Flood forecasting using support vector machines

D. Han, L. Chan and N. Zhu

ABSTRACT

This paper describes an application of SVM over the Bird Creek catchment and addresses some important issues in developing and applying SVM in flood forecasting. It has been found that, like artificial neural network models, SVM also suffers from over-fitting and under-fitting problems and the over-fitting is more damaging than under-fitting. This paper illustrates that an optimum selection among a large number of various input combinations and parameters is a real challenge for any modellers in using SVMs. A comparison with some benchmarking models has been made, i.e. Transfer Function, Trend and Naive models. It demonstrates that SVM is able to surpass all of them in the test data series, at the expense of a huge amount of time and effort. Unlike previous published results, this paper shows that linear and nonlinear kernel functions (i.e. RBF) can yield superior performances against each other under different circumstances in the same catchment. The study also shows an interesting result in the SVM response to different rainfall inputs, where lighter rainfalls would generate very different responses to heavier ones, which is a very useful way to reveal the behaviour of a SVM model.

Key words | artificial intelligence, flood forecasting, model response, over-fitting, support vector machines, under-fitting

D. Han (corresponding author)
L. Chan
N. Zhu
Department of Civil Engineering,
University of Bristol,
Bristol BS8 1TR,
UK
E-mail: *d.han@bristol.ac.uk*

INTRODUCTION

The foundation of Support Vector Machines (SVM) was given by Vapnik, a Russian mathematician in the early 1960s (Vapnik 1995), based on the Structural Risk Minimisation principle from statistical learning theory and gained popularity due to its many attractive features and promising empirical performance. SVM has been proved to be effective in classification by many researchers in many different fields such as electric and electrical engineering, civil engineering, mechanical engineering, medical, financial and others (Vapnik 1998). Recently, it has been extended to the domain of regression problems (Kecman 2001). In the river flow modelling field, Liong & Sivapragasam (2002) compared SVM with Artificial Neural Networks (ANN) and concluded that SVM's inherent properties give it an edge in overcoming some of the major problems in the application of ANN (Han *et al.* 2006). Nonlinear modelling of river flows of the Bird Creek catchment in the USA with SVM was reported to have its limitations (Han & Yang 2001; Han *et al.* 2002). Dibike *et al.* (2001) presented some results showing that Radial Basis Function (RBF) is the best kernel function to be used in SVM models. However, Bray (2002) found linear kernel outperformed other popular kernel functions (radial basis, polynomial, sigmoid). Bray & Han (2004) illustrated the difficulties in SVM identification for flood forecasting problems. It is clear that, due to its short history, there are still many knowledge gaps in applying SVM in flood forecasting and some conflicting results from different researchers are a good indication that this technique is still in its infancy and more exploratory work is necessary to improve our understanding of this potentially powerful tool from the machine learning community.

doi: 10.2166/hydro.2007.027

FUNDAMENTALS OF SUPPORT VECTOR MACHINE

Details of SVM theory have been documented by many authors (Vapnik 1998; Kecman 2001) and only a brief introduction is given here. Unlike former learning machines, the hypothesis space of SVM is limited to linear functions, in a high-dimensional feature space. These hypotheses are trained by a learning algorithm, which is based on optimisation theory. These algorithms implement a learning bias derived from statistical learning theory. By fine tuning the learning machine in this way the aim of optimising the machines' ability to generalise is achieved. The problem of linear regression is finding a linear function $y = f(x) = \langle \mathbf{w}{\cdot}\mathbf{x} \rangle + b$ that best interpolates a set of training points. The least squares approach prescribes choosing the parameters $(\mathbf{w},b)$ to minimise the sum of the squared deviations of the data, $\sum_{i=f1}^{l} (y_i - \langle \mathbf{w}{\cdot}\mathbf{x} \rangle - b)^2$ (Cristianini *et al.* 1999). To allow for some deviation ϵ between the eventual targets y_i and the function $f(x) = \langle \mathbf{w}{\cdot}\mathbf{x} \rangle + b$, the following constraints are applied: $y_i - \mathbf{w}{\cdot}\mathbf{x} - b < \varepsilon$ and $y_i - \mathbf{w}{\cdot}\mathbf{x} + b \leq \varepsilon$. This can be visualised as a band or a tube around the hypothesis function $f(x)$ with points outside the tube regarded as training errors, otherwise called slack variables ξ_i. These slack variables are zero for points inside the tube and increase progressively for points outside the tube. This approach to regression is called ε-SV regression (Vapnik 1998). It is the most common approach although it is not the only one. The task is now to minimise $\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\left(\xi_i + \xi_i^{*}\right)$ subject to: $y_i - \mathbf{w}{\cdot}\mathbf{x} - b \leq \varepsilon + \xi_i$ and $(\mathbf{w}{\cdot}\mathbf{x} + b) - y_i \leq \varepsilon + \xi_i^{*}$. An alternative form of SVM is called ν-SV regression (Smola & Schölkopf 1998). This model uses ν to control the number of support vectors. Given a set of data points, $\{(x_1,z_1),\ldots(x_l,z_l)\}$, such that $x_i \in R^n$ is an input vector and $z_i \in R^l$ the corresponding target, the form is

$$\min_{w,b,\xi,\xi^*} \frac{1}{2} w^T w + C\left(\nu\varepsilon + \frac{1}{l}\sum_{i=1}^{l}\left(\xi_i + \sum_{i=1}^{l}\xi_i^{*}\right)\right)$$

subject to $w^T\phi(x_i) + b - z_i \leq \varepsilon + \xi_i$ and $z_i - w^T\phi(x_i) - b \leq \varepsilon + \xi_i^{*}$ with ξ the upper training bound and ξ_i^{*} the lower training bound.

The role of the kernel function simplifies the learning process by changing the representation of the data in the input space to a linear representation in a higher-dimensional space called a feature space. A suitable choice of kernel allows the data to become separable in the feature space despite being non-separable in the original input space. This allows us to obtain nonlinear algorithms from algorithms previously restricted to handling linearly separable datasets. The kernel is defined to be a function $K(\mathbf{x},\mathbf{z})$, which computes the inner product $\langle \phi(\mathbf{x}){\cdot}\phi(\mathbf{z}) \rangle$ directly from the input points. Four standard kernels are usually used in classification problems and also used in regression cases: linear, polynomial, radial basis and sigmoid. Based on the work by other researchers (Dibike *et al.* 2001; Han & Yang 2001; Liong & Sivapragasam 2002; Bray 2002; Bray & Han 2004), only two kernel functions (linear and radial basis) have been explored further in this study since they generated most of the conflicting results (see Table 1).

THE CATCHMENT

The data used in this study were collected in a region called Bird Creek in the USA. The data formed part of a real-time hydrological model intercomparison exercise conducted in Vancouver, Canada in 1987 and reported by WMO (WMO 1992). The dataset is divided into two parts: a calibration (training) period and a verification (testing) period. The rainfall values were derived from 12 rain gauges situated in/near the catchment area. The river flow values were obtained from a continuous stage recorder. The period used for model calibration spanned some eight years from October 1955 to September 1963 and the verification period ranged from November 1972 to November 1974. During the calibration period the discharge at the basin

Table 1 | Formula of kernels

Kernel	Formula
Linear	$K(x, x) = x{\cdot}z$
Polynomial	$K(x, z) = (1 + (x, z))^a$
Multi-layer	$K(x, z) = \tanh(\alpha\ (x, z) + \beta)$
Radial Basis Function (RBF)	$K(x, z) = \exp(-\alpha\lvert x - z\rvert^2)$
Exponential RBF	$K(x, z) = \exp(-\alpha\lvert x - z\rvert)$

outlet ranged from 0 to 2540 m^3/s and rainfall up to 153.8 mm/d. The highest recorded discharge during the verification period was 1506 m^3/s (Hajjam 1997).

The Bird Creek catchment covers an area of 2344 km^2 and is located in Oklahoma, close to the northern state border with Kansas. The outlet of the basin is near Sperry about ten kilometres north of Tulsa. The catchment is relatively low lying, with altitudes ranging from 175 m up to 390 m above the mean sea level. There are no mountains or large water surfaces to influence local climatic conditions. Some 20% of the catchment surface is covered by forest while the main vegetative cover is grassland. The storage capacity of the soil is very high (Georgakakos & Smith 1990).

The catchment receives significant rainfall in most years, and the catchment can be classified as humid although extended periods with very low rainfall can occur. Well-defined rainy seasons occur in the spring and summer, with rain in the form of showers and thundershowers of convective origin. Snowfall remains on the ground for only a very short time. From the latter part of July to September air temperatures are high (38°C is common) and, as a result, significant evapotranspiration occurs during this time. At the same time, relative humidity is low and southerly breezes are common (Georgakakos *et al.* 1988). The river basin and the stream network are shown in Figure 1 and the training/test data are depicted in Figures 2 and 3.

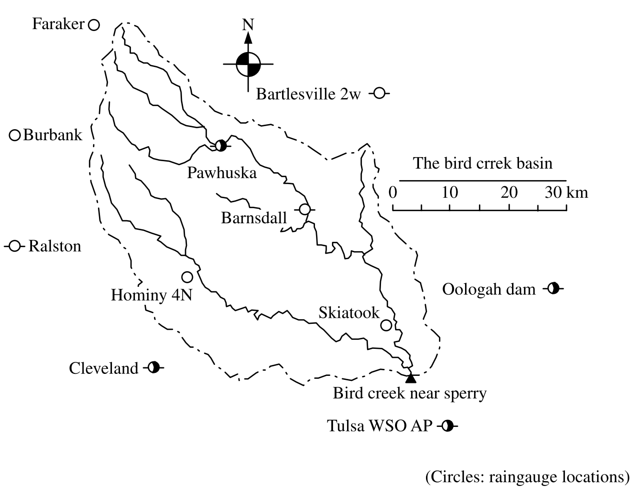

Figure 1 | The Bird Creek drainage basin (WMO 1992).

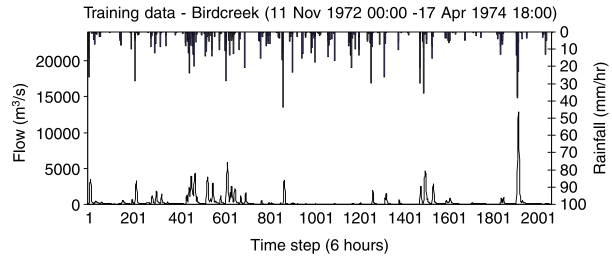

Figure 2 | Hydrograph and hyetograph of Bird Creek catchment – training data.

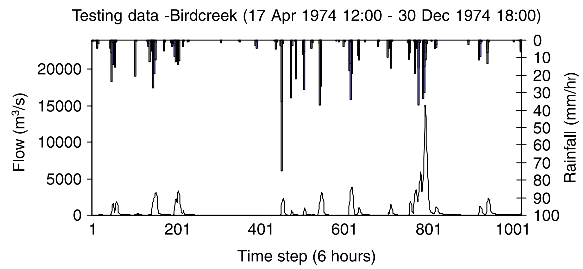

Figure 3 | Hydrograph and hyetograph of Bird Creek catchment – testing data.

MODEL CONSTRUCTION

A number of support vector machine software packages are now available. The tools used in this project are from LIBSVM, a freeware package, developed by Chih-Chung Chang and Chih-Jen (Chang & Lin 2004*a*), coupled with Gunn's toolbox (Gunn 2004) for data normalisation. The basic algorithm is a simplification of both SMO by Platt and SVMLight by Joachims (Platt 1999; Joachims 1999). LIBSVM is capable of C-SVM classification, one-class classification, ν-SV classification, ν-SV regression and ε-SV regression. In this study, ε-SVR has been used to investigate rainfall–runoff modelling. There are four main parameters, which could influence the behaviour of this model: γ, cost (C), ε-p and ε-e.γ is only essential when the kernel is not linear. Cost controls the model's tolerance to the errors. When the C value is too large, the model could be in danger of over-fitting. ε-p is a parameter in loss function of ε-SVR while ε-e would set the error tolerance as a termination criterion.

The critical issue in developing an AI model is its generalisation ability: how well will the model make predictions for events that are not in the training set? SVM, like other flexible estimation methods, can suffer from either under-fitting or over-fitting. A major problem in any model training is the decision about the complexity of the model's structure. If a SVM is not sufficiently complex to cope with the modelled process, it would fail to fully detect supportive points in the training data. This would be leading to the under-fitting case. In contrast, if a model is too complex that it can remember any single data point in training even the noise, it is considered as over-fitting. The danger of over-fitting is that the SVM can't predict anything beyond the points appeared in the training data set. In Figure 4, it demonstrates the relationship between prediction error and the complexity of a model. Therefore, choosing a suitable model structure is critical.

Figure 5 illustrates the procedures adopted in this study. In the first step, training and testing data are to be normalised to avoid the dominance of the scale problem caused by the different units used in rainfall and flow records, otherwise the vector distance will be biased towards the variables with large values. Each lead time has its own target series: hence, if six lead time forecasts are needed, there will be six SVM models, with each of them specialising for a specific lead time prediction. The selection of the past rainfall and flow as inputs to the model is quite tedious. The information content of each rainfall and flow plays an important role here. For example, if the record of the past 12 rainfall steps could provide all the information required for predicting flow in the next step, there is no need to use any rainfall beyond 12 steps. However, rainfall and runoff processes are very complicated and intertwined, and it is clear that measured flow data contains some information of the past rainfall record, since all flow data is a result of past rainfall events. If flow data is used in the model, less rainfall data would be selected. The lead time may also influence the data input combinations. For short lead time, the latest flow would dominate the prediction and, when the lead time is increased, rainfall data would play a more decisive role. Various parameters with two candidate kernel functions are altered to optimise their values. The final decision about the optimum models is not based on the training data, but on the testing data, as illustrated in Figure 4.

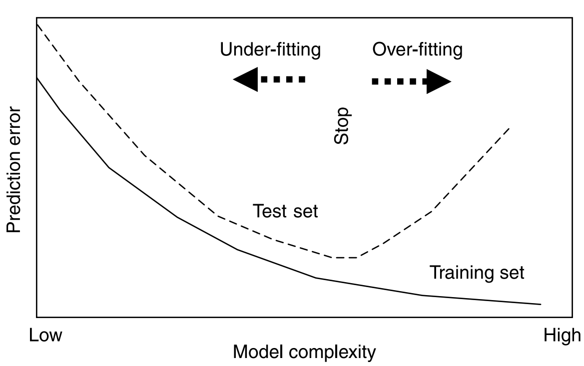

Figure 4 | The influence of model complexity (Nelles 2001).

OUTCOMES OF THE MODELLING PROCESS

One-dimensional modelling

In order to observe the behaviour of each kernel function, a SV machine is trained with three different simple models: sine, linear and quadratic curves training data. The results demonstrate that the radial basic function is ideal for a waving sine curve data; likewise, the linear function is very effective for a linear training data. However, when both functions are applied to a quadratic curve model, the linear function has superior performance, since the radial basic function is weak in extrapolation prediction. Furthermore, the extrapolation results in the RBF model was capped to give constant outputs whereas the linear function would give a trend line result. It is interesting to use sine curves in observing the behaviour and sensitivity of various SV parameters. For support vector machines' regression, gamma is crucial in the Radial Basic Function (RBF) model, which can lead to under-fitting and over-fitting in prediction. Gamma has a default value in LIBSVM ($1/k$, where $k =$ number of input records). The best fitting gamma value can be obtained by trial and error. In this study, the gamma parameter is set to several values (0.001, 0.01, 0.03, 0.05, 0.07, 0.09, 0.1, 0.3, 0.7 and 0.9) while the others are set to default ones (Chang & Lin 2004*b*). Under-fitting happens when the models are unable to predict the data that have been trained. Conversely, over-fitting occurs when the models tend to memorise all the training data but are unable to generalise for unseen data: hence, only trained data points can be predicted. A massive increase in the gamma will cause the risk of over-fitting because all the support vectors distances are taken into account; thus a complex model is

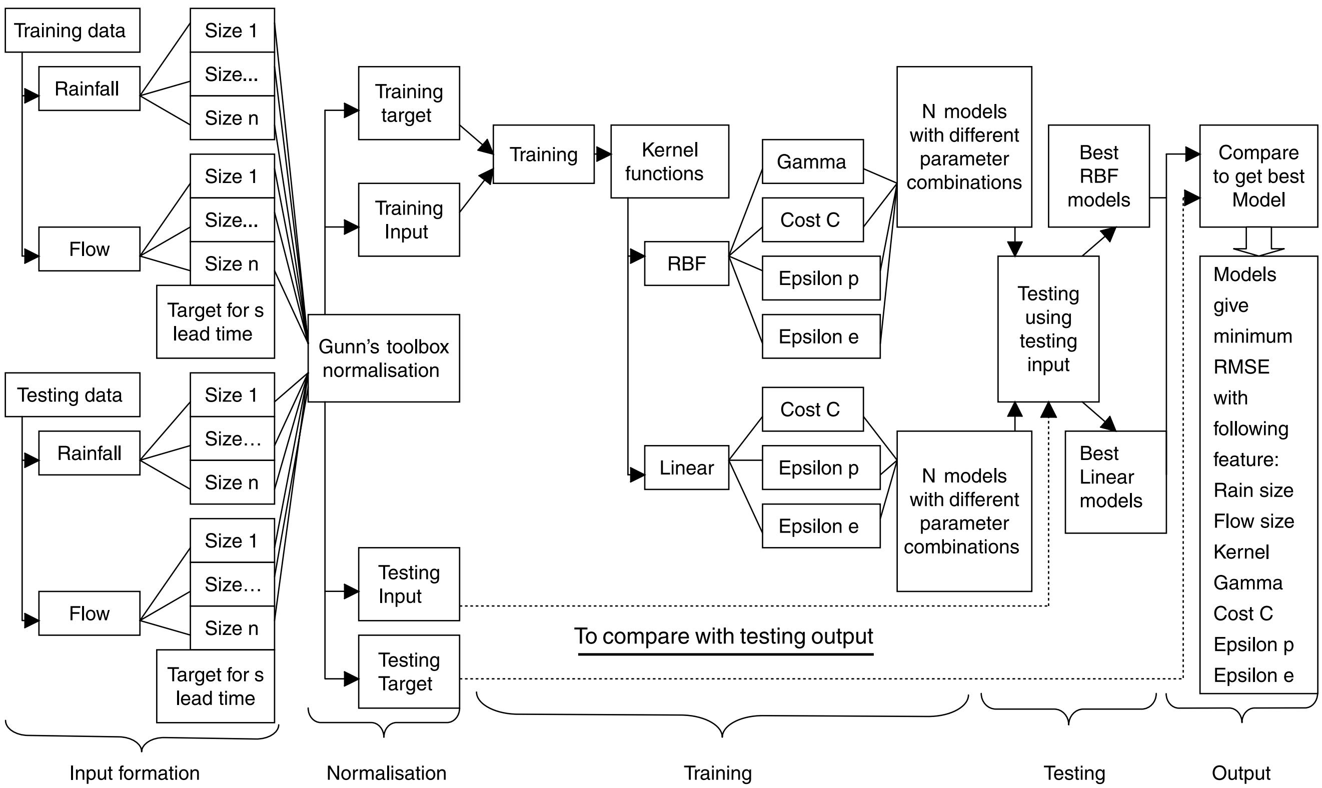

Figure 5 | Flow chart of model developing process for each lead time.

built, as mentioned previously. Conversely, when the gamma value is changed to an extremely small value, the machine would ignore most of the support vectors and hence lead to a failure in the trained point prediction, known as under-fitting. The extremely under- and over-fitted SVM models in the test cases are illustrated in Figures 6 and 7, which clearly demonstrate that over-fitting can be more damaging in a model's performance than under-fitting.

The cost has a default value of 1 and the values assigned in the model have been set as 1, 5, 10, 20, 40 and 80. A penalty is assigned for the number of support vectors falling between the hyperplanes; therefore, data that consist of lots of noises should have a smaller cost value in order to avoid penalisation to the support vectors. The ε-e and ε-p are another two parameters, which are not as sensitive as gamma and cost, and after

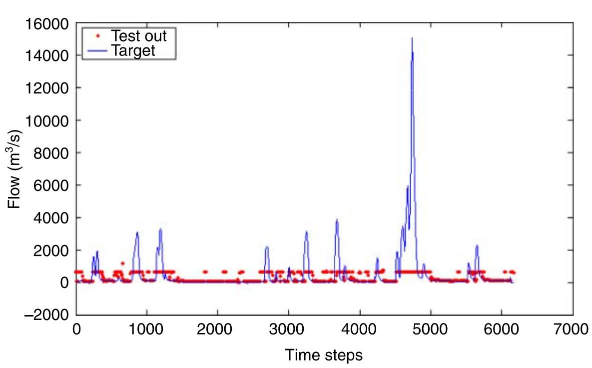

Figure 6 | Over-fitting in flood forecasting for input 4 rainfall, 3 flow, step 6.

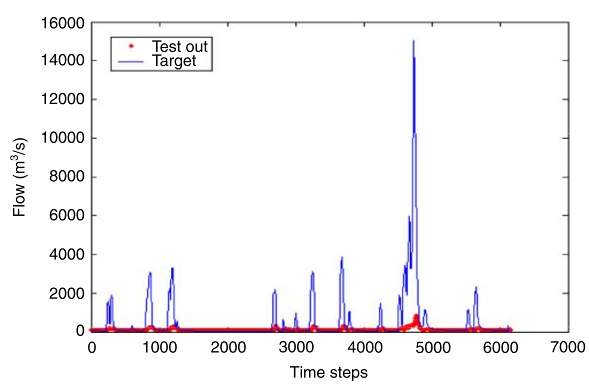

Figure 7 | Under-fitting in flood forecasting for input 4 rainfall, 3 flow, step 6.

some exploration, they have been set as ε-e = 0.000 001 and ε-p = 0.001.

Benchmarking

In judging the effectiveness of a new model, it is important to compare it with some benchmark models. The basic benchmark models are the 'naive model' and 'trend model'. In a naive model, the next step is assuming the same as the value one step before. A trend model is defined so that the future flow values are predicted from the linear extrapolation of the last two flow values. In addition, a linear transfer function (TF) model is also used in the benchmarking in this study. The TF model is based on the traditional unit hydrograph technique. It is considered that rainfall is nonlinear to stream discharge, but storm runoff may be more linearly related to an effective rainfall (Beven 2000). Due to the difficulties in estimating effective rainfall in real time, it is quite common to use total rainfall as input to TF models, as is the case in this study.

A transfer function (TF) model has been built using the same training and testing data as SVM so that the output is comparable. The theory of the model is expressed as

$$y_t = a_1 y_{t-1} + \ldots + a_N y_{t-N} + b_0 u_{t-lag} + b_1 u_{t-1-lag} + \ldots + b_M u_{t-M-lag} + e_t$$

where

a_i, b_i = model parameters;
y_t = total river flow at time t;
u_t = total rainfall rate at time t.
lag = time lag
e_t = model noise at t.

With the input of four rainfalls and three flows, the target is the runoff of 1–6 step lead times. From the RMSE values of each model in Table 2, the TF model produces the best output so the SVM is compared with the TF model in the subsequent comparisons.

The application of SVM in flood forecasting

Normalisation or scaling is crucial in flood forecasting prediction since SVM predict floods by considering the weight (distance) between the input data and the support vectors. The input data is scaled down to −1 and 1 for the entire models built throughout the study.

Table 2 | RMSE values in different models

Lead time	TF	Naïve	Trend
1	166	295	196
2	394	557	498
3	619	776	851
4	799	954	1215
5	925	1097	1580
6	1008	1212	1953

The single time step models were trained by using both radial basic and linear kernel functions with different parameter values. The parameter values in each combination are found by using the trial and error method as described by Bray & Han (2004): running the model by changing one parameter to several values while the others are set to default. Although fivefold cross-validation has been carried out in the training data as guidance to the model's training performance, the final model selection is done by the testing data. The overall performance of the final support vector machine is effective, and it has managed to learn the time and magnitude of the peak flows (Figure 8). Among all the models with different combinations of rainfall and flow inputs (rain × flow as 1 × 1,

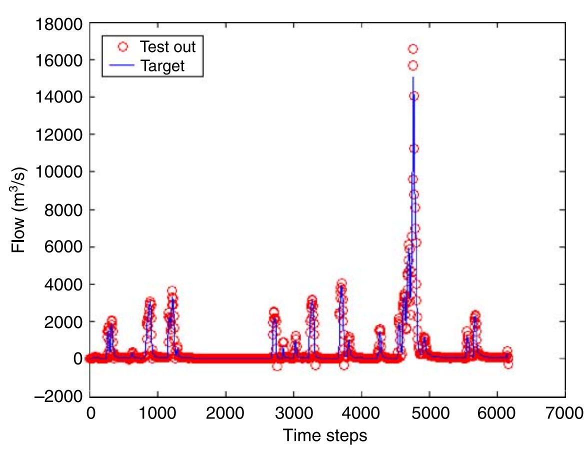

Figure 8 | Prediction result for 4 rainfall, 3 flow combination (single time step).

2×2, 3×2, 4×1, 4×3, 5×1, 5×3, 5×5, 6×2, 6×4, 6×6, 7×2, 7×4, 7×6, 8×1, 8×3, 8×5, 8×7, 9×1, 9×3, 9×5, 9×7 and 9×9), a combination of 4 rainfall and 3 flow with a linear kernel function generate the least root mean square errors results (RMSE = 162.14).

In order to predict the flow at different leading time steps, the multi-step models were built. The parameter searching method used is as before and all the inputs are normalised. There are two different types of input used to predict the flood: by assuming the rainfall of the future can be predicted (input format I) and unpredicted (input type II, i.e. zero future rain). The models were trained with different combinations of training and testing data as well as time steps (steps 2–6). The performance of the machine is very satisfactory and the flood prediction results had higher accuracy compared to the TF model (Table 3). The performance of input format I models (known future rainfalls) is better than input format II (unknown future rainfalls) and all the models generated higher RMSE values as the time step increases. The performance of the models showed that the RBF function is capable of generating lower RMSE results compared to the linear function in input format I. Conversely, in input format II, the result suggests that the linear function is better than the RBF function, aside from the single flow combinations. Therefore, the observation concludes that the RBF function could perform better when the predicted rainfall data are available and dominate the whole process, while the linear function could work better when flow data carry more important information. However, further research and study is needed to verify this hypothesis.

MODEL RESPONSE TO RAINFALL

Previously, a support vector machine is assumed to be a black-box machine learning system, which simply transforms input into output. A modeller has no idea what is inside the model and how the model is going to behave if an unforeseen input is present: hence it is important to test the model's characteristics in response to various rainfall inputs. In this study, the machine model is tested with 0 mm, 1 mm, 2 mm, 4 mm, 50 mm and 100 mm of rainfall and the results are shown in Figures 10 and 11. For lighter rains, SVM generates flows with a ramp curve and becomes flat after a certain step (Figure 9). This clearly contradicts the hydrological principle that a limited amount of rainfall couldn't generate an unlimited amount of flow. It is interesting to note that, despite this problem, the model works well when both rainfall and flow data are fed into it. With higher rainfall (Figure 10), the responses from 50 mm and 100 mm are more like a traditional unit hydrograph, although clear nonlinearity could be observed between them. It is quite logical that 50 mm rain would generate a lower peak and longer duration, but the difference between them seems quite large. Finally, if an extremely large rainfall is fed into the model which is

Table 3 | RMSE results for 4 rainfalls, 3 flows with different time step

Lead time	Perfect future rainfall Input RMSE	Kernel function	Unknown future rainfall Input RMSE	Kernel function
1	162.1	Linear	162.1445	Linear
2	376.0	RBF	386.0941	Linear
3	583.5	RBF	596.7335	Linear
4	698.8	RBF	768.5272	Linear
5	762.2	RBF	924.1675	Linear
6	828.5	RBF	1051.5	Linear

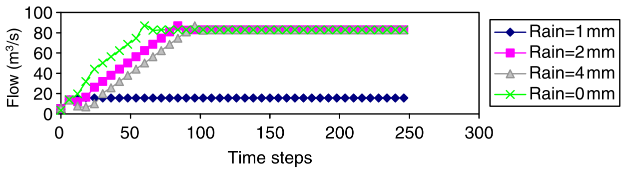

Figure 9 | Model response for 0 mm, 1 mm, 2 mm and 4 mm of rainfall.

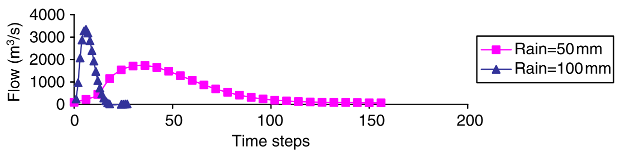

Figure 10 | Model response for 50 mm and 100 mm of rainfall.

beyond the scale limit, the SVM model clearly becomes unstable (Figure 11) and this demonstrates that SVM also suffers the same problems as artificial neural network models.

DISCUSSION

There are two important features in SVM theory: the VC dimension and structural risk minimisation which were developed by Vladimir Vapnik and Alexey Chervonenkis during 1960–1990 (Vapnik 1995). Basically, the VC dimension represents the power of a mathematical model and structural risk minimisation is used to choose the best among the candidate models. For a given model and let h be its VC dimension, Vapnik showed that with probability 1-η, the upper bound for the structural risk is

$$\text{structural risk} = \text{training error} + \sqrt{\frac{h(\log(2N/h) + 1 - \log(\eta/4))}{N}}.$$

Such an equation gives us a way to estimate the error on future data based only on the training error and the VC dimension. As models become more powerful with high VC dimensions (i.e. large h), their training error would be smaller (like the training set curve in Figure 4), but the second part of the equation related to VC dimensions will be larger; hence there is a minimum point along the upper structural risk curve which is similar to the test set curve in Figure 4. In theory, no test data are needed if the structural risk upper bound curve could be computed and an optimum model can be selected from this curve alone. It is clear that the higher the VC dimension, the more powerful (i.e. more complex) a model it is. However, the more power in a model could lead to higher tendency to overfitting and the less power might increase the tendency to under-fitting. In this sense, linear kernels with smaller VC dimensions (with a VC dimension of $n+1$, where n being the number of variables) are less likely to overfit but more likely to underfit. On the other hand, RBF kernels have high VC dimensions (infinite) and are more prone to overfitting. This is an interesting hypothesis but we are unable to prove it since this study has not been carried out to find the tendency of over/under-fitting between the linear kernels and RBF kernels and we suggest that this should be attempted in the future. It should be pointed out that, although the VC dimension has provided a useful theoretical guidance to model selections, in practice, the structural risk minimisation with the VC dimension is too conservative and other methods are more widely used (Moore 2001). Usually the method of cross-validation on the training data and test data is more popular. However, such a method is very computing intensive and has its own pitfalls (e.g. cross-validation could still fail under certain circumstances and there is uncertainty about the optimum number of folds to be used for each specific problem). In this study, fivefold cross-validation with the training data is used as a guide and the test data are used to finally select the model settings. This is quite tedious and in the future it may be useful to investigate the adoption of AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) in hydrological SVM selections which would be much more efficient and practical than the method adopted with cross-validation.

It has been found that the linear kernel SVM outperforms the nonlinear RBF kernel SVM for one lead time step prediction. This could be due to the near linear effect of the

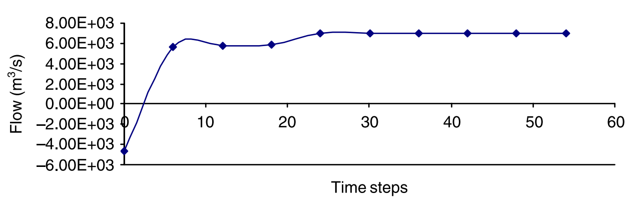

Figure 11 | Predicted flow when input rain is beyond the scale limit.

rainfall–runoff system for small time steps. For larger time steps, the nonlinear effect could not be coped well with by the linear kernel and RBF could perform better due to its higher nonlinear ability. This result seems coincidentally as good as with another study on ANN models (Han *et al.* 2006) where nonlinear ANNs showed no advantage over the linear model at short prediction ranges and outperformed the linear model at longer lead times. It is interesting to note in this study that the linear kernel models outperform RBF models for no-rainfall cases and this could be due to the reduced nonlinear effect from the missing rainfall. Further study over other catchments should be carried out to check if such a phenomenon exists elsewhere.

Another area worth exploring is the application of the model response testing to single-step rainfall with different amounts. If a undesirable response from a certain rainfall has been found, a desirable response curve (e.g. derived by a physically realisable unit hydrograph type, as shown by Yang & Han (2006)) might be used to train SVM so that such unrealistic responses could be removed in the SVM model and, as a result, the model would become more reliable for different situations. It is recommended that further explorations on this should be carried out.

CONCLUSIONS

Despite its success in many other fields, SVM is still in its infancy in hydrological applications. There are many conflicting results in its applications so far (e.g. which kernel function is more suitable in flood modelling?). This study demonstrates that, like artificial neural network models, SVMs also have over-fitting and under-fitting problems, and the over-fitting is more damaging than the under-fitting, which has not been properly addressed by the research community so far. Unlike previous published results, this paper shows that linear and nonlinear kernel functions (i.e. RBF) can yield superior performance against each other under different circumstances in the same catchment. It is not a simple task to simply declare one kernel is better than another one in complicated hydrological simulations. This study also shows an interesting result in the SVM response to different rainfall inputs, where lighter rainfalls would generate very different responses than heavier ones, which is a very useful way to reveal the behaviour and shortcomings of a SVM model. It is still early days for us to understand the implication of this important response feature and future research work will undoubtedly improve our knowledge on this issue and enable modellers to make more use of this information in SVM's development in hydrology. Although SVMs perform better than the benchmark models in this study, it should be noted that a huge amount of time and effort is needed to achieved this (e.g. with trial and error for different input combinations and parameter optimisation) and the result could be very different for other catchments (and indeed it could be different if different test data are used in the same catchment). There is still a long way before this type of model can have any practical impact in the hydrological community, especially among practising hydrologists.

ACKNOWLEDGEMENTS

The comments from two reviewers have been very helpful in improving the paper (resulting in the insertion of the discussion section) and we are grateful for their effort and time in providing those valuable suggestions.

REFERENCES

Beven, K. J. 2000 *Rainfall-runoff Modelling*. John Wiley & Sons. Chichester.

Bray, M. 2002 *Identification of Support Vector Machines for Runoff Modelling*. MEng thesis, Department of Civil Engineering, University of Bristol.

Bray, M. & Han, D. 2004 Identification of support vector machines for runoff modelling. *J. Hydroinf.* **6**, 265–280.

Chang, C. C. & Lin, C. J. 2004*a*. *LIBSVM – A Library for Support Vector Machines*. Available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html.

Chang, C. C. & Lin, C. J. 2004*b*. *A Practical Guide to Support Vector Classification*. Available at: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

Cristianini, N., Campbell, C. & Taylor, J. S. 1999 Dynamically adapting kernels in support vector machines. In *Advances in Neural Information Processing Systems*, vol 12, pp. 204–210. MIT Press, Cambridge, MA.

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

Georgakakos, K. P. & Smith, G. F. 1990 On improved hydrological forecasting - results from a WMO real-time forecasting experiment. *Journal of Hydrology* **114** (1–2), 17–45.

Georgakakos, K. P., Rajaram, H. & Li, S. G. 1988 *On improved operational hydrological forecasting of stream flows*. IAHR Report No. 325.

Gunn, S. 2004 *Matlab SVM Toolbox*. Available at: http://www.ecs.soton.ac.uk/ ~ srg/publications/

Hajjam, S. 1997 *Real Time Flood Forecasting Model Intercomparison and Parameter Updating Using Rain Gauge and Weather Radar Data*. PhD thesis, Telfard Research Institute, University of Salford.

Han, D., Cluckie, I. D., Kang, W. & Xiong, Y. 2002 River flow modelling using reference vector machines. In *Hydroinformatics Proceedings, Cardiff*, vol. B, pp. 1429–1434. IWA Publishing, London.

Han, D., Kwong, T. & Li, S. 2006 Uncertainties in real-time flood forecasting with neural networks. *Hydrol. Process* DOI: 10.1002/hyp.6184, 8 June.

Han, D. & Yang, Z. 2001 River flow modelling using support vector machines. In *XXIX IAHR Congress, Beijing, China, 17–21 September*, pp. 494–499. Qinghua University Press, China.

Joachims, T. 1999 *Estimating the Generalization Performance of a SVM Efficiently*, pp. 25. Universität Dortmund, LS VIII.

Kecman, V. 2001 *Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models*. The MIT Press, Cambridge, MA.

Liong, S. Y. & Sivapragasam, C. 2002 Flood stage forecasting with support vector machines. *J. AWRA* **38** (1), 173–186.

Moore, A. 2001 VC-dimension for characterizing classifiers. *Statistical Data Mining Tutorials*. Available at: http://www.autonlab.org/tutorials/.

Nelles, O. 2001 *Nonlinear System Identification*. Springer-Verlag, Berlin.

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

Smola, A. J. & Schölkopf, B. 1998 *A Tutorial on Support Vector Regression*. NeuroCOLT2 Technical Report Series, NC2-TR-1998-030.

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

Vapnik, V. 1998 *Statistical Learning Theory*. John Wiley & Sons, New York.

WMO 1992 *Simulated Real-Time Intercomparison of Hydrological Models*. WMO Report 779. WMO, Geneva, Switzerland.

Yang, Z. & Han, D. 2006 Derivation of unit hydrograph using a transfer function approach. *Wat. Res. Res.* **42** W01501.