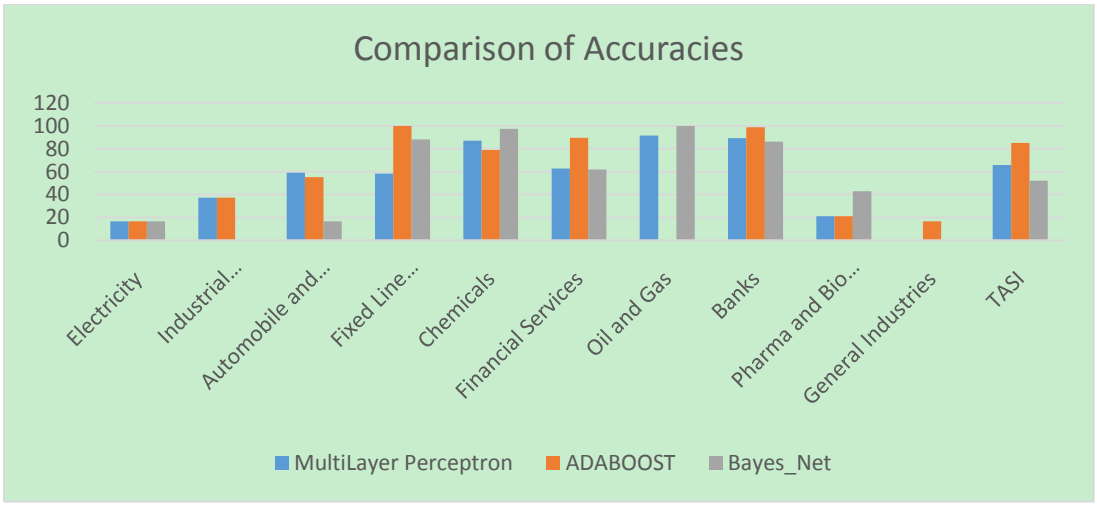

Fig. 4(C): Comparison of accuracies.

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 Ne) out of ten classifers 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.

APPENDIX A:

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

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

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

[1] R. K. Daseand D. D. Pawar, "Application of Artificial Neural Network for stock market predictions: A review of literature," *International Journal of Machine Intelligence*, vol. 2, issue 2, pp. 14-17, 2010.

[2] H. White," Economic prediction using neural networks: The case of IBM daily stock returns," Department of Economics University of California, San Diego.

[3] J. T. Yao and C. L. Tan, *Guidelines for Financial Prediction with Artificial Neural Networks*.

[4] H.-K. Y. Tiffany and K.-H. Huarng, "A neural network-based fuzzy time series model to improve forecasting," Elsevier, pp. 3366-3372, 2010.

[5] A. T. Akinwale, O. T. Arogundade, and A. F. Adekoya, "Translated Nigeria stock market price using artificial neural network for effective prediction," *Journal of Theoretical and Applied Information Technology*, 2009.

[6] D. Enke and S. Thawornwong, "The use of data mining and neural networks for forecasting stock market returns," 2005.

[7] A. S. Olaniyi *et al*., "Stock trend prediction using regression analysis – A data mining approach," *AJSS Journal*, 2010.

[8] K. Y. Huang and C.-J. Jane, "A hybrid model stock market forecasting and portfolio selection based on ARX, grey system and RS theories," *Expert Systems with Applications*, pp. 5387-5392, 2009.

[9] M. R. Hassan, B. Nath, and M. Kirley, "A fusion model of HMM, ANN and GA for stock market forecasting," *Expert Systems with Applications*, pp. 171-180, 2007.

[10] Y.-F. Wang, S. M. Cheng and M.-H. Hsu, "Incorporating the Markov chain concepts into fuzzy stochastic prediction of stock indexes," *Applied Soft Computing*, pp. 613-617, 2010.

[11] H.-L. Wong, Y.-H. Tu and C.-C. Wang, "Application of fuzzy time series models for forecasting the amount of Taiwan export," *Experts Systems with Applications*, pp. 1456-1470, 2010.

[12] B. Johan, H. Mao, and X. Jun Zeng, "Twitter mood predicts the stock market." Journal of Computational Science 2.1 (2011): 1-8.
[13] S. R. Shen, H. M. Jiang, and T. D. Zhang, *Stock Market Forecasting Using Machine Learning Algorithms*, 2012.
[14] H. Osman, O. S. Soliman, and M. A. Salam, "A machine learning model for stock market prediction," *International Journal of Computer Science and Telecommunications*, vol. 4, issue 12, December 2013.
[15] W. E. N. Fenghua *et al.*, "Stock price prediction based on SSA and SVM," *Procedia Computer Science*, vol. 31, pp. 625-631, 2014.
[16] G. M. Ali and P.-B. Adam, "The advantage of careful imputation sources in sparse data-environment of recommender systems: generating improved SVD-based recommendations," Informatica, vol. 37, no. 1, pp. 61-92, 2013.
[17] S. Ying, Z. Fengting, and Z. Tao, "China's stock index futures regression prediction research based on SVM," *China Journal of Management Science*, vol. 3, pp. 35-39, 2013.
[18] G. M. Ali and P.-B. Adam, "Exploiting context in kernel-mapping recommender system algorithms," in *Proc. Sixth International Conference on Machine Vision (ICMV 13)*, Nov. 2013, Italy.
[19] G. M. Ali, P.-B. Adam and S. Sandor, "Kernel mapping recommender systems," *Information Sciences*, vol. 208, pp. 81-104, 2012.

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 Mustaqeem 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.