HDM: A Hybrid Data Mining Technique for Stock Exchange Prediction

M. Saeedmanesh[1], T. Izadi[1], E. Ahvar[2]

***Abstract*— This paper[3] addresses the accuracy of predictions in stock exchange using data mining methods. To do this we modeled the problem by means of a time series. After this, a novel data mining technique is used to classify data. The proposed technique combines the advantages of time series analysis and data mining approaches in order to enhance the prediction accuracy. In order to evaluate the proposed technique, it is compared with the well known data mining techniques. In comparisons we used the *Dow Jones* Industrial data for all methods to have fair comparison. Results show that the proposed technique has at least 34% improvement in prediction accuracy.**

Keywords-stock exchange; data mining; prediction;

I. INTRODUCTION

The problem of stock exchange is particularly complex. The basic question is how to select from the large number of features and indicators available those that are most significant and relevant. Although many propositions of stock models exist [5, 6, 7, 8, 9], until now no efficient solution has been found. Many experts take the view that the model should be looked for in historical data, and the search process should have a continuous and incremental character depending on the efficiency of the currently used model. In practice, this means that at a given moment of the financial time series, it should be possible to design a model that provides a better forecast than any other model in terms of the evaluation function.

There have many studies using artificial neural networks (ANNs) in stock area. The early days of these studies focused on application of ANNs to stock market prediction, such as [13][14][15]. Recent research tends to hybridize several artificial intelligence (AI) techniques [16]. Some researchers tend to include novel factors in the learning process. Kohara et al. [17] incorporated prior knowledge to improve the performance of stock market prediction. Tsaih et al. [18] integrated the rule based technique and ANN to predict the direction of the S&P 500 stock index futures on daily basis. Similarly, Quah and Srinivasan [19] proposed an ANN stock selection system to select stocks that are top performers from the market and to avoid selecting under performers. They concluded that the portfolio of the proposed model outperformed the portfolios of the benchmark model in terms of compounded actual returns overtime. Kim and Han [20] proposed a genetic algorithms approach to feature decartelization and the determination of connection weights for ANN to predict the stock price index. They suggested that their approach reduced the dimensionality of the feature space and enhanced the prediction performance.

On the other hand, there has been a recent interest in using decision tree and nearest neighbor techniques to learn simple trading rules in financial markets. These techniques have proved surprisingly effective, and have proved to be an interesting starting point for further research.

This paper proposes a hybrid technique by combination of data mining prediction techniques. The hybridization causes to increase accuracy and fault tolerance. *In order to overcome the above main limitations of ANN, a* Hybrid Fault-tolerant Data Mining Technique for stock exchange prediction*, named HDM; is proposed in this study.*

The rest of the paper is organized as follows. Section II discusses our motivation. Section III presents some preliminaries about stock markets. Section IV presents description of well known mining techniques. In section V we present simulation details and Section VI presents the results. Section VII is our conclusion.

II. MOTIVATION

Although a large number of successful applications have shown that ANN can be a very useful tool for stock market modeling and forecasting [21], some of these studies, however, showed that ANN had some limitations in learning the patterns because stock market data has high volatility and noise. ANN often exhibits inconsistent results on noisy data [22]. Furthermore, ANN also suffers from difficulty in trapping into local minima, over fitting and selecting relevant input variables [23].

In order to overcome the above main limitations of ANN, a Hybrid Fault-tolerant Data Mining Technique for stock exchange prediction, named HDM; is proposed in this study.

III. PRELIMINERIES

A. Cross-Validation

Cross-Validation is a statistical method of evaluating and comparing learning algorithms by dividing data into two segments. One used to learn or train a model and the other used to validate the model. In typical cross-validation, the training and validation sets must crossover in successive rounds such that each data point has a chance of being

[1] Computer Engineering Department, Azad University, Iran, m.saeedmanesh@iau-aligudarz.ac.ir, t-izadi@iau-arak.ac.ir.
[2] Information Technology and Communication Department Payame Noor University, Iran, ehssana2000@yahoo.com, IAENG member.

[3] This work is partly supported by Iran Young Researchers Club (IYRC).

validated against. The basic form of cross-validation is k-fold cross-validation.

B. *K-fold cross-validation*

Other forms of cross-validation are special cases of k-fold cross-validation or involve repeated rounds of k-fold cross-validation. In k-fold cross-validation the data is first partitioned into k equally (or nearly equally) sized segments or folds. Subsequently, k iterations of training and validation are performed such that within each iteration a different fold of the data is held-out for validation while the remaining k-1 folds are used for learning.

C. *Stacking method*

Stacking is concerned with combining multiple classifiers generated by using different learning algorithms L1…LN on a single data set S, which consists of examples si = (xi, yi), i.e., pairs of feature vectors (xi) and their classifications (yi). In the first phase, a set of base-level classifiers C1, C2, …CN is generated, where Ci= Li(S). In the second phase, a meta-level classifier is learned that combines the outputs of the base-level classifiers. To generate a training set for learning the meta-level classifier, a leave-one-out or a cross validation procedure is applied.

D. *Voting*

In contrast to stacking, no learning takes place at the meta-level when combining classifiers by a voting scheme (such as plurality, probabilistic or weighted voting). The voting scheme remains the same for all different training sets and sets of learning algorithms (or base-level classifiers). The simplest voting scheme is the plurality vote. According to this voting scheme, each base-level classifier casts a vote for its prediction.

IV. DESCRIPTION OF TECHNIQUES

A. *Neural network*

Neural network mainly address the classification and regression tasks of data mining. Similar to decision trees, neural networks can find nonlinear relationships among input attributes and predictable attributes. Neural networks, however, find smooth rather than discontinues nonlinearities. On the negative side, it usually takes longer to learn to use a neural network than it dose to use decision trees and Naïve Bayes. Another drawback of neural networks is the difficulty in interpreting results. a neural network model contains no more than a set of weight for network. it is difficult to see the relationships in the model and why they are valid [1].

B. *Decision tree*

The decision tree is probably the most popular data mining technique. The most common data mining task for a decision tree is classification; for example, to identify the credit risk for each customer of a bank or to find those customers who are likely to be online buyers. The principle idea of a decision tree is to split your data recursively into subsets so that each subset contains more or less homogeneous states of your target variable (predictable attribute). At each split in the tree, all input attributes are evaluated for their impact on the predictable attribute. When this recursive process is complete, a decision tree is formed. The classification and regression tree (CART) proposed by professor Briemann is a popular decision tree algorithm for classification and regression [1]. We use CART algorithm for our experiments.

C. *K-nearest neighbor (KNN)*

The *K*-nearest neighbor (KNN) classifier has been both a workhorse and benchmark classifier [10, 11, 12, 25, 26]. Given a query vector *x*0 and a set of *N* labeled instances {xi, yi}$^{N}_{1}$, the task of the classifier is to predict the class label of x_0 on the predefined *P* classes. The *K*-nearest neighbor (KNN) classification algorithm tries to find the *K* nearest neighbors of x_0 and uses a majority vote to determine the class label of x_0. Without prior knowledge, the KNN classifier usually applies Euclidean distances as the distance metric. However, this simple and easy-to-implement method can still yield competitive results even compared to the most sophisticated machine learning methods.

The performance of a KNN classifier is primarily determined by the choice of *K* as well as the distance metric applied [2].

V. SIMULATION MODEL

Generally, we use TSTOOL for analysis. TSTOOL [4] is a software package for nonlinear time series analysis. It is implemented mainly in MATLAB, with some time-critical parts written in C/C++ (as MEX-functions).

A. *Time series selection*

Many researchers select their series time and data randomly. They do not have a method for selecting data. We use Hurst exponent to determine best predictable period. The Hurst exponent (H) is use to determine the underlying distribution of a particular time series. As a rule of thumb:

- 0.50 < H < 1.0 implies a persistence time series. The larger the H indicates a stronger trend. (Strong position on long)
- 0 < H < 0.5 implies ant persistence. (Trade on reversal)
- H more or less equal to 0.5 indicates random time series. (No position taken)

Hurst exponent (H) can be analyzed with Rescaled Range Analysis. The Rescaled Range analysis is a statistical methodology used to detect the presence or absence of trend in time series by finding the Hurst exponent. For example, it is generally known that time series like stock prices, indexes of stock market; sunspot etc does exhibit the persistence of trends. R/S analysis is also highly data intensive. Basically, this method is used to identify when a stock price is persistence i.e. the tendency of the price to continue its current direction and also ant persistence i.e. the tendency of the price to reverse itself rather than to continue its current direction. Or it is random and unpredictable.

Our research is based on *Dow Jones* Industrial Index and series time is 4 years. Figure 1 shows data form 1930 till 2004 and figure 2 shows Hurst exponent for this series time. Figure.2 shows that the Hurst exponent is variable form 0.42 till 0.6804. Maximum Hurst exponent is about 1973. Therefore, we select 1969-1973 as our series time.

B. *Our pattern*

We need a pattern to produce model for ANN, KNN and Decision tree techniques.

We use Chaotic Theory to determine embedded dimension (d) and time delay (~). Chaotic behavior has been reported in abroad range of scientific disciplines including astronomy, biology, chemistry, ecology, engineering, and physics.

Based on the chaotic theory, it is well known that a random-like (i.e., noisy) series or fluctuated behavior can be attributed to deterministic rules. From such random-like signals, chaotic techniques are capable of exploring the inherent non-linear dynamic and deterministic features in such signals. We use Auto Mutual Information to compute *d* and Chaos False Nearest Neighborhood methods to determine ~. Therefore, each 4 days must predict fifth day.

1) *the results*

- The time delay: 1 day
- Embedded dimension: 4
- Input Vector: x1=(x1, x2,….., xd), x2=(x2,x3,………,xd+1)
- Our pattern: (ri-3, ri-2, ri-1, ri,di+1)

VI. Simulation results

In this section we evaluate and compare the various schemes. The performance measure of interest in this study is *error rate*. We consider the following scenarios for simulations. In Scenario I, we compute error rate of each technique individually; Tables I, II and III. Then the results are compared. In Scenario II, we divide series to equal blocks. Then each block is normalized individually. Scenario III compares stacking, simple voting and weighted voting methods and scenario IV compute correlation coefficient of different techniques with each other. Finally, in scenario V, we combine different techniques and compare the results.

In all experiments presented here, classification errors are estimated using 5-fold stratified cross validation. Cross validation is repeated ten times using different random generator seeds resulting in ten different sets of folds. The same folds (random generator seeds) were used in all experiments. The classification error of a classification algorithm for a given data set is estimated by averaging over the ten runs of 5-fold cross validation. In scenario I, the neural network technique shows good performance compared with decision tree and nearest neighbor techniques.

In scenario II, however the neural network technique has good performance compared with decision tree and nearest neighbor techniques in Test data. But decision tree technique has a best error rate in prediction data compared with neural network and nearest neighbor techniques. Therefore, the single method is not sufficient for stock exchange prediction. This is our motivation to combine data mining techniques.

In scenario III, Table V shows that the simple voting scheme is better than stacking scheme. Table VI is related to scenario IV and indicates correlation coefficient. Finally, we combine different techniques. The results in Table VII show that the combination of three techniques has very good performance especially with 5-fold-average-prediction method.

VII. Conclusions

This paper addressed to improve stock exchange prediction. We studied and analyzed three well known prediction techniques. The results showed that a single technique such as neural network has some weakness. We could improve error rate by using 5-fold-average-prediction and combination of techniques.

VIII. Refrences

[1] ZhaoHui Tang, Jame Maclennan "Data mining with SQL server 2005" Wiley Publishing, Inc.2005.

[2] Latourrette M "Toward an explanatory similarity measure for nearest-neighbor classification." Proceedings of the 11th European Conference on Machine Learning, London UK ,2000, 238-245. I. S.

[3] D.J.E. Baestaens, W.M. van den Bergh, and H. Vaudrey, "Market inefficiencies, technical trading and neural networks", In C. Dunis, editor, Forecasting Financial Markets, Financial Economics and Quantitative Analysis, John Wiley & Sons, Chichester, England, pp.254-260, 1996

[4] http://www.physik3.gwdg.de/tstool/, STOOL is a software package for nonlinear time series analysis. It is implemented mainly in MATLAB, with some time-critical parts written in C/C++ (as MEX-functions).

[5] R.J. Bauer, Genetic Algorithms and Investment Strategies, Wiley, 1994.

[6] R. Caldwell, Nonlinear Financial Forecasting, Proc. of the First INFFC, Finance and Technology, 1997.

[7] J. Creedy, L.M.Vance, Nonlinear Economic Models, Edward Elgar Publ., 1997.

[8] J.O. Katz, D.L. McCormick, Neurogenetics and its Use in Trading System Development, NeuroVe$t Journ., Vol.2, No.4, pp. 5-11, 1994.

[9] A.S. Weigend, N.A. Gershenfeld , Time Series Prediction: Forecasting the Future and Understanding the Past , Addison-Wesley, 1993.

[10] Athitsos,V.,Alon,J.,Sclaroff,S."Effcient nearest neighbor classification using a cascade of approximate similarity measures." In proc of CVPR , pp. 486–493, ComputerSociety,Washington,DC,USA(2005)

[11] Athitsos,V.,Sclaroff,S."Boosting nearest neighbor classifiers for multi class recognition. "proc CVPR’05,IEEE Computer Society, Washington,DC, USA(2005)

[12] Cover,T.,Hart,P."Nearest neighbor pattern classification." IEEE Transactionson Information Theory 13(1),21–27(1967).

[13] Kamijo, K., Tanigawa, T.: Stock Price Pattern Recognition: A Recurrent Neural Network Approach. In: Proceedings of the International Joint Conference on Neural Networks, San Diego, CA (1990) 215-221

[14] Yoon, Y., Swales, G.: Predicting Stock Price Performance: A Neural Network Approach. In: Proceedings of the 24th Annual Hawaii International Conference on System Sciences.Hawaii (1991) 156-162

[15] Trippi, R.R., DeSieno, D.: Trading Equity Index Futures with a Neural Network. Journal of Portfolio Management 19 (1992) 309-317

[16] Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers, San Francisco, CA (2000)

[17] Kohara, K., Ishikawa, T., Fukuhara, Y., Nakamura.: Stock Price Prediction Using Prior Knowledge and Neural Networks. International Journal of Intelligent Systems in Accounting, Finance and Management 6 (1997) 11-22

[18] Tsaih, R., Hsu, Y., Lai, C.C.: Forecasting S&P 500 Index Futures with a Hybrid AI system. Decision Support Systems 23 (1998) 161-174

[19] Quah, T.S., Srinivasan, B.: Improving Returns on Stock Investment through Neural Network Selection. Expert Systems with Applications 17 (1999) 295-301

[20] Kim, K., Han, I.: Genetic Algorithm Approach to Feature Discretization in Artificial Neural Networks for the Prediction of Stock Price Index. Expert Systems with Applications 19 (2000) 125-132

[21] Zhang, G., Patuwo, B.E., Hu, M.Y.: Forecasting with Artificial Neural Networks: the State of the Art. International Journal of Forecasting 14 (1998): 35-62

[22] Hann, T.H., Steurer, E.: Much Ado about Nnothing? Exchange Rates Forecasting: Neural Networks vs. Linear Models Using Monthly and Weekly Data. Neurocomputing 10 (1996): 323-339

[23] Yu, X.H.: Can Backpropagation Error Surface not have Local Minima? IEEE Transactions on Neural Networks 3 (1992): 1019–1021

[24] Mehmed Kantardzic,"Data Mining (Concepts,Models,Methods and Algorithms)",University of Louisvilla,IEEE Press,ISBN 0-471-22852,2003

[25] Peng,J.,Heisterkamp,D.R.,Dai,H.K.:LDA/SVMdrivennearestneighbor classification.In CVPR'01,p. 58.IEEEC omputer Society, LosAlamitos, CA, USA (2001)

[26] Zhang,H.,Berg,A.C.,Maire,M.,Svm knn,J.M. "Discriminative nearest neighbor classification for visual category recognition" .In CVPR'06,pp.2126–2136.IEEE Computer Society, LosAlamitos, CA, USA (2006)

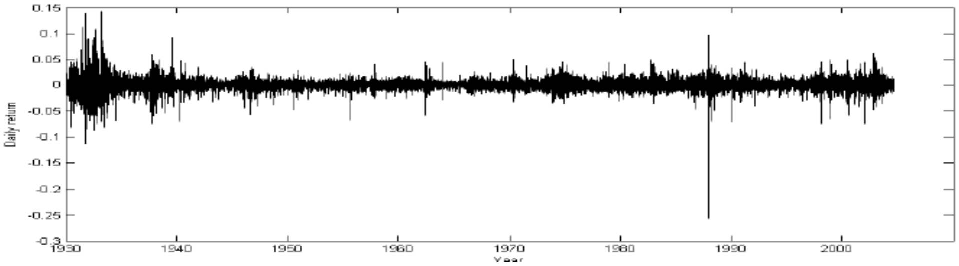

Figure 1. *Dow Jones* Industrial Index used in our simulations [3].

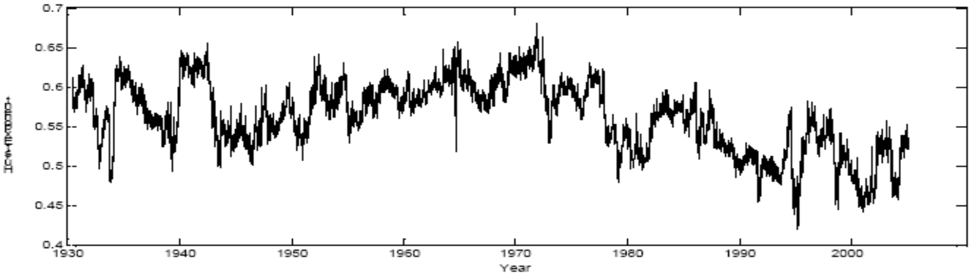

Figure 2. Hurst exponent extracted by the proposed method.

TABLE I. NURAL NETWORK ERROR RATES.

10	9	8	7	6	5	4	3	2	1	
39.49	39.62	39.00	39.88	39.26	39.48	38.64	39.36	39.39	39.25	low
41.36	42.22	41.10	41.23	41.98	40.86	41.88	40.85	42.58	41.58	high
40.29	40.78	40.22	40.73	40.74	40.32	40.02	39.91	40.62	40.51	medium

TABLE II. NERAREST NEIGHBOR ERROR RATES

10	9	8	7	6	5	4	3	2	1	
42.45	42.59	42.95	43.21	42.84	42.09	42.74	41.96	42.35	44.19	Low
45.19	44.94	45.80	44.94	46.44	44.21	45.80	45.32	45.44	46.55	high
43.87	43.48	44.34	44.32	44.73	43.36	43.99	44.20	44.26	44.98	medium

TABLE III. DECISION TREE ERROR RATES

10	9	8	7	6	5	4	3	2	1	
43.20	43.59	43.32	43.72	43.00	41.97	43.59	42.61	43.47	42.96	low
43.20	45.81	46.06	45.06	45.06	45.67	48.42	45.56	46.06	44.56	high
44.59	44.50	44.47	44.47	44.00	43.63	44.95	43.93	44.40	43.62	medium

TABLE IV. ERROR RTAES OF THE STACKING, SIMPLE VOTING AND WEIGHTED VOTING METHODS.

medium	Decision tree		Nearest neighbor		Neural network		
	Standard deviation	medium	Standard deviation	medium	Standard deviation	medium	
41.79	2.91	42.16	2.87	42.90	4.37	40.29	test(Mj)
41.45	2.47	39.21	2.38	43.76	2.51	41.39	prediction(Mj)
41.57	1.23	42.63	1.24	43.53	1.32	38.57	5-fold-test(Mj(-k))
41.46	1.43	38.68	0.69	43.81	1.16	41.87	5-fold-prediction(Mj(-k))
40.71	1.81	38.02	1.28	43.07	1.29	41.04	5-fold-average-prediction(Mj')

TABLE V. ERROR RATE OF THE PROPOSED METHOD IN DIFFERENT WORKING CONDITIONS

voting		Weighted voting		Stacking		
variance	medium	variance	medium	variance	medium	
1.62	36.12	1.90	40.79	1.56	41.49	prediction
1.82	*34.64*	1.03	39.50	1.17	40.99	5-fold-average-prediction

TABLE VI. CORRELATION COEFFICNET

5-fold-average-prediction(Mj')			prediction			
Decision tree	Nearest neighbor	Neural network	Decision tree	Nearest neighbor	Neural network	
0.4447	0.7108	0.9610	0.5007	0.7853	0.9281	Neural network
0.4903	0.9854	0.7108	0.5251	0.8352	0.7853	Nearest neighbor
0.9854	0.4903	0.4447	0.8020	0.5251	0.5007	Decision tree

TABLE VII. ERROR RATE FOR COMBINED TECHNIQUES

3 techniques	Nearest neighbor Decision tree	Decision tree Neural network	Neural network Nearest neighbor	
%36.12-1412	%37.25-1576	%37.21-1535	%39.69-1736	Prediction out
%34.64-1147	%35.83-1334	%36.88-1391	%38.39-1589	5-fold-average-prediction out