GBP/USD Currency Exchange Rate Time Series Forecasting Using Regularized Least-Squares Regression Method

Hongxing LI, Zhaoben FANG, Dongming ZHAO

***Abstract* —Kernel-based Regularized Least-squares Regression (RLSR）is a technique originally from Statistical Learning (SL) theory. RLSR can deal with non-linear problem through mapping the samples into a higher dimension space using a kernel function. This paper adopts the RLSR to time series forecasting and the resulted model is termed RLS-TS model getting the idea from applying neural network and support vector regression to time series forecasting. This paper applies the RLS-TS model to GBP/USD Exchange Rate forecasting. RLS-TS performs better than random walk, linear regression, autoregression integrated moving average, and artificial neural network model in predicting GBP/USD currency exchange rates. A grid search is used to choose the optimal parameters.**

***Index Terms*—exchange rate, regularized least-squares, time series, forecasting.**

I. Introduction

Over the past two decades, exchange rates have exhibited substantial short-term volatility. The amount traded exceeds a trillion US dollars in transactions executed each day in the foreign exchange market. At present, both translation and conversion of foreign currency involve the use of exchange rates. In this increasingly challenging and competitive market, investors and traders need tools to analyze their data from the vast amounts of data available to them to help them make good decisions. This paper specifically describes a new and effective approach to forecast the currency exchange rate between U.S. and GB.

Many models have been built to predict the exchange rate. Refenes [1] developed a constructive learning algorithm to predict the exchange rate between U.S. dollar and the Deutsche mark. Kuan and Liu [2] examined performance of feed-forward and recurrent neural. Diebold and Nason [3] investigated ten weekly spot rates and did not find any significant difference in both in-sample fit and out-of-sample forecasting across these exchange rate series.

Verkooijen [4] forecasts monthly U.S. dollar/Deutsche mark exchange rate using neural networks. He finds that the neural network performance is very similar to the linear structural models in out-of-sample forecasting. Hann and Steurer [5] compare neural network models with linear monetary model in forecasting U.S. dollar/Deutsch mark exchange rate.

Sfetsos and Siriopoulos[6] compare four methods including random walk (RW), linear regression (LR), auto regression integrated moving average (ARIMA), and artificial neural network (ANN) in forecasting exchange rate between US dollar and GB pound. Our work is based on Sfetsos and Siriopoulos's research, and we adopt regularized least-squares method to time series field and build our forecasting model named RLS-TS model. The idea is from applying neural network and support vector regression to time series forecasting. The RLS-TS model performs better than other models in references in forecasting USD/GBP exchange rate.

This paper is organized as follows. In Section 2, we describe regularized least-squares time series (RLS-TS) theory. The results are illustrated in Section 3. Some discussions are presented in Section 4.

II. THEORY

Regularized least-square regression (RLSR) is based on statistical learning (SL) theory. SL theory was introduced in the late 1960's. Vapnik and Chervonenkis had done much initial and fundamental work. Until the 1990's, it was a purely theoretical analysis of the problem of function estimation from a given collection of data. In the middle of the 1990's, the statistical theory was used as a tool for creating practical algorithms for estimating multidimensional functions.

A. Learning Model

The statistical learning model can be described using three components:

(1) Random vector x is drawn independently from a fixed but unknown distribution $P(x)$.

Manuscript received March 6, 2007.

Hongxing LI was with Department of Statistics & Finance, University of Science & Technology of China. Hefei, Anhui 230026 CHINA. He is now a research associate at Electrical and Computer Engineering, University of Michigan-Dearborn, MI 48128 USA (e-mail:hongxing@umd.umich.edu).

Zhaoben Fang is with the Department of Statistics & Finance, University of Science & Technology of China, Hefei, Anhui 230026 CHINA (e-mail: zbfang@ustc.edu.cn).

Dongming Zhao is with the Department of Electrical & Computer Engineering, The University of Michigan-Dearborn, MI 48128 USA (e-mail: dmzhao@umich.edu).

(2) A supervisor that returns an output vector y for every input vector x, according to a conditional distribution function $P(y \mid x)$, is also fixed but unknown.

(3) A learning machine capable of implementing a set of learning functions $f(x,\alpha), \alpha \in \Lambda.$ where Λ is the parameter space.

The problem of learning is that of choosing from the given set of functions $f(x,\alpha), \alpha \in \Lambda$,.the one which predicts the supervisor's response in the best possible way. The selection is based on a training set of l random independent identically distributed observations according to $P(x,y) = P(x)P(y \mid x)$

$$(x_i, y_i).\ i = 1, 2, \cdots, l \qquad (1)$$

In order to choose the best available approximation to the supervisor's response, one measures the loss or discrepancy $L(y, f(x,\alpha))$ between the response y of the supervisor to a given input x and the response $f(x,\alpha)$ provided by the learning machine. Consider the expected value of the loss, given by the risk functional

$$R(\alpha) = \int L(y, f(x,\alpha))dP(x,y). \qquad (2)$$

The goal is to find the function $f(x,\alpha_0)$ which minimizes the risk functional $R(\alpha)$ (over the class of functions $f(x,\alpha), \alpha \in \Lambda.$) in the situation where the joint probability distribution $P(x,y)$ is unknown and the only available information is contained in the training set (1). This formulation of the learning problems is rather general. Considering different lost functions it could be different problems [7].

(a) Pattern recognition:

$$L(y, f(x,\alpha)) = \begin{cases} 0 & if\ \ y = f(x,\alpha) \\ 1 & if\ \ y \neq f(x,\alpha) \end{cases}$$

(b) Regress estimation:

$$L(y, f(x,\alpha)) = (y - f(x,\alpha))^2$$

(c) Density estimation

$$L(p(x,\alpha)) = -\log p(x,\alpha)$$

With, $\alpha \in \Lambda$, $p(x,\alpha)$ is density.

B. Regularized Least-Squares Regression (RLSR)

In regression, we are given a training set (1) and { y_i }'s are real-valued. The goal is to learn a function f to predict the y values associated with new observed x value. For regularized least-square regression (RLSR), we pose our regression task as finding the function f that solves a Tikhonov Regularization problem:

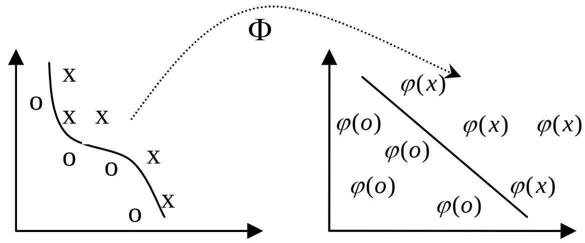

Fig.1 The function φ embeds the data into a feature space where the nonlinear pattern now appears linear. The kernel computes inner products in the feature space directly from the inputs.

$$f = \arg\min_{f \in H_K} \frac{1}{l} \sum_{i=1}^{l} L(f(x_i), y_i) + \lambda \|f\|_K^2 \qquad (3)$$

where x_i , $f(.)$ is the decision function, λ is the regularization parameter, and $\|f\|_K^2$ is the norm of the function in the Reproducing Hilbert Space [8, 9]. The choice of the lost function is the square loss:

$$L(y, f(x,\alpha)) = (y - f(x,\alpha))^2 \qquad (4)$$

RLSR solves the problem that in some cases there isn't a linear separating hyper-plane in an ingenious way: mapping the samples to a higher dimension space using a kernel function, and seeking a hyper-plane in that space (Fig.1). A nonlinear function (left) in the original space is mapped into the feature (right) where the function becomes linear. This is done by replacing x_i mapping into feature space $\varphi(x_i)$ which linearizes the relation between x_i and y_i [10, 11]

$$x = (x_1, ..., x_n) \to \Phi(x) = (\varphi_1(x), ..., \varphi_N(x)) \qquad (5)$$

There are some kernel functions that can be chosen to solve problems in different conditions [12], which linearizes the relation between x_i and y_i :

(a) Dot kernel: $k(x_i, x_j) = x_i \square x_j$.

(b) Polynomial kernel: $k(x_i, x_j) = (xi \square xj + 1)^d$

(c) Gaussian kernel: $k(x_i, x_j) = \exp(-\frac{1}{2\sigma^2}\|x_i - x_j\|^2)$

Representer Theorem: The solution to the Tikhonov regularization problem (3) can be written in the form:

$$f(x) = \sum_{i=1}^{l} c_i K(x_i, x) \qquad (6)$$

This theorem says that to solve the Tikhonov regularization problem, we need only find the best function of form (6) . Put differently, all we have to do is find the c_i .

Notation: We will use the symbol K_{ij} for the kernel function K:

$$K_{ij} \equiv K(x_i, x_j) \qquad (7)$$

Using this definition, consider the output of function (6) at the training point x_j

$$f(x_j) = \sum_{i=1}^{l} K(x_i, x_j) c_i = (Kc)_j \qquad (8)$$

With this notation, we apply the Representer Theorem to our Tikhonov minimization problem, reformulation it as: using the square loss, our problem becomes

$$f = \arg\min_{f \in H_K} \frac{1}{l}(Kc - y)^2 + \lambda \| f \|_K^2 \qquad (9)$$

For the norm of a Represented Function , recall that if we have the function (6), then we have

$$\| f \|_k^2 = c^T K c \qquad (10)$$

Substituting (10) into (9), our Tikhonov minimization problem becomes a problem of finding c (Ref. [13]):

$$f = \arg\min_{c \in R^l} \frac{1}{l}(Kc - y)^2 + \lambda c^T K c \qquad (11)$$

C. Solving the Problem

The solution to the problem developed above becomes the minimization of function

$$g(c) = \frac{1}{l}(Kc - y)^2 + \lambda c^T K c \qquad (12)$$

This is a convex, differentiable function of c , so we can minimize it simply by setting to zero of the derivative of *g(c)* with respect to c .

$$\frac{\partial g(c)}{\partial c} = \frac{2}{l} K(Kc - y) + 2\lambda K c = 0 \qquad (13)$$

Function (13) can be simplified as

$$c = (K + \lambda l I)^{-1} y \qquad (14)$$

The matrix $K + \lambda l I$ is positive definite and will be well-conditioned, if λ is not too small. The conjugate gradient algorithm is a popular algorithm for solving positive definite linear systems [14, 15].

D. Regularized Least-Squares Time Series Model Building

In this section, we adopt RLSR to time series forecasting and give the mathematics formulation of the RLS-TS model. Time series prediction can be seen as autoregression in time, for this reason, a regression method can be used for this task [16]-[19].

Given a time series $\{x_1, x_2, ..., x_n\}$, in order to make a prediction on it using regularized least-squares regression, it must be transferred into an autocorrected dataset.

That is, if $\{x_{t+1}\}$ is the value to predict, the previous values $\{x_t, x_{t-1}, ..., x_{t-p+1}\}$ should be the input variables. Then we can map the autocorrected input variables $\{x_t, x_{t-1}, ..., x_{t-p+1}\}$ to the goal variable $y_{t+1} = \{x_{t+1}\}$, p are called embedded dimension.

After transferring the data, we can get the samples which are suitable for RLSR learning, with the following matrix form:

$$X = \begin{bmatrix} x_n & x_{n+1} & \cdots & x_{n+p} \\ x_{n+1} & x_{n+2} & \cdots & x_{n+p+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n+k-1} & x_{n+k} & \cdots & x_{n+p+k+1} \end{bmatrix} \qquad Y = \begin{bmatrix} x_{n+p+1} \\ x_{n+p+2} \\ \vdots \\ x_{n+p+k} \end{bmatrix} \qquad (15)$$

Table.1 Definitions of performance criteria for time series forecasting

Metrics	Calculation
RMSE	$\left[\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \right]^{1/2}$
MAPE	$100 * \frac{1}{n} \sum_{i=1}^{n} \left\| \frac{\hat{y}_i - y_i}{y_i} \right\|$

Thus, we can predict $y_{t+1} = \{x_{t+1}\}$ using regularized least-squares regression [20]. Suppose the future value x_{t+1} will be predicted through function h with the history data $\{x_t, x_{t-1}, ..., x_{t-p+1}\}$. The function will be

$$\hat{x}_{t+1} = h\,(x_t, x_{t-1}, \cdots, x_{t-p+1}) \qquad (16)$$

Then, Our goal becomes to find the suitable function h .

The RLSR theory we described in Section 2 has provided the answer.

Suppose $V_t = \{x_t, x_{t-1}, ..., x_{t-p+1}\}$ then we have

$$\begin{aligned} \hat{x}_{t+1} &= h\,(x_t, x_{t-1}, \cdots, x_{t-p+1}) \\ &= h(V_t) \\ &= \sum_{i=1}^{l} c_i K(V_t, V_{i+p-1}) \end{aligned} \qquad (17)$$

As it is shown in 2.B and 2.C, the problem is treated by solving a positive linear equation with the conjugate gradient algorithm [14].

E. RLS-TS for Nonstationary Time Series Forecasting

For time series forecasting, one key problem is that time series are nonstationary. Given a stationary time series $\{x_t\}$, we can model it by using

$$\hat{x}_{t+1} = f\,(x_t, x_{t-1}, \cdots, x_{t-p+1})$$

p is the embedding dimension. However, for the non-stationary time series there is no unique f for the whole sequence[21].

The nonstationarity will lead to gradual changes in the actual relationship between independent and dependent variables. And, the nonstationarity can occur in many different ways.

Parzen [22] introduce the idea of modeling the nonstationary time series by estimating the models on the series in levels, i.e. by avoiding the differencing as in Box and Jenkis [23]. Since then, the issue of nonstationarity has been widely discussed in the literature [24-26]. And many papers have been tracked this issue by using NN and SVM [27-28, 31].

Many successful approaches are developed with considering the structural changes of the data. There are mainly two categories: regime-switching approaches and time-varying parameter approaches.

The regime-switching approaches consider the structural changes by using a given model for a limited time and then constructing a new model using the recent data points whenever the underlying data distribution is detected to be changed.

The time-varying parameter approaches make use of all available data points and handle the nonstationarity by using dynamic parameters. The advantage of the time-varying parameter approaches is that the long-term relationship inherent in the old data points can be retained [29]. The disadvantage is the parameters have to be chosen over time, the algorithm will be much slower.

In this paper, as far as RLS-TS concerned, we try to balance the time and the performance to the model. At the first step, the optimal p, kernel parameter, and regularization parameter are chosen. For the every next step, only c is updated without changing the other parameters (table 3). And the function (17) will be adjusted to the form (18).

$$\begin{aligned}\hat{x}_{t+1} &= h_t\ (x_t, x_{t-1}, \cdots, x_{t-p+1}) \\ &= h_t(V_t) \\ &= \sum_{i=1}^{l} c_{it} K(V_t, V_{i+p-1})\end{aligned} \tag{18}$$

F. Performance Criteria

There are many statistical metrics to evaluate the prediction performance, namely, the normalized mean square error (NMSE), root mean square error (RMSE), mean absolute error (MAE), directional symmetry (DS), weighted directional symmetry (WDS)[30]. In order to compare with Sfetsos and Siriopoulos' models[6], we use RMSE and MAPE to evaluate the model's accuracy. In Table.1, y and $\hat{y}$ represent the actual and predicted output values respectively, and n is the total number of data patterns.

III. EXPERIMENTAL RESULT

In this section, we describe our experimental results for RLS-TS as well as Sfetsos and Siriopoulos' models including Random Walk (RW), Linear Regression (LR), Atuoregression Integrated Moving Average (ARIMA), and Artificial Neural Network (ANN), and provide the comparisons to these models.

A. Data Set

We download data of daily closing values between US dollar and GB pound from Federal Reserve Statistical Release: http://www.federreserve.gov/releases/h10/Hist/. In order to make our result comparable, we used the same data as that in [6]. The data covers from January 2nd, 1990 to January 15th, 2001(see Fig. 2). Weekly and biweekly data are calculated from daily data.

The data from each series was split into three subsets namely the training set, the validation set, and the test set. These were formed using approximately the 70%, 19%, and 11%, respectively, of the entire set, irrespective of the number of data in each series (Table.2).

We split the data set into three subsets. Training set is for training the model, Validation set is for choosing the optimal parameters, and Test set is preserved as unknown data to test the forecasting accuracy of the RLS-TS model. In order to make our results comparable we use the same setting as that in [6], approximately 70%, 19%, and 11% respectively.

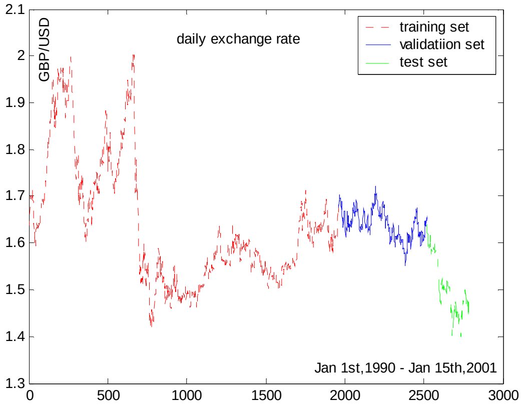

Fig. 2: Daily currency exchange rate between GBP/USD.

Table 2: GBP/USD data for RLS-TS model

Data	Training set	Validation set	Test set
Daily	1960	560	288
Weekly	392	102	54
Biweekly	188	56	24

Table 3: Process for parameter selection

(a) Give the initial values of p, kernel parameter, and regularization parameter respectively.
(b) With (14) and the training set, get the c.
(c) With c and the initial kernel parameter, apply (8) to the validation set, so to get the predicted value of validation set.
(d) With the predicted value of validation set, calculate the MAPE of the validation set.
(e) Use the grid search to repeat from (a) to (d), and get a series of MAPE corresponding to different p, kernel parameters, and regularization parameters.
(f) The parameters that produce the minimum MAPE will be the optimal parameters.
(g) For every next step, update the c without changing p, kernel parameter, and the regularization parameter.

B. RLS-TS

We have biweekly, weekly, and daily dataset. Also in order to compare our result to others, we use the RMSE and MAPE to evaluate our model. We use biweekly dataset and the criteria of MAPE to illustrate the process of model building for convenience.

After transferring the original dataset into the matrix form (15), we create the sample dataset (X, Y), and the dimension of X is set to *p*.

We developed our code in MATLAB based on the program ManifoldLearn, available at: http://manifold.cs.uchiicago.edu. With conjugate algorithm [14], we built the RLS-TS model to

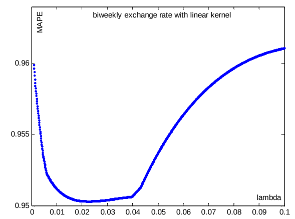

Fig.3: dot kernel

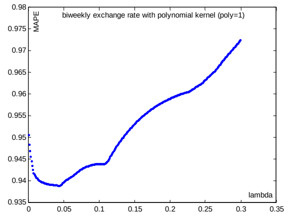

Fig.4: poly polynomial kernel (d=1)

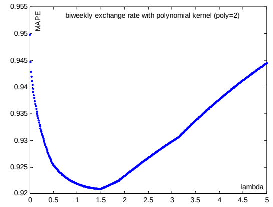

Fig.5: polynomial kernel (d=2)

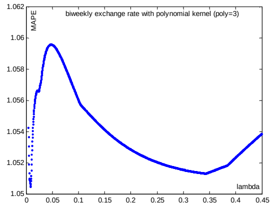

Fig.6: polynomial kernel (d=3)

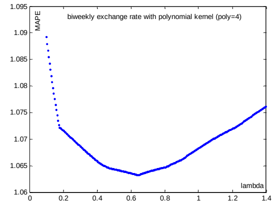

Fig.7: polynomial kernel (d=4)

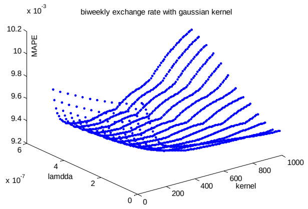

Fig.8: Gaussian kernel

Table.4 Daily USD/GBP currency exchange rate

Model Builder	Model	RMSE(*100)	MAPE
Proposed	RLS-TS-poly	0.7632	0.3772
Sfetsos and Siropoulos[6]	RW	0.7839	0.4090
	LR	0.7876	0.4122
	ARIMA	0.7833	0.4087
	ANN	0.7843	0.4080

Table.5 weekly USD/GBP currency exchange rate

Model Buidlier	Model	RMSE(*100)	MAPE
Proposed	RLS-TS-linear	1.3862	0.5827
	RLS-TS-poly	1.3755	0.7451
Sfetsos and Siropoulos[6]	RW	1.5495	0.7732
	LR	1.5353	0.8148
	ARIMA	1.4917	0.7739
	ANN	1.5019	0.7899

Table.6 weekly USD/GBP currency exchange rate

Model Buidlier	Model	RMSE(*100)	MAPE
Proposed	RLS-TS-linear	2.1804	0.6713
	RLS-TS-poly	2.2055	1.1964
	RLS-TS-rbf	2.2023	1.2707
Sfetsos and Siropoulos[6]	RW	2.3095	1.1914
	LR	2.4172	1.3357
	ARIMA	2.2885	1.2264
	ANN	2.3765	1.2447

predict the currency exchange rate between US dollar and GB pound.

C. *Parameter Selection*

In RLS-TS models, the kernel functions are restricted into three categories: the dot kernel, the polynomial kernel and the Gaussian kernel. We use a grid search method to find the optimal parameters. The optimal values of kernel parameter and regularization parameter are chosen based on smaller MAPE on the validation set. For a given kernel function Table 3 gives the details on how to choose the optimal parameters.

We restrict p in the interval [2, 20], the regularized and gaussion kernel parameters both in the interval [1×10^{-5}, 1×10^{5}]. For the kernel parameter, dot kernel does not have parameter. We restrict kernel parameter d in the interval of [1, 6] for polynomial kernel. Figures 3-8 show the relations between the parameters and the MAPEs. The optimal parameters are the one that gives the minimum MAPE.

With the procedure, we model the daily, weekly, and biweekly data using both RMSE and MAPE as evaluation criteria.

D. *Results Comparison*

The results of predicting USD/GBP currency exchange rate using RLS-TS model are given in Tables 4-6. From them we can see that using our RLS-TS to forecasting the USD/GBP exchange rate is better than all the models used by Sfetsos and Siropoulos[6] based on the criteria of RMSE and MAPE. When using the polynomial kernel in the RLS-TS model for forecasting, we found that there are more than one extreme value. Proper interval should be chosen to search for the optimal parameter in case getting the local minimum value. Tay and Cao [31] suggest that two small values of regularized parameter caused under-fitting the training data while too large a value of the parameter cause over-fitting the training data using the SVM for time series forecasting. Further research should be done on parameter choosing of the RLS-TS forecasting model.

Our RLS-TS model takes the full advantage of Statistical Learning (SL) theory using Reproducing Kernel Hilbert Space (RKHS) and adopts the RLSR theory directly in the time series forecasting. The performance of prediction is better than all the models in the referenced literature[6].

IV. CONCLUSION AND DISCUSSION

The RLS-TS model provides better performance than Random Walk (RW), Linear Regression (LR), Atuoregression Integrated Moving Average (ARIMA), or Artificial Neural Network (ANN) built by Sfetsos and Siriopoulos [6] when forecasting currency exchange rate between US dollar and GB pound.

Kernel-based regularized least-square regression (RLSR) is a new technique originally from the statistical learning (SL) theory. RLSR can deal with non-linear problems through mapping the samples into a higher dimension space using a kernel function [7]. We have developed a RLS-TS model, and use this model to predict the currency exchange rate between US dollar and GB pound. We also provide the details of using grid search method to choose the optimal kernel parameter and regularization parameter based on the MAPE or RMSE of the validation set.

Although this paper shows the effectiveness of the RLS-TS in forecasting exchange rate, there are only three kernel functions have been investigated. One of future work is to explore more useful kernel functions for further improving the performance of the RLS-TS models.

The Regularization Least-Squares method is originated from statistical learning theory and used for classification and function estimation issues. The hypothesis is that the training set is drawn from the random independent identically distributed observations. But for the times series forecasting, the observations are definitely dependent. So, the future work should give the new answer, in dependent conditions, of the four questions proposed by Vapnik [7] in statistical theory.

a) What are the conditions for consistency of the ERM (empirical risk minimization) principle?
b) How fast does the sequence of smallest empirical risk values converge to the smallest actual risk?
c) How can one control the rate of convergence (the rate of generalization) of the learning machine?
d) How can one construct algorithms that can control the rate of generalization?

References

[1] A. N. Refenes, "Constructive learning and its application to currency exchange rate forecasting," in Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-World Performance, R. R.Trippi and E. Turban, Eds, IL: Probus, 1993, pp. 465–493.

[2] C. M. Kuan and T. Liu, "Forecasting exchange rates using feedforward and recurrent neural networks," J. Appl. Econometrics, vol. 10, 1995, pp. 347–364.

[3] F. X. Diebold and J. A. Nason, "Nonparametric exchange rate pediction?," J. Int. Econom., vol. 28, 1990, pp. 315–332.

[4] W. Verkooijen, "A neural network approach to long-run exchange rate prediction," Computat. Econom., vol. 9, 1996, pp. 51–65.

[5] T. H. Hann and E. Steurer, "Much ado about nothing? Exchange rate forecasting: Neural networks vs. Linear models using monthly and weekly data," Neurocomputing, vol. 10, 1996, pp. 323–339.

[6] Athanasios Sfetsos and Costas Siriopoulos, "Time Series Forecasting of Averaged Data With Efficient Use of Information", IEEE Ansactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 35, NO. 5 September 2005, pp. 738-745.

[7] Vladimir N. Vapnik, "An overview of statistical learning theory", IEEE transactions on neural networks, 10 (5), 1999, pp.988-989.

[8] Gene Yeo, and Alex Rakhlin, "Feature Selection for (Nonlinear) Regularized Least-Squares Classification", 2002, pp. 1-2. Available : http://cbcl.mit.edu/projects/cbcl/res-area/abstracts/2002-abstracts/yeo-abstract-2.pdf

[9] Cristian Preda, "Regression models for functional data by reproducing kernel Hilbert spaces methods", Journal of statistical planning and inference, 137, 2007, pp. 829-840.

[10] John Shawe-Taylor, and Nello Cristianini, "Kernel methods for pattern analysis", Cambridge University Press, 2004, pp. 26-29.

[11] FANG Zhao-ben, LI Hong-xing, LI De-hui, LI Yong, "The Application of Regularized Least-Square Classification in Credit Scoring of Mortgage Loan Applicants", Lecture Notes in Decision Sciences, vol.7, 2006, pp. 320-327.

[12] Rong Zhang, and Alexander I. Rudnicky, "Word level confidence annotation using combinations of features", European conference on speech communication and technology, 2001, pp. 2105-2108.

[13] Ryan Rifkin, "Regression and least-square classification", 2006, pp. 1-12. Avaliable: http://www.mit.edu/~9.520/Classes/class04.pdf

[14] Jonathan Richard Shewchuck, "An introduction to the conjugate gradient method without the agonizing pain", Technical Report CMU-CS-94-125, School of Computer Science, Carnegie Mellon University, 1994, pp. 1-48.

[15] LI Hong-xing, FANG Zhao-ben, LI Yong, SONG Jia-shan, The Application of Regularized Least-Square Classification to Credit Scoring of Auto Loan Applicants, Journal of Data Analysis, vol.1, No.05, 2006.10, pp.53-64.

[16] U. Thissen, R. van brakel, A.P. De Weijer, W.J. Melssen, L.M.C. Buydens, "Using support vector machines for time series prediction", Chemometrics and intelligent laboratory systems 69, 2003, pp. 35-49.

[17] Tomas G. Dietterich. "Machine learning for sequential data: A review", InT. Caelli, editor, Structural, Syntactic, and Statistical Pattern Recognition, volume 2396 of Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 15-30.

[18] Kyung-jae Kim, "Finacial time series forecasting using support vector machines", Neurocomputing 55, 2003, pp. 307-319.

[19] N. Ancona, R. Maestri, D. marinazzo, L. Nitti, M Pellicoro, G.D. Pinna and Stramaglia, "Leave-one-out prediction error of systolic arterial pressure time series under paced breathing", Physiological Measurement 26, 2005, pp. 363-372.

[20] Ding-zhou Cao, Su-lin Pang, Yuan-huai Bai, "Forecasting exchange rate using support vector machines", Proceedings of the fourth international conference on machine learning and cybernetics, 2005, pp.3488-3452.

[21] M. Chang and C. Lin and R. Weng, "Analysis of nonstationary time series using support vector machines", in Pattern Recognition with Support Vector Machines --- First International Workshop, SVM 2002. Available:http://citeseer.ist.psu.edu/612183.html

[22] Fore Parzen, E., "AR–ARMA models for time series analysis casting, Journal of Time Series Analysis", Journal of Business and forecasting. Journal of Forecasting, 1, 1982, pp.67–82.

[23] Box, G. E. P., & Jenkins, G. M., "Time Series Analysis: Forecasting and Control", Holden–Day, San Francisco, CA. 1976.

[24] W.A. Barnett, E.R. Berndt and H. White, "Dynamic Econometric Modeling", Cambridge University Press, Cambridge, 1988.

[25] G. Deco, R. Neuneier and B. Schurmann, "Non-parametric data selection for neural learning in non-stationary time series", Neural Networks 10(3), 1997, pp.401–407.

[26] S. Wang, "An adaptive approach to market development forecasting", Neural Computing & Applications 8, 1999, pp.3–8.

[27] K. Pawelzik, J. Kohlmorgen, and K.-R. Muller. "Annealed competition of experts for a segmentation and classification of switching dynamics", Neural Computation, 8(2), 1996, pp.340-356.

[28] A. Kehagias and V. Petridis. "Time-series segmentation using predictive modular neural networks", Neural Computation, 9, 1997, pp.1691-1709.

[29] Lijuan Cao and Qingming Gub,"Dynamic support vector machines for non-stationary time series forecasting", Intelligent Data Analysis 6, IOS Press, 2002, pp.67–83.

[30] Jan G. De Gooijer and Rob J. Hyndman, "25 years of time series forecasting", International Journal of Forecasting 22, 2006, pp. 443– 473.

[31] Francis E.H. Tay, Lijuan Cao, "Application of support vector machines in financial time series forecasting", The international journal of management science, Omega 29, 2001, pp.309-317.