© IWA Publishing 2004 Journal of Hydroinformatics | 06.3 | 2004

EC-SVM approach for real-time hydrologic forecasting

Xinying Yu, Shie-Yui Liong and Vladan Babovic

ABSTRACT

This study demonstrates a combined application of chaos theory and support vector machine (SVM) in the analysis of chaotic time series with a very large sample data record. A large data record is often required and causes computational difficulty. The decomposition method is used in this study to circumvent this difficulty. The various parameters inherent in chaos technique and SVM are optimised, with the assistance of an evolutionary algorithm, to yield the minimal prediction error. The performance of the proposed scheme, EC-SVM, is demonstrated on two daily runoff time series: Tryggevælde catchment, Denmark and the Mississippi River at Vicksburg. The prediction accuracy of the proposed scheme is compared with that of the conventional approach and the recently introduced inverse approach. This comparison shows that EC-SVM yields a significantly lower normalised RMSE value of 0.347 for the Tryggevælde catchment runoff and 0.0385 for the Mississippi River flow compared to 0.444 and 0.2064, respectively, resulting from the conventional approach. A slight improvement in accuracy was obtained by analysing the first difference or the daily flow difference time series. It should be noted, however, that the computational speed in analysing the daily flow difference time series is significantly much faster than that of the daily flow time series.

Key words | chaotic technique, support vector machine, decomposition method, evolutionary algorithm

Xinying Yu
Shie-Yui Liong (corresponding author)
Department of Civil Engineering,
National University of Singapore,
10 Kent Ridge Crescent,
Singapore 119260
Tel: +65 68742155
E-mail: *cvelsy@nus.edu.sg*

Vladan Babovic
Tectrasys AG,
Sihleggstrasse 23,
883 Wollerau,
SZ,
Switzerland

INTRODUCTION

Evidence of chaotic behaviour has been observed in various studies on hydrological data. Taken's theorem (1981) showed that the time delay vector approach can be used to reconstruct the phase space of time series, which can subsequently be utilised for prediction. The approach outlined in the theorem, however, does not provide concrete guidance as to how to optimally choose the two key parameters, namely time delay t and embedding dimension d. Conventional techniques, for example, apply the average mutual information (AMI) and false nearest neighbours (FNN) to select τ and d. Abarbanel (1996) proposed that the time delay τ should be chosen when AMI occurs at its first minimum. With this τ value, the embedding dimension d is selected associated with the minimum false nearest neighbours. The prediction is then based on such embedding and utilises a local model through the interpolation of k nearest neighbours. The drawback with the above scheme is that the prediction accuracy may not be as desired since the values of d, t and k are not derived with the minimal prediction error in mind.

Recently Babovic *et al*. (2000) and Phoon *et al*. (2002), for example, circumvented the above mentioned difficulty. Search schemes (such as genetic algorithms) were implemented to define the optimal embedding structure: the optimal embedding structure is obtained when the least prediction error has been achieved. While Babovic *et al*. (2000) were concerned mainly with obtaining the optimal (d, τ, k) which yields the least prediction error, Phoon *et al*. (2002) searched for the optimal (d, τ, k) which yields a very small prediction error, if not the smallest, and yet retains the chaotic signature.

In this study the Support Vector Machine (SVM) (Vapnik 1992, 1995), is considered as the prediction tool in the chaotic time series analysis. SVM is a relatively new statistical learning technique. Due to its strong theoretical statistical basis, SVM has been demonstrated in various studies to be much more robust, particularly for noise mixed data, than the local model commonly used in traditional chaotic techniques. One problem associated with SVM, for example, is its difficulty in dealing with the large training record required in chaotic time series analysis. To overcome processing large data storage, a decomposition algorithm developed very recently (Platt 1999; Joachims 1999; Collobert & Bengio 2001) is demonstrated in this study. Associated parameters in SVM, together with the parameters describing the embedding structure of chaotic time series, are optimally determined in this study with an evolutionary algorithm. The proposed approach, EC-SVM, which couples SVM with an evolutionary algorithm and applies in a chaos-based reconstructed phase space is first described and then applied to two real river flow data: a daily runoff time series of the Tryggevælde catchment in Denmark and a daily flow time series of the Mississippi River.

TRADITIONAL CHAOTIC TECHNIQUES: BRIEF REVIEW

Many hydrological time series have been found to possess a chaotic signature (Jayawardena & Lai 1994; Sivakumar *et al*. 1998). One of the signatures of chaotic time series is its broadband power spectrum. The Fourier transform, as shown below, is used to determine the type of the spectrum:

$$F(s) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi xs}\mathrm{d}x \qquad (1)$$

where $f(x)$ is the original space, $F(s)$ is known as the frequency space and the power spectrum is the magnitude of $F(s)$, i.e. $H = |F(s)|^2$. H is a decreasing function as s increases. When s is larger than a certain large number, $H \simeq 0$; the signal is a limited band signal. However, when s is very high and H is still significantly different from zero, the signal is a broadband signal. Both stochastic and chaotic time series have a broadband power spectrum.

A low attractor dimension is another signature of a chaotic system while that of a stochastic system is infinite. Thus, the attractor dimension can be used as a measure to distinguish the chaotic from the stochastic time series. The basic idea of the determination of the attractor dimension was suggested by Grassberger & Procaccia (1983). The correlation dimension can be estimated from a given data set. If the time series provides a low correlation dimension, which normally is not an integer dimension as in Euclidean space, known as the fractal dimension, the time series can be regarded as being from a chaotic system. The embedding techniques of using lag vectors can reconstruct the phase space in common Euclidean space and the forecasting for such time series can be carried out.

Embedding theory (Takens 1981; Sauer *et al*. 1991) provides a scheme to detect the evolution of the system and to reconstruct the chaotic attractor. For a time series y_t, given a delay time τ, a time lag vector $\boldsymbol{Y}$ of d dimensions can be defined as

$$\boldsymbol{Y}_t = [y_t, y_{t-\tau}, \ldots, y_{t-(d-1)\tau}]. \qquad (2)$$

The embedding theory does not, however, suggest the time lag τ and the embedding dimension d. The commonly used techniques to determine the time lag τ and the dimension d are the average mutual information (AMI) and the false nearest neighbour (FNN), respectively.

The mutual information, $I(\tau)$, between $y(t)$ and its delay time series $y(t+\tau)$ can be determined for a given time series. The probabilities and joint probabilities can be estimated from the given data. $I(\tau)$ is greater than zero. As τ gets significantly larger, $I(\tau)$ tends to go to zero since the chaotic signals $y(t)$ and $y(t+\tau)$ become independent from each other. Thus, the τ value at the first minimum of $I(\tau)$ is commonly suggested to be chosen as the time lag (Fraser & Swinney 1986; Abarbanel 1996).

In practice, it is very common to choose d with the minimal false nearest neighbours. The basic idea of the false nearest-neighbour (FNN) method is that, if the embedding dimension is d, then the neighbour points in R^d are also the neighbour points in R^{d+1}. If this is not the case these points are then called false neighbour points. If the false neighbour points are negligible then this d can be chosen as the embedding dimension d (Kennel *et al*. 1992). The Euclidean distance can be used as a measure of the distance between the two neighbour points $\boldsymbol{Y}(t)$ and $\boldsymbol{Y}'(t)$ in R^d. Empirically, if the additional distance in R^{d+1} compared with the distance in R^d is larger than a threshold value of approximately 15, then we are in the presence of a false neighbour. This number, 15, is an experimental value. It may change due to the nature of the sample data set.

Since the lag vector $\boldsymbol{Y}$ can reconstruct the phase space, the evolution of y follows the evolution of the system. Short-term predictions in a chaotic system can be considered as a function of the lag vector $\boldsymbol{Y}$ as

$$y(t+1) = F(\boldsymbol{Y}(t)). \tag{3}$$

F is approximated by certain functions and is mainly a local model (Farmer & Sidorowich 1987). The local model considers a local function for each local region. This set of local functions builds up the approximation of F for the whole domain. Usually each region covers only a limited number of nearest-neighbour points in the data set. Abarbanel (1996) and Zaldívar *et al.* (2000), for example, have shown that a higher-order nonlinear model may not yield better results than a simple linear model due to the small number of sample size. The number of nearest-neighbour points, k, is commonly set as 2 times the embedding dimension plus 1 (Farmer & Sidorowich 1987) or the embedding dimension plus 1 (Abarbanel 1996).

There is no guarantee, however, that conventional embedding techniques as described above will perform well for all kinds of real-world noisy time series. Babovic *et al*. (2000) employed a genetic algorithm to search for the embedding parameters, time delay τ, embedding dimension d and the number of nearest neighbours k, based on the idea that good forecasting implies good embedding properties. Using this approach, the prediction for the validation set was improved by about 20–35% compared to the results using a conventional approach for water level prediction in Venice Lagoon, Italy. Phoon *et al*. (2002) proposed an inverse approach and applied it on the daily runoff time series of the Tryggevælde catchment. Data were divided into three subsets, instead of two, i.e. state space reconstruction set, calibration set and prediction set. The calibration set is used to ensure that the results are meaningful and self-consistent. Phoon *et al*. (2002) used a brute force method to search for the optimal set. The reported NRMSE was 0.369 with (d, τ, k) as (3,1,10) for the years 1992–1993. Liong *et al*. (2002) later extended the work of Phoon *et al*. (2002) by implementing a genetic algorithm (GA) to derive the optimal parameter set. The NRMSE was reduced to 0.361 with (d, τ, k) being equal to (2,1,11) for the years 1992–1993. These two studies show that the choice of k in the local linear model may influence the choice of embedding parameters.

A local linear model has been the norm in fitting the function F given in Equation (3). The reason is that local models can be implemented easily and more conveniently, especially for a large historical data set. The application of SVM as a global model, for example, has not been encountered in the chaotic hydrological time series analysis. The main reason is its difficulty in dealing with large data sets. This study thus shows the application of SVM as a forecasting tool (regression engine), operating in a chaotic-based phase space, on hydrological time series.

SUPPORT VECTOR MACHINE

The Support Vector Machine (SVM) is based on statistic learning theory and is an approximation implementation of the method of structural risk minimisation with a good generalisation capability. SVM has been proven to be a robust and efficient algorithm for both classification (Vapnik 1995) and regression (Drucker *et al*. 1997; Vapnik *et al*. 1997; Collobert & Bengio 2001). Recently, a decomposition method has also been successfully developed on a regression problem with a large data record; this makes SVM much more suitable, particularly for chaotic time series analysis.

SVM for regression

SVM converts an input $\boldsymbol{Y}$ into a feature space through a nonlinear function $\varphi(\boldsymbol{Y})$. The original complex nonlinear function, $F(\boldsymbol{Y})$, is converted into a linear function, $\boldsymbol{w}^T\varphi(\boldsymbol{Y})$, in the feature space. In the chaos technique, a key function is Equation (3), i.e. the relationship between the l-lead time prediction $y(t+l)$ and the lag vector $\boldsymbol{Y}(t)$. For a training set $\{\boldsymbol{Y}(t), y(t+l)_d\}$, $t=1,2,\ldots, N$, the task is to find the best fit function:

$$y(t+l) = F(\boldsymbol{Y}) = f(\varphi(\boldsymbol{Y})) = \sum\nolimits_{i=1}^{m} w_i\varphi_i(\boldsymbol{Y}) + b$$
$$= w^T\boldsymbol{\varphi}(\boldsymbol{Y}) + b. \qquad (4)$$

A typical loss function used in SVM is the ε-insensitive loss function $L_\varepsilon(y(t+l),\ y(t+l)_d)$ (Vapnik 1995), i.e. $L_\varepsilon(y,d) = |y-d| - \varepsilon$ for $|y-d| > \varepsilon$ and $L_\varepsilon(y,d)=0$ for $|y-d| \leq \varepsilon$. Based on the structural risk minimisation scheme, the objective is to minimise the empirical risk and to minimise $\|\boldsymbol{w}\|^2$ as shown below:

$$\text{Minimise: } \Phi(\boldsymbol{w},\xi,\xi') = C\sum\nolimits_{i=1}^{N}(\xi_i + \xi_i') + \frac{1}{2}\left\|\boldsymbol{w}\right\|^2$$
$$\text{subject to: } y_{(i+l)_d} - \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{Y}_i) - b \leq \varepsilon + \xi_i$$
$$\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{Y}_i) + b - y_{(i+l)_d} \leq \varepsilon + \xi_i'$$
$$\xi_i \geq 0,\ \xi_i' \geq 0,\ i = 1, 2, \ldots, N \qquad (5)$$

where ξ_i, ξ'_i are slack varibles. The above problem can be converted into a dual problem where the task is to optimise the Lagrangian multipliers, α_i and α'_i. The dual problem contains a quadratic objective function of α_i and α'_i with one linear constraint:

Maximise:

$$Q(\alpha,\alpha') = \sum\nolimits_{i=1}^{N} y_{(i+l)d}(\alpha_i - \alpha_i') - \varepsilon\sum\nolimits_{i=1}^{N}(\alpha_i + \alpha_i')$$
$$- \frac{1}{2}\sum\nolimits_{i,j=1}^{N}(\alpha_i - \alpha_i')(\alpha_j - \alpha_j')k(\boldsymbol{Y}_i, \boldsymbol{Y}_j)$$
$$\text{subject to: } \sum\nolimits_{i=1}^{N}(\alpha_i - \alpha_i') = 0$$
$$0 \leq \alpha_i,\alpha_i' \geq C,\ i = 1, 2, \ldots, N \qquad (6)$$

where $k(\boldsymbol{Y}_i,\boldsymbol{Y}_j) = \boldsymbol{\varphi}(\boldsymbol{Y}_i)^T\boldsymbol{\varphi}(\boldsymbol{Y}_j)$ is the inner-product kernel. The above problem can be stated as standard formulized qudratic programming $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x}$, where $\boldsymbol{H}$ is the Hessian matrix. Using

$$\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\alpha} \\ -\boldsymbol{\alpha}' \end{pmatrix} \quad \widetilde{\boldsymbol{K}} = \begin{pmatrix} \boldsymbol{K} & \boldsymbol{K} \\ \boldsymbol{K} & \boldsymbol{K} \end{pmatrix} \quad \boldsymbol{b} = \begin{pmatrix} -\boldsymbol{y} - \boldsymbol{1}\varepsilon \\ -\boldsymbol{y} + \boldsymbol{1}\varepsilon \end{pmatrix}$$

the dual problem can then be expressed as the following standard quadratic programming form, denoted as **OP1**:

$$\text{Minimise: } Q(\boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{\beta}^T\widetilde{\boldsymbol{K}}\boldsymbol{\beta} - \boldsymbol{\beta}^T\boldsymbol{b}$$
$$\text{subject to: } \beta^T\boldsymbol{1} = 0$$
$$0 \leq \delta_i\beta_i \leq C,\ i = 1, 2, \ldots, 2N \qquad (7)$$

where $\delta_i = 1$ for $1 \leq i \leq N$ and $\delta_i = -1$ for $N+1 \leq i \leq 2N$. Lagrangian multipliers α_i and α'_i are first solved through Equation (7) and $y(t+l)$ is solved by the following:

$$y(t+l) = \sum\nolimits_{i=1}^{N}(\alpha_i - \alpha_i')\varphi(\boldsymbol{Y}_i)^T\varphi(\boldsymbol{Y}) + b$$
$$= \sum\nolimits_{i=1}^{N}(\alpha_i - \alpha_i')k(\boldsymbol{Y}_i, \boldsymbol{Y}) + b \qquad (8)$$

The requirement for the kernel $k(\boldsymbol{Y}_i,\boldsymbol{Y})$ is to satisfy Mercer's theorem. Several functions have been shown to qualify as kernel functions. Studies such as Dibike *et al.* (2001) and Liong & Sivapragasm (2002) show the suitability of the Gaussian kernel function in hydrological time series. The Gaussian kernel function is expressed as follows:

$$k(\boldsymbol{Y}_i,\boldsymbol{Y}) = \mathbf{exp}(-\|\boldsymbol{Y}_i - \mathbf{Y}\|^2/2\sigma^2). \qquad (9)$$

The selection of appropriate values for the three parameters (C, ε, σ) in the above expressions has been proposed by various researchers. Cherkassky & Mulier (1998) suggested the use of cross-validation for the SVM parameter choice. Mattera & Haykin (1999) proposed the parameter C to be equal to the range of output values. They also proposed the selection of the ε value to be such that the percentage of support vectors in the SVM regression model is around 50% of the number of samples. Smola *et al.* (1998) assigned optimal ε values as proportional to the noise variance, in agreement with general sources on SVM. Cherkassky & Ma (2004) proposed the selection of ε parameters based on the estimated noise. Different approaches yield different values for the three parameters. As shown later, this study finds the optimal parameter set simultaneously by minimising the prediction error as the objective function.

Since SVM solves a quadratic programming, there is a unique optimal solution to this quadratic programming. The Karush–Khun–Tucker (KKT) condition is the necessary and sufficient condition for the optimal solution. The derivative of the Lagrangian is equal to zero at the optimal solution. Thus, the derivative is used for checking whether the algorithm has achieved the optimal solution.

Handling of large data records with SVM

SVM deals with solving a quadratic programming with one linear constraint and bound constraints. Even though this type of optimisation problem is well understood and algorithms are well developed, a serious obstacle is faced when it deals with a large training data set. The Hessian matrix, Equation (10), becomes tremendously large with increasing training sample size:

$$\boldsymbol{K} = \begin{bmatrix} k(\boldsymbol{Y}_1,\boldsymbol{Y}_1) & k(\boldsymbol{Y}_1,\boldsymbol{Y}_2) & \dots & k(\boldsymbol{Y}_1,\boldsymbol{Y}_N) \\ k(\boldsymbol{Y}_2,\boldsymbol{Y}_1) & k(\boldsymbol{Y}_2,\boldsymbol{Y}_2) & \dots & k(\boldsymbol{Y}_2,\boldsymbol{Y}_N) \\ \dots & \dots & \dots & \dots \\ k(\boldsymbol{Y}_N,\boldsymbol{Y}_1) & k(\boldsymbol{Y}_N,\boldsymbol{Y}_2) & \dots & k(\boldsymbol{Y}_N,\boldsymbol{Y}_N) \end{bmatrix}. \qquad (10)$$

For instance, a 20 years daily flow time series has about 7300 records. The Hessian matrix in **OP1** has the size of the square of 2 times the sample size, i.e. 213,160,000. If each element of the Hessian matrix is stored as an 8-byte double-precision number, the total memory capacity required is 1705 megabytes. Common PCs have a RAM size of 256 megabytes. Since the Hessian matrix is required to be stored, this requirement poses a serious problem for SVM to solve problems, such as chaotic time series analysis, requiring large training data records.

Most recently a decomposition method was developed to overcome the above-mentioned problem. This allows SVM to deal with the large data record problem. For classification problems, Platt (1999) developed sequential minimal optimisation (SMO) and Joachims (1999) developed SVM$^{\text{light}}$. For the regression problem, Collobert & Bengio (2001) successfully implemented the decomposition method in SVM$^{\text{light}}$.

The basic idea of the decomposition method is to decompose the quadratic programming into a small quadratic programming series of only 2 unknown variables while the remaining variables are fixed. The memory requirement is then decreased to be only linear to the sample size. Since a quadratic programming with 2 variables can be solved analytically, the whole algorithm becomes very efficient. The basic algorithm is as follows:

1. Set an initial value $\boldsymbol{\beta}^0$.
2. Select 2 working variables from $2N$ variables, e.g. β_1, β_2, among $\boldsymbol{\beta}$.
3. Solve the quadratic programming having only 2 variables analytically. $Q(\boldsymbol{\beta}^{k+1}) < Q(\boldsymbol{\beta}^k)$ is guaranteed.
4. Check the optimal conditions. If the KKT conditions are verified, the optimum is achieved; otherwise, go to step (2) and repeat the remaining steps.

The decomposition method splits the variables into a fixed set F and a working set S. Noting that

$$\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\beta}s \\ \boldsymbol{\beta}_F \end{pmatrix} \quad \widetilde{\boldsymbol{K}} = \begin{pmatrix} \widetilde{\boldsymbol{K}}ss & \widetilde{\boldsymbol{K}}_{SF} \\ \widetilde{\boldsymbol{K}}_{FS} & \widetilde{\boldsymbol{K}}_{FF} \end{pmatrix} \quad \boldsymbol{b} = \begin{pmatrix} \boldsymbol{b}s \\ \boldsymbol{b}_F \end{pmatrix} \qquad (11)$$

$\boldsymbol{\beta}s$ contains 2 variables, e.g. β_1, β_2, which are chosen as working variables among $\boldsymbol{\beta}$. **OP1** becomes

$$Q\left(\boldsymbol{\beta}\right) = \frac{1}{2}\boldsymbol{\beta}^T \widetilde{\boldsymbol{K}}\boldsymbol{\beta} - \boldsymbol{\beta}^T \boldsymbol{b} = \frac{1}{2}\boldsymbol{\beta}_S^T \widetilde{\boldsymbol{K}}ss\boldsymbol{\beta}s - \boldsymbol{\beta}_S^T \left(\boldsymbol{b}s - \widetilde{\boldsymbol{K}}_{SF}\boldsymbol{\beta}_F\right)$$

$$+ \boldsymbol{\beta}_F^T\widetilde{\boldsymbol{K}}_{FF}\boldsymbol{\beta}_F - \boldsymbol{\beta}_F^T\boldsymbol{b}_F. \qquad (12)$$

Denoting, $\boldsymbol{h} = \boldsymbol{b}s - \tilde{\boldsymbol{K}}_{SF}\beta_F$, it is equivalent to the following standard quadratic programming form **OP2**:

$$\text{Minimise: } Q(\boldsymbol{\beta}s) = \frac{1}{2}\boldsymbol{\beta}_s^T \widetilde{\boldsymbol{K}}ss\boldsymbol{\beta}s - \boldsymbol{\beta}_S^T \boldsymbol{h}$$

$$\text{subject to: } \boldsymbol{\beta}_S^T \boldsymbol{1} = -\boldsymbol{\beta}_F^T \boldsymbol{1}$$

$$0 \le \delta_i\beta_i \le C. \qquad (13)$$

The memory requirement is reduced from the square of the sample size, N^2, to 2 times the sample size, $2N$. Instead of requiring the storing the total large $\boldsymbol{K}$ matrix, the memory requirement in the decomposition method is to store only the components of the 2 lines corresponding to the 2 selected working variables.

$\boldsymbol{\beta}s$ contains only 2 variables: β_1, β_2. **OP2** can then be described as follows:

$$\boldsymbol{\beta}s = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} \quad \tilde{\boldsymbol{K}}ss = \begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix} \quad \zeta = -\boldsymbol{\beta}_F^T \boldsymbol{1} \tag{14}$$

$$\text{Minimise: } Q(\beta_1,\beta_2) = \frac{1}{2}(\beta_1\,\beta_2)\begin{pmatrix} k_{11} & k_{12} \\ k_{11} & k_{22} \end{pmatrix}\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} - (\beta_1\,\beta_2)\begin{pmatrix} h_1 \\ h_2 \end{pmatrix}$$
$$\text{subject to: } \beta_1 + \beta_2 = \zeta$$
$$0 \le \delta_1\beta_1, \delta_2\beta_2 \le C. \tag{15}$$

Using $\beta_2 = \zeta - \beta_1$, the above objective function becomes

$$\text{Minimise: } Q(\beta_1) = \frac{1}{2}(k_{11} - 2k_{12} + k_{22})\beta_1^2 + \left[(k_{12} - k_{22})\zeta - h_1 - k_2\right]\beta_1. \tag{16}$$

This is a simple quadratic program with one variable as the standard form: $f(x) = \frac{1}{2}ax^2 + bx$. $a = k_{11} - 2k_{12} + k_{22} > 0$ always holds for the Gaussian kernel. The function has a unique minimum when $\beta_1 = a/b$. The solution is dependent on the bound constraints of β_1. If a/b is inside the bound constraints, a/b is the solution of **OP2**. Otherwise the solution is one of the boundary points as shown in Figure 1.

Selecting 2 variables as the working variables among $2N$ variables gives a number of different choices. The total number of choices is

$$\frac{1}{2}C_{2N}^2 = \frac{1}{2}\frac{(2N)!}{(2N-2)!}. \tag{17}$$

Choosing a good working set is very essential to ensure a rapid convergence. Thus, an efficient and effective selection method is a key procedure in minimising the objective function $Q(\beta)$ step by step. The steepest feasible descent strategy is used to choose a good pair of working variables, which will guarantee that the variables chosen have the largest potential to decrease to the objective function.

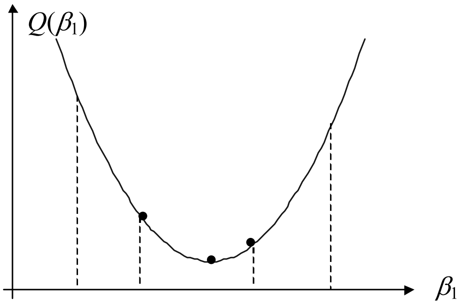

Figure 1 | Optimisation problem for a working set of 2 variables.

Figure 2(a) shows how the optimisation of the decomposition method progresses. $\boldsymbol{\beta}$ is varied within the feasible region as $\boldsymbol{\beta}$ approaches the optimum. At an iteration k, for example, $\boldsymbol{\beta}^k = (\beta_1^k, \beta_2^k, \beta_3^k, \ldots, \beta_{2N}^k)$ and only β_1^k and β_2^k are chosen as working variables. Thus, only β_1^k and β_2^k become $\beta_1^{k+1}$and β_2^{k+1} at iteration $(k+1)$ while the rest $\beta_3^k, \beta_4^k, \ldots, \beta_{2N}^k$ remain unchanged, i.e. $\boldsymbol{\beta}^{k+1} = (\beta_1^{k+1}, \beta_2^{k+1}, \beta_3^k, \ldots, \beta_{2N}^k)$. Denoting $\boldsymbol{d}$ as the difference between $\boldsymbol{\beta}^{k+1}$ and $\boldsymbol{\beta}^k$:

$$\boldsymbol{d} = \boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k \tag{18}$$

where $\boldsymbol{d}$ has only 2 nonzero components, i.e. $\boldsymbol{d} = (d_1, d_2, 0, \ldots, 0)$. Since the linear constraint $\boldsymbol{\beta}^T\mathbf{1} = 0$ must hold $(\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k)^T\mathbf{1} = 0$ is true, i.e. $\boldsymbol{d}^T\mathbf{1} = 0$ or $d_1 + d_2 = 0$. When the problem is projected into $\beta_1\beta_2$ space, the feasible region is a tangential line equal to $-1(\beta_1 + \beta_2 = \text{constant})$ and there are only 2 possible directions the solution points can move, as illustrated in Figure 2(b).

To choose a good set of working variables, the steepest feasible descent strategy is employed. The less the dot product of the gradient $\nabla Q(\boldsymbol{\beta})$ and $\boldsymbol{d}$ is, the closer $\boldsymbol{d}$ is to the negative gradient: this means that the working variables will reduce the objective function $Q(\boldsymbol{\beta})$ further. For instance, direction 2 in Figure 2(b) will be chosen from among the four possible directions. A good working set can be found by solving

$$\text{Minimise: } \nabla Q(\boldsymbol{\beta}^k)\boldsymbol{d}/\|\boldsymbol{d}\|. \tag{19}$$

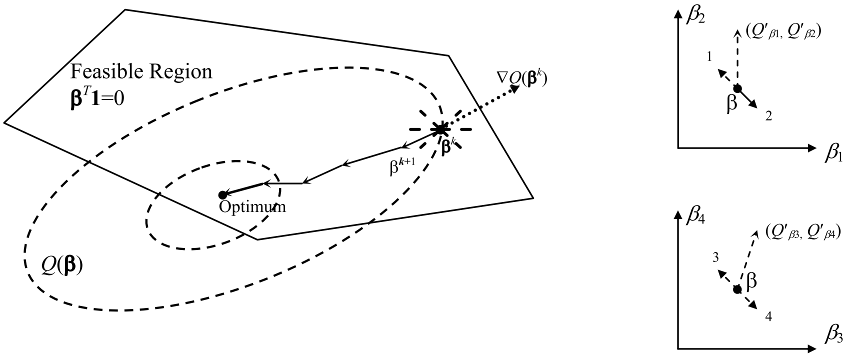

Figure 2 | Illustration of the decomposition algorithm.

Since $\boldsymbol{d}$ has only 2 nonzero components and $d_1 + d_2 = 0$, the above problem is reduced to

$$\text{Minimise: } \left(Q'_{\beta_1{}^k} - Q'_{\beta_2{}^k}\right) \Big/ \sqrt{2}. \tag{20}$$

This optimisation problem is with the objective to achieve a minimum $(Q'_{\beta1} - Q'_{\beta2})$. Therefore, the two working variables should be such that one variable (β_1) has the smallest first-order derivative $Q'_{\beta1}$ among the total $2N$ variables while the other variable (β_2) has the largest first-order derivative $Q'_{\beta2}$.

The decomposition method illustrated here is highly effective and efficient for a large scale training set due to the two key strategies employed in the algorithm: 2 working variables and the steepest feasible direction in selecting the 2 working variables. The problem is then decomposed into a series of quadratic programming steps, each having only 2 variables and 1 linear constraint. Thus, SVM equipped with the decomposition method could deal with the large data record requirement easily, such as that in the chaotic time series analysis. The software used in this study is SVMTorch II.

SVM APPLIED IN PHASE SPACE

In this study SVM is not applied on the original time series but on the series in the phase space. The appropriate embedding structure (d, τ) and the SVM parameters (C, ε, σ) are all determined simultaneously to minimise the prediction error. The search algorithm used in this study is the Shuffled Complex Evolution (Duan *et al.* 1992), which is described in the following section.

With a user-defined time lag, τ, and an embedding dimension, d, a phase space reconstruction is created. The lag vector is then used as the input while a l-lead time prediction is the desired output. The regression form used is as follows:

$$y^d_{t+l} = f\left(\boldsymbol{Y}_t\right) + v = f\left(y_t, y_{t-\tau}, \ldots, y_{t-(d-1)\tau}\right) + v. \tag{21}$$

The regression problem is concerned with deriving a regression function in the reconstructed phase space $\boldsymbol{Y}$. The regression problem is solved by SVM in this study.

Figure 3 demonstrates the process of deriving the optimal embedding structure. E_{test}, the root mean square error, is used as a measure for the optimal parameter set and is defined as

$$E_{test} = \sqrt{\frac{1}{N}\sum\nolimits_{t=1}^{N} \left(y(t+1)_p - \left(y(t+1)_o\right)^2\right.} \tag{22}$$

where $y(t+1)_p$ and $y(t+1)_o$ are the predicted and observed values, respectively.

SEARCH OF OPTIMAL PARAMETERS WITH EVOLUTIONARY ALGORITHM

In this study the coupling of SVM and the techniques inspired by chaos theory are proposed. The decomposition

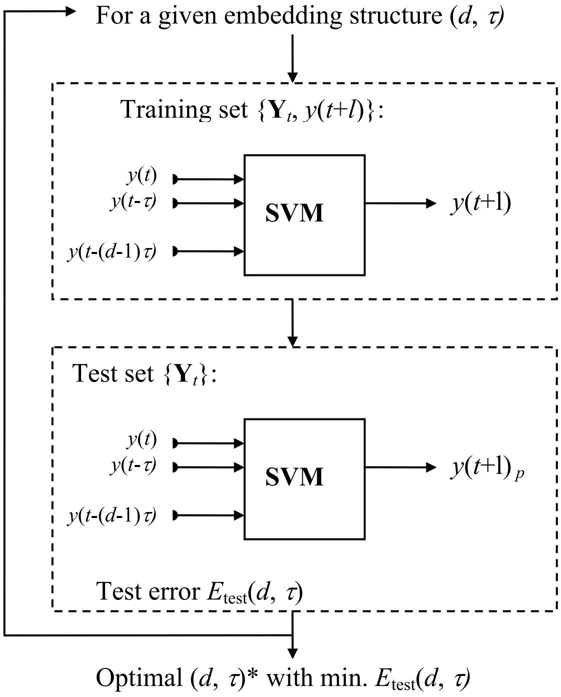
For a given embedding structure (d, τ)
Training set {Y_t, y(t+l)}:
y(t)
y(t-τ)
y(t-(d-1)τ)
SVM
y(t+1)
Test set {Y_t}:
y(t)
y(t-τ)
y(t-(d-1)τ)
SVM
y(t+1)_p
Test error E_test(d, τ)
Optimal (d, τ)* with min. E_test(d, τ)

Figure 3 | Algorithm for the optimal algorithm used in optimal embedding structure search.

method serves as the inner key module for the regression problem. There are a total of five parameters to be determined in this approach. They are the time delay (τ, the embedding dimension (d) and the three SVM parameters (C, ε, σ). These five parameters are optimised simultaneously.

The parameter search scheme used in this study is the Shuffled Complex Evolution (SCE) algorithm (Duan *et al.* 1992). SCE combines the simplex procedure of Nelder and Mead with concepts of a controlled random search. SCE is a category of evolution algorithm. It begins with a population of points sampled from the feasible space. The population is partitioned into n parts, known as complexes. Each complex has $2n+1$ individuals. Each complex evolves for $2n+1$ steps separately, after which the population is shuffled. Each evolution of the complex generates a new offspring by using the operations of selection, reflection, contraction and mutation. Selection is based on fitness and good individuals have higher probabilities to be chosen. The offspring generated replace the worst points in the complex. If the offspring generated by reflection fail to be better than the worst individual, then a contraction offspring is generated. If the contraction offspring fail then a random offspring is generated. There are three stopping criteria used in SCE: (1) the population is converged into a small zone, e.g. 0.001 of the search space; (2) the change of objective function is ignorable, e.g. less than 0.0001 during the last 10 generations, for example; and/or (3) the number of evolutions exceeds a user-specified number.

In this study, the chromosomes in EC-SVM comprise the following genes (d, τ, C, ε, σ). The fitness measure is the prediction error of the test set. For a given chromosome, the test set error can be calculated following the algorithm as shown in Figure 3.

The proposed EC-SVM algorithm can be summarized as follows:

1. Generate initial populations in the feasible parameter space. The number of populations $= n \times (2n+1)$, where n is the number of parameters.
2. Calculate the fitness value of each chromosome, E_{test}, resulting from vectors in the phase space for a given embedding structure (d, τ) for a given SVM's parameters (C, ε, σ) and for a user-defined prediction horizon.
3. Rank each chromosome in descending order.
4. Distribute the orderly ranked chromosomes into n numbers of complexes in such a manner that the first complex receives the first rank chromosome, the second complex receives the second rank chromosome, etc. Thus, each complex contains $2n+1$ chromosomes.
5. Evolve each complex for $2n+1$ evolutions. Each evolution removes the worst point in the complex either by reflection, contraction or mutation, in that order.
6. Check the stopping criteria. The algorithm ceases when one of the stopping criteria is met; otherwise, go to step (3) and continue with the rest of the steps.

Figure 4 illustrates the EC-SVM scheme when it is applied to a univariate time series in the phase space reconstruction. In complex nonlinear dynamical systems, where multivariate time series may have to be considered for a

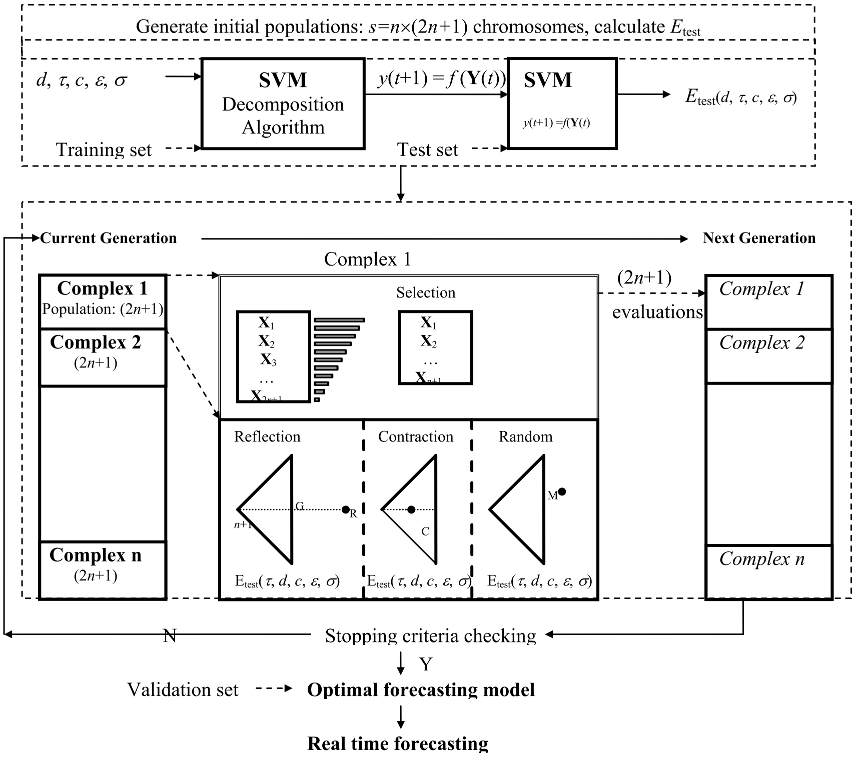
Generate initial populations: $s=n\times(2n+1)$ chromosomes, calculate E_{test}
$d, \tau, c, \varepsilon, \sigma$
Training set
SVM
Decomposition
Algorithm
$y(t+1)=f(\mathbf{Y}(t))$
Test set
SVM
$y(t+1)=f(\mathbf{Y}(t))$
$E_{test}(d, \tau, c, \varepsilon, \sigma)$
Current Generation
Next Generation
Complex 1
Complex 1
Population: $(2n+1)$
Complex 2
$(2n+1)$
Complex n
$(2n+1)$
Selection
X_1
X_2
X_3
…
X_{2n+1}
X_1
X_2
…
X_{n+1}
Reflection
G
R
n+1
Contraction
C
Random
M
$E_{test}(\tau, d, c, \varepsilon, \sigma)$
$E_{test}(\tau, d, c, \varepsilon, \sigma)$
$E_{test}(\tau, d, c, \varepsilon, \sigma)$
$(2n+1)$
evaluations
Complex 1
Complex 2
Complex n
N
Stopping criteria checking
Y
Validation set
Optimal forecasting model
Real time forecasting

Figure 4 | Diagram of the EC-SVM algorithm: an optimal forecasting model.

higher prediction accuracy, the EC-SVM scheme may also be applied to the following procedures, such as those shown by Cao *et al.* (1998) and Porporato and Ridolfi (2001). The scheme illustrated in Figure 4 will then search the optimal embedding parameters set (d_1, τ_1, d_2, τ_2, . . ., d_i, τ_i), where i is the number of time series considered. SCE can also easily deal with chromosomes with dozens of parameters.

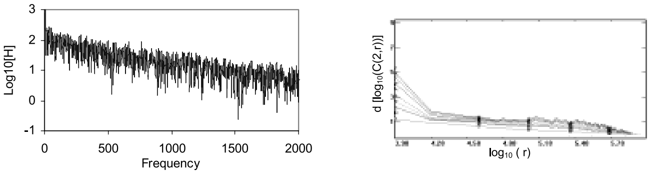

Figure 5 | Runoff data of the Tryggevælde catchment.

APPLICATIONS

The application of the EC-SVM approach is now demonstrated on two daily flow time series: (1) the Tryggevælde catchment, Denmark; and (2) the Mississippi River at Vicksburg. EC-SVM is also applied to the daily flow difference time series. The results are compared with each other and those originating from traditional chaotic techniques and the naive prediction approach.

Tryggevælde catchment runoff

The time series used in this study is the daily runoff of the Tryggevælde catchment, Denmark, with an area of 130.5 km^2. The data is from 1 January 1975 to 31 December 1993. The basic statistics are: (1) mean = 0.977 m^3/s; (2) standard deviation = 1.367 m^3/s; (3) maximum = 11.068 m^3/s and minimum = 0.014 m^3/s.

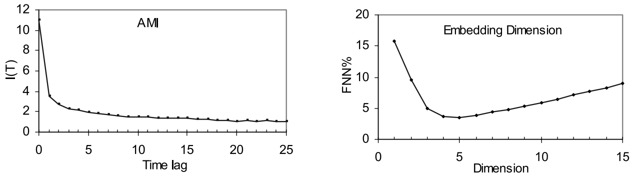

Figure 6 | Embedding structure achieved from traditional chaotic techniques.

In the traditional chaotic technique, data are divided into two parts. One part (e.g. from 1975 to 1991) serves for chaotic behaviour detection and phase space reconstruction, while the other part (from 1992 to 1993) serves for forecasting. The Fourier transform analysis conducted provides a broadband power spectrum and a low dimension, around 1.2–1.5 is observed in the correlation dimension calculation (Figure 5). The embedding structures resulting from AMI and FNN are a time lag of 12 and an embedding dimension of 4, as shown in Figure 6.

Using $d = 4$, $\tau = 12$ and $k = 5$, the normalised root mean square error (NRMSE) for 1-lead day prediction for the forecasting set (year 1992–1993) is 0.444. The normalised root mean square error (NRMSE) is defined as

$$\text{NRMSE} = \sqrt{\left(\sum_{t=1}^{N}\left(y_i - \hat{y}_i\right)^2\right) / \sum_{i=1}^{N}\left(y_i - \bar{y}\right)^2} \qquad (23)$$

where y_i is the observed value, y is the average of the measured values and $\hat{y}_i$ is the predicted value.

Naive forecasting is a simple and yet often potentially effective time series forecasting technique, particularly for short lead times. The forecast at time $(t + 1)$, for example, is assumed to be equal to the observed values at time t, i.e. $y_{t+1} = y_t$. Using naive forecasting the NRMSE for 1-lead day prediction, applied to the same data set (1992–1993), is 0.396. Thus, naive forecasting performs better than the traditional chaotic technique.

ARIMA(p, d, q) model is the mixed autoregressive-moving average model of order (p, q) on the dth differences of the time series. The NRMSE values for 1-lead day prediction, for the validation set, resulting from the first-order ARIMA(1,0,1) and ARIMA (1,1,1) are 0.367 and 0.373, respectively.

Phoon *et al.* (2002) and Liong *et al.* (2002) in their study of the inverse approach used the same data set. However, they separated the entire data set into three subsets, i.e. training, testing and validation sets. Years

Table 1 | NRMSE resulting from various techniques for the Tryggevælde catchment

Approach	Time series	RMSE	NRMSE	(d,τ)
Standard chaos technique	*Q*	0.647	0.444	(4,12)
Naive	*Q*	0.577	0.396	–
ARIMA(1,0,1)	*Q*	0.535	0.367	–
Inverse approach*	*Q*	0.527	0.361	(3,1)
EC-SVM	***Q***	**0.514**	0.352	(3,1)
Standard chaos technique	*dQ*	0.598	0.410	(4,8)
ARIMA(1,1,1)	*Q*	0.543	0.373	–
EC-SVM	***dQ***	**0.504**	0.347	(2,1)

*Liong *et al*. (2002).

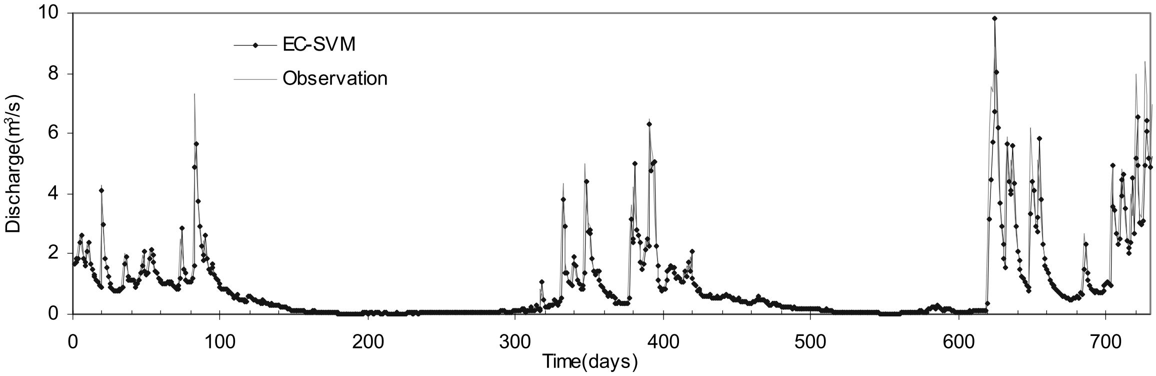

Figure 7 | Observed and predicted hydrographs for the validation set of the Tryggevælde catchment runoff.

1975–1989 were for training, years 1990–1991 for testing, while years 1992–1993 were for validation. The purpose of creating a testing set is quite obvious in their study since their objective function is to minimise the prediction error. The NRMSE values resulting from Phoon *et al.* (2002) and Liong *et al.* (2002), using an optimal parameters set, for the validation set are 0.369 and 0.361, respectively, which are, as expected, lower than that of the traditional chaotic technique (0.444). It should be noted that the optimal (d, τ, k) set resulting from Phoon *et al.* (2002) and Liong *et al.* (2002) are (3,1,10) and (2,1,11), respectively.

The proposed EC-SVM adopts the same data sets as those used in the study of Phoon *et al.* (2002) and Liong *et al.* (2002). Results from the validation set (years 1992–1993) show that the NRMSE is 0.352, which is thus far the best among the various techniques available. The embedding structure resulting from the search yields $d = 3$ and $\tau = 1$.

The success of the proposed EC-SVM on the original flow time series brings the authors to look at the daily flow difference time series—an attempt to further improve the prediction accuracy. The daily flow difference, or the first difference, $dQ(t)$, is expressed as

$$dQ(t) = Q(t+1) - Q(t). \tag{24}$$

The focus is now on predicting the $dQ(t)$ value. Similar to the Q time series, the phase space of the dQ time series is first reconstructed, followed by dynamic reconstruction. The following function serves as the predictor of $dQ(t)$:

$$dQ_{t+1} = f(dQ_t, dQ_{t-\tau}, \ldots, dQ_{t-(d-1)\tau}). \tag{25}$$

The error prediction, dQ, is first calculated and is then incorporated into $Q(t+1) = Q(t) + dQ(t)$.

Table 1 shows the results from Q and dQ time series, using various techniques, on the Tryggevælde catchment runoff. The results show that EC-SVM has a better performance than the other techniques shown in Table 1. The EC-SVM scheme on Q time series yields a 20.7% improvement over the standard chaotic techniques applied on Q time series. EC-SVM achieves further improvements by the analysis being conducted on the dQ time series. EC-SVM on the dQ time series provides the highest prediction accuracy, with a NRMSE value of 0.347. Figure 7 shows the hydrograph comparison between EC-SVM simulated (based on the dQ time series) and that actually observed.

Table 2 | Training time for the *Q* and *dQ* time series for the Tryggevælde catchment runoff

	Time (s)	Iterations	NRMSE (validation)
Q	207.67	151,668	0.352
dQ	53.34	11,800	0.347

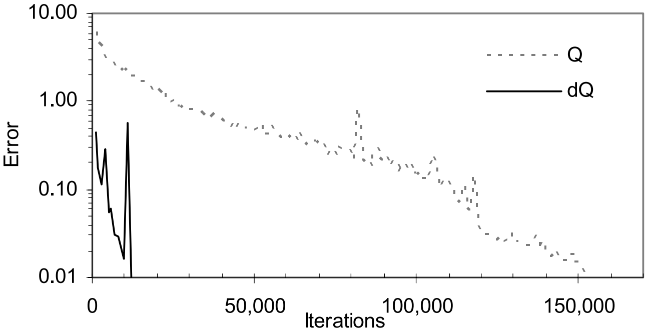

Figure 8 | Training on the *Q* and *dQ* time series for the Tryggevælde catchment runoff.

Moreover, the training time for the *dQ* time series is much shorter than that for the *Q* time series directly (about 4 times faster), as shown in Table 2 and Figure 8. The results shown are based on the programme running under Linux on a Pentium II at 333 MHz.

Mississippi River flow

In this section the daily flow time series of the Mississippi River at Vicksburg, Station No. 07289000 (Hydrological Region 08 of the USGS), is considered. This data covers the period from 1 January 1975 to 31 December 1993. Its basic statistics are: (1) mean = 18,456.54 m^3/s; (2) standard deviation = 9727.72 m^3/s; (3) maximum = 52,103.00 m^3/s and minimum = 3907.72 m^3/s.

Similar to the approach shown in the section on the traditional chaotic technique, data are divided into two parts: (1) 1975–1991 for phase space reconstruction and (2) 1992–1993 for forecasting. Fourier analysis provides a broadband power spectrum while the correlation dimension calculation yields a low dimension, around 1.9. An embedding structure of $d = 4$ and $\tau = 13$ results from the AMI and FNN approaches. With $d = 4$, $\tau = 13$ and $k = 5$ the resulting NRMSE from the forecasting set (1992–1993) is 0.2064.

Liong *et al.* (2002) also analysed the same set of Mississippi River data and divided them into training (1975–1989), test (1990–1991) and validation (1992–1993) sets. Liong *et al.* (2002) reported an NRMSE value of 0.0452, with the optimal parameters set (d, τ, k) as (2,1,5), for the validation set (1992–1993). Using naive forecasting the NRMSE yields an value of 0.0771. The proposed EC-SVM approach, however, gives an NRMSE of 0.0387, the best thus far among the various techniques. With the *dQ* time series, the NRMSE is slightly improved, to 0.0385.

Table 3 | NRMSE resulting from various techniques for Mississippi River flow

Approach	Time series	RMSE	NRMSE	(d,τ)
Standard chaos technique	*Q*	1738.95	0.2064	(6,13)
Naive	*Q*	608.70	0.0771	–
ARIMA(1,0,1)	*Q*	435.00	0.0551	–
Inverse approach*	*Q*	356.89	0.0452	(2, 1)
EC-SVM	***Q***	**306.58**	0.0387	(2, 1)
Standard chaos technique	*dQ*	365.26	0.0462	(4, 6)
ARIMA(1,1,1)	*Q*	322.69	0.0409	–
EC-SVM	***dQ***	**304.26**	0.0385	(3, 1)

*Liong *et al*. (2002).

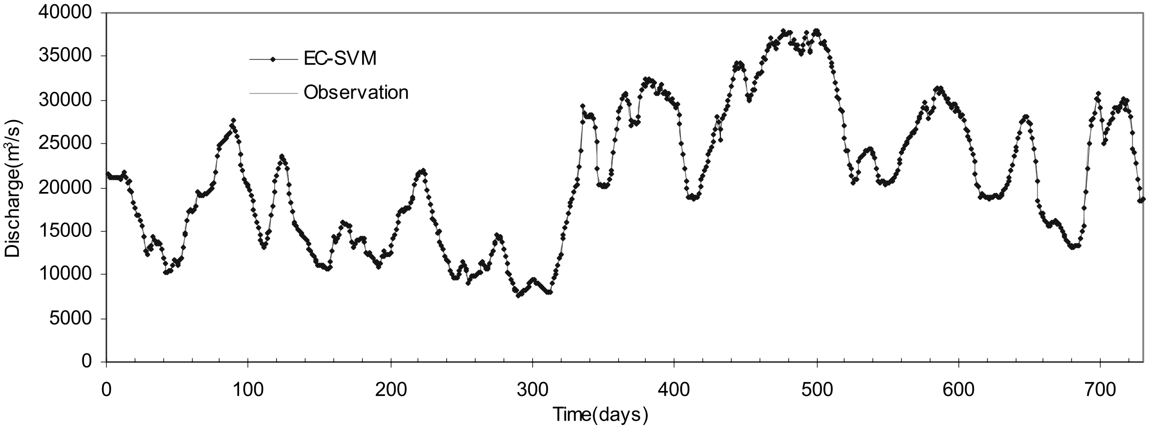

Figure 9 | Observed and predicted hydrographs for the validation set for the Mississippi River, Vicksburg.

Table 3 shows the prediction accuracies resulting from the Q and dQ time series, using various techniques, for the Mississippi River. EC-SVM on the dQ time series yields the highest prediction accuracy. It can be seen that the EC-SVM approach on the Q time series yields a significant improvement, 81.2%, over the standard chaotic approach, a 49.8% improvement over the naive approach, a 29.5% improvement over the ARIMA(1,0,1) model and a 14.3% improvement over the inverse approach with a local model. EC-SVM applying on the dQ time series yields a marginally better prediction performance than that of the Q time series. Figure 9 shows the hydrograph comparison between EC-SVM simulated (with dQ time series analysis) and that actually observed.

Although the improvement using the dQ time series is not spectacular, as shown in Tables 3 and 4, the application of EC-SVM is still suggested to be applied to the dQ time series since the performance speed, as shown in Table 4 and Figure 10, is significantly much faster (about 70 times) than that of the Q time series analysis. The results shown are based on the programme running under Linux on a Pentium II at 333 MHz.

Table 4 | Training time for the Q and dQ time series for Mississippi River flow

	Time (s)	Iterations	NRMSE (validation)
Q	3,235.52	1,732,579	0.0387
dQ	45.77	47,590	0.0385

CONCLUSIONS

The proposed EC-SVM, a forecasting tool SVM operating in the chaos-inspired phase space and optimised with an evolutionary algorithm, has been described and applied to two real daily flow time series, the Mississippi River and runoff from the Tryggevælde catchment. A recently developed decomposition method has been found to be most suitable in chaos time series analysis since the method is able to deal with large data records, a signature of the chaos technique.

The study shows that the proposed EC-SVM provides a more accurate prediction than the traditional chaos technique and naive forecasting. For the Tryggevælde catchment runoff, the NRMSE is reduced from 0.444 to 0.347, while for the Mississippi River, the NRMSE is reduced from 0.2064 to 0.0385.

The study further suggests considering the daily flow differences time series instead of the flow time series since the computational speed is significantly faster.

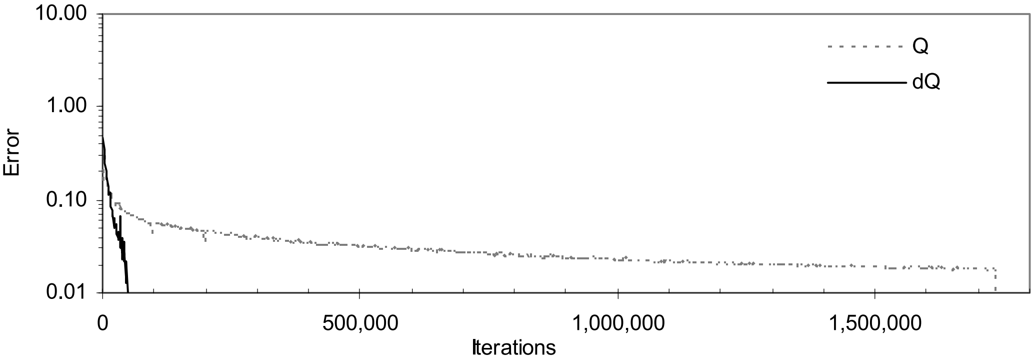

Figure 10 | Training on the *Q* and *dQ* time series for Mississippi River flow.

ACKNOWLEDGEMENTS

The first author gratefully acknowledges the financial support provided by the National University of Singapore through their Research Scholarship. The authors wish to thank the Danish Hydraulic Institute (DHI) for the daily runoff data of the Tryggevælde catchment. The authors would like to thank the reviewers for their valuable comments and suggestions.

ABBREVIATIONS AND NOTATION

Abbreviations

KKT	Karush–Khun–Tucker (KKT) conditions
NRMSE	normalised root mean square error
RMSE	root mean square error
SCE	shuffled complex evolution method
SMO	sequential minimal optimisation
SVM	support vector machine

Notation

τ	time lag
d	embedding dimension
$\boldsymbol{Y}$	lag vector
C	tradeoff between empirical error and model complexity
ε	insensitive loss function
σ	width of Gaussian kernel function
$Q(\beta)$	objective function of the quadratic dual problem in SVM
N	number of training examples
$\boldsymbol{K}$	kernel matrix
$\boldsymbol{\beta}_S$	selected working variables
$\boldsymbol{\beta}_F$	fixed variables
$\boldsymbol{d}$	difference between β^k and β^{k+1}
n	number of parameters to be optimised
E_{test}	prediction error of testing set
$Q(t)$	runoff time series
$dQ(t)$	daily flow difference

REFERENCES

Abarbanel, H. D. I. 1996 *Analysis of Observed Chaotic Data*. Springer Verlag, New York.

Babovic, V., Keijzer, M. & Stefasson, M. 2000 Optimal embedding using evolution algorithms. *Proc. 4th International Conference on Hydroinformatics, Iowa City, USA*. CD-ROM, IHAR Publications.

Cao, L., Mees, A., & Judd, K. 1998 Dynamics from multivariate time series. *Physica D* **121,** 75–88.

Cherkassky, V. & Ma, Y. 2004 Practical selection of SVM parameters and noise estimation for SVM regression. *Neurocomputing* **17** (1), 113–126.

Cherkassky, V. & Mulier, F. 1998 *Learning from Data: Concepts, Theory and Methods*. John Wiley and Sons, New York.

Collobert, R. & Bengio, S. 2001 SVMTorch: support vector machines for large-scale regression problems. *J. Machine Learning Res.* **1**, 143–160.

Dibike, Y. B., Velickov, S., Solomatine, D. P. & Abbott, M. B. 2001 Model induction with support vector machines: introduction and applications. *J. Comput. Civil Engng. ASCE* **15** (3), 208–216.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J. & Vapnik, V. 1997 Support vector regression machines. In *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, MA, pp. 155–161.

Duan, Q., Sorooshian, S. & Gupta, V. K. 1992 Effective and efficient global optimization for conceptual rainfall-runoff models. *Wat. Res. Res.* **28** (4), 1015–1031.

Farmer, J. D. & Sidorowich, J. J. 1987 Predicting chaotic time series. *Phys. Rev. Lett.* **59**, 845–848.

Fraser, A. & Swinney, H. 1986 Independent coordinates for stranger attractors from mutual information. *Phys. Rev. A* **33**, 1134–1140.

Grassberger, P. & Procaccia, I. 1983 Characterization of strange attractors. *Phys. Rev. Lett.* **50**, 346–349.

Jayawardena, W. & Lai, F. 1994 Analysis and prediction of chaos in rainfall and stream flow time series. *J. Hydrol.* **153**, 23–52.

Joachims, T. 1999 Making large-scale SVM learning practical. *Advances in Kernel Methods—Support Vector Learning* (ed. Schölkopf, B., Burges, C. & Smola, A.), pp. 169–183. MIT Press, Cambridge, MA.

Kennel, M. B., Brown, R. & Abarbanel, H. D. I. 1992 Determining embedding dimension for phase space reconstruction using a geometrical construction. *Phys. Rev. A* **45**, 3403–3411.

Liong, S. Y., Phoon, K. K., Pasha, M. F. K. & Doan, C. D. 2002 A robust and efficient scheme in search for optimal prediction parameters set in chaotic time series. *First Asia Pacific DHI Software Conference*, *Bangkok* Keynote paper. http://www.dhisoftware.com/Bangkok2002/Proceedings/keynotes/keynotespks.htm/

Liong, S. Y. & Sivapragasm, C. 2002 Flood stage forecasting with SVM. *J. Am. Wat. Res. Assoc.* **38** (1), 173–186.

Matterra, D. & Haykin, S. 1999 Support vector machines for dynamic reconstruction of a chaotic system. In *Advances in Kernel Methods* (ed. Schölkopf, B., Burges, C. J. C. & Smola, A. J.), pp. 211–241. MIT Press, Cambridge, MA.

Phoon, K. K., Islam, M. N., Liaw, C. Y. & Liong, S. Y. 2002 Practical inverse approach for forecasting nonlinear hydrological time series. *J. Hydrol. Engng. ASCE* **7** (2), 116–128.

Platt, J. C. 1999 Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods -Support Vector Learning* (ed. Schölkopf, B., Burges, C. & Smola, A.), pp. 185–208. MIT Press, Cambridge, MA.

Porporato, A. & Ridolfi, L. 2001Multivariate nonlinear prediction of river flows. *J. Hydrol.* **248**, 109–122.

Sauer, T., Yorke, J. & Casdagli, M. 1991 Embedology. *J. Stat. Phys.* **65**, 579–616.

Sivakumar, B., Liong S. Y. & Liaw, C. Y. 1998 Evidence of chaotic behaviour in Singapore rainfall. *J. Am. Wat. Res. Assoc.* **34** (2), 301–310.

Smola, A. J., Murata, N., Schölkopf, B. & Müller, K. 1998 Asymptotically optimal choice of e-loss for support vector machines. In *Proceedings of the 8th International Conference on Artificial Neural Networks*. Springer Verlag, Berlin, pp. 105–110.

Takens, F. 1981 In *Dynamical Systems and Turbulence. Lecture Notes in Mathematics (Warwick) Vol. 898* (ed. Rand, A. & Young, L. S.), p. 366. Springer Verlag, Berlin.

Vapnik, V. N. 1992 Principle of risk minimization for learning theory. *Adv. Neural Inf. Processing Syst.* **4,** 831–838.

Vapnik, V. N. 1995 *The Nature of Statistical Learning Theory*. Springer Verlag, New York.

Vapnik, V. N., Golonich, S. & Smola, A. 1997 Support vector method for function approximation, regression estimation, and signal processing. *Advances in Neural Information Processing Systems* **9** (ed. Mozer, M., Joradan, M. & Petsche, T.), pp. 281–287. MIT Press, Cambridge, MA.

Zaldívar, J. M., Gutiérrez, E., Galván, I. M., Strozzi, F. & Tomasin, A. 2000 Forecasting high waters at Venice Lagoon using chaotic time series analysis and nonlinear neural networks. *J. Hydroinf.* **2**, 61–84.