

Time Series Prediction

Using Support Vector Machines: A Survey

Abstract:

Time series prediction techniques have been used in many real-world applications such as financial market prediction, electric utility load forecasting, weather and environmental state prediction, and reliability forecasting. The underlying system models and time series data generating processes are generally complex for these applications and the models for these systems are usually not known *a priori*. Accurate and unbiased estimation of the time series data produced by these systems cannot always be achieved using well known linear techniques, and thus the estimation process requires more advanced time series prediction algorithms. This paper provides a survey of time series prediction applications using a novel machine learning approach: Support Vector Machines (SVM). The underlying motivation for using SVMs is the ability of this methodology to accurately forecast time series data when the underlying system processes are typically nonlinear, non-stationary and not defined *a-priori*. SVMs have also been proven to outperform other non-linear techniques including neural-network based non-linear prediction techniques such as multi-layer perceptrons. The ultimate goal is to provide the reader with insight into the applications using SVM for time series prediction, to give a brief tutorial on SVMs for time series prediction, to outline some of the advantages and challenges in using SVMs for time series prediction, and to provide a source for the reader to locate books, technical journals, and other online SVM research resources.

Introduction

The purpose of this paper is to present a general survey of Support Vector Machine (SVM) applications for time series prediction. This survey is based on publications and information found in technical books and journals as well as other informative data sources such as SVM technology-oriented websites such as <http://www.support-vector.net> and <http://www.kernel-machines.org>. SVMs used for time series prediction span many practical application areas from financial market prediction to electric utility load forecasting to medical and other scientific fields. Table 1 summarizes the number of SVM time series prediction publications in this survey paper with respect to application:

As noted in Table 1, the two predominant (published) research activities are financial market prediction and electric utility forecasting. There are several other applications listed—from control system applications, environment and weather forecasting and other applications involving non-linear processes.

It should be noted that the focus of this survey is on the applications and the general numerical accuracy of the SVM techniques associated with time series prediction. Where applicable, notes are made in this survey with respect to the training methodologies used to “tune” the SVMs for specific applications. Although training time (numerical computation time) is an important design criteria, most of the applications listed in this survey use data sets that are mainly static or change slowly with time (such as stock forecasting using

TABLE 1 Number of SVM time series prediction publications listed by application.

APPLICATION	NUMBER OF PUBLISHED PAPERS SUMMARIZED IN THIS SURVEY
FINANCIAL MARKET PREDICTION	21
ELECTRIC UTILITY FORECASTING	17
CONTROL SYSTEMS AND SIGNAL PROCESSING	8
MISCELLANEOUS APPLICATIONS	8
GENERAL BUSINESS APPLICATIONS	5
ENVIRONMENTAL PARAMETER ESTIMATION	4
MACHINE RELIABILITY FORECASTING	3

daily closing prices). The reader is directed to the web based references listed in the reference section of this survey to learn more about training techniques.

Traditionally SVMs, as well as other learning algorithms such as Neural Networks, are used for classification in pattern recognition applications. These learning algorithms have also been applied to general regression analysis: the estimation of a function by fitting a curve to a set of data points. The application of SVMs to general regression analysis case is called Support Vector Regression (SVR) and is vital for many of the time series prediction applications described in this paper. For comparison, Table 2 below contrasts selected attributes and challenges associated with some of the most common classic methods, artificial neural network (ANN) based time series prediction methods, and SVR:

A more detailed performance summary of intelligent “tools” (i.e., ANN based methods) for time series prediction, specifically financial market time series prediction applications, can be found in Table 1 of [48].

The primary objective of this paper is to provide a survey of SVM time series prediction literature and data sources accompanied by the following: 1) a brief summary of the broad range

of application(s) using SVM time series prediction methods, 2) a brief discussion of the observations generated from this survey with respect to the technical merits and challenges associated with SVM time series prediction, and 3) a resource for the reader to locate and research SVMs and their applications.

Time Series Prediction Summary

The purpose of this section is to provide references and a general outline for time series prediction theory. There are vast amounts of technical references, books, and journal articles detailing time series prediction algorithms and theory for both linear and non-linear prediction applications. The reader is encouraged to research classical publications such as Orfanidis [1] and Kalman [2] for more details.

Fundamentally, the goal of time series prediction is to estimate some future value based on current and past data samples. Mathematically stated:

$$\hat{x}(t + \Delta_i) = f(x(t - a), x(t - b), x(t - c), \dots), \quad (1)$$

where, in this specific example, \hat{x} is the predicted value of a (one dimensional) discrete time series x .

The objective of time series prediction is to find a function $f(x)$ such that \hat{x} , the predicted value of the time series at a future point in time is *unbiased* and *consistent*. Where i is an index to a discrete time series value and N is the total number of samples. It should be noted that another measure of a predictor's goodness is *efficiency* as related to *bias*. The Cramér-Rao bound provides the lower bound for the variance of *unbiased* estimators [1]. If the estimator achieves this bound, then it is said to be *efficient*. This analysis was not provided in any of the papers summarized in this survey.

Estimators generally fall into two categories: linear and non-linear. Over the past several decades, a vast amount of technical literature has been written about linear prediction: the estimation

TABLE 2 Summary of advantages and challenges of classical, ANN based, and SVR time series prediction methods.

TIME SERIES PREDICTION METHOD	ADVANTAGES	CHALLENGES
AUTOREGRESSIVE FILTER	CAN BE COMPUTATIONALLY EFFICIENT FOR LOW ORDER MODELS CONVERGENCE GUARANTEED MINIMIZES MEAN SQUARE ERROR BY DESIGN	ASSUMES LINEAR, STATIONARY PROCESSES CAN BE COMPUTATIONALLY EXPENSIVE FOR HIGHER ORDER MODELS
KALMAN FILTER	COMPUTATIONALLY EFFICIENT BY DESIGN CONVERGENCE GUARANTEED MINIMIZES MEAN SQUARE ERROR BY DESIGN	ASSUMES LINEAR, STATIONARY PROCESSES ASSUMES PROCESS MODEL IS KNOWN
MULTI-LAYER PERCEPTRON	NOT MODEL DEPENDENT NOT DEPENDENT ON LINEAR, STATIONARY PROCESSES CAN BE COMPUTATIONALLY EFFICIENT (FEED FORWARD PROCESS)	NUMBER OF FREE PARAMETERS LARGE SELECTION OF FREE PARAMETERS USUALLY CALCULATED EMPIRICALLY NOT GUARANTEED TO CONVERGE TO OPTIMAL SOLUTION CAN BE COMPUTATIONALLY EXPENSIVE (TRAINING PROCESS)
SVM/SVR	NOT MODEL DEPENDENT NOT DEPENDENT ON LINEAR, STATIONARY PROCESSES GUARANTEED TO CONVERGE TO OPTIMAL SOLUTION SMALL NUMBER OF FREE PARAMETERS CAN BE COMPUTATIONALLY EFFICIENT	SELECTION OF FREE PARAMETERS USUALLY CALCULATED EMPIRICALLY CAN BE COMPUTATIONALLY EXPENSIVE (TRAINING PROCESS)

of a future value based on the linear combination of past and present values. Real world time series prediction applications generally do not fall into the category of linear prediction. Instead, they are typically characterized by non-linear models.

Support Vector Machines for Time Series Prediction

Support Vector Machines and Support Vector Regression are based on statistical learning theory, or VC theory (VC – Vapnik, Chervonenkis), developed over the last several decades. Many books, journal publications, and electronic references currently exist. The reader is directed to Vapnik's reference books [3, 4] and an introductory reference book by Cristianini/Shawe-Taylor [5] for further study. Brief, general descriptions of Vapnik's learning theory and SVM regression can be found in references [6–8]. Finally, many publicly available websites exist (at the time this paper was written) that also offer an extensive amount of information and software for SVMs. See references [94–104].

The Support Vector Machine (SVM), developed by Vapnik and others in 1995, is used for many machine learning tasks such as pattern recognition, object classification, and in the case of time series prediction, regression analysis. Support Vector Regression, or SVR, is the methodology by which a function is estimated using observed data which in turn "trains" the SVM. This is a departure from more traditional time series prediction methodologies in the sense there is no "model" in the strict sense – the data drives the prediction.

Given a set of time series data $x(t)$, where t is a series of N discrete samples: $t = \{0, 1, 2, \dots, N-1\}$, and $y(t + \Delta)$ is some predicted value in the future (t greater than or equal to N). For a time series prediction algorithm, equation (1) defines a function $f(x)$ that will have an output equal to the predicted value for some prediction horizon. By using regression analysis, equations (2) and (3) both define these prediction functions for linear and non-linear regression applications respectively:

$$f(x) = (w \cdot x) + b \quad (2)$$

$$f(x) = (w \cdot \phi(x)) + b. \quad (3)$$

If the data is not linear in its "input" space, the goal is to map the data $x(t)$ to a higher dimension "feature" space, via $\phi(x)$ (referred to as a Kernel Function), then perform a linear regression in the higher dimensional feature space [11].

The goal is to find "optimal" weights w and threshold b as well as to define the criteria for finding an "optimal" set of weights. First is the "flatness" of the weights, which can be measured by the Euclidean norm (i.e. minimize $\|w\|^2$). Second is the error generated by the estimation process of the value, also known as the empirical risk, which is to be minimized. The overall goal is then the minimization the regularized risk $R_{\text{reg}}(f)$ (where f is a function of $x(t)$) as defined as:

$$R_{\text{reg}}(f) = R_{\text{emp}}(f) + \frac{\lambda}{2} \|w\|^2. \quad (4)$$

The scale factor λ is commonly referred to as the regularization constant and this term is often referred to as the capacity control term. Its function is to reduce "over-fitting" of data and minimize bad generalization effects. The empirical risk is defined as:

$$R_{\text{emp}}(f) = \frac{1}{N} \sum_{i=0}^{N-1} L(x(i), y(i), f(x(i), w)), \quad (5)$$

where, i is an index to a discrete time series $t = \{0, 1, 2, \dots, N-1\}$ and $y(i)$ is the "truth" data (training set) of the predicted value being sought. $L(\cdot)$ is a "loss function" or "cost function" to be defined.

Two of the more common loss functions that are used are the ϵ -insensitive loss function defined by Vapnik and the quadratic loss function typically associated with Least Squares Support Vector Machine (LS-SVM). The details of the LS-SVM development can be found in [9, 10]. To solve for the optimal weights and minimize the regularized risk, a quadratic programming problem is formed (using the ϵ -insensitive loss function):

$$\text{minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n L(y(i), f(x(i), w))$$

where

$$L(y(i), f(x(i), w)) = \begin{cases} |y(i) - f(x(i), w)| - \epsilon & \text{if } |y(i) - f(x(i), w)| \geq \epsilon \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Equation (9) is referred to as the regularized risk function. The constant "C" also includes the $(1/N)$ summation normalization factor and ϵ is the "tube size," referring to the precision by which the function is to be approximated. It should be noted both ϵ and C are both user defined constants and are typically computed empirically. It is inherently assumed that a function $f(x)$ actually exists and the optimization problem is feasible; however, errors may have to be accepted to make the problem feasible. To account for errors, "slack variables" are typically introduced.

Solving for the optimal weights and bias values is an exercise in convex optimization, which is made much simpler by using Lagrange multipliers and forming the dual optimization problem given by (7):

$$\begin{aligned} \text{Maximize: } & -\frac{1}{2} \sum_{i,j=1}^N (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \langle x(i), x(j) \rangle \\ & - \epsilon \sum_{i=1}^N (\alpha_i - \alpha_i^*) + \sum_{i=1}^N y(i) (\alpha_i - \alpha_i^*) \\ \text{Subject to: } & \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0: \alpha_i, \alpha_i^* \in [0, C]. \end{aligned} \quad (7)$$

The solution for the weights is based on the Karush-Kuhn-Tucker conditions that state at the point of the optimal solution, the product of the variables and constraints equal zero. Thus the approximation of the function $f(x)$ is given as the sum of the optimal weights times the dot products between the data points as:

$$f(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \langle x, x(i) \rangle + b. \quad (8)$$

Those data points on or outside the ϵ tube with non-zero Lagrange multipliers α are defined as Support Vectors. As can be seen, the optimal weights associated with having non-zero Lagrange multipliers is typically less than the entire data set, meaning one does not need the entire data set to define $f(x)$. The sparseness of this solution is one of several advantages of using this methodology.

To carry out the non-linear regression using SVR, it is necessary to map the input space $x(i)$ into a (possibly) higher dimension feature space $\phi(x(i))$. Noting that the solution of the SVR relies on the dot products of the input data, a kernel function that satisfies Mercer's conditions can be generated as:

$$k(x, x') = \langle \phi(x), \phi(x') \rangle, \quad (9)$$

which can be directly substituted back into equation (8) and the optimal weights w can be computed in feature space in exactly the same fashion. There are several kernel functions that satisfy Mercer's conditions (required for the generation of kernel functions) such as Gaussian, polynomial, and hyperbolic tangent. The use of kernels is the key in SVM/SVR applications. It provides the capability of mapping non-linear data into “feature” spaces that are essentially linear, where the optimization process can be duplicated as in the linear case. The use of Gaussian kernels appears to be the most prevalent choice, but typically empirical analyses are necessary in selection of an appropriate kernel function. SVR and its derivation are described in detail in publications found in [11-27], especially Smola and Schölkopf [13].

The resulting SVR architecture is given below in Figure 1 (reproduced here from Figure 2 in [13]).

There are several Quadratic Programming (QP) methods that can be used for training SVMs and most of the algorithms are publicly available (see the text references [3-5] and the general web based references found in [80-87]). The Sequential Minimization Optimization (SMO) algorithm is one of the most popular methods used for solving the QP problem (developed by Platt in 1999—a

description of the algorithm is given in detail in [5, 13]). This is the most popular among the various methods available that are described in the application summaries. It is beyond the scope of this survey paper to analyze and compare training algorithms, but the algorithms play an important role in the implementation of the SVR for practical applications and are mentioned in the summaries of these applications.

Financial Data Time Series Prediction Using Support Vector Regression

Of all the practical applications using SVR for time series prediction, financial data time series prediction appears to be the most studied along with electrical load forecasting. Twenty one research papers are listed in the references (in chronological order) detailing SVR applications for specifically predicting stock market index (time series) values and miscellaneous financial market time series data. The inherent noisy, non-stationary and chaotic nature of this type of time series data appears to lend itself to the use of non-traditional time series prediction algorithms such as SVR. Many different variations of SVR and combinations of SVRs with other learning techniques are found for financial time series prediction and are summarized below.

Trafalis and Ince [28] compared SVR to more traditional Neural Network architectures including the feed forward multilayer perceptrons using back propagation and radial basis functions for the prediction of stock price indices. Using the ϵ -insensitive loss function and several different quadratic optimization algorithms, the authors demonstrated the SVR's superior performance over the other NN based applications for a very small time window of three samples and a very small prediction horizon of one sample.

Tay and Cao [29] studied the use of SVR for predicting five specific financial time series sources including the S&P 500 and several foreign bond indices. The results were compared to a feed-forward MLP using back propagation for a prediction horizon of five samples (days). The data was “pre-processed” by applying an exponential moving-average window and outliers (identified as data beyond two standard deviations) were replaced with relatively close values. The data was broken down into three sets: training set, validation set, and test set (typical for neural network training methodologies). The SVR significantly outperformed the BP NN. They conclude that the ability of the SVR to appropriately fit the data (as compared to over-fitting issues related to MLP based NNs), is one key reason for better performance. They published several other related SVR applications for financial data time series prediction [31, 32, 33, 39]. An alternative architecture using a “mixture of experts (ME)” approach is presented in [31]. This is a two stage approach with the first stage being a self-organizing feature map (SOM) and the second stage containing a set of SVR “experts”. The parameters for the kernel functions used, such as C and ϵ , were essentially derived empirically

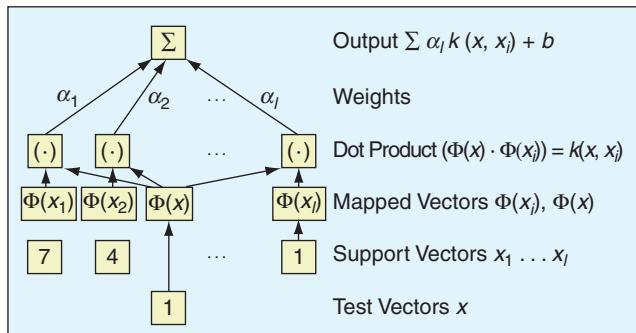


FIGURE 1 SVM architecture.

and the overall approach was shown to have not only better prediction performance as compared to a single SVR approach, but also superior convergence speed. In [32], Tay and Cao proposed a modified version of SVR for financial series prediction called C-ascending

SVMs. The goal of this approach is to weight the most current (in time) ε -insensitive errors and de-weight the more distant ones – analogous to the discounted least squares approach. Both linear and exponential weighting functions were tested against several stock indices including the S&P 500. They conclude that better performance (for five sample prediction horizon) can be obtained using this method as compared to a standard SVR implementation. They proposed another adaptive approach and modification to the SVM – the ε -Descending Support Vector Machine [33]. Instead of the regularization constant changing with time, the tube width was varied with time and was weighted exponentially with the most recent data points being penalized the most. Every training data point will use a different tube size ε . For both simulated data (weighted sinusoids) and financial data sets (stock indices including S&P 500), a better overall performance in NMSE was found using the ε -DSVM with a five sample prediction horizon. In [39], Cao and Tay proposed the Adaptive SVM (ASVM) which modifies both the tube size ε (see [33]) and the regularization constant C (see [32]). Increasing ε will decrease the number of support vectors (support vectors in SVR are the points on or outside the ε tube – the larger ε , the smaller the number of support vectors). The decrease in support vectors represents a more sparse solution, with a tradeoff in prediction accuracy. The more recent time samples were given more weight and had greater influence on the solution. As compared to a weighted back propagation MLP, the ASVM showed better performance for five selected stock indices.

Van Gestel *et al.* [30] proposed the use of an LS-SVM used in a Bayesian evidence framework. Both a point time series prediction and volatility models for financial stock index prediction are developed in this paper. A marginal improvement in MSE, MAE, and Negative Log Likelihood (NLL) was found using this method compared to other traditional methods such as auto regressive models using US short term T-bill and DAX30 market data.

Yang *et al.* [34] proposed a non-fixed and asymmetrical margin, along with the use of momentum, to improve the SVR's ability to predict the financial time series. The ε -insensitive loss function is modified to have different "upside" and "downside" margins (ε_u and ε_d) based on the standard deviation of the input data. The "margin" is the linear combination of this standard deviation and the momentum term. By applying these time varying parameters to the loss function, the authors showed that the MAE for one step ahead prediction of the Hang Seng Index (HSI) and Dow

Signal processing, control and communications systems applications face an additional challenge of being highly sensitive to computation timing, as expected in real time signal processing applications.

Jones Industrial Average (DJIA) was improved vs. using standard AR and RBF models. It is worth pointing out that they presented a similar discussion of the use of asymmetrical margin determination for SVR based on the standard deviation of the data in [35]. A more thorough discussion of this approach can be found in H. Yang's thesis [37]. In [44], the same authors propose a two phase SVR training method for detecting outliers in the data, thus reducing the prediction error (RMSE and MAE in this case). In contrast to, the "upside" (ε_u) and "downside" (ε_d) margins are adaptable. They extend their asymmetric margin approach in [34] to be adaptable relative to the slack variables (also time dependent). The results showed a small increase in prediction performance at the price of retraining the SVR.

Abraham *et al.* [36] compared the one-step ahead time series prediction performance of an ANN using the Levenberg–Marquardt algorithm, SVM, Takagi–Sugeno neuro-fuzzy model and a Difference Boosting Neural Network (DBNN). Only a brief description of SVM for classification applications was provided. For one step ahead prediction of the Nasdaq-100 index and the NIFTY index, the SVM performed marginally better. In [41], Abraham and Yeung extended the work in [36] by combining the outputs of the four approaches (ANN using the Levenberg–Marquardt algorithm, SVM, Takagi–Sugeno neuro-fuzzy model and a Difference Boosting Neural Network (DBNN)). The combining of the four outputs is done in two ways: 1) a direct approach by using source selection using the lowest absolute error of the four methods as the decision criteria and 2) by using a Genetic Algorithm (GA) to optimize the RMSE, MAP, and MAPE (see [41] for fitness function specifics of the GA). Again using the Nasdaq-100 index and the NIFTY index, the one-step-ahead prediction of these intelligent paradigm approaches showed that the direct approach outperformed the GA approach.

Ongsritrakul and Soonthronphisaj [38] combined several approaches including MLPs, decision trees, and SVRs to predict gold price. The decision tree feeds "factors" into the SVR time series prediction process which then serves as input to a linear regression model, an MLP, and an SVR to predict the gold price. The MSE, MAD, and MAPE were computed for the three models and the SVR appears to outperform the other two models.

Liang and Sun [40] proposed an adaptive method for modifying the kernel function during the training of an SVR. Using a Gaussian RBF kernel, the authors proposed to modify the kernel function based on the method of information geometry. Using a conformal mapping, a new method was introduced

which improves the precision of forecasting. An optimal partition algorithm (OPA) was used to modify the kernel, making the kernel data dependent. Results using S&P 500 time series data as well as Shanghai Stock Exchange (CISSE) data were presented and compared to an unmodified SVR.

Kim [42] used an SVM for prediction of the Korea composite stock price index (KOSPI), not an SVR. By selecting twelve “technical indicators” (i.e. features), he used the SVM to predict the direction of the daily price change in the stock price. This is a slightly different application in the sense that the SVM only predicts daily direction, not the actual index price itself (an SVR application). As compared to a three layer MLP using back propagation and a case-based reasoning (CBR – in this application, a nearest-neighbor approach implementation), the SVM approach provided better results than the other two approaches in predicting index direction of change.

Bao *et al.* [43] proposed the use of SVR using the ϵ -insensitive loss function, an RBF kernel function and predict, five days (samples) ahead, the stock price of Haier, Inc. (Shanghai Stock Exchange). The procedure and normalization of the data is very similar to the work of Cao and Tay.

Similar to Kim [42], Huang *et al.* [45] proposed the use of an SVM for predicting the direction of the NIKKEI 225 index based on several inputs including interest rates, CPI, and other market attribute data. They compared this performance to linear discriminate analysis (LDA), quadratic discriminant analysis (QDA), and an Elman back propagation neural network (EBNN). In addition, they proposed a “combining model” that weights the output of the SVM with the other classification methods to produce a prediction. The authors stated that, individually, the SVM performed the best (highest “hit” ratio) and the “combined model” performed slightly better than the SVM.

Bao *et al.* [46] proposed a Fuzzy Support Vector Machines Regression (FSVMR) method for predicting the stock composite index of the Shanghai Stock Exchange. They stated that for two state classification problems, some of the input data points are corrupted by noise and should be (possibly) discarded while others that are marginal but important should be assigned to a class. Using a ϵ -insensitive loss function and an RBF kernel function, the FSVMR was trained using a cross-validation method to find the variable parameters. They showed that the NMSE using this approach was effective than stand alone SVR approaches.

Cao *et al.* [47] used an SVR using the ϵ -insensitive loss function to make one step ahead predictions for the British Pound and American Dollar exchange rate. By selecting the optimal parameters empirically using a validation data set, the authors stated that this method performed well, but resulting offsets (time shifts) moves the regression curve to the right (noted by the author for future study). The last financial paper by Quek and Ng [48] described a Genetic Complementary Learning (GCL) method for stock market prediction. Although the focus of this paper is on GCL, Quek made financial time series performance measurements against other traditional NN approaches including the SVM, and SVM performed as well as

other methods with respect to the root mean square error (RMSE) except the proposed GCL method. There is a comprehensive table in this reference that shows twenty alternative methods for stock prediction methodologies that have been published within the last ten years.

General Business Applications Using Support Vector Regression for Time Series Prediction

Following are summaries of five papers that describe the use of SVR for time series prediction relative to the following general business applications: (1) Electricity Price Forecasting, (2) Credit Rating Analysis, (3) Customer “Churning” – Auto Insurance Market Prediction, (4) Financial Failure of Dotcoms – Financial Analysis, and (5) Production Value Prediction of the Taiwanese Machinery Industry.

Sansom *et al.* [49] compared the performance of an SVR vs. an MLP for predicting Australian national Electricity Market Prices one week ahead (seven samples). Using 90 days of electricity forecasting prices, they showed that an SVR outperformed an MLP in training time, but obtained similar results in accuracy (MAE) which may have been due to the way training data was selected and used.

Huang *et al.* [50] investigated the use of an SVM to estimate corporate credit ratings. This is not a traditional SVR application, but an SVM classification problem. They used two data sets, Taiwan and US corporate rating data, for training both an SVM with 21 variables as input and an ANN using back propagation. The authors stated that the SVM outperformed the BP results for both data sets.

Hur and Lim [51] compared an SVM for predicting customer “churn” ratio for auto insurance market prediction. In this application, “churn” represents a customer changing from one auto insurance company to another due to the increased accessibility of customers to “on-line” insurance vendors. As in the business application for Huang *et al.* [50], this application does not use an SVR methodology. Fifteen variables were selected as input to an SVM, and the SVM was trained to predict the “churn” ratio (see [51]). SVM was shown to outperform ANN.

Bose and Pal [52] analyzed the fate of failed dotcoms using an SVM (as in Huang *et al.* [50] and Hur and Lim [51]), not an SVR. The goal is to train the SVM to determine if the dotcom would succeed (a “1” classification) or fail (a “0” classification) based on using twenty four “financial ratios” such as total debt to total assets ratio, net income to total assets ratio, etc. as the vector input to the SVM. The SVM parameters were determined empirically and the results show that it was easier to classify a survived dotcom company than a failed one.

Pai and Lin [53] discussed the use of an SVR for predicting the one-step ahead production values of the Taiwanese machinery industry. They compared the performance of the standard SVR using the ϵ -insensitive loss function and Gaussian kernel to a Seasonal Auto-Regressive Integrated Moving Average (SARIMA) method and a general regression neural network (GRNN). The SARIMA model (developed by Box and

Jenkins) was used to model long time period variations such as seasonal dependencies. Using the MAE, MAPE, RMSE and NMSE as performance metrics, they showed that the SVR approach outperformed the other two methods, especially in the MAE and MAPE metrics.

Environmental Parameter Estimation Using Support Vector Regression for Time Series Prediction

SVR has been used for the prediction of environmental parameters such as air quality parameters rainfall estimation and detection, and weather forecasting. The following section summarizes the four papers for environmental parameter estimation:

Lu *et al.* [54] proposed the use of SVR to forecast air quality parameters. The determination of short term air quality relies on the use of non-linear regression methods such as SVR. The input data was respirable suspend particles (RSP) as collected with other major pollutants such as nitrogen oxides, etc. Using SMO for training and a Gaussian kernel function with arbitrarily selected parameters, the SVR performance (MAE) for predicting one week ahead outperformed an RBF network using the same data set. A sensitivity analysis was provided for the free parameters (regularization constant, kernel constants, etc.) and there was no set heuristic for determining these parameters.

Wang *et al.* [56] extended their environmental pollution prediction work from [54] and compare an SVR approach to an adaptive radial basis function (ARBF) network and an ARBF using principle component analysis (PCA). Again, they tried to forecast respirable suspended particulate (RSP) concentrations with a 72 hour forecast horizon. The free parameters of the SVR were determined empirically using MAE, RMSE, and Wilmott's index of agreement (WIA) as metrics for indicating the most accurate predictions. For the presented data, the SVM outperformed the ARBF network and the ARBF/PCA network for the three day ahead prediction horizon. They implied, as in [54], the challenge remains to find a suitable heuristic to determine the free parameters of the SVR.

Trafalis *et al.* [55] applied both SVR and Least Squares (LS) SVR methodologies to predict rainfall estimation and the presence of rain using WSR-88D radar data (also known as NEXRAD). For the rainfall rate estimation problem, the LS-SVR using a polynomial kernel (refer to [55] as well as work from Suykens [9, 10] for more information on LS-SVR) outperformed the SVR using a Gaussian RBF kernel and a linear regression technique in the MSE. For the detection of rainfall (SVM type classification problem), they mentioned that the SVR slightly outperformed the LS-SVR for the detection (classification of rain existence) of rain in a geographic grid. The authors pointed out that the use of the LS-SVR loses the sparseness quality of the representation of the solution as compared to the SVM, noting that the solution of

Of all the practical applications using SVR for time series prediction, financial data time series prediction appears to be the most studied along with electrical load forecasting.

the optimization problem of the LS-SVR is essentially a matrix calculation.

Prem and Srinivasa Raghavan [57] applied SVR in use with the Network Weather Services – a “grid” of computational nodes used for weather prediction. Their goal was to optimize the network parameters such that the final weather forecast output is the most accurate given the constraints of the computational architecture and network topology (i.e., QoS). By accurately predicting the need for different resources required, the overall system can adapt more efficiently and provide better forecasting results (essentially, this is a dynamic scheduling problem for providing and maintaining Weather Prediction Services). As compared to other AR methods, the SVR outperformed the other methods, especially in multi-step ahead prediction of CPU time and network bandwidth.

Electric Utility Load Forecasting Applications Using Support Vector Regression for Time Series Prediction

A non-linear prediction problem found in power systems research is the forecasting of electrical power consumption demands by consumers. There are many beneficial aspects to the accurate prediction of electrical utility load forecasting including proper maintenance of electrical energy supply, the efficient utilization of electrical power resources, and the proper administration and dissemination of these resources as related to the cost of these resources to the consumer. Seventeen research papers concerning electricity load forecasting are summarized below:

Chang *et al.* [58] proposed an SVR approach for the EUNITE Network Competition which is the prediction of daily maximal electrical load of January 1999 based on temperature and electricity loading data from 1997 to 1998. It is interesting to note that there is clearly a periodic component within the data set due to the seasonal variation of consumer electricity demand, “holiday” effects (use of less electricity during major holidays), and the impact of weather on electricity demand. Their inputs were several attributes, including binary attributes for indicating which day of the week it is, is it a holiday, etc. From these attributes, they formulated the predicted max load, which is a numerical value. They concluded that the use of the temperature data did not work as well because of the inherent difficulty in predicting temperature and they also concluded that this SVR approach was feasible for determining an accurate prediction model. Chen *et al.* [62] approach described in [58] was the winning approach for the EUNITE Network Competition. The paper describes the SVM implementation. With respect to the

SVMs used for time series prediction span many practical application areas from financial market prediction to electric utility load forecasting to medical and other scientific fields.

design details, it is interesting to note that the use of temperature in their model actually decreases the accuracy of their predictions, which they state as sensitivity to the variance of the output to improper temperature estimations. They experimented with inputs excluding the previous (in time) load data and found poorer performance (note that the inputs to the SVR are not only time series load data).

Mohandes [59] compared the results of a standard SVR using a sigmoid kernel function and the ϵ -insensitive loss function to a standard autoregressive model of order one for short term forecasting electrical load forecasting (short term meaning less than one week prediction horizon). The preprocessing of the data included the elimination of annual periodicity and linearly increasing trends from the data. The author showed that the performance of the SVR, with respect to the RMSE, was much lower than the AR method, especially as the number of training points was increased.

The electricity supply industry (ESI), generator companies, and other electrical utility load entities depend on load forecasting to maximize revenues. So, Sansom and Saha [60] proposed the use of an SVM and not SVR for forecasting the wholesale (spot) electricity prices and compares performance to a linear regression (as a sum of sine waves) and an ANN trained using back propagation. In their research, the inputs to these prediction methodologies were a set of 14 “attributes” including spot price at different previous time samples, require capacity, etc. Using the SVM approach appeared to work well as compared to the other methodologies, but the authors stated that this approach, under certain circumstances where selected data points (attributes) were removed from the problem, performed far worse than the other approaches with respect to MAE. They mentioned that the superior SVM performance with all the data may have been “luck” and recommended further research.

Tian and Noore [61] proposed the use of an SVM to predict a wide span of forecast horizons (hourly and several days ahead) of electrical load using measurements from Berkeley, California. Their modeling also takes into account temperature and humidity as factors into training the SVM as well as a normalization of the data to a range of $[0, 1]$. As compared to a cosine radial basis function neural network and a feed forward neural network, the SVM approach using electrical load, temperature and humidity outperformed the other methods in MSE, RMSE, and Durbin-Watson d statistic.

Dong *et al.* [63] discussed the use of SVM to predict “landlord energy consumption” – the electrical load necessary for large commercial buildings to operate normally (use of air conditioning, elevators, etc.). Their work considers other factors

associated with the electrical load forecasting problem such as temperature, humidity, and global solar radiation. Their selection of an RBF kernel is based on stated shortcomings of other kernel functions such as polynomial or sigmoid (complexity as example).

The authors stated that this approach is superior to other NN based approaches in performance and small model parameter selection.

Bao *et al.* [64] proposed the use of a self-organizing map (SOM) along with an SVM to predict short term electrical load forecasts based on EUNITE competition data. The purpose of the SOM is to cluster training data, based on time sample (i.e. day) and correlate the same weather conditions found on the training day(s) to the present day’s weather conditions. In terms of performance, the authors stated that this hybrid approach outperforms the SVM by itself. It should also be noted that smoothing (preprocessing) the data, in their case, worsened the MAPE performance.

Pai and Hong [65] proposed a Recurrent Support Vector Machine with Genetic Algorithms (RSVMG) for the forecasting of electrical loads. The Genetic Algorithms (GA’s) were used to determine the free parameters of the SVMs, specifically the regularization constant (C), the tube size (ϵ), and the Gaussian kernel parameter σ . A recurrent SVM (RSVM) was detailed as one of their approaches, which uses a standard MLP with back propagation combined with the SVM architecture. The output of the ANN was fed back (recurrent) to the inputs of the MLP prior to the SVM architecture. The authors compared the RSVMG approach to the SVMG model and the ANN model and results show, with respect to MAE, the superior performance of using the GA approach to select model parameters as well as the introduction of feedback (recursion) into the NN architecture.

Ji *et al.* [66] proposed the use of mutual information (MI – the computation of Shannon’s entropy) to select the “best” input variables, i.e., the data points that maximize MI. Then, an LS-SVM is trained to make the prediction up to six samples ahead. The first two-thirds of the data set (Poland Electricity Dataset) were used to train the LS-SVM. There were two methodologies compared, “direct” forecast where the prediction horizon is calculated directly from N samples and “recursive” forecast where one-step ahead prediction is calculated up to the desired prediction horizon (six samples in this case). The authors stated that direct prediction performed better in MSE than recursive.

Zhang [67] discussed the use of SVM for short-term load forecasting. The author stated that most linear models such as Kalman filtering, AR, and ARMA models are not typically sufficient to model the nonlinearities associated with short term load forecasting processes. The use of SVR, with both electrical load data and corresponding weather time series data, appears to outperform other NN based techniques including a back-propagation neural network. The author also used cross validation to select the free parameters of the RBF kernel function as

well as the regularization constant. The MAPE of the SVM approach was lower than that of the BPNN.

Li *et al.* [68] proposed the use of both SVR and a “Similar Day Method” to predict the next day forecasting of electrical loads.

The purpose of the “Similar Day Method” was to identify days where the sampled data (electrical load, weather, etc.) is similar to the present day and use this information to “amend” the result given by the SVR result. This essentially “corrects” the output of the SVR. The authors stated that this method is an effective short term load forecasting method as compared to using SVM alone.

Pai and Hong [69] discussed the use of SVM for short term load forecasting using a simulated annealing (SA) algorithm, which is based on the annealing process of material physics, to select the SVM parameters. The simulated annealing algorithm combined with SVR is called SVMSA. Essentially, initial values of σ , C , and ϵ (the free parameters associated with the kernel function and the loss function) are set and the SA algorithm selects a “provisional” state by randomly adjusting these free parameters. A repetition of this procedure is executed until a final state is found where the MAPE of the SVM is found to be at some acceptable level. This technique is compared to the ARIMA and general regression neural network (GRNN) [53] and for Taiwanese electricity load data, this technique significantly outperformed the two other methods.

Wu and Zhang [70] presented a hybrid of several approaches for forecasting electrical load. Based on the assumption that the electrical load data exhibits both chaotic and periodic behavior, they employed wavelet transforms as well as an average mutual information (AMI) algorithm (based on chaos theory) along with a Least Squares Support Vector Machine (LS-SVM) to predict the maximal electrical load of EUNITE competition data. Based on other EUNITE publications of prediction results, this technique was claimed by the authors to be superior in performance relative to MAPE and maximal error (ME). The authors concluded that the selection of LS-SVM parameters is “tough” (i.e., assumed selected empirically).

Espinosa *et al.* [71] discussed an alternative solution to solving Least Square Support Vector Machines using electrical load forecasting as an example application (note Suykens is a co-author for this publication). The goal was not to solve the LS-SVM in dual space, but rather in primal space using eigen value decomposition. The authors also stated that typically there are large data sets associated with these kinds of applications and using a sparse representation of the data could provide computational benefits. The entropy maximization was proposed as one possible technique for generating subsamples of the data. Using an RBF kernel function and cross-validation technique for parameter selection, the authors showed that the maximum MSE found was less than 3% for one hour and 24 hour ahead prediction. The authors present a more detailed version of [71] in [73].

General business applications of SVR span credit rating analysis to auto insurance market prediction to production rate prediction.

Similar to the work by Pai and Hong [65], Hsu *et al.* [72] described an alternative genetic algorithm based approach to selecting SVR parameters (GA-SVR). Using the real-valued genetic algorithm (RGA), They designed a GA-SVR using the same data as used in the EUNITE Network competition as described in Chang *et al.* [58, 62]. The GA was used to adaptively select the regularization parameter and the sigma value of the Gaussian kernel function. Using MAPE (the same metric used for the EUNITE competition), RMSE, and Max Error metrics, the authors showed that the use of a genetic algorithm to adaptively select the parameters of the SVR outperformed the winners of the EUNITE competition [58, 62].

He *et al.* [74] proposed a novel hybrid algorithm for short term electrical load forecasting. They proposed using an ARIMA model to estimate the linear portion of the electrical load time series data and an SVM to estimate the nonlinear residual, where the residual is the difference between the load data and the linear estimation. The underlying assumption was that the system model can be divided (equally) into a sum of a linear and non-linear representation. Using MAPE as the accuracy criteria, the single sample prediction horizon results were several percentage points better than the time series model by itself.

Machine Reliability Forecasting Applications Using Support Vector Regression for Time Series Prediction

Three papers are summarized below for the prediction of machine reliability from mechanical vibration time series signals, automotive related reliability measures, and engine reliability via prediction of MTBF using SVR. The prediction of machine reliability is typically non-linear and several traditional (ARIMA as an example) and ANN approaches have been studied regarding this application; however, the use of SVR for this particular application has not been widely studied.

Yang and Zhang [75] compared the use of an SVR and LS-SVM vs. a back propagation neural network (BPNN), an RBF network, and a GRNN for predicting vibration time series signals related to the mechanical condition of machinery. For short term prediction (one step ahead prediction), the SVR using a Gaussian kernel outperformed all of the other methods including the LS-SVM. For long term prediction (24 samples), the RBF network performed better with respect to the NMSE as compared to the two SVM methods. Hong *et al.* [76] discussed the use of SVMG and RSVMG [65] models for predicting the “period reliability ratio” for the automotive industry based on time series data containing vehicle damage incidents and the number of damages repaired. For one-step ahead forecasting, the RSVMG model outperformed ARIMA, BPNN, ICBPNN and SVMG (no feedback) methods with

SVR has been used for the prediction of environmental parameters such as air quality parameters rainfall estimation and detection, and weather forecasting.

respect to the RMSE. The key to this approach was the use of both a genetic algorithm and the use of feedback (recurrent network architecture) to aid in the selection of the free parameters of the SVR. Hong and Pai [77] compared the SVR to three other models (Duane, ARIMA, and GRNN) for the prediction of engine failure. The authors noted that the prediction of engine failure is critical in both the repair and design process of mechanical engines. The data set used as input was the engine age at the time of unscheduled maintenance actions and the outputs of the different models were the predicted engine age of the next unscheduled maintenance action per maintenance period. The authors noted that the use of SVR exceeds performance with respect to the NRMSE for all other models.

Control System and Signal Processing Applications Using Support Vector Regression for Time Series Prediction

There are several research papers using SVR for time series prediction in the fields of control systems and signal processing. These applications include: mobile position tracking, Internet flow control, adaptive inverse disturbance cancelling, narrowband interference suppression, antenna beamforming, elevator traffic flow prediction, and dynamically tuned gyroscope drift modeling. These diverse applications face the same nonlinear prediction challenges as all of the other applications described in this survey. In addition, some of these applications face an additional challenge of being highly sensitive to computation timing, as expected in real time signal processing applications. Summarized below are eight publications related to control theory and SVR time series prediction:

Suykens *et al.* [78] provided a detailed summary with real world (simplified) examples of non-linear control system theory using Least Squares Support Vector Machines (LS-SVM). Important discussion topics related to closed loop control theory such as local stability analysis were included. Several real world examples were given: state space estimation for non-linear system, inverted pendulum problem, and a ball and beam example.

Gezici *et al.* [79] proposed the use of SVR to improve the position estimation of users of wireless communications devices. Multi-path, non-line-of-sight propagation, and multiple access interference are the main sources of geo-location error. They proposed the use of a two step process to estimate the position of the mobile user. First, an SVR (ϵ -insensitive loss function and Gaussian kernel function) is used to predict an initial location. This process is followed by a Kalman-Bucy (K-B) filter to refine the geo-location. Although this application used the K-B filter for position estimation, it is not a specific time series prediction application (rather, a tracking problem). However, the processes

described in this paper could be used to predict time series data specifically.

Huang and Cheng [80] proposed two different algorithms for admission control and traffic scheduler schemes for internet web servers. The process of web client servicing is usually first come first serve, a technique that is not well suited to handle “bursty” loads, which, in turn, can negatively impact cost of internet sales vendors in the form of lost transactions. The authors proposed a prediction mechanism to forecast total maximum arrival rate and maximum average waiting time for priority groups using SVR and a fuzzy logic system. Using an event driven simulator, the authors showed significant increase in average throughput for two different priority task groups using SVR vs. the fuzzy logic system and the legacy first come first serve paradigm.

Liu *et al.* [81] discussed methods to control plant responses and plant disturbances, treated as separate process, using LS-SVM. The goal was to combine the plant output, which includes the plant disturbance, with the output of the LS-SVM (plant model approximation) to produce an estimate of the disturbance and feed back this estimate through an “inverse” LS-SVM to negate the disturbance via the input of the actual plant. For a non-linear modeled plant and a one-step-ahead prediction horizon, the authors successfully demonstrated the use of both SVR and an adaptive method for determining the free parameters of the SVM. The key aspect of this approach was the use of a Bayesian Evidence Framework for the adaptive selection of LS-SVM free parameters.

Yang and Xie [82] proposed the use of SVR to reduce the effects of high-power narrowband interference (NBI) in spread spectrum systems. Adaptive filters used to solve this problem were time-domain nonlinear LMS adaptive filters (TDAF) and frequency-domain nonlinear LMS adaptive filters (FDAF) which both have sensitivity to noise in estimating NBI. For this specific application, cross validation methods were too time-costly to train the SVR and to determine the SVR free parameters. Using a Gaussian kernel function, the authors noted that NBI suppression using SVR is a viable approach for NBI suppression where computational time is a more critical aspect of this application.

Ramon *et al.* [83] used an SVR approach to adaptively change antenna beam patterns (beamforming) in the presence of interfering signals arriving at an arbitrary angle of arrival. This particular application requires the use of complex variables (real and imaginary components of the objective function associated with the signal weighting for the individual antenna elements) for the solution which required separate Lagrange multipliers for the real and imaginary components of the solution. Because this is an adaptive beam forming problem, there is also a computational time constraint. The authors used an alternative optimization method: the iterative reweighted least squares (IWRRLS). Using a modified cost function (quadratic for data and linear for “outliers”), the authors demonstrated a

significant decrease in bit error rate (BER) as compared to a minimum mean square error based algorithm.

Luo *et al.* [84] proposed the use of an LS-SVM for the prediction of elevator traffic flow. ANNs have been used to study this problem and the LS-SVM was used here to improve the control system's ability to predict traffic flow in order to improve elevator service quality. Using three different groups of elevator traffic data, the authors demonstrated the feasibility of the LS-SVM for predicting traffic flow. There is a significant computational tradeoff between the sparseness of the LS-SVM solution compared to a standard SVR using other non-quadratic loss functions and the computational complexity associated with the training of the LS-SVM.

Xu *et al.* [85] compared the use of an SVR using accumulated generated operation (AGO) based on grey theory to an RBF neural network, a grey model, and a standard SVR to predict the drift of a dynamically tuned gyroscope. The AGO algorithm was used to pre-process the drift data in order to reduce noise and complexity of the original data set. Then, the SVM was trained and an inverse AGO algorithm (IAGO) was applied after the SVM training to compute the model. A B-spline kernel function was used for this application. As compared to the RBF network, the AGO-SVM approach showed superior performance in both the MAE and NMSE by almost an order of magnitude.

Miscellaneous Applications Using Support Vector Regression for Time Series Prediction

There are eight other research papers describing the use of SVR for time series prediction that were not specifically associated with any of the discussed categories in this survey. One paper pertains to a biological neuron application, two papers describe the use of SVR to Kalman filtering methods, an application of switching dynamics associated with unsupervised segmentation, two papers on SVM application to natural gas load forecasting, transportation travel time estimation, and the use of particle swarm optimization (PSO) used in conjunction with SVR. The references [87, 88, 89] do not actually use SVR to directly solve a time series prediction problem, but rather embed the use SVR into their respective approaches. The work presented in the last paper is not specifically a financial time series prediction application, although financial time series data was used to evaluate an alternative SVM training methodology. These eight publications are summarized below.

Frontzek *et al.* [86] used SVR to learn the dynamics of biological neurons of Australian crayfish. To model this biological neural network, they used time series data of a pyloric dilator neuron and SVR, with the ϵ -insensitive loss function and a Gaussian kernel function, for one-step ahead prediction of these time series data and compared results to an RBF network. The authors concluded that the Gaussian kernel function outperformed other kernels, the SVR approach "learned" faster than the RBF networks, and more data points (i.e. support vectors) produced better results.

Ralaivola and d'Alche-Buc [87] discussed the modeling of non-linear dynamic systems using SVR in conjunction with Kalman filters. The discussion is based on the transformation of the non-linear time series equation into a linear equation by the use of kernel functions. The authors proposed the use of SVR to map the transformed data from the feature space back into the input space, noting that they use one SVR for each dimension of the input space (with the kernel transformed data as inputs to the SVR). Using both a one-step-ahead prediction horizon as well as a 100 sample prediction horizon for Mackey-Glass time series data and laser time series data from the Santa Fe competition (see [87] for details), the authors showed results using both polynomial and Gaussian kernel functions and state that this approach could be comparable to other Kalman filtering approaches processes such as Extended Kalman Filters (EKF) or Unscented Kalman Filters (UKF).

Ralaivola and d'Alche-Buc [88] extended their work from [87] and proposed the Kernel Kalman Filter (KKF) where the non-linear input space is transformed to a higher dimension feature space by the use of kernels described in [87]. This Kalman filtering method was used for both the filtering and smoothing functions. The authors proposed the use of an Expectation-Maximization (EM) approach to determine the free parameters of the model, including the kernel parameters. Using Mackey-Glass data, Ikeda series data (laser dynamics), and Lorenz attractor data, the authors state the KKF accuracy in the RMSE sense are superior to the MLP and SVR prediction methods for both one-step-ahead prediction as well as multiple step ahead prediction.

Chang *et al.* [89] proposed applying an SVR to an unsupervised learning problem, specifically the unsupervised segmentation of time series data. Unsupervised segmentation can be applied to many time series applications such as speech recognition, signal classification, and brain data. The authors used the SVR as one component of a "competing" SVM architecture which is based on an "annealed competition of experts" (ACE) methodology. They were specifically, predicting weighting coefficients based on error terms. Simulated chaotic time series data, Mackey-Glass data, and Santa-Fe data were used as input to this methodology and the proposed architecture appears feasible for the prediction of non-linear time series data.

Liu *et al.* [90] applied SVR to natural gas load forecasting including factors related to natural gas loading such as weather related parameters (i.e., temperature, etc.), day of the week, holidays, etc. The results of using SVR to predict seven day ahead load forecasts were compared to a multi-layer perceptron ANN using a self-organizing feature map (SOFM) and the SVR outperformed the hybrid ANN approach by several percentage points in the MAPE. The same authors examined natural gas load forecasting [91] using a Least Squares Support Vector Machine and compared to an ANN using SOFM. The LS-SVM implementation had similar performance characteristics as found in [90]. A commercial software package (Natural Gas Pipeline Simulation and Load Forecasting – NGPSLF) based

The prediction of machine reliability is typically non-linear and several traditional (ARIMA as an example) and ANN approaches have been studied regarding this application; however, the use of SVR for this particular application has not been widely studied.

on LS-SVM was developed and implemented specifically for this application.

Wu *et al.* [92] used an SVR to analyze and predict travel time for highway traffic. Travel time prediction is essential for travel information systems, including the estimation of en-route times. Using a Gaussian kernel function and a standard SVR implementation, their SVR showed improved RME and RMSE results as compared to two other travel time prediction methods: current travel time prediction and historical mean prediction methods.

Zhang and Hu [93] proposed the use of Particle Swarm Optimization (PSO) for selecting certain features of data to reduce the inputs to an SVM (essentially data pruning). Also, the PSO was used to optimize the SVM free parameters as well. Using a financial time series data set as input (CBOT-US), the authors showed that the PSO feature selection procedure was comparable to other genetic feature selection algorithms in terms of minimizing the prediction error. The main advantage demonstrated with this approach is the great improvement in computation time as compared to the other methodologies.

Discussion

In the wide spectrum of time series prediction applications using SVR techniques, the fundamental reason for considering SVR as an approach for time series prediction is the non-linear aspect of the prediction problem. This non-linear aspect of the applications is common throughout all of the discussed applications. There are several broad observations and generalizations that can be made based on the brief summaries presented in this paper.

Traditional (and more sophisticated) model-based techniques generally do not perform as well as the SVR in predicting time series generated from non-linear systems. This is based on the fact that the SVR “lets the data speak for itself” whereas the model-based techniques typically can not model the non-linear processes well.

- ❑ Traditional Artificial Neural Network (ANN) based techniques such as Multi-Layer Perceptrons do not necessarily perform as well as the SVR. This can be due to their inherent limitation in not being able to guarantee a global minimum for the optimization of the network. By design, the SVR guarantees this global minimum solution and are typically superior in their ability to generalize.
- ❑ There is no predetermined heuristic for the choice of several parameters and designs for the SVR – it appears to be very application specific (as well as individual designer specific). This appears to be one of the largest challenges in

maturing this technology, as many of the papers presented offer different approaches for improving SVR performance through the adaptation of free parameters.

- ❑ There are several choices for solving the convex optimization problem inherent in the solution of the SVM.
- ❑ There are no measures of prediction uncertainty (i.e. covariance) associated with the predictions (note Relevance Vector Machines address this issue, see [105] as example).

For further information regarding this (and other) SVR based techniques, the authors encourage readers to start with the introductory papers, texts, and especially the publicly available websites [94–104]. These websites serve a valuable purpose in advancing and disseminating this viable technology to the scientific community and can act as a viable resource and database for current SVM/SVR applications.

Challenges Associated with Using SVR for Time Series Prediction

From the technical literature reviewed in this paper, there appears to be several challenges associated with the use of SVR in prediction of highly non-linear time series prediction applications. Below is a summary of some of these technical challenges and issues:

- 1) *Selection of kernel function:* The choice of kernel function appeared somewhat arbitrary, although the vast majority of the applications listed above use the Gaussian kernel. Some efforts empirically determine that the use of the Gaussian kernel is superior to other kernel functions, but in general, there appears to be no formal proof of optimality.
- 2) *Free parameter selection:* Some research has been done with respect to adaptively changing the free parameters associated with SVR training to improve prediction results, including the use of sophisticated genetic-based algorithms. Again, there is no “optimal” method for adaptation of the SVR parameters.
- 3) *Use of SVR in “real time” applications:* For the vast amount of the applications mentioned in this paper, all but two of them required some sort of “real time” computational demands. There is very little mention of the computational cost of deriving the results, most likely due to the static nature of the datasets being analyzed.
- 4) *Managing the trade space complexity of technical advantages:* The technical tradeoffs and nuances between the design of the SVR system, the sparseness of the solution, the accuracy of the solution, and the computational efficiency in finding the solution have not been summarily defined in any of the papers reviewed. Several of these aspects have been analyzed together, but not all of them as a whole.
- 5) *Selection of SVR optimization techniques:* Several QP optimization packages exist (publicly available) to train the SVR and the SMO algorithm appears to be the most

popular. The reader is referred to the web based references of this survey paper for more information on a selected set of training methodologies.

- 6) *Determining when to use the Least Squares SVM technique:* LS-SVM approaches are sometimes more efficient to implement at the expense of sparseness of the solution. LS-SVM did not always outperform the SVR approaches in some of the listed applications.
- 7) *Selection of performance metrics and benchmarks:* There are no sets of metrics and benchmarks for SVR approaches, although several publicly available data sets are used to compare performances. RMSE and MAPE are the most typical metric for goodness of the solution.

Conclusion

Support Vector Machines/Support Vector Regression (SVR) are powerful learning mechanisms that have been developed and matured over the last 15 years. They provide a method for predicting and forecasting time series for a myriad of applications including financial market forecasting, weather and environmental parameter estimation, electrical utility loading prediction, machine reliability forecasting, various signal processing and control system applications, and several other applications detailed in this survey paper. Non-traditional time series prediction methods are necessary for these types of applications due to the highly non-linear aspects of the data and processes.

In conclusion, SVR research continues to be a viable approach in the prediction of time series data in non-linear systems. Many methods and alternatives exist in the design of SVRs and a great deal of flexibility is left to the designer in its implementation. This survey presents a summary of these methods with their associated applications' papers references.

References

- [1] S. J. Orfanidis, *Optimum Signal Processing: An Introduction*, 2nd ed. New York: MacMillan, 1988.
- [2] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Eng., ser. D*, vol. 82, pp. 35–45, 1960.
- [3] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
- [4] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
- [5] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
- [6] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sept. 1999.
- [7] H. Drucker, C. C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Adv. Neural Inform. Process. Syst.*, no. 9, pp. 155–161, 1997.
- [8] B. Schölkopf, A. J. Smola, and C. Burges, *Advances in Kernel Methods—Support Vector Learning*. Cambridge, MA: MIT Press, 1999.
- [9] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, June 1999.
- [10] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. Singapore: World Scientific, 2002.
- [11] K.-R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Predicting time series with support vector machines," in *Proc. Int. Conf. on Artificial Neural Networks*, Springer, 1997.
- [12] S. Mukherjee, E. Osuna, and F. Girosi, "Nonlinear prediction of chaotic time series using support vector machines," in *Proc. 1997 IEEE Workshop – Neural Networks for Signal Processing VII*, Sept. 24–26, 1997, pp. 511–520.
- [13] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," Royal Holloway College, London, UK, NeuroCOLT Tech. Rep., 1998.
- [14] N. de Freitas, M. Milo, P. Clarkson, M. Niranjan, and A. Gee, "Sequential support vector machines," in *Proc. 1999 IEEE Signal Processing Society Workshop – Neural Networks for Signal Processing IX*, Aug. 23–25, 1999, pp. 31–40.
- [15] S. Rüping, "SVM kernels for time series analysis," CS Dept., AI Unit, University of Dortmund, Dortmund, Germany, 2001.
- [16] K. L. Lee and S. A. Billings, "Time series prediction using support vector machines, the orthogonal and regularized orthogonal least-squares algorithms," *Int. J. Syst. Sci.*, vol. 33, no. 10, pp. 811–821, 2002.
- [17] L. Cao and Q. Gu, "Dynamic support vector machines for non-stationary time series forecasting," *Intell. Data Anal.*, vol. 6, no. 1, pp. 67–83, 2002.
- [18] J.-Y. Zhu, B. Ren, H.-X. Zhang, and Z.-T. Deng, "Time series prediction via new support vector machines," in *Proc. 1st Int. Conf. on Machine Learning and Cybernetics*, Nov. 2002, vol. 1, pp. 364–366.
- [19] S. Rüping and K. Morik, "Support vector machines and learning about time," *2003 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003)*, Apr. 6–10, 2003, vol. 4, pp. IV-864–IV-867.
- [20] L. J. Cao, "Support vector machine experts for time series prediction," *Neurocomputing*, vol. 51, pp. 321–339, Apr. 2003.
- [21] Y. Mei-Ying and W. Xiao-Dong, "Chaotic time series prediction using least squares support vector machines," *Chin. Phys.*, vol. 13, no. 4, pp. 454–458, Apr. 2004.
- [22] J. M. Górriz, C. G. Puntonet, M. Salmerón, R. Martín-Clemente, and S. Hornillo-Mellado, "Using confidence interval of a regularization network," in *Proc. 12th IEEE Mediterranean Electrotechnical Conf. (MELECON 2004)*, May 12–15, 2004, pp. 343–346.
- [23] A. Hornstein and U. Parlitz, "Bias reduction for time series models based on support vector regression," *Int. J. Bifurcation Chaos*, vol. 14, no. 6, pp. 1947–1956, 2004.
- [24] J. M. Górriz, C. G. Puntonet, M. Salmerón, and J. J. G. de la Rosa, "A new model for time series forecasting using radial basis functions and exogenous data," *Neural Comput. Appl.*, vol. 13, no. 2, pp. 101–111, June 2004.
- [25] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.
- [26] Y.-F. Deng, X. Jin, and Y.-X. Zhong, "Ensemble SVR for prediction of time series," in *Proc. 4th Int. Conf. on Machine Learning and Cybernetics*, Aug. 18–21, 2005, pp. 3528–3534.
- [27] J. An, Z.-O. Wang, Q. Yang, and Z. Ma, "A SVM function approximation approach with good performances in interpolation and extrapolation," in *Proc. 4th Int. Conf. on Machine Learning and Cybernetics*, Aug. 18–21, 2005, pp. 1648–1653.
- [28] T. B. Trafalis and H. Ince, "Support vector machine for regression and applications to financial forecasting," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. on Neural Networks 2000 (IJCNN 2000)*, vol. 6, July 24–27, 2000, pp. 348–353.
- [29] F. E. H. Tay and L. J. Cao, "Application of support vector machines in financial time series forecasting," *Omega*, vol. 29, pp. 309–317, 2001.
- [30] T. Van Gestel, J. A. K. Suykens, D.-E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor, and J. Vandewalle, "Financial time series prediction using least squares support vector machines within the evidence framework," *IEEE Trans. Neural Netw.*, vol. 12, no. 4, pp. 809–821, July 2001.
- [31] F. E. H. Tay and L. J. Cao, "Improved financial time series forecasting by combining support vector machines with self-organizing feature map," *Intell. Data Anal.*, vol. 5, no. 4, pp. 339–354, 2001.
- [32] F. E. H. Tay and L. J. Cao, "Modified support vector machines in financial time series forecasting," *Neurocomputing*, vol. 48, pp. 847–861, Oct. 2002.
- [33] F. E. H. Tay and L. J. Cao, "ε-descending support vector machines for financial time series forecasting," *Neural Process. Lett.*, vol. 15, no. 2, pp. 179–195, 2002.
- [34] H. Yang, I. King, and L. Chan, "Non-fixed and asymmetrical margin approach to stock market prediction using support vector regression," in *Proc. 9th Int. Conf. on Neural Information Processing (ICONIP '02)*, Nov. 18–22, 2002, vol. 3, pp. 1398–1402.
- [35] H. Yang, L. Chan, and I. King, "Support vector machine regression for volatile stock market prediction," in *Proc. 3rd Int. Conf. on Intelligent Data Engineering and Automated Learning (IDEAL'02)*, Springer-Verlag, 2002, pp. 391–396.
- [36] A. Abraham, N. S. Philip, and P. Saratchandran, "Modeling chaotic behavior of stock indices using intelligent paradigms," *Int. J. Neural, Parallel, Scientific Computat.*, vol. 11, no. 1/2, pp. 143–160, 2003.
- [37] H. Yang, "Margin variations in support vector regression for the stock market prediction", Ph.D. dissertation, Chinese Univ. of Hong Kong, June 2003.
- [38] P. Ongsitrakul and N. Soonthornphais, "Apply decision tree and support vector regression to predict the gold price," in *Proc. Int. Joint Conf. on Neural Networks – 2003*, July 20–24, 2003, vol. 4, pp. 2488–2492.
- [39] L. J. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1506–1518, Nov. 2003.
- [40] Y. Liang and Y. Sun, "An improved method of support vector machine and its application to financial time series forecasting," *Prog. Natural Sci.*, vol. 13, no. 9, pp. 696–700, 2003.
- [41] A. Abraham and A. A. Yeung, "Integrating ensemble of intelligent systems for modeling stock indices," *Lect. Notes Comput. Sci.*, vol. 2687, pp. 774–781, 2003.
- [42] K.-J. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, pp. 307–319, 2003.
- [43] Y. Bao, Y. Lu, and J. Zhang, "Forecasting stock price by SVMS regression," *Lect. Notes Comput. Sci.*, vol. 3192, pp. 295–303, 2004.
- [44] H. Yang, K. Huang, L. Chan, I. King, and M. R. Lyu, "Outliers treatment in support vector regression for financial time series prediction," in *Proc. 11th Int. Conf. on Neural Information Processing (ICONIP 2004)*, Lecture Notes in Computer Science, 2004, vol. 3316, pp. 1260–1265.
- [45] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Comput. Operat. Res.*, vol. 32, no. 10, pp. 2513–2522, Oct. 2005.
- [46] Y.-K. Bao, Z.-T. Liu, L. Guo, and W. Wang, "Forecasting stock composite index by fuzzy support vector machines regression," in *Proc. 4th Int. Conf. on Machine Learning and Cybernetics*, Aug. 18–21, 2005, pp. 3535–3540.

- [47] D.-Z. Cao, S.-L. Pang, and Y.-H. Bai, "Forecasting exchange rate using support vector machines," in *Proc. 4th Int. Conf. on Machine Learning and Cybernetics*, Aug. 18–21, 2005, pp. 3448–3452.
- [48] T. Z. Tan, C. Quek, and G. S. Ng, "Brain-inspired genetic complimentary learning for stock market prediction," in *Proc. 2005 IEEE Congress on Evolutionary Computation*, Sept. 2–5, 2005, vol. 3, pp. 2653–2660.
- [49] D. C. Sansom, T. Downs, and T. K. Saha, "Evaluation of support vector machine based forecasting tool in electricity price forecasting for australian national electricity market participants," presented at the Australian Universities Power Engineering Conf., 2002.
- [50] Z. Huang, H. Chen, C.-J. Hsu, W.-H. Chen, and S. Wu, "Credit rating analysis with support vector machines and neural networks: A market comparative study," *Decision Support Syst.*, vol. 37, pp. 543–558, 2004.
- [51] Y. Hur and S. Lim, "Customer churning prediction using support vector machines in online auto insurance service," in *Proc. 2nd Int. Symp. on Neural Networks (ISNN 2005)*, Lecture Notes in Computer Science, May 30–June 1, 2005, vol. 3497, pp. 928–933.
- [52] I. Bose and R. Pal, "Using support vector machines to evaluate financial fate of dotcoms," in *Proc. Pacific Asia Conf. on Information Systems 2005*, July 7–10, 2005, pp. 521–528.
- [53] P.-F. Pai and C.-S. Lin, "Using support vector machines to forecast the production values of the machinery industry in Taiwan," *Int. J. Adv. Manuf. Technol.*, vol. 27, no. 1/2, pp. 205–210, Nov. 2005.
- [54] W. Lu, W. Wang, A. Y. T. Leung, S.-M. Lo, R. K. K. Yuen, Z. Xu, and H. Fan, "Air pollutant parameter forecasting using support vector machines," in *Proc. 2002 Int. Joint Conf. on Neural Networks (IJCNN '02)*, May 12–17, 2002, vol. 1, pp. 630–635.
- [55] T. B. Trafalis, B. Santosa, and M. B. Richman, "Prediction of rainfall from WSR-88D radar using kernel-based methods," *Int. J. Smart Eng. Syst. Design*, vol. 5, no. 4, pp. 429–438, Oct.–Dec. 2003.
- [56] W. Wang, Z. Xu, and J. W. Lu, "Three improved neural network models for air quality forecasting," *Eng. Computat.*, vol. 20, no. 2, pp. 192–210, 2003.
- [57] H. Prem and N. R. Srinivasa Raghavan, "A support vector machine based approach for forecasting of network weather services," *J. Grid Computing*, vol. 4, no. 1, pp. 89–114, Mar. 2006.
- [58] M.-W. Chang, B.-J. Chen, and C.-J. Lin, "EUNITE network competition: Electricity load forecasting," Nov. 2001.
- [59] M. Mohandes, "Support vector machines for short-term load forecasting," *Int. J. Energy Res.*, vol. 26, no. 4, pp. 335–345, Mar. 2002.
- [60] D. C. Sansom and T. K. Saha, "Energy constrained generation dispatch based on price forecasts including expected values and risk," in *Proc. IEEE Power Energy Society General Meeting 2004*, June 6–10, 2004, vol. 1, pp. 261–266.
- [61] L. Tian and A. Noore, "A novel approach for short-term load forecasting using support vector machines," *Int. J. Neural Syst.*, vol. 14, no. 5, pp. 329–335, Aug. 2004.
- [62] B.-J. Chen, M.-W. Chang, and C.-J. Lin, "Load forecasting using support vector machines: A study on EUNITE competition 2001," *IEEE Trans. Power Syst.*, vol. 19, no. 4, pp. 1821–1830, Nov. 2004.
- [63] B. Dong, C. Cao, and S. E. Lee, "Applying support vector machines to predict building energy consumption in tropical region," *Energy Buildings*, vol. 37, no. 5, pp. 545–553, May 2005.
- [64] Z. Bao, D. Pi, and Y. Sun, "Short term load forecasting based on self-organizing map and support vector machine," *Lect. Notes Comput. Sci.*, vol. 3610, pp. 688–691, 2005.
- [65] P.-F. Pai and W. C. Hong, "Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms," *Electric Power Syst. Res.*, vol. 74, no. 3, pp. 417–425, 2005.
- [66] Y. Ji, J. Hao, N. Reyhani, and A. Lendasse, "Direct and recursive prediction of time series using mutual information selection," in *Proc. IWAAN 2005*, June 8–10, 2005, pp. 1010–1017.
- [67] M.-G. Zhang, "Short-term load forecasting based on support vector machine regression," in *Proc. 4th Int. Conf. on Machine Learning and Cybernetics*, Aug. 18–21, 2005, vol. 7, pp. 4310–4314.
- [68] X. Li, C. Sun, and D. Gong, "Application of support vector machine and similar day method for load forecasting," *Lect. Notes Comput. Sci.*, vol. 3611, pp. 602–609, 2005.
- [69] P.-F. Pai and W.-C. Hong, "Support vector machines with simulated annealing algorithms in electricity load forecasting," *Energy Conv. Manage.*, vol. 46, no. 17, pp. 2669–2688, Oct. 2005.
- [70] H.-S. Wu and S. Zhang, "Power load forecasting with least squares support vector machines and chaos theory," in *Proc. Int. Conf. on Neural Networks and Brain*, Oct. 13–15, 2005, vol. 2, pp. 1020–1024.
- [71] M. Espinoza, J. A. K. Suykens, and B. De Moor, "Load forecasting using fixed-size least squares support vector machines," *Lect. Notes Comput. Sci.*, vol. 3512, pp. 1018–1026, 2005.
- [72] C.-C. Hsu, C.-H. Wu, S.-C. Chen, and K.-L. Peng, "Dynamically optimizing parameters in support vector regression: An application of electricity load forecasting," in *Proc. 39th Annu. Hawaii Int. Conf. on System Sciences (HICSS 2006)*, Jan. 4–7, 2006, vol. 2, pp. 1–8.
- [73] M. Espinoza, J. A. K. Suykens, and B. De Moor, "Fixed-size least square support vector machines: A large scale application in electrical load forecasting," *Computat. Manage. Sci.*, vol. 3, no. 2, pp. 113–129, Apr. 2006.
- [74] Y. He, Y. Zhu, and D. Duan, "Research on hybrid ARIMA and support vector machine model in short term load forecasting," in *Proc. Int. Conf. on Intelligent Systems Design and Applications (ISDA '06)*, Oct. 2006, vol. 1, pp. 804–809.
- [75] J. Yang and Y. Zhang, "Application research of support vector machines in condition trend prediction of mechanical equipment," in *Proc. 2nd Int. Symp. on Neural Networks (ISNN 2005)*, Lecture Notes in Computer Science, May 30–June 1, 2005, vol. 3498, pp. 857–864.
- [76] W.-C. Hong, P.-F. Pai, C.-T. Chen, and P.-T. Chang, "Recurrent support vector machines in reliability prediction," *Lect. Notes Comput. Sci.*, vol. 3610, pp. 619–629, 2005.
- [77] W.-C. Hong and P.-F. Pai, "Predicting engine reliability using support vector machines," *Int. J. Adv. Manuf. Technol.*, vol. 28, no. 1/2, pp. 154–161, Feb. 2006.
- [78] J. A. K. Suykens, J. Vandewalle, and B. De Moor, "Optimal control by least squares support vector machines," *Neural Netw.*, vol. 14, no. 1, pp. 23–35, 2001.
- [79] S. Gezici, H. Kobayashi, and H. V. Poor, "A new approach to mobile position tracking," in *Proc. IEEE Sarnoff Symp. on Advances in Wired and Wireless Communications*, Mar. 11–12, 2003, pp. 204–207.
- [80] C.-J. Huang and C.-L. Cheng, "Application of support vector machines to admission control for proportional differentiated services enabled internet servers," in *2004 Proc. Int. Conf. on Hybrid Intelligent Systems*, Dec. 5–8, 2004, pp. 248–253.
- [81] X. Liu, J. Yi, and D. Zhao, "Adaptive inverse disturbance cancelling control system based on least square support vector machines," in *Proc. 2005 American Control Conf.*, June 8–10, 2005, pp. 2625–2629.
- [82] Q. Yang and S. Xie, "An application of support vector regression on narrow-band interference suppression in spread spectrum systems," *Lect. Notes Comput. Sci.*, vol. 3611, pp. 442–450, Aug. 2005.
- [83] M. Martínez Ramón, N. Xu, and C. G. Christodoulou, "Beamforming using support vector machines," *IEEE Antennas Wireless Propagat. Lett.*, vol. 4, pp. 439–442, 2005.
- [84] F. Luo, Y.-G. Xu, and J.-Z. Cao, "Elevator traffic flow prediction with least squares support vector machines," in *Proc. 4th Int. Conf. on Machine Learning and Cybernetics*, Aug. 18–21, 2005, pp. 4266–4270.
- [85] G. Xu, W. Tian, and Z. Jin, "An AGO-SVM drift modeling method for a dynamically tuned gyroscope," *Measure. Sci. Technol.*, vol. 17, no. 1, pp. 161–167, Jan. 2006.
- [86] T. Frontzek, T. N. Lal, and R. Eckmiller, "Learning and prediction of the nonlinear dynamics of biological neurons with support vector machines," in *Proc. Int. Conf. on Artificial Neural Networks (ICANN 2001)*, Lecture Notes in Computer Science, 2001, vol. 2130, pp. 390–398.
- [87] L. Ralaivola and F. d'Alché-Buc, "Dynamical modeling with kernels for nonlinear time series prediction," in *Proc. Neural Information Processing Systems (NIPS 2003)*, Dec. 8–13, 2003.
- [88] L. Ralaivola and F. d'Alché-Buc, "Nonlinear time series filtering, smoothing, and learning using the kernel Kalman filter," Université Pierre et Marie Curie, Paris, France, Tech. Rep., 2004.
- [89] M.-W. Chang, C.-J. Lin, and R. C.-H. Weng, "Analysis of switching dynamics with competing support vector machines," *IEEE Trans. Neural Netw.*, vol. 15, no. 3, pp. 720–727, May 2004.
- [90] H. Liu, D. Liu, G. Zheng, and Y. Liang, "Research on natural gas load forecasting based on support vector regression," in *Proc. 5th World Congress on Intelligent Control and Automation*, June 15–19, 2004, pp. 3591–3595.
- [91] H. Liu, D. Liu, Y.-M. Liang, and G. Zheng, "Research on natural gas load forecasting based on least squares support vector machine," in *Proc. 3rd Int. Conf. on Machine Learning and Cybernetics*, Aug. 26–29, 2004, pp. 3124–3128.
- [92] C.-H. Wu, J.-M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Trans. Intell. Transport. Syst.*, vol. 5, no. 4, pp. 276–281, Dec. 2004.
- [93] C. Zhang and H. Hu, "Using PSO algorithm to evolve an optimum input subset for a SVM in time series forecasting," in *Proc. 2005 IEEE Conf. on Systems, Man, and Cybernetics*, Oct. 10–12, 2005, vol. 4, pp. 3793–3796.
- [94] Support Vector Machines. [Online]. Available: <http://www.support-vector.net/index.html>
- [95] Kernel Machines. [Online]. Available: <http://www.kernel-machines.org/index.html>
- [96] International Neural Network Society. [Online]. Available: <http://www.inns.org/>
- [97] Neural Computation. [Online]. Available: <http://neco.mitpress.org/>
- [98] European Neural Network Society. [Online]. Available: <http://www.snn.ru.nl/enns/>
- [99] Asia Pacific Neural Network Assembly. [Online]. Available: <http://www.apnna.net/apnna>
- [100] Japanese Neural Network Society. [Online]. Available: <http://www.jnns.org/>
- [101] Journal of Artificial Intelligence Research. [Online]. Available: <http://www.jair.org/>
- [102] S. Rüping. [Online]. Available: <http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/index.html>
- [103] R. Collobert and S. Bengio, "SVM Torch: Support vector machines for large-scale regression problems," *J. Machine Learning Res.*, vol. 1, pp. 143–160, Sept. 2001.
- [104] K. Pelckmans, J. A. K. Suykens, T. Van Gestel, J. De Brabanter, L. Lukas, B. Hamers, B. De Moor, and J. Vandewalle. [Online]. Available: http://www.esat.kuleuven.be/sista/lssvmlab/tutorial/lssvmlab_paper0.pdf
- [105] M. E. Tipping, S. A. Solla, T. K. Leen, and K.-R. Müller, "The relevance vector machine," in *Advances in Neural Information Processing Systems 12*. Cambridge, MA: MIT Press, 2000.

