Rainfall and runoff forecasting with SSA–SVM approach

C. Sivapragasam, Shie-Yui Liong and M. F. K. Pasha

ABSTRACT

Real time operation studies such as reservoir operation, flood forecasting, etc., necessitates good forecasts of the associated hydrologic variable(s). A significant improvement in such forecasting can be obtained by suitable pre-processing. In this study, a simple and efficient prediction technique based on Singular Spectrum Analysis (SSA) coupled with Support Vector Machine (SVM) is proposed. While SSA decomposes original time series into a set of high and low frequency components, SVM helps in efficiently dealing with the computational and generalization performance in a high-dimensional input space. The proposed technique is applied to predict the Tryggevælde catchment runoff data (Denmark) and the Singapore rainfall data as case studies. The results are compared with that of the non-linear prediction (NLP) method. The comparisons show that the proposed technique yields a significantly higher accuracy in the prediction than that of NLP.

Key words | forecasting, nonlinear prediction, singular spectrum analysis, support vector machine

C. Sivapragasam
M. F. K. Pasha
Department of Civil Engineering,
National University of Singapore,
10 Kent Ridge Crescent,
Singapore 119260

Shie-Yui Liong (corresponding author)
Department of Civil Engineering,
National University of Singapore,
10 Kent Ridge Crescent,
Singapore 119260
Tel: (+65) 8742155.
E-mail: *cvelsy@nus.edu.sg*

INTRODUCTION

In the last decade or so, machine learning techniques such as Artificial Neural Networks (ANN), fuzzy logic, genetic programming, etc., have been widely used in the modeling and prediction of hydrologic variables. One common way to improve the prediction accuracy is to perform some pre-processing of the inputs. Changing the representation of data is one such technique, for example. Ideally, a pre-processing that most matches the specific learning problem should be chosen. In this study, Singular Spectrum Analysis (SSA) is proposed as a novel pre-processing technique for the deterministic chaotic systems, e.g. the rainfall and runoff processes, and the resulting input representation is trained with Support Vector Machine (SVM) for forecasting.

SSA, as proposed by Vautard *et al*. (1992), is generally seen as an adaptive noise-reduction algorithm. It is used to perform a spectrum analysis on the input data, eliminate the 'irrelevant features' (high-frequency components) and invert the remaining components to yield a 'filtered' time series. This approach of filtering a time series to retain desired modes of variability is based on the idea that the predictability of a system can be improved by forecasting the important oscillations in time series taken from the system. The general idea is to filter the record first and then use some model to forecast on the filtered series (Elsner & Tsonis, 1997). For example, Lisi *et al*. (1995) applied SSA to extract the significant components in their study on Southern Oscillation Index (SOI) time series and used a back-propagation neural network for prediction. They reconstructed the original series by summing up the first '*p*' significant components.

Although SSA is seen as an adaptive noise-reduction (filtering) algorithm, in this study SSA is used as an efficient pre-processing algorithm which results in the modified representation of the input vectors where new features are linear functions of the original attributes. This is because for deterministic chaotic systems like the rainfall and runoff processes (Jayawardena & Lai, 1994; Sharifi *et al*., 1990; Sivakumar *et al*., 1998; Islam *et al*., 2000), it is difficult to precisely demarcate signal and noise components and the suppression of certain high frequency components may alter the resulting filtered output signal. Thus, the prediction accuracy may be better when the learning machine is presented with all components of

the spectrum analysis for training. However, such an approach has an obvious disadvantage in terms of the cost one has to pay for the computational and generalization performance of the learning machine, which degrades rapidly with the growth in the number of input features. SVM is proposed to overcome this problem. SVM offers an efficient way to deal with the computational and generalization performance in a high-dimensional input space owing to the dual representation of the machine in which the training patterns always appear in the form of scalar products between pairs of examples.

In summary, this paper addresses the forecasting problem in two steps: (1) pre-processing the input time series based on Singular Spectrum Analysis (SSA) into a set of high and low frequency signals resulting in a high dimensional input space; (2) training the Support Vector Machine (SVM) to learn this preprocessed data and subsequent prediction. Further, a new 'kernel' function is proposed to improve the efficiency of the SVM prediction.

The paper is organized as follows. First, SVM and SSA techniques are described. Then the proposed technique is applied for single-lead day prediction of Singapore rainfall and that of the Tryggevælde catchment runoff. Finally, a brief discussion concludes the paper.

SUPPORT VECTOR MACHINE

Introduction

SVM is basically a linear machine, which can be seen as a statistical tool that approaches the problem similar to Artificial Neural Networks (ANN). It is an approximate implementation of the principle of Structural Risk Minimization (SRM) which helps it to generalize well on unseen data. While on one hand it has all the strengths of ANN, yet on the other hand it overcomes some of the basic lacunae as reported in the application of ANN (ASCE Task Committee, 2000*a, b*). In this paper, a brief discussion on the strengths of SVM over ANN is presented.

Although most of the research work till now has been focused on the SVM classifiers and its applications, recently some applications have been seen in the regressions and time series predictions as well. Mukherjee *et al.* (1997) applied SVM for non-linear prediction of chaotic time series (the Mackey–Glass time series, the Ikeda map and the Lorenz time series) and compared the results with different approximation techniques (ANN, polynomial, RBFs, local polynomial and rational). They concluded that SVM gave excellent performance in chaotic time series, outperforming all other techniques. Dibike (2000) concluded that SVM does generalize better than both ANN and genetic programming in his case study of rainfall-runoff modeling. Babovic *et al.* (2000) concluded that SVM produced consistently better results over 12 lead periods than ANN for water level forecasting in the city of Venice. Liong & Sivapragasam (2000) and Sivapragasam & Liong (2000) demonstrated that SVM shows good generalization performance in their applications on flood forecasting and rainfall-runoff modeling, respectively.

SVM theory

According to the Structural Risk Minimization (SRM) principle, the generalization ability of learning machines depends more on capacity concepts than merely the dimensionality of the space or the number of free parameters of the loss function (as espoused by the classical paradigm of generalization). Thus, for a given set of observations $(\boldsymbol{x}_1,y_1)$, ..., $(\boldsymbol{x}_n,y_n)$, the SRM principle chooses the function f^*_β in the subset $\{f_\beta: \beta\in\Lambda\}$, for which the guaranteed risk bound, as given by Equation (1) below, is minimal. In other words, the actual risk is controlled by the two terms in Equation (1):

$$R(\beta) \le R_{emp}(\beta) + \Omega\left(\frac{n}{h}\right) \tag{1}$$

where the first term is an estimate of the risk and the second term is the confidence interval for this estimate. The parameter h is called the VC dimension (named after Vapnik and Chervonenkis) of a set of functions. It can be seen as the capacity (or the flexibility of the functional class in fitting the underlying learning problem) of a set of functions implementable by the learning machine. If the

function is too complex (for the given amount of training data), then chances of overfitting arise. In the case of ANN, for a chosen architecture, the capacity is fixed and we try to minimize the empirical risk, whereas in SVM, the empirical risk term and the capacity term are simultaneously controlled.

SVM is an approximate implementation of the SRM principle. The final approximating function for SVM for regression is of the form

$$f(x) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) K(x_i, x) + b \tag{2}$$

where $K(x_i, x) = \langle \varphi(x).\varphi(x_i) \rangle$ is called the kernel function, which performs the inner product in feature space, $\varphi(x)$. To act as a kernel, a function needs to satisfy Mercer's condition (discussed in the subsection on the proposed kernel function). Kernel representation offers a powerful alternative for using linear machines for hypothesizing complex real world problems as opposed to Artificial Neural Network based learning paradigms, which uses multiple layers of threshold non-linear functions (Cristianini & Shawe-Taylor, 2000).

The approximating function is designed to have the smallest ε deviation (given by Vapnik's ε-insensitive loss function) from measured targets, d_i, for all training data. Slack variables, ξ and ξ^*, are introduced to account for outliers in the training data. The algorithm computes the value of Lagrange multipliers, α_i and α_i^*, by minimizing the following objective function:

$$\textit{Minimize} \quad \frac{1}{2} \| a \|^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^*) \tag{3}$$

$$\textit{Subject to} \quad d_i - (a.x_i + b) \le \varepsilon + \xi_i$$
$$(a.x_i + b) - d_i \le \varepsilon + \xi_i^*$$
$$\xi_i \, \xi_i^* \ge 0$$

expressed in the dual form as

$$\textit{Maximize} \quad -\frac{1}{2} \sum_{i,j=1}^{N} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \phi_i, \phi_j \rangle - \varepsilon \sum_{i=1}^{N} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i (\alpha_i - \alpha_i^*) \tag{4}$$

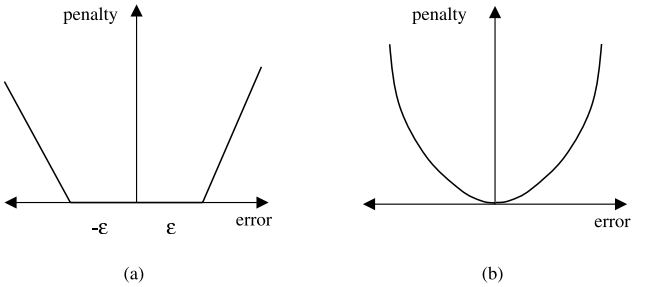

Figure 1 | Illustrative figures for (a) ϵ-insensitive loss function and (b) quadratic loss function.

subject to the following constraints:

$$\sum_{i=1}^{n} (\alpha_i - \alpha_i^*) = 0$$

$$0 \le \alpha_i \le C, \quad i = 1,2,....., N$$
$$0 \le \alpha_i^* \le C, \quad i = 1,2,....., N$$

where C is a user specified constant and it determines the trade-off between the flatness of $f(x)$ and the amount up to which the deviation can be tolerated. It should be noted that, both in the objective function given by Equation (4) and in the approximating function given by Equation (2), the training patterns appear as dot products between the training pairs.

The solution of the above problem yields α_i and α_i^* for all $i = 1$ to N. It can shown that all the training patterns within the ε-insensitive zone yields α_i and α_i^* as zeros. The remaining non-zero coefficients essentially define the final decision function. The training examples corresponding to these non-vanishing coefficients are called support vectors.

The values of ε, C and the kernel-specific parameters must be tuned to their optimum by the user to get the final regression estimation. At the moment, identification of optimal values for these parameters is largely a trial and error process. Further, other than ε-insensitive loss function, quadratic loss function (Figure 1) may also be used in which case $\varepsilon = 0$. In this study, the quadratic loss function is preferred over the ε-insensitive loss function, as the former is less computer memory intensive. Details on SVM can be found, for example, in Vapnik (1995), Drucker

et al. (1997), Smola & Scholkopf (1998), Haykin (1999), Vapnik (1999) and Cristianini & Shawe-Taylor (2000).

In this study, SVM is implemented on the Singapore Rainfall data and Tryggevælde catchment data using a MATLAB tool developed by Gunn (1997).

Proposed kernel function

Kernel representation essentially involves two steps, viz. transforming the data to a feature space, $\varphi(x)$, by non-linear mapping and performing linear regression in the feature space. These two steps can be merged into a single step through the use of kernel functions, which compute the inner product, $\langle\varphi(x).\varphi(x_i)\rangle$, in the feature space as a function of original input points. For a function to act as a kernel function, it must satisfy Mercer's theorem as defined below.

Let X be a finite input space with K(X,Z) a symmetric function on X. K(X,Z) is a kernel function if and only if the matrix,

$$[K] = (K(x_i,x_j))_{i,j=1}^{N} \tag{5}$$

is positive semi-definite.

Previous works (Babovic *et al.*, 2000; Sivapragasam & Liong, 2000; Dibike, 2000) indicate the superiority of the Radial Basis Function (RBF) kernel for hydrologic variables. However, in the present study it is found that a combination of RBF and linear (simple dot product) kernels are more robust than the RBF kernel only. It can be easily proved that the proposed kernel function satisfies Mercer's theorem to qualify as a kernel.

Consider a finite set of points $\{\boldsymbol{x}_1,\boldsymbol{x}_2 \ldots, \boldsymbol{x}_n\}$. Let $[K_1]$ and $[K_2]$ be the matrices obtained by using two kernels K_1 and K_2. If K_1 and K_2 individually satisfies Mercer's theorem, it can easily be proved that the linear combination of such kernels $K_1 + K_2$ will also satisfy Mercer's theorem and therefore will also be a kernel.

If ψ is any vector such that $\psi \in R^n$, $[K]$ is positive semi-definite if $\psi^T[K]\psi \geq 0$.

Now, for the combination of two kernels, $\psi^T[K_1 + K_2]\psi \geq 0$, i.e. $\psi^T[K_1]\psi + \psi^T[K_2]\psi \geq 0$, which is true.

Since the RBF kernel and the linear kernel are shown to be kernel functions individually, for example, in Cristianini & Shawe-Taylor (2000), the combination is also a kernel function. The proposed kernel, K_{new}, is as expressed below:

$$K_{new} = \exp(-((x_i - x_j)(x_i - x_j)^T)^{1/2}/2\sigma^2) + x_i x_j \tag{6}$$

where σ is the width of the RBFs.

Strengths of SVM over ANN

This section presents a brief discussion of the advantages of SVM over ANN, particularly in terms of the nature of the model, arriving at the optimal architecture and dealing with multi-dimensional inputs. A more detailed discussion on comparison between SVM and ANN can be found in Liong & Sivapragasam (2000).

(a) SVM is not a black-box model: SVM is founded on principles from computational learning theory. Unlike ANN, where the final set of optimal weights and threshold of the trained network cannot be interpreted, the final values of Lagrange multipliers in SVM show the relative importance of the training patterns in arriving at the final decision.
(b) Optimal architecture: arriving at the optimal architecture of the network is a time consuming and laborious task in ANN. In contrast, SVM gives the optimal architecture as a solution of quadratic optimization problem.
(c) Multi-dimensional inputs: multi-dimensional input vectors result in more complicated ANN architecture with more number of tunable parameters. However, in SVM there is no increase in the number of tunable parameters with the size of input dimension. Since in dual representation, the dot product of two vectors can be easily estimated, SVM can handle multi-dimensional inputs more efficiently and easily than ANN.

SINGULAR SPECTRUM ANALYSIS

SSA, commonly known as Karhunen–Loeve expansion, is widely used in digital signal processing. Its utility in time series analysis and prediction is attributed to the data

adaptive nature of the basis functions (eigenelements) on which it is based. In contradistinction from classical spectral analysis, where the basis functions are prescribed sines and cosines, SSA determines the shape of the oscillations adaptively from the data (Vautard *et al.*, 1992).

SSA extracts as much reliable information as possible from short and noisy time series without using prior knowledge about the underlying physics or biology of the system (Vautard *et al.*, 1992). It is based on principal component analysis (PCA) in the time domain of a univariate time series. The first step in SSA is to construct the so-called 'trajectory matrix'. The dynamics of the underlying system is generally described by a continuous variable and its derivatives. Alternatively, it can also be described as a discrete time series *xi* together with its successive shifts by a lag parameter τ. For SSA, this method is the procedure that takes a univariate time record and makes it a multivariate set of observations (Elsner & Tsonis, 1997). Thus, the 'trajectory matrix' gives the vector space of delay coordinates for a time series denoted by:

$$X=\frac{1}{\sqrt{N}}\begin{bmatrix} x_1 & x_{1+\tau} & \cdots\cdots & x_{1+(m-1)\tau} \\ x_2 & x_{2+\tau} & \cdots\cdots & x_{2+(m-1)\tau} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ x_{N-(m-1)\tau} & x_{N-(m-2)\tau} & \cdots\cdots & x_N \end{bmatrix} \quad (7)$$

where m is the embedding dimension (described under the heading 'Selection of embedding dimension') and τ is the delay time (also called time lag, described under the heading 'Selection of time delay'). The individual series in the trajectory matrix is reduced to a length, $N'=N-(m-1)\tau$. In the next step, Singular Value Decomposition (SVD) is applied to the lagged-covariance matrix, $Z=X^TX$. It can be shown that X is decomposed into $X=LDR^T$, where $L(N'\times m)$ and $R(m\times m)$ are the left and right eigenvectors and $D[\mathrm{diag}(m\times m)]$ is the corresponding singular values ($\lambda_1, \lambda_2, \ldots, \lambda_m$). The Principal Components (PCs) are the projection of the trajectory onto the columns in R, called the 'empirical orthogonal functions' (EOFs), i.e. the columns of the matrix $P=XR$ are the PCs. The elements of P are given as:

$$p_i^k=\sum_{j=1}^{m} x_{i+j-1}e_j^k. \quad (8)$$

The singular values are ordered as $\lambda_1\geq\lambda_2\geq\ldots\geq\lambda_m\geq 0$. Each λ_i^2 explains the variance of the ith PC.

As mentioned above, the eigenvectors can be used to compute the principal components of the time record. In turn, by choosing a small number of principal components, 'p' ($p\leq m$) and their associated eigenvectors, the original record can be filtered through a convolution in order to reflect oscillatory modes of interest (Elsner & Tsonis, 1997). This is the principle behind using SSA as a noise-reduction algorithm. In the present study, however, all the components are used simultaneously to form input vectors for training. In other words, the 'm' PCs from SSA is used to form the 'm' dimensional input vector. The fact that each individual PC is of length N' instead of N means they cannot be used for prediction directly. A series of length N, called Reconstructed Components (RCs), corresponding to a given set of eigenelements, is extracted as suggested by Vautard *et al.* (1992). The main advantage of using RCs instead of PCs is the recovery of the epochs.

Selection of embedding dimension (m)

It has been suggested, for example, by Penland *et al.* (1991) that the results are not greatly sensitive to 'm' as long as 'm' is considerably smaller than N. In fact, variations of the window length (embedding dimension) about a sufficiently large 'm' only stretch or compress the spectrum of the eigenvalues, leaving the relative magnitudes of the eigenvalues unchanged (Elsner & Tsonis, 1997).

In the present study, the embedding dimension is calculated, as adopted by Lisi *et al.* (1995), from Takens' theorem. According to the embedding theorem of Takens (1981) to characterize a dynamic system with an attractor (dissipative dynamical systems are characterized by the attraction of all trajectories toward a geometric object called an attractor) dimension c (correlation dimension), an m ($m=2c+1$) dimensional phase space is adequate.

'c' is found using the Grassberger–Procaccia correlation dimension algorithm (e.g. Grassberger and Procaccia, 1983*a, b*; Theiler, 1987).

Selection of time delay (τ)

SSA is based on the autocorrelation structure in the data. The autocorrelation function method is applied to determine the value of τ. The value of time lag corresponds to the value when the autocorrelation function first crosses the zero line of the lag series.

Nonlinear Prediction Method (NLP)

In the Nonlinear Prediction Method (NLP) the basic idea is to set a functional relationship between the current state X_t and future state X_{t+T}, $X_{t+T}=f_T(X_t)$ from the attractor in a phase space of an univariate time series. At time t for an observation value x_t the current state of the system is X_t, where $X_t=[x_t,x_{t-\tau}, \ldots x_{t-(m-1)\tau}]$ and the future state at time $t+T$ is X_{t+T}, where $X_{t+T}=[x_{t+T}, x_{t+T-\tau} \cdots x_{t+T-(m-1)\tau}]$, where T is the lead time. For a chaotic system the predictor F_t, which approximates f_T, is necessarily nonlinear. There are two approaches to find f_T: one is a global approximation and the other is a local approximation. According to Farmer & Sidorowich (1987), only states near (Euclidean near) the current state are used for prediction. To find the k nearest neighbors of current state X_t a Euclidean metric is imposed on phase space so that one can construct a local predictor by projecting the nearest state $X_{t'}$ to a state $X_{t'+T}$, for example through averaging $\hat{x}_{t+T}=\frac{1}{k}\sum_{i=1}^{k} x_{t'+T}$, where $\hat{x}_{t+T}$ is the predicted value.

SINGAPORE RAINFALL PREDICTION

The proposed SSA–SVM approach is first applied to a single lead day prediction of Singapore rainfall data. The island of Singapore lies only 1°20′ north of the equator. The average annual rainfall over the island is 2700 mm, a large share of which is caused by the northeast monsoon.

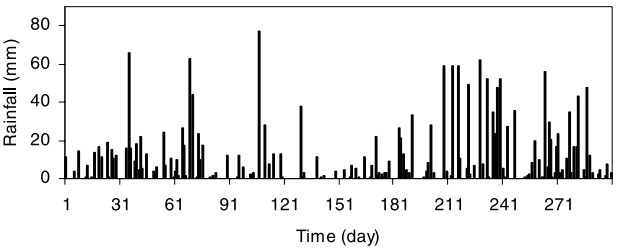

Figure 2 | Variation of Singapore daily rainfall at station 23 (December, 1965–September, 1966).

There are a total of 64 rainfall stations located on the main island of Singapore. The collection device used is the tipping bucket with drum autorecorder. Daily rainfall data from Station 23 is considered for this study. Figure 2 shows the variation of daily rainfall depth for Station 23.

Of the total available data, 3000 data are used for training while 100 data are used for validation. The prediction performance is evaluated using two goodness-of-fit measures, the correlation coefficient (*CC*) and the root-mean-square-error (*RMSE*) as defined below:

$$CC=\frac{\sum_{i=1}^{n}[(X_m)_i-(X_m)_{avg}][(X_s)_i-(X_s)_{avg}]}{\sqrt{\sum_{i=1}^{n}[(X_m)_i-(X_m)_{avg}]^2}\ \sqrt{\sum_{i=1}^{n}[(X_s)_i-(X_s)_{avg}]^2}} \quad (9)$$

$$RMSE=\sqrt{\frac{1}{n}\sum[(X_m)_i-(X_s)_i]^2} \quad (10)$$

where the subscripts m and s represent the measured and simulated values, the subscript '*avg*' represents the average value of the associated variable and n is the total number of events considered.

The study is carried out in two stages. In the first stage, raw data are used for training and prediction but the results are far from satisfactory. In the second stage, the SSA pre-processed data are used. This results in the decomposition of the original time series into a set of high and low frequency components with the disappearance of the discontinuity characterized by many 'zeros' (dry periods) existed in the original rainfall time series. The efficiency of any prediction method is greatly affected, as is widely known, by the existence of discontinuities in the

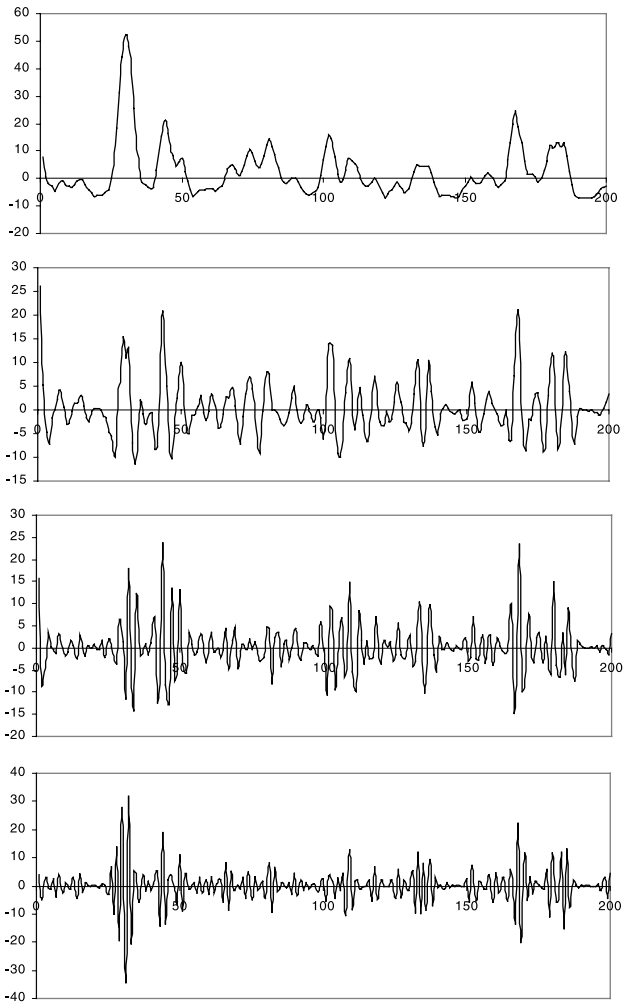

Figure 3 | Reconstructed components of the original rainfall series (sample plot).

time series. The decomposition results in a significant improvement in the prediction accuracy.

In this study an optimal set of $m = 4$ (using $m = 2c + 1$ for the next higher integer value of $c = 1.1$) and $\tau = 7$ (autocorrelation function first approaches zero at 7 of the lag series) are obtained. These values are then used in the SSA decomposition. The original time series is centered by subtracting its mean before computing the covariance matrix. The resulting RCs are shown in Figure 3 (sample plot of 200 vectors). The first two components in Figure 3 are low frequency components as compared to the remaining two. SVM is now implemented as defined in Equation (6), with a spread σ. The best performance on training data is obtained for $C = 15$ and $\sigma = 25$.

Table 1 | Prediction accuracy of various techniques (Singapore rainfall data)

		SVM	
Items	**NLP**	**Raw data**	**Preprocessed data with SSA**
Number of training samples	3000	700	700
Number of verification samples	100	100	100
Correlation coefficient			
Training	0.57	0.18	0.80
Verification	0.51	0.10	0.70
RMSE			
Training	14.57	22.9	9.35
Verification	8.50	20.12	6.11

The result is then compared to the best prediction from NLP method with $m = 4$, $\tau = 7$ and the number of nearest neighbors, $k = m + 1 = 5$ [as suggested by Farmer and Sidorowich (1987) and Cao and Soofi (1999)]. Table 1 compares the prediction accuracy resulting from NLP and SVM (raw data only, and pre-processed data with SSA) for a single lead day forecasting. It can be seen that the pre-processing with SSA drastically improves the prediction result over forecasting with raw data. The discontinuity in the rainfall series (raw data) characterized by multiple dry periods ('zeros') causes SVM to yield a very poor prediction (correlation coefficient $CC = 0.10$). Applying SVM on the preprocessed data, however, gives a *CC* of 0.70 while NLP yields a *CC* of 0.51 for the verification set. The *RMSE* for SVM with the preprocessed data is 6.11 as opposed to 8.50 obtained by NLP. It should noted that, although the number of training data used for SVM is considerably less than that for NLP (700 against 3000), the generalization error is still good in spite of the fact that the univariate input vectors become four-dimensional after SSA pre-processing. Computational

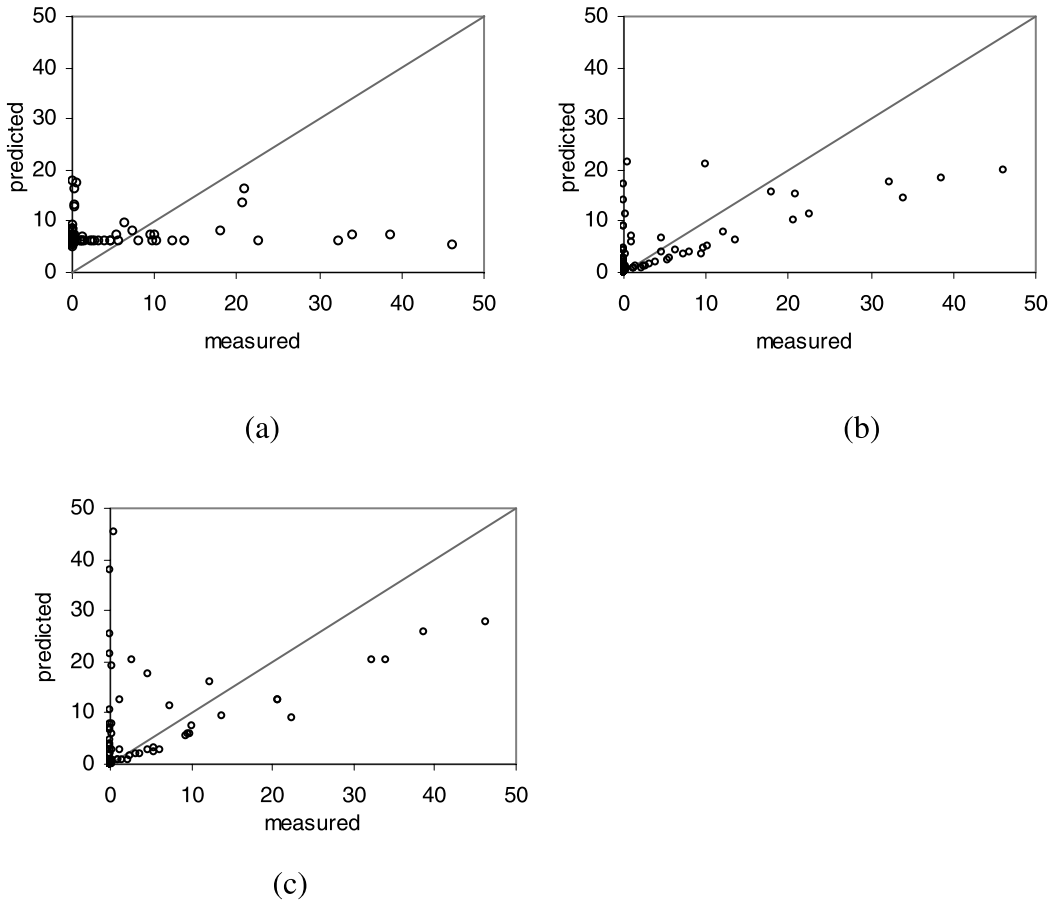

Figure 4 | Scatter plots for (a) SVM with raw input, (b) SSA–SVM and (c) NLP.

efficiency also remains unaffected. This is because SVM derives its desirable property of better generalization from the SRM principle, as explained earlier. Figure 4(a–c) show the scatter plots for 3 cases, viz. NLP prediction, SVM prediction with raw data and SVM prediction with pre-processed data, respectively.

TRYGGEVÆLDE CATCHMENT RUNOFF PREDICTION

The proposed SSA–SVM method is now applied to forecast the runoff from Tryggevælde catchment similar to the previous section. The Tryggevælde catchment (with an area of 130.5 square km) is situated in the eastern part of Sealand, north of the village Karise. The soils in the catchment are predominated by clay, implying a very flashy flow regime. Daily data of meteorological input (precipitation, potential evapotranspiration and mean temperature) and observed runoff data are available for the period 1 Jan 1975 to 31 Dec 1993.

In this study $m = 4$ ($m = 2c + 1$ where $c = 1.4$ for this time series) and $\tau = 9$ are used in the SSA decomposition. It is noted that runoff at time t, Q_t, closely depends on the difference, $Q_t - Q_{t-1}$. This deviation term is included with respect to the first RC as one of the inputs along with the other 4 RCs. This results in a five-dimensional input vector. Figure 5 shows the 4 RCs as obtained by SSA decomposition. It can be seen that the first RC accounts

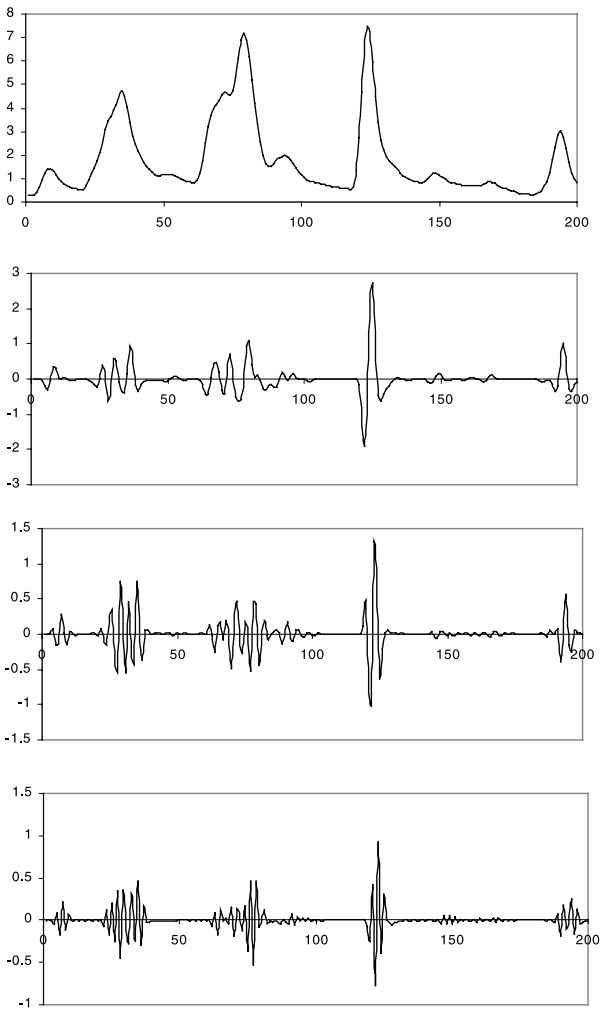

Figure 5 | Reconstructed components of the original runoff series (sample plot).

Table 2 | Prediction accuracy of various techniques (Tryggevælde runoff data)

		SVM	
Items	**NLP**	**Raw data**	**Preprocessed data with SSA**
Number of training samples	3288	700	700
Number of verification samples	365	365	365
RMSE			
Training	0.428	0.398	0.152
Verification	0.737	0.726	0.304
Correlation coefficient			
Training	0.891	0.940	0.992
Verification	0.919	0.924	0.983

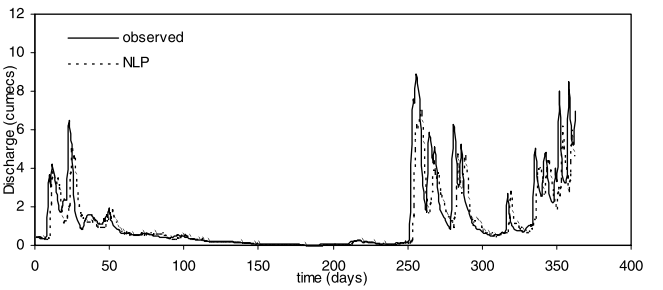

Figure 6 | Comparison between measured and the predicted flows with NLP: verification.

for the maximum variance. SVM is now applied with the proposed new kernel, as defined in Equation (6). The best performance on training data is obtained for $C = 25$ and $\sigma = 3$. The study carried out in Islam *et al.* (2000) using NLP is used for comparison with the results of the present study.

Table 2 compares the prediction performance of the single lead day forecasting on training and verification data resulting from NLP and SVM methods. It can be seen that data pre-processing with SSA coupled with SVM improves the prediction result significantly. While the *RMSE* as obtained by NLP is 0.736, SSA–SVM yields a *RMSE* of 0.304, resulting in an improvement of 58.75% in the verification data. The correlation coefficient also shows a significant improvement from 0.919 for NLP to 0.983 for SSA–SVM. It should be noted that the total number of training patterns used in SVM is significantly lower than that in NLP. Figures 6 and 7 show the comparison of measured and predicted flows for the verification data resulting from NLP and SSA–SVM methods. The peak flow prediction is significantly improved with the proposed technique. The low flows are also better predicted by the SSA–SVM approach by NLP.

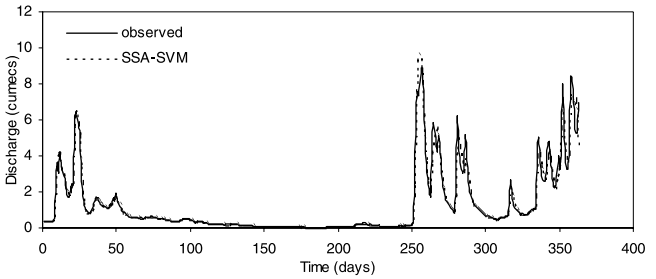

Figure 7 | Comparison between measured and the predicted flows with SSA–SVM: verification.

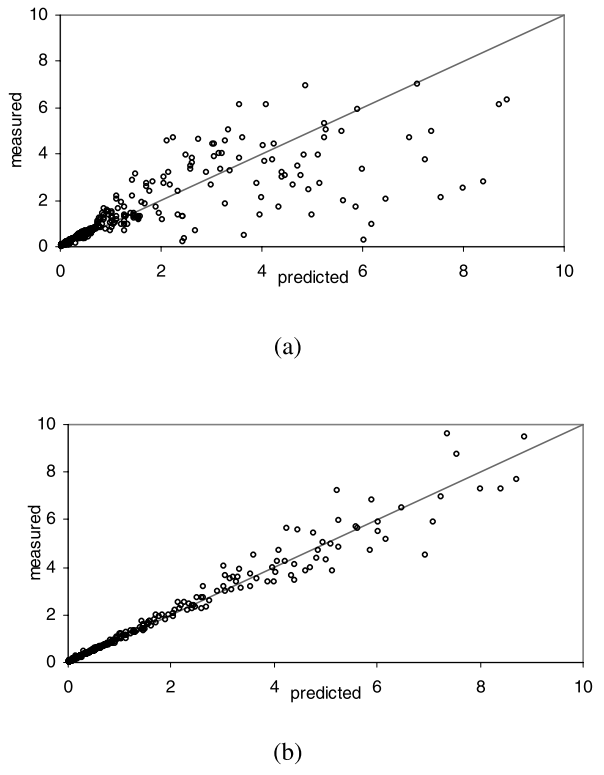

Figure 8 | Scatter plots of verification data for (a) NLP and (b) SSA–SVM.

Figure 8(a, b) show the scatter plots for the verification data with NLP and SSA–SVM, respectively.

CONCLUSIONS

In this study, it has been demonstrated that the proposed approach, SSA–SVM, could yield significantly higher prediction accuracy of hydrologic variables than that of the non-linear prediction (NLP) method. SSA–SVM results in a significant improvement in the case study on Singapore rainfall prediction with a correlation coefficient of 0.70 as opposed to 0.51 obtained by NLP. Similarly, SSA–SVM yields 58.75% improvement (in terms of *RMSE*) over NLP in the runoff prediction for Tryggevælde catchment.

Moreover, the predictions from SVM offer special advantages as compared to other machine learning techniques like ANN. Unlike ANN, SVM does not require the architecture to be defined *a priori*. The structural risk minimization principle gives SVM the desirable property to generalize well in the unseen data. The dual representation offers the unique advantage of ease in dealing with the high-dimensional input vectors without loss of both generalization accuracy and computational efficiency. The optimization problem formulated for SVM is always uniquely solvable and, thus, does not suffer from the limitation of ways of regularization as in ANN, which may lead them to local minima.

ACKNOWLEDGEMENTS

The authors wish to thank Mr Marco Sandri of Centro di Calcolo–C.I.C.A., Università di Verona, Italy for his constructive discussions on SSA. Also, the authors wish to express their sincere thanks to Danish Hydraulic Institute (DHI) and Meteorological Service Singapore for providing the daily runoff data of Tryggevælde catchment and Singapore rainfall data, respectively.

NOTATION

The following symbols are used in this paper:
$R(\beta)$ = actual risk
$R_{emp}(\beta)$ = empirical risk
Ω = confidence interval
h = VC dimension
x = input data

φ = the non-linear mapping function
f = the linear function in feature space
a and b = coefficients to be estimated
d = measured targets
ξ and ξ^* = slack variables
ε = insensitive loss function
α and α^* = Lagrange multipliers
C = user specified constant,
N = the number of training samples
l = the number of support vectors
K = the kernel function
K_{new} = proposed kernel function
$[K_1]$ = matrix obtained using kernel K_1
$[K_2]$ = matrix obtained using kernel K_2
σ = width of the RBFs
n = total number of events considered for prediction
ψ = any vector$\in R^n$
m = embedding dimension
τ = time delay
X' = trajectory matrix
N' = reduced training set after forming trajectory matrix
L, R = left and right eigenvector matrix
P = matrix of principal components
p = individual principal component
c = correlation dimension
λ = square root of variance
CC = correlation coefficient
$RMSE$ = root mean square error.

Subscripts

i, j, k = positive integer index
m = measured
s = simulated.

ABBREVATIONS

The following abbreviations are used in this paper:
ANN = Artificial Neural Network
NLP = Non-Linear Prediction
PCA = Principal Component analysis
PC = Principal Components
RC = Reconstructed Components
RBF = Radial Basis Function
SOI = Southern Oscillation Index
SRM = Structural Risk Minimization
SSA = Singular Spectrum Analysis
SVD = singular Value Decomposition
SVM = Support Vector Machine.

REFERENCES

ASCE Task Committee. 2000a Artificial neural networks in hydrology-1: preliminary concepts. *J. Hydrol. Engng* **5**(2), 115–123.

ASCE Task Committee. 2000b Artificial neural networks in hydrology-2: hydrologic applications. *J. Hydrol. Engng* **5**(2), 124–137.

Babovic, V., Keijzer, M. & Bundzel, M. 2000 From global to local modelling: a case study in error correction of deterministic models. *Hydroinformatics'2000*, Iowa Institute of Hydraulic Research, Iowa, USA (CD-ROM).

Cao, L. & Soofi, A. S. 1999 Nonlinear deterministic forecasting of daily dollar exchange rates. *Int. J. Forecast.* **15,** 421–430.

Cristianini, N. & Shawe-Taylor, J. 2000 *An Introduction to Support Vector Machines.* Cambridge University Press, Cambridge, UK.

Dibike, Y. B. 2000 Machine learning paradigms for rainfall-runoff modelling. *Hydroinformatics 2000*, Iowa Institute of Hydraulic Research, Iowa, USA (CD-ROM).

Drucker, H., Burges, C., Kaufman, L., Smola, A. & Vapnik, V. 1997 Support vector regression machines, *Advances In Neural Information Processing Systems 9*. MIT Press, Cambridge, MA, 156-161.

Elsner, J. B. & Tsonis, A. A. 1997 *Singular Spectrum Analysis: A New Tool in Time Series Analysis*. Plenum Press, New York.

Farmer, J. D. & Sidorowich, J. J. 1987 Predicting chaotic time series. *Phys Rev. Lett.* **59**(8), 845–848.

Grassberger, P. & Procaccia, I. 1983a Measuring the strangeness of strange attractors. *Physica D* **9,** 189–208.

Grassberger, P. & Procaccia, I. 1983b Characterisation of strange attractors. *Phys. Rev. Lett.* **50**(5), 346–349.

Gunn, S. R. 1997 Support vector machines for classification and regression. *Technical Report. Image Speech and Intelligent Systems Research Group,* University of Southampton.

Haykin, S. 1999 *Neural Networks: A Comprehensive Foundation.* Prentice Hall International, Inc., New Jersey.

Islam, N., Liong, S. Y., Phoon, K. K. & Liaw, C. Y. 2000 Forecasting of river flow data with general regression neural network. Accepted for publication in IAHS red book on *International Symposium on Integrated Water Resources Management* University of California, Davis.

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

Liong, S. Y. & Sivapragasam, C. 2000 Flood stage forecasting with SVM. Submitted for publication in *J. Am. Water Res. Assoc*.

Lisi, F., Nicolis, O. & Sandri, M. 1995 Combining singular-spectrum analysis and neural networks for time series forecasting. *Neural Processing Lett.* **2**(4), 6–10.

Mukherjee, S., Osuna, E. & Girosi, F. 1997 Nonlinear prediction of chaotic time series using support vector machines. *Neural Networks for Signal Processing—Proceedings of the IEEE Workshop.* IEEE, Piscataway, NJ, pp. 511–520.

Penland, C., Ghil M. & Weickmann, K. M. 1991 Adaptive filtering and maximum entropy spectra with application to changes in atmospheric angular momentum. *J. Geophys. Res.* **96,** 659–671.

Sharifi, M. B., Georgakakos, K. P. & Rodriguez-Iturbe, I. 1990 Evidence of deterministic chaos in the pulse of storm rainfall. *J. Atmos. Sci.* **47**(7), 888–893.

Smola, A. J. & Scholkopf, B. 1998 *A tutorial on support vector regression. Technical Report NeuroCOLT Technical Report NC-TR-98-030*, Royal Holloway College, University of London, UK.

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

Sivapragasam, C. & Liong. S. Y. 2000 Improving higher lead period runoff forecasting accuracy: the SVM approach. *12th Congress of the Asia and Pacific Regional Division of the IAHR*. Asian Institute of Technology, Bangkok, Vol 3, pp. 855–861.

Takens, F. 1981 Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence* (ed. Rand, D.A. & Young, L.S.). Springer-Verlag, London, pp. 366–381.

Theiler, J. 1987 Efficient algorithm for estimating the correlation dimension from a set of discrete points. *Phys. Rev. A* **36**(9), 4456–4462.

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

Vapnik, V. 1999 An overview of statistical learning theory. *IEEE Trans. Neural Networks* **10**(5), 988–999.

Vautard, R., Yiou, P. & Ghil, M. 1992 Singular-spectrum analysis: a toolkit for short, noisy and chaotic signals. *Physica D* **58,** 95–126.