Kernel Principal Component Analysis and Support Vector Machines for Stock Price Prediction

Huseyin Ince
Faculty of Business Administration
Gebze Institute of Technology
Cayırova Fab. Yolu No:101 P.K:141 41400
Gebze,Kocaeli, TURKEY
Email: h.ince@gyte.edu.tr

Theodore B. Trafalis
School of Industrial Engineering,
University of Oklahoma
202 West Boyd, Room 124, Norman, OK 73019 USA
Email: ttrafalis@ou.edu

Abstract- **Financial time series are complex, non stationary and deterministically chaotic. Technical indicators are used with Principal Component Analysis (PCA) in order to identify the most influential inputs in the context of the forecasting model. Neural networks (NN) and support vector regression (SVR) are used with different inputs. Our assumption is that the future value of a stock price depends on the financial indicators although there is no parametric model to explain this relationship. This relationship comes from the technical analysis. Comparison shows that SVR and MLP networks require different inputs. Besides that the MLP networks outperform the SVR technique.**

Keywords: Support Vector Regression, Kernel Principal Component Analysis, Financial Time Series, Forecasting.

I. INTRODUCTION

Technical indicator analysis has attracted many researchers from different fields. For example, Ausloos and Ivanova have investigated the relationship between momentum indicators and kinetic energy theory [2]. Genetic algorithms have been used with technical indicators in order to discover some patterns [12].

One of the goals of financial methods is asset evaluation. The behavior of an asset can be analyzed by using technical tools, parametric pricing methods or combination of these methods. Prediction of financial time series is one of the most challenging applications. Since the financial market is a complex, non stationary and deterministically chaotic system, it is very difficult to forecast by using deterministic (parametric) techniques because of the assumptions behind the parametric techniques. For example, linear regression models assume normality, serial correlation etc. Therefore, nonparametric techniques such as support vector regression (SVR), neural networks (NN), and time series models are good candidates for financial time series forecasting. Over the past decades, neural networks and radial basis function networks have been used for financial applications ranging from option pricing, stock index trading to currency exchange [3, 6, 8, 11,21]. Neural networks are universal function approximators that can map any nonlinear function without any assumption about the data.

The SVR algorithm that was developed by Vapnik [5, 25] is based on statistical learning theory. In the case of regression [1,5,15], the goal is to construct a hyperplane that lies "close" to as many of the data points as possible. This determines the trend line of training data, hence the name support vector regression (SVR). Most of the points deviate with at most ε precision from the target defining an ε-band. Therefore, the objective is to choose a hyperplane where w has a small norm (providing good generalization) while simultaneously minimizing the sum of the distances from the data points, outside the ε -band, to this hyperplane [14, 18].

NN models have been applied to stock index and exchange rate forecasting [8, 11, 24]. SVR has been applied to stock price forecasting and option price prediction [21, 22]. Recent papers have showed that SVR outperforms the MLP networks [19, 22]. This can be explained with the theory of SVR. SVR has a small number of free parameters and is a convex quadratic optimization problem. We will compare the SVR with MLP networks for short term forecasting.

The tendency in the field of financial forecasting is to use state variables which are fundamental or macro economic variables. On the other hand, technical analysis, also known as charting, is used widely in real life. It is based on some technical indicators. These indicators help the investors to buy or sell the underlying stock. Our goal is to try to understand the relationship between stock price and these indicators. These indicators are computed by using stock price and volume overtime. In addition to this, more than 100 indicators have been developed to understand the market behavior. Identification of the right indicators is a challenging problem. Two different approaches will be given in this paper. The first approach is using Principal Component Analysis (PCA) to identify the most important indicators. The second approach is based on heuristic models.

Section I explains the theory of support vector regression briefly. In section II, SVR and PCA are discussed. Brief review of technical indicators is given in section III. In section IV computational experiments and comparison of the methods is discussed. Section V concludes the paper.

II. METHODOLOGY

In this part, principal component analysis and support vector regression are explained briefly.

A. Kernel Principal Component Analysis

The objective of PCA is to reduce the dimensionality of the data set while retaining as much as possible variation in the data set and to identify new meaningful underlying variables. In addition to PCA analysis, other techniques can be applied to reduce to dimension of the data. They can be classified into two categories; linear algorithms such as factor analysis, and PCA and nonlinear algorithms such as kernel principal component analysis that is based on support vector machines theory. These methods are also called feature extraction algorithms. They are widely used in signal processing, statistics, neural computing and meteorology [13, 17, 20].

The basic idea in PCA is to find the orthonormal features $s_1, s_2, \ldots, s_n$ with maximum variability. PCA can be defined in an intuitive way using a recursive formulation. Define the direction of the first principal component, say, v_1 by

$$v_1 = \arg\max_{\|v\|=1} E\{(v^T X)^2\} \quad (1)$$

where v_1 is of the same dimension m as the random data vector X. Thus the first principal component is the projection on the direction in which the variance of the projection is maximized. Having determined the first k-1 principal components, the k-th principal component is determined as the principal component of the residual:

$$v_k = \arg\max_{\|v\|=1} E\left\{\left[v^T\left(x - \sum_{i=1}^{k-1} v_i v_i^T X\right)\right]^2\right\} \quad (2)$$

The principal components are then given by $s_i = v_i^T X$. In practice, the computation of the v_i can be simply accomplished using the (sample) covariance matrix $E\{XX^T\} = C$ of the training data. The v_i are the eigenvectors of C that correspond to the n largest eigenvalues of C.

In real life, it is hard to find linear relationships. Therefore, instead of linear PCA, one can use the nonlinear PCA analysis which is called kernel principal component analysis (kPCA). kPCA is closely related to the Support Vector Machines (SVMs) method. It is useful for various applications such as pre-processing in regression and de-noising [17].

kPCA is a nonlinear feature extraction method. The data is mapped from the input space to a feature space. Then linear PCA is performed in feature space. The linear PCA can be expressed in terms of dot products in feature space [17, 20].

The nonlinear mapping $\Phi : \Re^n \to F$ can be defined as $k(x,y) = \Phi(x)\cdot\Phi(y)$, in terms of vector dot products. Given a dataset, $\{x\}_{i=1}^{l}$, in the input space we have the corresponding set of mapped dataset in the feature space, defined as $\{\Phi_i = \Phi(x_i)\}_{i=1}^{l}$. The PCA problem in feature space F can be formulated as the diagonalization of an l-sample estimate of the covariance matrix[20], defined as

$$\hat{C} = \frac{1}{l}\sum_{i=1}^{l} \Phi(x_i)\Phi(x_i)^T, \quad (3)$$

where $\Phi(x_i)$ are centered nonlinear mappings of input variables $x_i \in \Re^n$. The centralization mapping is explained in [17]. We need to solve the following eigenvalue problem

$$\lambda V = \hat{C} V, \quad V \in F, \lambda \ge 0 \quad (4)$$

Note that, all the solutions V with $\lambda \ge 0$ lie in the span of $\Phi(x_1), \Phi(x_2), \ldots, \Phi(x_l)$. Schölkopf et al. [13] derived the equivalent eigenvalue problem

$$n\lambda\alpha = K\alpha,$$

where α denotes the column vector such that $V = \sum_{i=1}^{l} \alpha_i \Phi(x_i)$, and K is a kernel matrix which satisfies the following conditions

$$\iint K(x,y) g(x) g(y)\, dx dy > 0, \quad \int g^2(x) dx < \infty \quad (5)$$

Then we can compute the k-th nonlinear principal component of x as the projection of $\Phi(x)$ onto the eigenvector V^k

$$\beta(x)_k = V^k \Phi(x) = \sum_{i=1}^{l} \alpha_i^k K(x_i, x) \quad (6)$$

Then the first $p < l$ nonlinear components are chosen, which have the desired percentage of data variance. By doing this, we reduce the size of the dataset. Then, we perform support vector regression or the multi layer perceptron algorithm on the reduced dataset.

B. Support Vector Regression (SVR)

In SVR our goal is to find a function $f(x)$ that has ε deviation from the actually obtained target y_i for all training data and at the same time is as flat as possible. Suppose $f(x)$ takes the following form,

$$f(x) = w \cdot x + b, w \in X, b \in \Re \tag{7}$$

Then we solve the following [18] :

$$\begin{aligned} &\min \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{Subject to} \\ &\mathbf{y_i} - \mathbf{wx_i} - b \le \varepsilon \\ &\mathbf{wx_i} + b - \mathbf{y_i} \le \varepsilon \end{aligned} \tag{8}$$

In the case of infeasibility we introduce slack variables ξ_i, ξ^*_i and we solve the following problem:

$$\begin{aligned} &\min \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ &\text{Subject to} \\ &\quad y_i - \mathbf{wx}_i - b \le \varepsilon + \xi_i \\ &\quad \mathbf{wx}_i + b - y_i \le \varepsilon + \xi_i^* \\ &\quad \xi_i, \xi_i^* \ge 0 \\ &\quad C \succ 0 \end{aligned} \tag{9}$$

where C determines the trade-off between the flatness of the $f(x)$ and the amount up to which deviations larger than ε are tolerated.

Next we construct the dual problem. The reason is that solving the primal problem is more difficult due to too many variables. If we use the dual problem formulation, we can decrease the number of variables and the size of the problem becomes smaller. Specifically,

$$\begin{aligned} &\max -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\lambda_i - \lambda_i^*)(\lambda_j - \lambda_j^*)\mathbf{x}_i\mathbf{x}_j \\ &-\varepsilon\sum_{i=1}^{l}(\lambda_i + \lambda_i^*) + \sum_{i=1}^{l} y_i(\lambda_i - \lambda_i^*) \\ &\text{Subject to} \quad \sum_{i=1}^{l}(\lambda_i - \lambda_i^*) = 0 \\ &\qquad \lambda_i, \lambda_i^* \in (0, C) \end{aligned} \tag{10}$$

Now we consider the non-linear case. First of all, we need to map the input space into the feature space and try to find a regression hyperplane in the feature space. We can accomplish that by using the kernel function $k(x,y)$. In other words we replace k as follows:

$$k(x, y) = \Phi(x) \cdot \Phi(y) \tag{11}$$

Therefore, we can replace the dot product of points in the feature space by using kernel functions. Then, the problem becomes

$$\begin{aligned} &\max -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\lambda_i - \lambda_i^*)(\lambda_j - \lambda_j^*)K(\mathbf{x}_i, \mathbf{x}_j) \\ &-\varepsilon\sum_{i=1}^{l}(\lambda_i + \lambda_i^*) + \sum_{i=1}^{l} y_i(\lambda_i - \lambda_i^*) \\ &\text{Subject to} \quad \sum(\lambda_i - \lambda_i^*) = 0 \\ &\qquad \lambda_i, \lambda_i^* \in (0, C) \end{aligned} \tag{12}$$

At the optimal solution, we obtain

$$\begin{aligned} &w^* = \sum_{i=1}^{l}(\lambda_i - \lambda_i^*)K(x_i, x), \quad \text{and} \\ &f(x) = \sum_{i=1}^{l}(\lambda_i - \lambda_i^*)K(x_i, x) + b \end{aligned} \tag{13}$$

According to [5,18], any symmetric positive semi-definite function, which satisfies Mercer's conditions, can be used as a kernel function in the SVRs context (see equation 5). Different functions which satisfy the above conditions can be used as kernel functions. Polynomial and RBF kernel functions are very common.

In order to solve the optimization problem given in equation (12), one needs an efficient algorithm. Recently, several decomposition techniques have been developed [4, 9, 10, 16, 25] to solve this large scale problem. Bender's decomposition technique and sequential minimum optimization technique has been developed to solve linear and quadratic programming problems [4, 22].

III. Brief Review of Technical Indicators

There are more than 100 technical indicators that have been developed to gain some insight about the market behavior. Some of them catch the fluctuations and some focus on where to buy and sell. The difficulty with technical indicators is to decide which indicators are crucial in order to determine the market movement. It is not wise to use all indicators in a model.

Another problem is that those indicators are very dependent on an asset price. For example, some indicators would provide excellent information for stock A, but they would not give any insight information for stock B. Thus, we need a tool to choose the right indicators (inputs) for each stock. This can be achieved by using data mining techniques which have been widely used for other areas.

Technical indicators can be used for short term and long term investment strategies. Usually, three or more indicators can be used together for identifying the trend of the market. In our heuristic models, we will consider several indicators as inputs in the model. Specifically the exponential moving average , volume % change, stochastic oscillator and relative strength index, bollinger bands width and Chaikin Money Flow. These indicators will be explained briefly.

Exponential Moving Average (EMA): The EMA is used to reduce the lag in simple moving average. Exponential moving averages reduce the lag by applying more weight to recent prices relative to older prices. The weighting applied to the most recent price depends on the length of the moving average [7].

Relative Strength Index (RSI): The RSI, developed by J. Welles Wilder, is an extremely useful and popular momentum oscillator [7]. The RSI compares the magnitude of a stock's recent gains to the magnitude of its recent losses and turns that information into a number that ranges from 0 to 100. It takes a single parameter, the number of time periods to use in the calculation. Like most true indicators, the RSI only needs one stock to be computed.

Bollinger Band (BB): BB is an indicator that allows users to compare volatility and relative prices levels over a period of time [7]. The indicator consists of three bands designed to encompass the majority of a security's price action.

Moving Average Convergence Divergence MACD: It is one of the simplest and most reliable indicators available [7]. MACD uses moving averages, which are lagging indicators, to include some trend-following characteristics. These lagging indicators are turned into a momentum oscillator by subtracting the longer moving average from the shorter moving average. The resulting plot forms a line that oscillates above and below zero, without any upper or lower limits. MACD is a centered oscillator and the guidelines for using centered oscillators apply.

Chaikin Money Flow (CMF): The CMF oscillator is calculated from the daily readings of the Accumulation/Distribution Line. The basic premise behind the Accumulation Distribution Line is that the degree of buying or selling pressure can be determined by the location of the close relative to the high and low for the corresponding period (Closing Location Value). There is buying pressure when a stock closes in the upper half of a period's range and there is selling pressure when a stock closes in the lower half of the period's trading range. The Closing Location Value multiplied by volume forms the Accumulation/Distribution Value for each period [7].

IV. Experiments

The combination of technical indicators and fundamentals are used for effective and efficient portfolio management. Portfolio managers use different techniques to identify which stocks have to put in their portfolio and which ones should sell. Generally, they focus on long term portfolio management. Here we will focus on short term portfolio management. Four different models have been used for stock price forecasting. Two of these models are based on kPCA and factor analysis. Our goal is to use these two techniques as an input selection algorithm for SVR and MLP networks. Others are heuristics models which are created by personal experience.

In the first stage of our experiments, our objective is to identify the important indicators that are used in SVR and MLP as inputs. PCA analysis and factor analysis are performed for reducing the size of input. These two techniques are used as a preprocessing algorithm. More than 100 technical indicators are used in this stage. Two different models have been identified, "PCA model" which has 7 inputs, and "factor model" which has 30 inputs. Since each technique is based on a different theory, the number of features/factors is also different.

In addition to these three models, two heuristic models are proposed. For the short term price movements, small numbers of indicators are used in practice. They give valuable information where the stock price will go for the next day. Experience with short term trading in the U.S. equity market suggests two simple models. In the first model which is called H1, we assume that the current stock price depends on the previous EMA, Volume, RSI, BB, MACD and CMF.

$$S_t = f(EMA_{t-1}, V_{t-1}, RSI_{t-1}, BB_{t-1}, MACD_{t-1}, CMF_{t-1})$$

The second model, called H2, is more complicated and has more independent variables. This is shown below.

$$S_t = f(EMA_{t-1}, EMA_{t-2}, V_{t-1}, V_{t-2}, RSI_{t-1}, RSI_{t-2}, RSI_{t-3}, BB_{t-1}, BB_{t-2}, MACD_{t-1}, MACD_{t-2}, CMF_{t-1}, CMF_{t-2})$$

We define the short term as 2-3 weeks. The last 3-4 years have shown that investment strategies have to change from long to short term, if the economy is in the recession. This move would save the investors from long term loses. For example, more than 90 percent of the companies traded in NASDAQ have lost at least half of their market capitalization in the last 3 years. Because of these difficulties, we focus on short term prediction. We used the daily stock prices of 10 companies traded on NASDAQ. The average training set contains 2500 observations and testing/validation set consists of 50 examples.

The relationship between stock price and these indicators are highly nonlinear and they are also correlated with each other. Instead of parametric models, non parametric data driven models, such as SVR and MLP will be applied with different settings. We have used two pre processing techniques and two heuristic model to apply SVR and MLP networks. Comparison of MLP and SVR are given in terms of the mean square error (MSE).

Two different comparisons are performed. The first comparison is to compare input selection processes. Determining the influential inputs for stock price forecasting is very crucial. In this point of view, we compare kPCA, factor analysis and two heuristic techniques for their out of sample performance. The performance is defined through the mean square error (MSE). Next, SVR and MLP networks are compared with each other. Since these two techniques have some free parameters, one has to chose the optimal or near optimal parameters. For example, for the SVR algorithms, we need to decide what kind of kernel function is good. After the selection of the kernel function, parameters of this function are also determined. In this study, we use a radial basis kernel function. In order to determine the free parameters, trade-off value and width (σ), 10 fold cross validation technique has been applied. After performing cross validation, we have decided that $C = 1000$ and $\sigma = 50$.

TABLE I.
MEAN SQUARE ERROR OF INPUT SELECTION ALGORITHMS FOR SVR. BOLD MSES ARE THE BEST VALUES FOR EACH STOCK. (KPCA: KERNEL PRINCIPAL COMPONENT ANALYSIS, FAC: FACTORIAL ANALYSIS)

	MODELS			
Stock	*kPCA*	*FAC*	*H1*	*H2*
MSFT	**0.555**	92.678	2.494	16.697
YAHOO	**5.662**	28.625	22.271	61.258
RMKR	0.019	0.196	**0.006**	0.027
INTEL	0.507	5.588	**0.429**	2.508
INCY	1.202	18.207	**0.761**	3.347
INAP	0.301	8.717	**0.126**	0.751
EBAY	**57.906**	456.941	64.772	96.638
AMZN	35.886	**27.021**	34.578	40.813
AMGN	**1.538**	6.969	1.864	3.458
AMCC	**0.364**	81.995	1.193	2.455

Table I shows the MSE for each input selection algorithms for 10 companies traded in NASDAQ stock exchange market. As it can be seen from table 1, kPCA and H1 models are superior to other two models. Table 2 shows the best models for MLP networks.

When we compare each preprocessing techniques with SVR and MLP networks, we have come out with the following interesting results. MSE errors for kPCA technique from tables 1 and 2 show that SVR method has better performance than MLP networks. This can be explained with the theory behind the kPCA and SVR method since these two methods are based on the structural risk minimization principle. On the other hand MLP networks use the empirical risk minimization principle. When we look at the performance of factorial analysis technique we see that the MLP networks outperforms the SVR method. The results of MSE for H1 and H2 model suggest that MLP networks have better performance than SVR technique.

TABLE II.
MEAN SQUARE ERROR OF INPUT SELECTION ALGORITHMS FOR MLP NETWORKS. BOLD MSE ARE THE BEST VALUES FOR EACH STOCK.

	MODELS			
Stock	kPCA	FAC	H1	H2
MSFT	2.590	0.261	0.990	**0.204**
YAHOO	8.264	1.501	1.117	**1.106**
RMKR	0.015	**0.005**	0.006	0.004
INTEL	0.688	0.465	0.434	**0.374**
INCY	0.455	0.153	0.570	**0.113**
INAP	0.887	0.135	0.037	**0.027**
EBAY	91.182	30.858	46.281	**24.666**
AMZN	26.899	3.647	3.094	**1.950**
AMGN	1.628	**1.114**	1.449	1.161
AMCC	2.434	0.968	0.424	**0.175**

In order to determine the number of neurons, some heuristic algorithms can be applied such as genetic algorithms. The number of neurons is could be selected as one half to three times the number of inputs [23].

Different network structures have been applied to these models. There are two problems with the backpropagation algorithm. BP algorithm converges to a local minimum. Secondly, the overfitting problem may be seen if the numbers of neurons are increased more than the network needs. We have performed cross validation algorithm to find the number of neurons in hidden layer. Multiple runs have been performed in order to find a better results for each networks. Table 2 shows the best results for each stock and model.

V. Conclusion

In this study, we used the kPCA analysis and factorial analysis as preprocessing or input selection techniques for SVR and MLP networks. kPCA and factor analysis are used to identify influential technical indicators. Two different models are used to explain the relationship between stock price and technical indicators. This study shows that SVR and MLP techniques are sensitive to the preprocessing techniques.

Our motivation is to provide a different approach for input selection in stock prediction. This study has showed that the right inputs vary with the preprocessing techniques and learning algorithm.

REFERENCES

[1] N. Ancona. Classification properties of support vector machines for regression. Technical report, 1999. RIIESI/CNR- Nr. 02/99.

[2] M. Ausloos, and K. Ivanova, Mechanistic approach to generalized technical analysis of share prices and stock market indices, *The European Physical Journal B (27)*: 177-187, 2002.

[3] A. Chen, M. T. Leung, and H. Daouk, Application of neural networks to an emerging market: forecasting and trading the Taiwan Stock Index, *Computers & Operation Research (30)*: 901-923, 2003.

[4] R. Collobert and S. Bengio. Svmtorch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research*, 1:143–160, 2001. 87

[5] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[6] J. Galindo. A framework for comperative analysis of statistical and machine learning methods: An application to the black scholes option pricing equations. Technical report, Banco de Mexico, Mexico, DF, 04930, 1998.

[7] A. Hill. Indicator Analysis, http://stockcharts.com/education/IndicatorAnalysis/index.html.

[8] J. M. Hutchinson, A. W. Lo, and T. Poggio. A nonparametic approach to pricing and hedging derivative securities via learning networks. *The Journal of Finance*, XLIX(3):851–889, 1994.

[9] T. Joachims. Making large-scale svm learning practical. In B. Sch"olkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, pages 169–184. MIT Press, 1999.

[10] Y.-J. Lee and O.L. Mangasarian. Rsvm: Reduced support vector machines. CD Proceedings of the First SIAM International Conference on Data Mining, 2001. 89

[11] W. Leigh, M. Paz, and R. Purvis, An analysis of a hybrid neural network and pattern recognition technique for predicting short-term increases in the NYSE, *The International Journal of Management Science (30)*: 69-76, 2002.

[12] M. Leus, D. Deugo, F. Oppacher, and R. Cattral, GA and financial analysis, in L. Monostori, J. Vancza, and M. Ali (Eds), IEA/AIE 2001, LNAI 2070, pp. 868-873,2001.

[13] B. Scholkopf, A. Smola, and K. R. Muller, Kernel Principal Component Analysis, in: B. Scholkopf, C.J.C. Burgers, A.J. Smola (Eds), Advances in Kernel Methods- Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 293-306

[14] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. *Proc. Computer Vision and Pattern Recognition'97*, pages 130–136, 1997.

[15] M. Pontil and A. Verri. Properties of support vector machines. Technical report, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1997.

[16] R. Rifkin. Svmfu a support vector machine package, 2000:http://fivepercentnation.mit.edu/PersonalPages/rif/SvmFu/index.html.

[17] R. Rosipal, M. Girolami, L.J. Trejo, A. Cichocki, Kernel PCA for Feature Extraction and De-Noising in Non-linear Regression. *Neural Computing & Applications*, 10(3): 231-243, 2001.

[18] A. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 1998. Invited paper, in press.

[19] F. E. H. Tay and L.J. Cao, Modified support vector machines in financial time series forecasting, *Neurocomputing(48)*: 847-861,2002.

[20] C.J. Twining and C.J. Taylor, The use of kernel principal component analysis to model data distribution, *Pattern Recognition* 36 (2003), 217-227.

[21] T.B. Trafalis, H. Ince, and T. Mishina, Support vector regression in option pricing, *Procedings of Conference on Computational Intelligence and Financial Engineering (CIFer 2003)*, March 20-23, 2003, Hong Kong.

[22] T.B. Trafalis and H. Ince. Benders decomposition technique for support vector regression. In *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, volume 3, pages 2767– 2772. IEEE, 2002.

[23] T.B. Trafalis, "Artificial Neural Networks Applied to Financial Forecasting", *Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, data Mining and Complex Systems*, (Dagli, Buczak, Ghosh, Embrechts and Ersoy, eds.), ASME Press, 1049-1054, 1999.

[24] R. Tsaih. Sensitivity analysis, neural networks and, the finance. pages 3830–3835, 1999.

[25] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.