Encapsulation of parametric uncertainty statistics by various predictive machine learning models: MLUE method

Durga L. Shrestha, Nagendra Kayastha, Dimitri Solomatine and Roland Price

ABSTRACT

Monte Carlo simulation-based uncertainty analysis techniques have been applied successfully in hydrology for quantification of the model output uncertainty. They are flexible, conceptually simple and straightforward, but provide only average measures of uncertainty based on past data. However, if one needs to estimate uncertainty of a model in a particular hydro-meteorological situation in real time application of complex models, Monte Carlo simulation becomes impractical because of the large number of model runs required. This paper presents a novel approach to encapsulating and predicting parameter uncertainty of hydrological models using machine learning techniques. Generalised likelihood uncertainty estimation method (a version of the Monte Carlo method) is first used to assess the parameter uncertainty of a hydrological model, and then the generated data are used to train three machine learning models. Inputs to these models are specially identified representative variables. The trained models are then employed to predict the model output uncertainty which is specific for the new input data. This method has been applied to two contrasting catchments. The experimental results demonstrate that the machine learning models are quite accurate. An important advantage of the proposed method is its efficiency allowing for assessing uncertainty of complex models in real time.

Key words | hydrological modelling, machine learning, MLUE, Monte Carlo, uncertainty analysis

Durga L. Shrestha (corresponding author)
CSIRO Land and Water,
Highett,
Australia
E-mail: *durgalal.shrestha@csiro.au*

Nagendra Kayastha
Dimitri Solomatine
Roland Price
UNESCO-IHE Institute for Water Education,
Delft,
The Netherlands

Dimitri Solomatine
Roland Price
Water Resources Section, Delft University of Technology,
Delft,
The Netherlands

INTRODUCTION

Hydrological models, in particular rainfall-runoff models, are simplified representations of reality and aggregate the complex, spatially and temporally distributed physical processes through relatively simple mathematical equations with parameters. The parameters of the rainfall-runoff models can be estimated in two ways (Johnston & Pilgrim 1976). First, they can be estimated from the available knowledge or measurements of the physical process, provided the model parameters realistically represent the measurable physical process. In the second approach, parameter values are estimated by calibration on the basis of the input and output measurements in situations when the parameters do not represent directly measurable entities or when it is too costly to measure them in the field. Conceptual rainfall-runoff models usually contain several parameters, which cannot be directly measured. Manual adjustment of the parameter values is labour intensive and its success is strongly dependent on the experience of the modeller. In the last two decades, a number of automated routines have been suggested (see e.g. Duan *et al.* 1992; Yapo *et al.* 1996; Solomatine 1999; Madsen 2000; Vrugt *et al.* 2005).

While considerable attention has been given to the development of calibration methods which aim to find a single best or Pareto set of values for the parameter vector, a realistic estimation of parameter uncertainty received

doi: 10.2166/hydro.2013.242

special attention over the last few years. It is now being broadly recognised that proper consideration of uncertainty in hydrologic predictions is essential for purposes of both research and operational modelling (Wagener & Gupta 2005). The value of hydrologic prediction for water resources-related decision-making processes is limited if reasonable estimates of the corresponding predictive uncertainty are not provided (Georgakakos *et al.* 2004). The research community has done quite a great deal in moving towards the recognition of the necessity of complementing point forecasts of decision variables by the uncertainty estimates, and nowadays it is widely recognised that along the modelling per se, there is a need to (i) understand and identify sources of uncertainty, (ii) quantify uncertainty, (iii) evaluate the propagation of uncertainty through the models, and (iv) find means to reduce uncertainty. Incorporating uncertainty into deterministic forecasts helps to enhance the reliability and credibility of the model outputs.

One may observe a significant proliferation of uncertainty analysis methods published in the academic literature, trying to provide meaningful uncertainty bounds of the model predictions. Pappenberger *et al.* (2006) provide a decision tree to find the appropriate method for a given situation. However, the methods to estimate and propagate this uncertainty have so far been limited in their ability to distinguish between different sources of uncertainty and in the use of the retrieved information to improve the model structure analysed. These methods range from analytical and approximation methods (see e.g. Tung 1996) to Monte Carlo (MC) sampling-based methods (e.g. Beven & Binley 1992; Kuczera & Parent 1998; Thiemann *et al.* 2001) with the use of Bayesian approaches to determine the posterior distributions; methods based on the analysis of model errors (e.g. Montanari & Brath 2004); machine learning methods (Shrestha & Solomatine 2008; Solomatine & Shrestha 2009; Shrestha *et al.* 2009), and methods based on fuzzy set theory (see e.g. Maskey *et al.* 2004).

Due to complexities, or even impossibility of using analytical methods to propagate uncertainty from parameters to outputs for complex models, MC-based (sampling) techniques have been widely applied in studying uncertainty of hydrological models. A version of the MC simulation method was introduced under the term 'generalised likelihood uncertainty estimation' (GLUE) by Beven & Binley (1992). GLUE is one of the popular methods for analysing parameter uncertainty in hydrological modelling and has been widely used over the past 15 years to analyse and estimate predictive uncertainty, particularly in hydrological applications (see e.g. Freer *et al.* 1996; Beven & Freer 2001; Montanari 2005). Users of the GLUE (and actually of any MC method in general) are attracted by its simple understandable ideas, relative ease of implementation and use, and its ability to handle different error structures and models without major modifications to the method itself. Despite its popularity, there are theoretical and practical issues related with the GLUE method reported in the literature. For instance, Mantovan & Todini (2006) argue that GLUE is inconsistent with the Bayesian inference processes such that it leads to an overestimation of uncertainty, both for the parameter uncertainty estimation and the predictive uncertainty estimation. For the account of different views at the methodological correctness of GLUE, readers are referred to the citation above and the subsequent discussions in the *Journal of Hydrology* in 2007 and 2008, and to the papers by Stedinger *et al.* (2008) and Vrugt *et al.* (2009).

Since MC-based methods require a large number of samples (or model runs), their applicability is sometimes limited to simple models. In the case of computationally intensive models, the time and resources required by these methods could be prohibitively expensive. Alternative approximation methods have been developed (e.g. moment propagation techniques), which under certain assumptions are able to calculate directly the first and second moments without the application of MC simulation (see e.g. Rosenblueth 1981; Harr 1989; Melching 1992). A number of methods allow for reducing the number of MC simulation runs, for instance, Latin hypercube sampling (see e.g. McKay *et al.* 1979) but they may fail to provide reliable estimates of uncertainty.

For models with a large number of parameters, the sample size from the respective parameter distributions must be very large in order to achieve a reliable estimate of uncertainties (Kuczera & Parent 1998) (it is worth mentioning that this is a problem for all methods based on sampling and multiple model runs). One of the ways to address the problem of computational complexity in optimisation, random search or MC simulation, is to use a limited number of samples of parameter vectors and run the

hydrologic or hydraulic model in order to generate a data set which is then used as the calibration set for building an approximating regression model. This latter (fast) model (also called a meta-model, or surrogate model) is then used instead of the (slower) original model; such approach is widely employed in industrial design optimisation, and in water resources problems was used, for example by Solomatine & Torres (1996) in model-based optimisation, and Khu & Werner (2003) in reducing the number of MC simulations.

Yet another approach is to use more efficient sampling strategies, as was done by Blasone *et al.* (2008) who used adaptive Markov chain Monte Carlo sampling within the GLUE methodology to improve the sampling of the high probability density region of the parameter space. Other examples of this approach are the delayed rejection adaptive Metropolis method (Haario *et al.* 2006), and the differential evolution adaptive Metropolis method, DREAM (Vrugt *et al.* 2009).

One of the practical observations concerning the GLUE method is that in many cases the percentage of observations falling within the prediction limits provided by GLUE is smaller than the given confidence level used to produce these prediction limits (see e.g. Montanari 2005). Xiong & O'Connor (2008) modified the GLUE method to somehow resolve this issue, so that the prediction limits would envelope the observations better.

There is, however, an issue which is not widely discussed in the literature, and this is the assessment of model uncertainty when it is used in operation, i.e. when the new input data are fed into the model, in other words, uncertainty prediction. The MC simulation provides only the averaged uncertainty estimates based on the past data, but in real time forecasting situations there may be simply little time to perform the MC simulations for the new input data in order to assess the model uncertainty for a new situation.

Recently, we proposed to use artificial neural network (ANN) to emulate the MC simulations results obtained for the past data, and named this method MLUE – machine learning in parameter uncertainty estimation (Shrestha *et al.* 2009). The idea of this method is to use the data from MC simulations to train a statistical or machine learning model to (with specially selected inputs) predict the quantiles of the model error distribution. MLUE method needs only a single set of MC simulations in off line mode and allows one to predict the uncertainty bounds of the model prediction when the new input data are observed and fed into hydrological models (whereas the standard MC approach requires new multiple model runs for each new input).

In a comparison with previous study of Shrestha *et al.* (2009), the main contributions of this study are to (i) provide an extensive review of the state art of the uncertainty analysis methods used in hydrology, (ii) generalise the methodology and to extend it further to approximate probability distribution function of the model outputs, (iii) apply methodology to different study area, (iv) employ and compare different machine learning models to emulate MC simulation results, and (v) compare the methodology with yet another uncertainty analysis method. The HBV (Hydrologiska Byråns Vattenbalansavdelning) hydrological models of the Brue catchment in UK and Bagmati catchment in Nepal are used as case studies.

MACHINE LEARNING METHODS

In this section, we introduce briefly the main notions of machine learning and the methods used. Major focus of machine learning is to automatically produce (induce) predictive models from data. A machine learning algorithm estimates an unknown mapping (or dependency) between the inputs (predictors) and outputs (predictands) of a physical system from the available data (Mitchell 1997). As such a dependency (model) is discovered, it can be used to predict the future outputs of the system from the known input values. Machine learning techniques, based on observed data $D = (X, \mathbf{y}) = \{\mathrm{x}_t, \mathbf{y}_t\}$, $t = 1, 2, \ldots, N$, try to identify (learn) the target function $f(\mathbf{x}_t, \mathbf{w})$ describing how the real system behaves, where X is the matrix ($\mathbf{x}$, vector) of the input data, $\mathbf{y}$ is the vector of systems' response, N is the number of data, $\mathbf{w}$ is the parameter vector of the function. Learning (or 'training') here is the process of minimising the difference between observed response y and model response $\hat{y}$ through an optimisation procedure. Such a model f is often called a 'data-driven model'. For a recent overview of data-driven modelling in water-related issues, see for example Solomatine & Ostfeld (2008), Maier *et al.* (2010), and Elshorbagy *et al.*

(2010a, b). A review of the application of machine learning techniques to estimate the uncertainty of rainfall-runoff models can be found in Shrestha & Solomatine (2008) and Shrestha *et al.* (2009).

Three machine learning methods namely ANN, model tree, and locally weighted regression (LWR), are used in this study. Among them, ANN is the most popular technique and has been extensively used in hydrological modelling over the past 15 years (see for example early papers by Minns & Hall (1996), Maier & Dandy (2000), Abrahart & See (2000), Govindaraju & Rao (2000), Dawson & Wilby (2001) and Dibike & Solomatine (2001)). The following sections present a brief overview of the other two methods which are less known in the water and environmental modelling community.

Model trees

A model tree (MT) is a hierarchical (or tree-like) modular model which has splitting rules in non-terminal nodes and linear regression functions at the leaves of the tree. In fact, it is a piece-wise linear regression model. In the mid 1980s, the Australian researcher Dr J. Ross Quinlan suggested the so-called M5 algorithm to build MT (Witten & Frank 2000); this is an iterative scheme that progressively splits the examples in the space of inputs $\{x_1, x_2, \ldots x_n\}$ using the criterion $x_i < A$, where i and A are the values chosen at each iteration according to the 'splitting criterion'. This criterion is based on the standard deviation of the output values (in rainfall-runoff models this is runoff) in the resulting subsets, which is used as a measure of the possible regression model error if it is built for this subset. All values of i and A are examined, and the M5 algorithm performs the splits that ensures the small standard deviation in the resulting subsets; the splitting iterations continue trying to perform the best possible split of each of the resulting subsets. At a certain moment splits are stopped, and linear regression models are built for each of the resulting subsets. The splitting procedure can be presented as a hierarchy, or a tree, where the splitting rules are in the intermediate nodes and the linear models are associated with the tree leaves. MT can tackle tasks with very high dimensionality, up to hundreds of variables. Compared to other machine learning techniques, MT learning is fast and the results are interpretable. An example of using MT in the role of a data-driven rainfall-runoff model can be found in Solomatine & Dulal (2003).

Locally weighted regression

LWR is a method that builds a regression model selecting only a limited number of examples close to the vector of input $\mathbf{x}_q$ (often called a query vector). The selected examples are assigned weights according to their distance to the query vector, and regression equations are generated using the weighted data. The word 'local' in the 'locally weighted regression' means that the function is approximated based on data in the locality of the query vector, and it is 'weighted' because the contribution of each training example is weighted by its distance from the query vector. The regression function f built for the neighbourhood of the query vector $\mathbf{x}_q$ can be a linear or non-linear function, an ANN, etc. Various distance-based weighting schemes can be employed (given in Appendix A, available online at http://www.iwaponline.com/jh/016/242.pdf). For a detailed description of LWR method, the readers are referred to Aha *et al.* (1991), and its application in rainfall-runoff modelling is reported in Solomatine *et al.* (2008).

METHODOLOGY

The original version of the main ideas of the MLUE method can be found in Shrestha *et al.* (2009) (in open access), here we present only a brief description of it, however in a more formalised and generalised fashion. The basic idea is to estimate the uncertainty of the hydrological model under a number of the following assumptions. First, that the model uncertainty is different in various hydro-meteorological conditions, and depends on the corresponding forcing input and the model states (e.g. rainfall, antecedent rainfall, soil moisture, etc.). Second, that the uncertainties associated with the prediction of the hydrological variables such as runoff in similar hydro-meteorological conditions are also similar. By 'hydrological conditions', we mean a vector of variables representing such conditions – the combination of the particular values of the input and state variables (possibly lagged and transformed), which are seen as the driving forces

generating runoff. This assumption is quite natural: e.g. typically prediction error (and hence uncertainty) is higher in case of peak flows during extreme events compared to the low flows – however, the proper statistical analysis to support the validity of this assumption is still to be done.

The flow chart of the MLUE methodology is presented in Figure 1. Let us assume that S various vectors of parameters or inputs are sampled and for each of them the hydrological model M is run generating a time series of the model output $\hat{y}$. The results are presented in the matrix form $Y = \{\hat{y}_{t,s}\}$, where $t = 1,\ldots,N$, $s = 1,\ldots,S$, N is the number of time steps, S is the number of simulations. Note that each row of the matrix Y corresponds to the particular forcing vector $\mathbf{x}'_t$, i.e.

$$\{\hat{y}_{t,1},\ldots,\hat{y}_{t,s}\} = \{M(\mathbf{x}'_t, \theta_1),\ldots,M(\mathbf{x}'_t, \theta_s)\} \quad (1)$$

where θ is the parameter vector of the model M. Similarly, each column of matrix Y, i.e. $\{\hat{y}_{1,s},\ldots,\hat{y}_{t,s}\}^{\mathrm{T}}$ is one realisation of MC simulations corresponding to the parameter set θ_s. Note that Equation (1) does not represent predictive uncertainty $P_t(y|\hat{y})$ which is the uncertainty related to the actual value given the model predictions and all the information and knowledge available up to the present (Mantovan & Todini 2006; Todini 2008). Rather, by and large it represents uncertainty of the model predictions due to the parameter

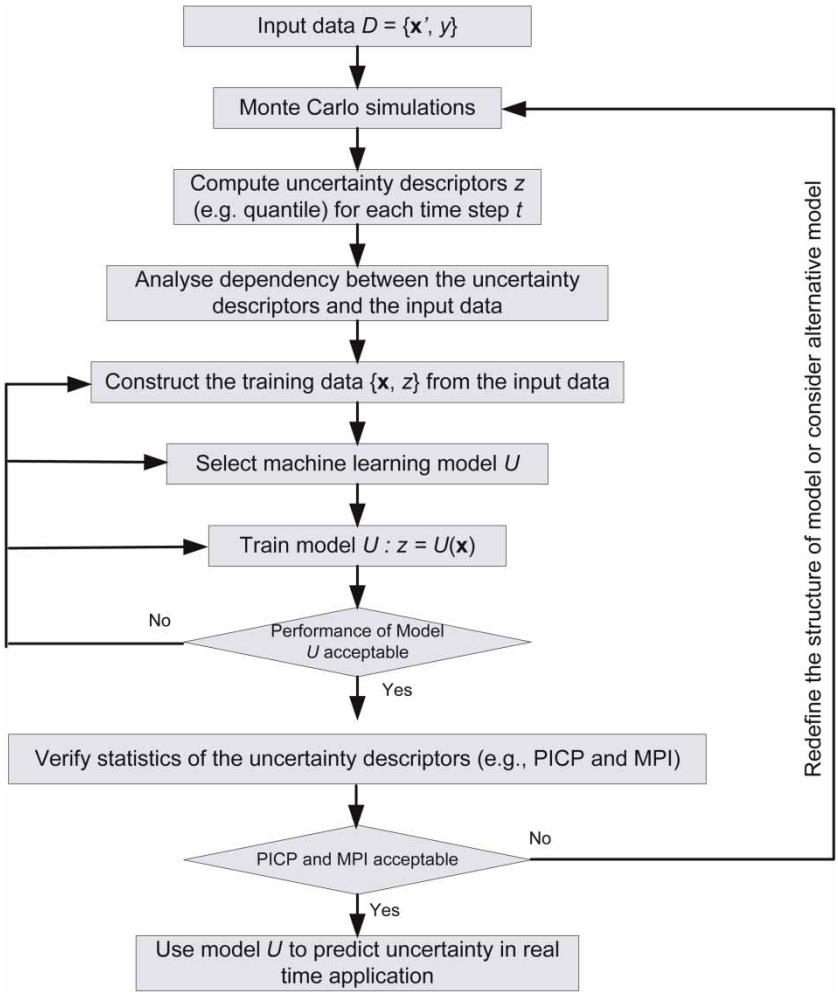

Figure 1 | Schematic diagram of the MLUE method.

uncertainty, i.e. $P_t(\hat{y}|\mathbf{x}_t, \theta)$. Estimating quantiles of the distribution of $\hat{y}$ or probability density $P_t(\hat{y}|\mathbf{x}_t, \theta)$ is not always practical in real time application (e.g. for computationally expensive environmental models). However, we can approximate $P_t(\hat{y}|\mathbf{x}_t, \theta)$ by estimating its quantiles using the MLUE method. Our intention is to build a regression (machine learning) model U which is relatively efficient (fast) and can encapsulate these uncertainty results in the following form:

$$\text{statistical properties } \mathbf{z} \text{ of } \hat{y}_t = U(\mathbf{x}_t) \tag{2}$$

where $\mathbf{z} = \{z_1, \ldots, z_K\}$ is a set of desired statistical properties; $\mathbf{x}$ is the input vector of the model U which is constructed from the forcing input variables $\mathbf{x}$', model state $\mathbf{s}$ and possibly model output $\hat{y}$ (all possibly combined, transformed and/or lagged). A way to construct the input space $\mathbf{x}$ is described in next section. To characterise the uncertainty of the model M prediction, the following uncertainty descriptors can be considered.

1. The prediction variance $\sigma_t^2(\hat{y}_t)$

$$\sigma_t^2(\hat{y}_t) = \frac{1}{S-1}\sum_{s=1}^{S}(\hat{y}_{t,s} - \bar{y}_{t,s})^2 \tag{3}$$

where $\bar{y}_{t,s}$ is the mean of MC realisations at the time step t.

2. The prediction quantile $Q_t'(p)$ of $\hat{y}_t$ corresponding to the pth [0, 1] quantile

$$P(\hat{y}_t < Q_t'(p)) = \sum_{s=1}^{S} w_s | \hat{y}_{t,s} < Q_t'(p) \tag{4}$$

where w_s is the weight given to the model output at simulation s, $\hat{y}_{t,s}$ is the value of model output at the time t simulated by the model $M(\mathbf{x},\theta_s)$. The use of weights is assumed in case of using GLUE framework.

3. The conditional prediction quantile $Q_t(p)$ corresponding to the pth quantile

$$Q_t(p) = Q_t'(p) - \hat{y}_t^{opt} \tag{5}$$

where $\hat{y}_t^{opt}$ is the output of the calibrated (optimal) model. Note that the quantiles $Q_t(p)$ obtained in this way are conditional on the model structure, inputs and other parameters (e.g. in case of using GLUE framework this is the likelihood weight vector w_s).

4. The prediction intervals $[PI_t^L(\alpha),\ PI_t^U(\alpha)]$ for the given confidence level of $1-\alpha$ $(0 < \alpha < 1)$

$$PI_t^L(\alpha) = Q_t(\alpha/2),\ PI_t^U(\alpha) = Q_t(1-\alpha/2) \tag{6}$$

where $PI_t^L(\alpha)$ and $PI_t^U(\alpha)$ are the differences between the model output and the lower and upper bounds of the prediction intervals (PI) respectively, corresponding to the $1-\alpha$ confidence level.

If U in Equation (2) is treated as a quantile, the general equation for calculating the conditional prediction quantile (Equation (5)) can be presented as

$$Q_t(p) = U(\mathbf{x}_t) + \xi \tag{7}$$

where ξ is the error between the target quantile and the predicted quantile by the machine learning model. In particular, the two quantiles that represent the bounds of the PI (Equation (6)) can be calculated as follows:

$$\begin{aligned} PI_t^L(\alpha) &= U_L(\mathbf{x}_t) + \xi_L \\ PI_t^U(\alpha) &= U_U(\mathbf{x}_t) + \xi_U \end{aligned} \tag{8}$$

Since these prediction quantiles are derived from the current value of the model output (Equation (5)), then the general model for the predictive quantile can be presented as

$$Q_t'(p) = U(\mathbf{x}_t) + \hat{y}_t^{opt} \tag{9}$$

In particular, the upper and lower bounds of the PI of the model output are given by

$$\begin{aligned} PL_t^L &= U_L(\mathbf{x}_t) + \hat{y}_t^{opt} \\ PL_t^U &= U_U(\mathbf{x}_t) + \hat{y}_t^{opt} \end{aligned} \tag{10}$$

where U_L and U_U are the machine learning models for the lower and upper bounds of the PIs, respectively. It is worthwhile to mention that Equation (10) is valid for the

uncertainty descriptors in Equation (6) and it is assumed that there is an optimal (calibrated) model M.

Model U, after being trained on the historical calibration data (generated by MC simulations), encapsulates the underlying dynamics of the uncertainty descriptors of the MC simulations and maps the input (or more precisely, vectors in space $\mathbf{x}$) to these descriptors. The model U can be of various types, from linear to non-linear regression models such as an ANN. The choice of model depends on the complexity of the problem to be handled and the availability of data. Once the model U is trained on the calibration data, it can be employed in operation to estimate the uncertainty descriptors such as quantiles for the new unseen input data vectors.

SELECTION OF INPUT VARIABLES FOR THE UNCERTAINTY MODEL

Selection of appropriate variables to serve as model inputs for the uncertainty model U is extremely important as they should be relevant for the particular modelling exercise and the type of the process model M and its inputs. For this, the domain (expert) knowledge and analysis of causal relationship between inputs and outputs should be used in combination. The following variables (or their combinations) of the process model M are considered as the candidates for being the input variables for model U: (i) input variables; (ii) state variables; (iii) outputs; (iv) time derivatives (rate of change) of the input data and state variables; (v) lagged variables of input, state and observed output; and (vi) other data from the physical system that may be relevant to the uncertainty descriptors. Since the nature of models M and U is very different, analysis techniques such as linear correlation or average mutual information between the uncertainty descriptors and the input data listed above may help in choosing the relevant input variables. Based on the domain knowledge and analysis of causal relationships, several structures of input data can be tested to select the optimal input data structure.

For example, if a model M is a conceptual hydrological model, it would typically use rainfall (R_t) and evapotranspiration (E_t) as input variables to simulate the output variable runoff (Y_t). However, the uncertainty model U, whose aim is to predict the error distribution of the simulated runoff, may be trained with the possible combination of rainfall and evapotranspiration (or effective rainfall), their several past (lagged) values, the lagged values of runoff, and, possibly, their derivatives and/or combinations.

VERIFICATION

The uncertainty model U can be validated in two ways: (i) measuring its predictive capability in approximating the uncertainty descriptors of the realisations of MC simulations; and (ii) measuring the 'quality' of representing uncertainty by using some indices.

Two performance measures, such as coefficient of correlation (CoC) and the root mean square error (RMSE), are widely used to measure the predictive capability of models, and they can be employed for the uncertainty model as well. Beside these numerical measures, the graphical plots such as scatter and time series plot of the uncertainty descriptors obtained from the MC simulations and their predicted values are used to judge the performance of the uncertainty model U.

For assessing the quality of model U, we use two measures (Shrestha & Solomatine 2006). Model U is considered to be good if PI coverage probability and mean PI calculated for U are close to those calculated for the MC simulation data from which is used to train U.

1. PI coverage probability (PICP). It measures the percentage of observations falling inside the PI and ideally should be equal to the confidence bounds used to generate these intervals. It is an indication of the quality of model U. PICP is given by:

$$\text{PICP} = \frac{1}{N}\sum_{t=1}^{N} C_t \quad \text{with } C = \begin{cases} 1, PL_t^L \le y_t \le PL_t^U \\ 0, \text{otherwise} \end{cases} \tag{11}$$

   where y_t is the observed model output at the time t.
2. Mean prediction interval (MPI). It measures the average width of the PIs (it gives an indication of how large the

uncertainty is) and given by:

$$\mathrm{MPI} = \frac{1}{N}\sum_{t=1}^{N}(PL_t^U - PL_t^L) \quad (12)$$

Besides these uncertainty statistics, a visual inspection of the plot of uncertainty bounds and of the observed model output can additionally provide significant information about how effective the uncertainty model is in enclosing the observed model outputs along the different input regimes (e.g. low, medium or high flows in hydrology). More detailed description of the performance measures can be found in Shrestha & Solomatine (2006) and Shrestha *et al.* (2009).

HYDROLOGICAL MODEL

A simplified version of HBV model (Bergström 1976) was used. This is a lumped conceptual hydrological model which includes conceptual numerical descriptions of the hydrological processes at catchment scale. The model comprises subroutines for snow accumulation and melt, soil moisture accounting procedure, routines for runoff generation, and a simple routing procedure. The model has 13 parameters; however only nine parameters (see Table 1) are effective when there is no snowfall.

STUDY AREA

The MLUE approach has been tested to two contrasting catchments: Brue and Bagmati. The Brue catchment is located in South West of England (Figure 2). It has a drainage area of 135 km^2 with the average annual rainfall of 867 mm and the average river flow of 1.92 $m^3\ s^{-1}$ (measured in a period from 1961 to 1990). The hourly rainfall, discharge, and the weather data (temperature, wind, solar radiation, etc.) are computed from the 15 minutes resolution data which are available from a period 1993 to 2000. The catchment average rainfall data are used in the study. The hourly potential evapotranspiration is computed using the modified Penman method recommended by FAO (Allen *et al.* 1998). One year hourly data from June 24 1994 to June 24 1995 is selected for calibration of the HBV model and data from June 25 1995 to May 31 1996 for the verification (testing) of the HBV model.

The Bagmati catchment is located in the central mountainous region of Nepal (Figure 3). Compared to the Brue, the size of the Bagmati catchment is bigger, the length of the data is larger, the temporal resolution of the data is coarse (daily), and the quality of the data is comparatively poorer. It encompasses nearly 3,700 km^2 within Nepal and reaches the Ganges River in India. The catchment area draining to the gauging station at Pandheradobhan is about 2,900 km^2. Two thousand daily records from 1 January

Table 1 | Ranges and the optimal values of the HBV model parameters

		Brue		Bagmati	
Parameter	**Description and Unit**	**Range**	**Value**	**Range**	**Value**
FC	Maximum soil moisture content (L)	100–300	160.335	50–500	450
LP	Limit for potential evapotranspiration (–)	0.5–0.99	0.527	0.3–1	0.90
ALFA	Response box parameter (–)	0–4	1.54	0–4	0.1339
BETA	Exponential parameter in soil routine (–)	0.9–2	1.963	1–6	1.0604
K	Recession coefficient for upper tank (/T)	0.0005–0.1	0.001	0.05–0.5	0.3
K4	Recession coefficient for lower tank (/T)	0.0001–0.005	0.004	0.01–0. 5	0.04664
PERC	Maximum flow from upper to lower tank (L/T)	0.01–0.09	0.089	0–8	7.5
CFLUX	Maximum value of capillary flow (L/T)	0.01–0.05	0.0038	0–1	0.0004
MAXBAS	Transfer function parameter (T)	8–15	12	1–3	2.02

Note: The uniform ranges of parameters are used both for calibrating the HBV model, and for analysis of the parameter uncertainty of the HBV model.

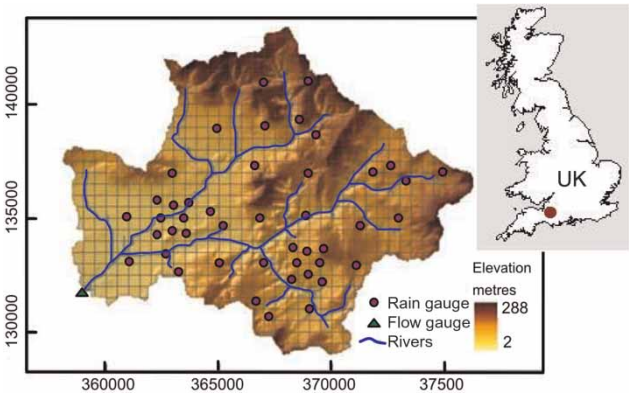

Figure 2 | The Brue catchment showing dense rain gauges network (reproduced from Shrestha & Solomatine (2008) with permission from the International Association for Hydraulic Research). The horizontal and vertical axes refer to the easting and northing in British national grid reference co-ordinates. Circles denote the rainfall stations and triangles denote the discharge gauging stations. The location of the Brue catchment (solid circle) in the map of UK is shown in the inset.

Figure 3 | Location map of the Bagmati catchment considered in this study. Discharge measured at Pandheradobhan is used for the analysis (adopted from Solomatine *et al.* (2008)).

1988 to 22 June 1993 are selected for calibration of the process model (HBV hydrological model) and data from 23 June 1993 to 31 December 1995 are used for the verification of the process model. The first two months of calibration data are used as the warming-up period and hence excluded in the study. In separation of the 8 years of data into calibration and verification sets, we follow the previous study of Solomatine & Shrestha (2009).

EXPERIMENTAL SETUP

Uncertainty analysis

Hydrological models are calibrated by using adaptive cluster covering (Solomatine 1999), an efficient randomised search method implemented in GLOBE software. The GLUE method is used in uncertainty analysis because it has now been widely used for uncertainty estimation in a variety of models of complex environmental systems (we do not discuss here how much it does (or does not) follow the Bayesian framework). No model is perfect (free from structural error), observation and input data are not free from errors, so Monte Carlo simulation results considering only parameter uncertainty are not free from these sources of error. We tried to reduce such errors as much as possible by selecting the best (automatically calibrated) model (which is reasonably accurate), and quality control of the input and observation data. Of course, uncertainty results only considering parameter uncertainty from GLUE are contaminated by other sources of error, again, we tried to minimise them. In this, we follow many researchers using parametric uncertainty analysis. Though Beven claimed that GLUE can be applied to other sources of error as well, we explicitly consider only parameter uncertainly in this study. So we can assume that the uncertainty results produced by GLUE represent mostly the parametric uncertainty per se, and neglecting the contamination by other sources of error seems to be reasonable thing to do. It is also worth noting that because of informal likelihood function and cut-off threshold value used in GLUE to select the behaviours parameter sets, GLUE does not consider complete parameter uncertainty in statistical sense (Vrugt *et al.* 2009).

The convergence of MC simulations is assessed to determine the number of samples required to obtain the reliable results by authors in the previous publication (see Shrestha *et al.* 2009) and is not reported here. The parameters of the HBV model are sampled using non-informative uniform sampling without prior knowledge of individual parameter distributions other than a feasible range of values (see Table 1). We use the sum of the squared errors as the basis to calculate the generalised likelihood measure (see

Freer *et al.* 1996) in the form:

$$L(\theta_s|D) = \left(1 - \frac{\sigma_e^2}{\sigma_{obs}^2}\right)^{\lambda} \quad (13)$$

where $L(\theta_s|D)$ is the generalised likelihood measure for the sth model (with parameter vector θ_s) conditioned on the observations D, σ_e^2 is the associated error variance for the sth model, σ_{obs}^2 is the observed variance for the period under consideration, λ is a user defined parameter. We set λ to 1, so Equation (13) is equivalent to the Nash–Sutcliffe coefficient of efficiency (CoE) (Nash & Sutcliffe 1970).

The threshold value of CoE =0 is selected to classify simulation as either behavioural or non-behavioural. The number of behavioural models is set to 25,000, which is based on the convergence analysis of MC simulations. Various uncertainty descriptors such as variance, quantiles, PIs and estimates of the probability distribution functions are computed from these 25,000 MC realisations. Note that these descriptors are computed using likelihood measure (Equation (13)) as weights w_s in Equation (4). The model parameters ranges used for MC sampling are given in Table 1. For Bagmati catchment, first 122,132 MC samples are generated by setting threshold value of 0.7 to obtain 25,000 behavioural samples. However, to make consistent with the Brue catchment experiment, model simulations with negative CoE are removed for further analysis, leaving 116,153 samples out of 122,132.

Input variables and data

Selection of input variables for the machine learning model U are based on the methods outlined in the previous section and publication of Shrestha *et al.* (2009) and is not discussed here; they are constructed from the forcing input variables (e.g. rainfall, evapotranspiration) used in the process models, and the observed discharge. The selected input variables are RE_{t-9a}, Y_{t-1}, ΔY_{t-1} for the Brue catchment and RE_{t-0}, RE_{t-1}, Y_{t-1}, Y_{t-2}, for the Bagmati catchment where

$RE_{t-\tau}$: effective rainfall at time $t-\tau$;
$Y_{t-\tau}$: discharge at time $t-\tau$; where τ is lag time;
RE_{t-9a}: the average of RE_{t-5}, RE_{t-6}, RE_{t-7}, RE_{t-8}, RE_{t-9};
$\Delta Y_{t-1} = Y_{t-1} - Y_{t-2}$.

For the Bagmati catchment, since the resolution of data is daily (as opposed to hourly for the Brue), we do not consider the derivative (stepwise difference) of the flow as input to the model.

The same data sets used for calibration and verification of the HBV model are used for training and verification of model U, respectively. However, for proper training of the machine learning models, the calibration data set is segmented into the two subsets: 15% of data sets for cross-validation (CV) and 85% for training per se. CV data set was used to identify the best structure of machine learning models.

Machine learning models

A multilayer perceptron neural network with one hidden layer is used; the Levenberg–Marquardt algorithm is employed for its training. The hyperbolic tangent function is used for the hidden layer, and the linear transfer function – for the output layer. The maximum number of epochs is fixed to 1000. Trial and error method is adopted to find the optimal number of neurons in the hidden layer; we tried the number of neurons ranging from 1 to 10. It was found that 7 and 8 neurons for lower and upper PI, respectively, gave the lowest CV error for the Brue. For the Bagmati catchment, the number of hidden neurons reduced to 5 and 7.

Experiments with MT are carried out with various values of the pruning factor that controls the complexity of the generated model (i.e. number of the linear models) and hence the generalising ability of the model. We report the results of the MT which have a moderate level of complexity. Note that CV data set has not been used in the MT, rather it uses the whole calibration data set to build the model.

In the LWR model, we vary two important parameters – number of neighbours and the weight functions (see Appendix A, available online at http://www.iwaponline.com/jh/016/242.pdf). Several experiments are done with different combination of these values and the best results are obtained with five neighbours and the linear weight function for the Brue and 11 neighbours and Tricube weight function for the Bagmati catchment.

Modelling the probability distribution function

In the previous study (Shrestha *et al*. 2009), we estimate the 90% PIs by building only two models predicting the 5 and 95% quantiles. In this paper, the methodology is extended to predict several quantiles of the model outputs to estimate the distribution functions (CDF) of the model outputs generated by the MC simulations. The methodology applied to estimate only two quantiles can be extended to approximate the full distribution of the model outputs. The procedures to estimate the CDF of the model outputs consists of (i) deriving the CDF of the realisations of the MC simulations in the calibration data, (ii) selecting several quantiles of the CDF in such a way that these quantile can approximate the CDF, (iii) computing corresponding prediction quantiles using Equation (5), (iv) constructing and training separate machine learning models for each prediction quantiles, (v) using these models to predict the quantiles for the new input data vector, and (vi) constructing a CDF from these discrete quantiles by interpolation. This CDF will be approximation to the CDF of the MC simulations.

We select 19 quantiles from 5 to 95% with uniform interval of 5%, and then an individual machine learning model is constructed for each quantile using the same structure of the input data and the model that was used for modelling two quantiles. In principle, the optimal set of input data and the model structure could be different for each quantile, but we leave this investigation to future studies.

RESULTS

The HBV model is calibrated maximising CoE. CoE values of 0.96 and 0.83 are obtained for the calibration period in the Brue and Bagmati catchment, respectively. We also experimented with more sophisticated performance measures taking into account different temporal scales and using step-wise line search (Kuzmin *et al*. 2008). The model is validated by simulating the flows for the independent verification data set, and CoE is 0.83 and 0.87 in the Brue and Bagmati catchments, respectively. HBV model is quite accurate for the Brue catchment but its error (uncertainty) is quite high during the peak flows. Note that for the Bagmati catchment, the standard deviation of the observed discharge in the verification period is 54% higher than that in the calibration period which apparently increases performance in the verification period.

Figure 4 shows a comparison of the 90% prediction bounds estimated by the GLUE and the three machine learning models in the verification period for the Brue catchment. One can see a noticeable difference among them for predicting the lower and upper bounds of PI. For example, in the second peak of Figure 4(a), the upper bound of PI is underestimated by ANN compared to the MT and LWR. However, the lower bound is well approximated by the ANN compared to the other models. Furthermore, in Figure 4(b), the ANN is overestimating two peaks, while the MT and LWR models underestimate them (Figure 4(d) and (f)). From Figure 4, it can be seen that the results of the three models are comparable. They reproduce the MC simulations uncertainty bounds reasonably well except for some peaks, in spite of the low correlation of the input variables with the PIs. The predicted uncertainty bounds follow the general trend of the MC uncertainty bounds although some errors can be noticed and the model fails to capture the observed flow during one of the peak events (Figure 4(a), (c), and (e)).

For the Bagmati catchment, it is found that only 49.79% of observed discharge data is inside the 90% prediction bounds computed by the GLUE method in the calibration period and 61.48% in the verification period. Therefore, we follow the modified GLUE method (denoted by mGLUE) (Xiong & O'Connor 2008) to improve the capacity of the prediction bounds to capture the observed runoff data. mGLUE method uses the bias corrected MC simulations to estimate the uncertainty bounds. Compared to the original GLUE method (Beven & Binley 1992), the mGLUE method includes two more procedural steps. Firstly, for each behavioural parameter set, a simulation bias curve is constructed on the basis of the simulation series that are obtained using the calibration data. Thus, for a number S of the behavioural parameter sets, there will be S different simulation bias curves. Secondly, at each time step, with the new data input, all the different prediction values for the same observation are corrected by dividing by a common median bias value, before the derivation of the prediction limits.

Figure 5 presents the 90% prediction bounds estimated by the mGLUE and the three machine learning models in

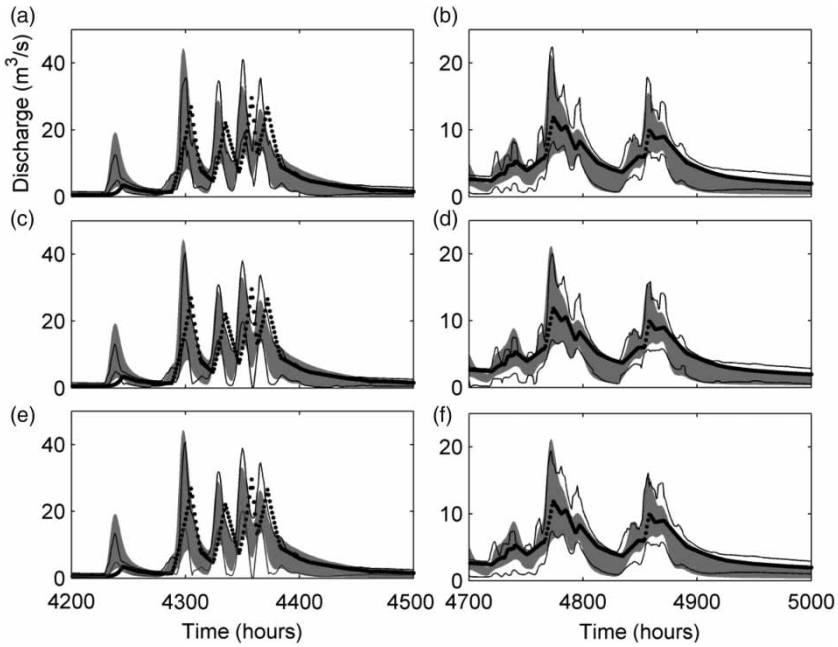

Figure 4 | Hydrograph of 90% prediction bounds estimated by GLUE and machine learning methods for the Brue catchment in parts of the verification period. The black dots indicate the observed discharges and the dark grey shaded area – the prediction uncertainty that results from GLUE. The black lines denote the prediction uncertainty estimated by neural networks ((a) and (b)), model trees ((c) and (d)) and locally weighted regression ((e) and (f)).

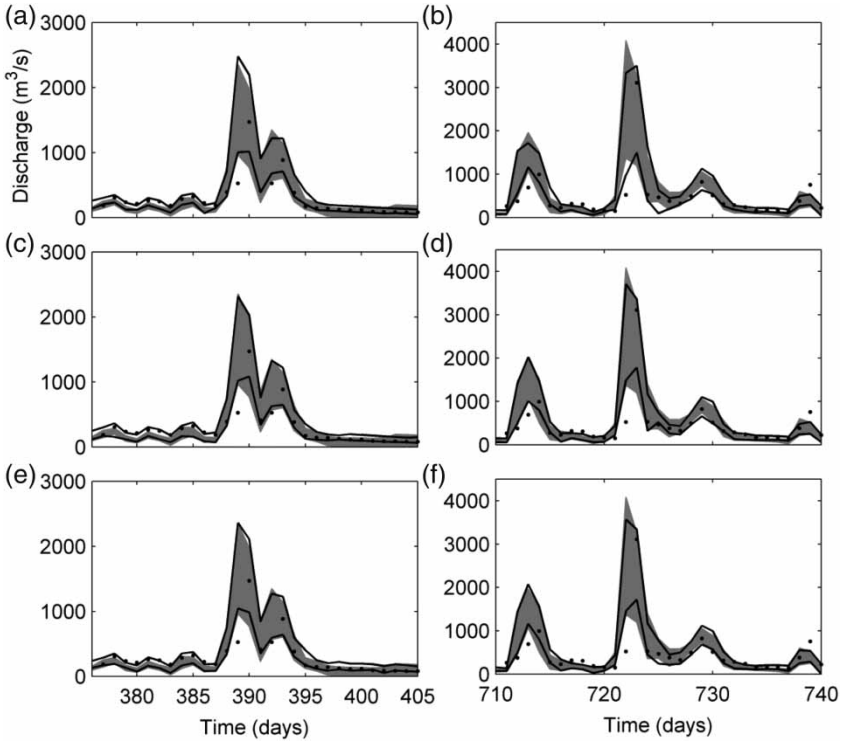

Figure 5 | Same as Figure 4, but for the Bagmati catchment.

the verification period. With mGLUE method, the percentage of the observation falling inside the bounds is increased to 65.26 and 67.52% in the calibration and verification periods, respectively. The machine learning models are able to approximate the mGLUE simulation results reasonably well. The results of the three machine learning models are comparable; however one can see a noticeable difference between them when predicting the peaks. The highest peak in Figure 5(a) is overestimated by the ANN model, while the other two peaks in Figure 5(b) are underestimated.

Figure 6 and Table 2 present a summary of statistics of the uncertainty estimation in the verification period. The ANN model is very close to the MC simulations results. The MT and LWR are better than the ANN with respect to MPI (note that lower MPI is the indication of better performance), however PICP shows that the prediction limits estimated by them enclose relatively lower percentage of the observed values compared to those of the ANN.

So far we have compared the performance of the three machine learning models by analysing the accuracy of the prediction only; however, there are other factors to be considered as well. These include computational efficiency, simplicity or ease of use, number of training parameters required, flexibility, transparency, etc. Computational efficiency is shown in Table 3. One can see that the time required to generate uncertainty results by MLUE methods in the verification period is significantly lower than that required by GLUE method. Table 4 shows linguistic variables to describe other factors mentioned above with parameters of machine learning models to be tuned. In ANN, we have only tuned one parameter – number of hidden neurons. MT also contains one parameter – pruning factor that has to be tuned. While in LWR, two parameters – number

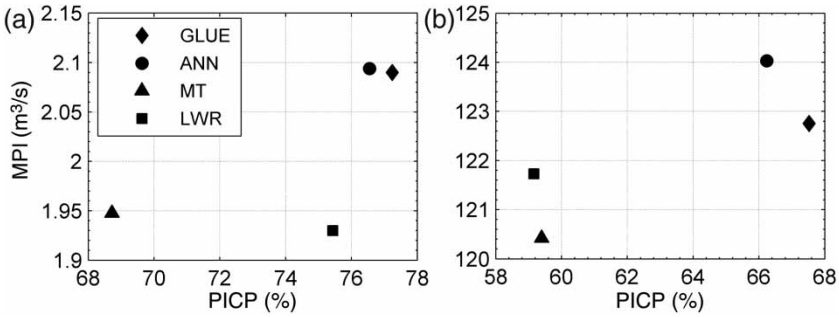

Figure 6 | A comparison of statistics of uncertainty (PICP and MPI) estimated with GLUE, neural networks (ANN), model trees (MT), and locally weighted regression (LWR) in the verification period. (a) Brue catchment; (b) Bagmati catchment.

Table 2 | Performances of the models measured by the coefficient of correlation (CoC), root mean squared error (RMSE), the prediction interval coverage probability (PICP) and the mean prediction interval (MPI) in the verification data set

		Lower prediction interval		Upper prediction interval			
Catchment	**Model**	**CoC**	**RMSE (m^3/s)**	**CoC**	**RMSE (m^3/s)**	**PICP (%)**	**MPI (m^3/s)**
Brue	ANN	**0.86**	**0.56**	**0.80**	**1.59**	**77.00**	2.09
	MT	0.84	0.61	0.79	1.63	68.72	1.95
	LWR	0.82	0.64	**0.80**	1.60	75.43	**1.93**
Bagmati	ANN	0.81	51.46	0.94	61.59	**66.24**	124.03
	MT	0.81	50.25	0.95	52.14	59.05	**120.59**
	LWR	**0.86**	**44.56**	**0.96**	**50.37**	59.16	121.73

Note: Bold type signifies the maximum value in each statistics.

Table 3 | Computational time for GLUE and MLUE

Catchments	Brue		Bagmati	
Period	Calibration	Verification	Calibration	Verification
Number of data used	8760	8217	2000	922
GLUE	16:34:00	11:45:00	7:45:00	6:41:00
ANN	2:07:00	0:04:00	1:03:00	0:01:30
MT	1:07:00	0:03:00	0:33:00	0:01:05
LWR	4:07:00	0:09:00	2:03:00	0:03:00

Note: The time (hh:mm:ss) is based on prediction of two quantiles (5% and 95%) and also includes data analysis and preparation time in the calibration period except for GLUE.

of neighbours and weighting functions have been tuned to get optimal results. Such parameters are optimised by exhaustive search during training the model. It can be observed that none of the models is superior with respect to all factors; however one may favour ANN if the ranking is done by giving equal weight to all factors.

Modelling the probability distribution function

Figure 7 and Figure 8 show comparison of the CDFs for the peak events estimated by the three machine learning methods for the Brue and Bagmati catchment, respectively. One can see that the CDFs estimated by the ANN, MT and LWR are comparable and are very close to the CDFs given by the GLUE simulations. It is observed that the CDFs estimated by the ANN, MT and LWR models deviate a little more near the middle of it for the peak event of 9 January 1996 in the Brue catchment (see Figure 7(b)). The CDFs estimated by the ANN, MT and LWR deviate a bit more at the higher percentiles values for the peak event of 13 August 1995 in the Bagmati catchment (see Figure 8(b)).

From the visual inspection one can see that the CDFs are reasonably approximated by the machine learning methods. However, it may require a rigorous statistical test to conclude if the estimated CDFs are not significantly different from those given by the GLUE simulations. In this study, since we have limited data (only 19 points) the results of the significance test (e.g. Kolmogorov–Smirnov) may not be reliable.

DISCUSSION

In this study, the uncertainty of the model output is assessed when the hydrological process model is used in simulation mode. However, this method can be used also in forecasting mode, provided that the process model is also run in forecasting mode. Note that we have not used the current observed discharge Q_t as an input to machine learning models because during the model application this variable is not available (indeed, the value of this variable is calculated by the HBV model, and the machine learning model assesses the uncertainty of this output).

It is observed that the results of machine learning models and the GLUE (or mGLUE) are visually closer to each other. The model prediction uncertainty caused by parameter uncertainty is rather large. There could be several reasons for this including the following ones (Shrestha *et al.* 2009): (i) the GLUE and mGLUE methods do not strictly follow the Bayesian inference process (Mantovan & Todini 2006) and overestimate the model prediction uncertainty; (ii) in the GLUE method, the uncertainty bound very much depends on the rejection threshold separating behavioural and non-behavioural models: in this study we use quite a low value of rejection threshold

Table 4 | Performance criteria of machine learning models indicated by linguistic variables

Models	Model parameters (optimised)	Accuracy		Efficiency	Transparency	Rank
		CoC	PICP and MPI			
ANN	Number of hidden nodes	High	High	Medium	Low	1
MT	Pruning factor	Medium	Low	High	Medium	2
LWR	Number of neighbours and weight functions	Low	Medium	Low	High	3

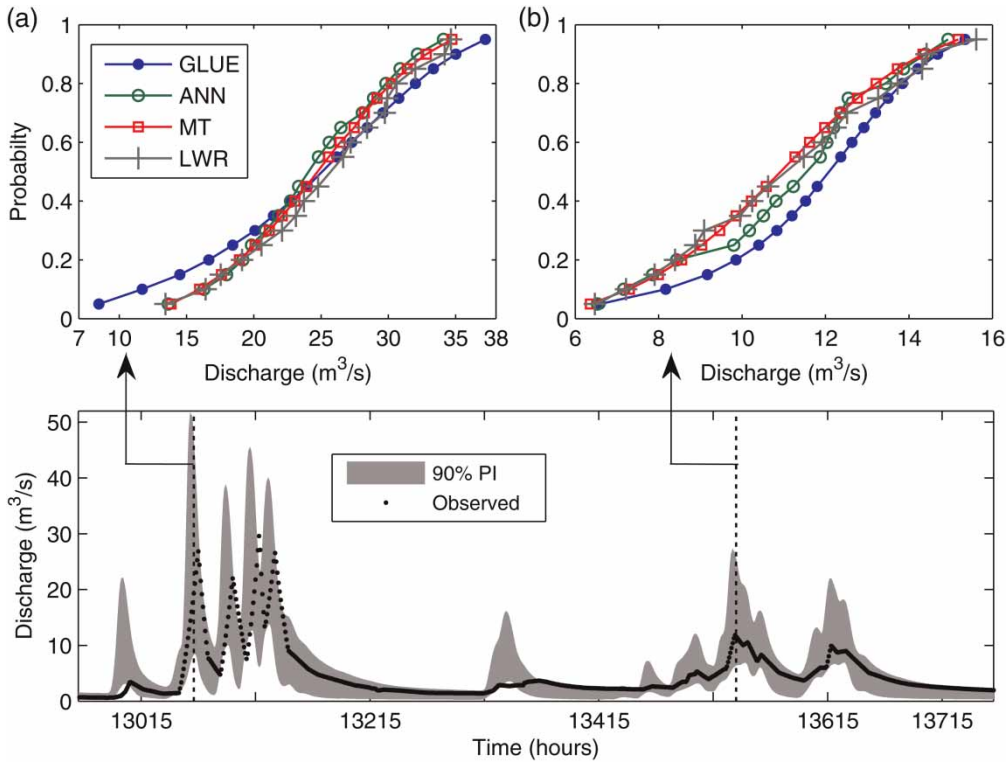

Figure 7 | A comparison of cumulative distribution function (CDF) estimated with GLUE and neural networks (ANN), model trees (MT), and locally weighted regression (LWR) for the Brue catchment in a part of the verification period. (a) Peak event of 20 December 1995; (b) peak event of 9 January 1996.

(CoE value of 0) which produces relatively wider uncertainty bounds; and (iii) we consider only parameter uncertainty, thus implicitly assuming no model structure and input data uncertainty.

It can be noticed that the performance of machine learning models to predict lower quantiles (5%, 10%, etc.) is relatively higher compared to those of the models for the upper quantiles (90%, 95%, etc.). This can be explained by the fact that the upper quantiles correspond to higher values of flow (where the HBV model is obviously less accurate) and higher variability, which makes prediction a difficult task. It is possible to develop a specific model only to simulate the peak observed data and their uncertainty as well as for the mean flows. In general, such model performs better than the global model. In this study, we have used MT and LWR models for uncertainty estimation which implicitly build the local models internally. It would be interesting to build the local models explicitly for high flow events for example. However, it is always not possible because of training data requirements for such rare and extreme events.

When comparing the percentage of the observed discharge data falling within the uncertainty bounds (i.e. PICP) produced by the GLUE, it can be seen that this percentage is much lower than the specified confidence level to generate these bounds. Low PICP value is consistent with the results reported in the literature (see e.g. Montanari 2005; Xiong & O'Connor 2008). The low 'quality' of the PIs obtained by the GLUE in enveloping the real-world discharge observations might be mainly due to the following three reasons (Shrestha *et al.* 2009): (i) by using GLUE method we investigate only the parametric uncertainty without consideration of uncertainty in the model structure, the input (such as rainfall, temperature data) and the output discharge data; (ii) we use uniform distribution and ignore the parameters correlation; (iii) results of the GLUE method depend on the (subjectively set) threshold value and likelihood measure for selecting the behavioural parameter sets.

To approximate CDF, an individual machine learning model is constructed for each quantile with the same structure of the input data and the model configuration. Thus, we

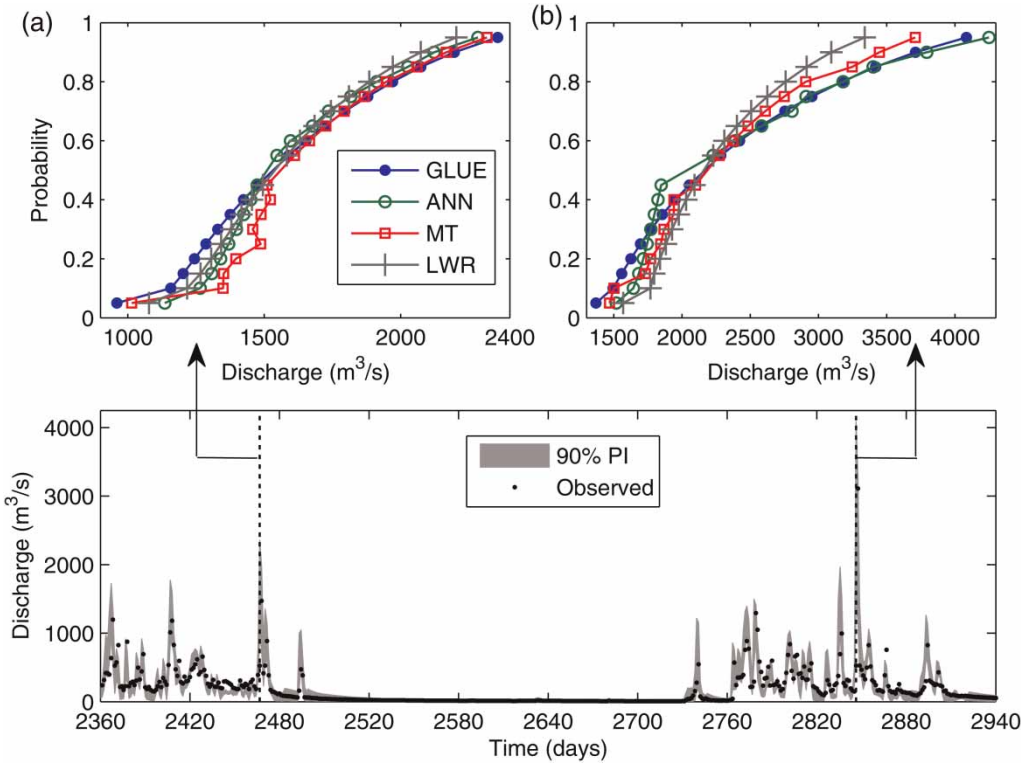

Figure 8 | Same as Figure 7, but for the Bagmati catchment in a part of the verification period. (a) Peak event of 14 September 1994; (b) peak event of 13 August 1995.

have not undertaken the full-fledged optimisation of the model and the input data structure of the machine learning models and there is a hope to improve the results. Furthermore one can notice that the CDFs estimated are not necessarily monotonously increasing (see e.g. 30% quantile of the MT model for the second case study). This is not surprising given that individual models are built for each quantile independently. This deficiency can be addressed by a correcting scheme (to be developed) that would ensure monotonicity of the overall CDF.

In this paper, the MLUE method is applied to emulate the results of the GLUE and mGLUE methods, however it can be used for other uncertainty analysis methods such as Markov chain Monte Carlo, Latin hypercube sampling, etc. Furthermore, the MLUE method can be applied in the context of other sources of uncertainty – input, structure or combined.

Since the machine learning technique is the core of the MLUE method, it may have a problem of extrapolation for extreme (rare) events. This means that the results are reliable only within the boundaries of the domain where the training data belong, and only a little beyond. In order to avoid the problem of extrapolation, an attempt should be made to ensure that the training data includes various possible combinations of the events including the extreme (such as extreme flood), however, this is not always possible since the extremes tend to be rather rare events. Like most of the uncertainty analysis methods, the MLUE method also presupposes the existence of a reasonably long, precise and relevant time series of measurements. As pointed out by Hall & Anderson (2002), uncertainty in extreme or unrepeatable events is more important than in situations where there are historical data sets, and this may require different approaches towards uncertainty estimation. The lack of sufficient historical data makes the uncertainty results from the model unreliable. This is actually true for all MC-based methods that use past data to make judgements about the future uncertainty.

The MLUE method is applicable only to systems whose physical characteristics do not change considerably with time. The results will not be reliable if the physics of the catchment (e.g. land use) and hydro-meteorological

conditions differ substantially from what was observed during the model calibration. If there is evidence of such changes, then the models should be re-calibrated.

The reliability and accuracy of the uncertainty analysis depend on the accuracy of the uncertainty models used, so attention should be given to these aspects as well. The proposed method does not consider the uncertainty associated with the model U itself. However, one could use CV data set to improve the accuracy of the model U by generalising its predictive capability.

CONCLUSIONS

This paper presents the further development, studying the relative performance and application of the MLUE method presented in its initial form by Shrestha *et al.* (2009), in predicting parameter uncertainty in rainfall-runoff modelling. The basic idea of the MLUE method is to encapsulate the computationally expensive MC simulations of a process model by an efficient machine learning model. (We used GLUE, a version of MC simulation method.) This model is first trained on the data generated by the MC simulations to encapsulate the relationship between the hydro-meteorological variables and the uncertainty statistics of the model output probability distribution, e.g. quantiles. Then the trained model can be used to estimate the latter for the new input data. The MLUE method is computationally efficient and can be used in real time applications when a large number of model runs are required.

We use three machines learning techniques, namely ANN, MT and LWR to predict several uncertainty descriptors of the rainfall-runoff model outputs. It is observed that the percentage of the observation discharge data falling within the prediction bounds generated by GLUE is much lower than the given certainty level used to produce these prediction bounds. Thus, we also apply mGLUE (Xiong & O'Connor 2008) method to improve the percentage of the observation falling within the prediction bounds.

On the two case studies we first demonstrate the application of the MLUE method to estimate the two quantiles (5 and 95%) forming the 90% PIs. Several performance indicators and visual inspection show that machine learning models are reasonably accurate to approximate the GLUE or mGLUE uncertainty bounds. It is also observed that the uncertainty bounds estimated by ANN, MT and LWR are comparable; however ANN is a bit better than the other two models. Second we extend the MLUE method to approximate the CDF of the model outputs, and the results demonstrate that the MLUE is performing quite well in estimating the CDF resulting from the GLUE (and mGLUE) methods.

It can be recommended to direct further studies at testing applicability of the MLUE approach with other sampling methods, ensuring compatibility of the models for multiple quantiles to achieve monotonicity of the resulting approximation of CDF, considering multiple sources of uncertainty, and testing the method on more complex models.

ACKNOWLEDGEMENTS

Most of this work has been completed during the first author's post doctorate research and second author's PhD research at UNSECO-IHE Institute for Water Education, Delft, The Netherlands; these were partly funded by the European Community's 7th Framework Research Program through the grants to the budget of the EnviroGRIDS, KULTURisk and WeSenseIt projects. WIRADA project (The Water Information Research and Development Alliances between CSIRO's Water for a Healthy Country Flagship and the Australian Bureau of Meteorology) partly supported the first author for completing this manuscript. The authors sincerely thank the editor and the three anonymous reviewers for providing helpful and constructive comments to improve the manuscript.

REFERENCES

Abrahart, R. J. & See, L. 2000 Comparing neural network and autoregressive moving average techniques for the provision of continuous river flow forecasts in two contrasting catchments. *Hydrological Processes* **14**, 2157–2172.

Aha, D., Kibler, D. & Albert, M. 1991 Instance-based learning algorithms. *Machine Learning* **6**, 37–66.

Allen, R. G., Pereira, L. S., Raes, D. & Smith, M. 1998 Crop Evapotranspiration: Guidelines for Computing Crop Water Requirements. Irrigation and Drainage Paper No. 56, FAO,

Rome. Available at: http://www.fao.org/docrep/X0490E/x0490e00.htm.

Bergström, S. 1976 Development and application of a conceptual runoff model for Scandinavian catchments. SMHI Reports RHO, No. 7, Norrköping, Sweden.

Beven, K. & Binley, A. 1992 The future of distributed models: Model calibration and uncertainty prediction. *Hydrological Processes* **6**, 279–298.

Beven, K. & Freer, J. 2001 Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems using the GLUE methodology. *Journal of Hydrology* **249**, 11–29.

Blasone, R., Vrugt, J., Madsen, H., Rosbjerg, D., Robinson, B. & Zyvoloski, G. 2008 Generalized likelihood uncertainty estimation (GLUE) using adaptive Markov Chain Monte Carlo sampling. *Advances in Water Resources* **31**, 630–648.

Dawson, C. W. & Wilby, R. L. 2001 Hydrological modelling using artificial neural networks. *Progress in Physical Geography* **25**, 80–108.

Dibike, Y. B. & Solomatine, D. P. 2001 River flow forecasting using artificial neural networks. *Journal of Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere* **26**, 1–8.

Duan, Q., Sorooshian, S. & Gupta, V. 1992 Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resources Research* **28**, 1015–1031.

Elshorbagy, A., Corzo, G., Srinivasulu, S. & Solomatine, D. P. 2010a Experimental investigation of the predictive capabilities of data driven modeling techniques in hydrology – Part 1: concepts and methodology. *Hydrology and Earth System Sciences* **14**, 1931–1941.

Elshorbagy, A., Corzo, G., Srinivasulu, S. & Solomatine, D. P. 2010b Experimental investigation of the predictive capabilities of data driven modeling techniques in hydrology – Part 2: application. *Hydrology and Earth System Sciences* **14**, 1943–1961.

Freer, J., Beven, K. & Ambroise, B. 1996 Bayesian estimation of uncertainty in runoff prediction and the value of data: an application of the GLUE approach. *Water Resources Research* **32**, 2161–2173.

Georgakakos, K., Seo, D.-J., Gupta, H. V., Schaake, J. & Butts, M. M. 2004 Towards the characterization of streamflow simulation uncertainty through multimodel ensembles. *Journal of Hydrology* **298**, 222–241.

Govindaraju, R. S. & Rao, A. R. 2000 *Artificial Neural Networks in Hydrology*. Kluwer Academic Publishers, Amsterdam, 348 pp.

Haario, H., Laine, M., Mira, A. & Saksman, E. 2006 DRAM: efficient adaptive MCMC. *Statistical Computation* **16**, 339–354.

Hall, J. & Anderson, M. G. 2002 Handling uncertainty in extreme or unrepeatable hydrological processes – the need for an alternative paradigm. *Hydrological Processes* **16**, 1867–1870.

Harr, M. 1989 Probabilistic estimates for multivariate analyses. *Applied Mathematical Modeling* **13**, 313–318.

Johnston, P. & Pilgrim, D. 1976 Parameter optimization for watershed models. *Water Resources Research* **12**, 477–486.

Khu, S.-T. & Werner, M. G. F. 2003 Reduction of Monte-Carlo simulation runs for uncertainty estimation in hydrological modelling. *Hydrology and Earth System Sciences* **7**, 680–692.

Kuczera, G. & Parent, E. 1998 Monte Carlo assessment of parameter uncertainty in conceptual catchment models: the Metropolis algorithm. *Journal of Hydrology* **211**, 69–85.

Kuzmin, V., Seo, D.-J. & Koren, V. 2008 Fast and efficient optimization of hydrologic model parameters using a priori estimates and stepwise line search. *Journal of Hydrology* **353**, 109–128.

Madsen, H. 2000 Automatic calibration of a conceptual rainfall–runoff model using multiple objectives. *Journal of Hydrology* **235**, 276–288.

Maier, H. R. & Dandy, G. C. 2000 Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental Modelling & Software* **15**, 101–124.

Maier, H. R., Jain, A., Dandy, G. C. & Sudheer, K. P. 2010 Methods used for the development of neural networks for the prediction of water resource variables in river systems: current status and future directions. *Environmental Modelling & Software* **25**, 891–909.

Mantovan, P. & Todini, E. 2006 Hydrological forecasting uncertainty assessment: Incoherence of the GLUE methodology. *Journal of Hydrology* **330**, 368–381.

Maskey, S., Guinot, V. & Price, R. K. 2004 Treatment of precipitation uncertainty in rainfall-runoff modelling: a fuzzy set approach. *Advance in Water Resources* **27**, 889–898.

McKay, M. D., Conover, W. J. & Beckman, R. J. 1979 A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239–245.

Melching, C. S. 1992 An improved-first-order reliability approach for assessing uncertainties in hydrologic modeling. *Journal of Hydrology* **132**, 157–177.

Minns, A. W. & Hall, M. J. 1996 Artificial neural networks as rainfall-runoff models. *Hydrological Science Journal* **41**, 399–417.

Mitchell, T. 1997 *Machine Learning*. McGraw-Hill, Singapore, 414 pp.

Montanari, A. 2005 Large sample behaviors of the generalized likelihood uncertainty estimation (GLUE) in assessing the uncertainty of rainfall-runoff simulations. *Water Resources Research* **41**, W08406.

Montanari, A. & Brath, A. 2004 A stochastic approach for assessing the uncertainty of rainfall-runoff simulations. *Water Resources Research* **40**, W01106.

Nash, J. & Sutcliffe, J. 1970 River flow forecasting through conceptual models – Part I – A discussion of principles. *Journal of Hydrology* **10**, 282–290.

Pappenberger, F., Harvey, H., Beven, K., Hall, J. & Meadowcroft, I. 2006 Decision tree for choosing an uncertainty analysis methodology: a wiki experiment http://www.floodrisknet.org.uk/methods http://www.floodrisk.net. *Hydrological Processes* **20**, 3793–3798.

Rosenblueth, E. 1981 Two-point estimates in probability. *Applied Mathematical Modelling* **5**, 329–335.

Shrestha, D. L. & Solomatine, D. P. 2006 Machine learning approaches for estimation of prediction interval for the model output. *Neural Networks* **19**, 225–235.

Shrestha, D. L. & Solomatine, D. P. 2008 Data-driven approaches for estimating uncertainty in rainfall-runoff modelling. *International Journal of River Basin Management* **6**, 109–122.

Shrestha, D. L., Kayastha, N. & Solomatine, D. P. 2009 A novel approach to parameter uncertainty analysis of hydrological models using neural networks. *Hydrology and Earth System Sciences* **13**, 1235–1248.

Solomatine, D. P. 1999 Two strategies of adaptive cluster covering with descent and their comparison to other algorithms. *Journal of Global Optimization* **14**, 55–78.

Solomatine, D. P. & Torres, L. A. A. 1996 Neural network approximation of a hydrodynamic model in optimizing reservoir operation. In: *Hydroinformatics '96* (A. Muller, ed.). Balkema, Rotterdam.

Solomatine, D. P. & Dulal, K. N. 2003 Model trees as an alternative to neural networks in rainfall–runoff modelling. *Hydrological Sciences Journal* **48**, 399–411.

Solomatine, D. P. & Ostfeld, A. 2008 Data-driven modelling: some past experiences and new approaches. *Journal of Hydroinformatics* **10**, 3–22.

Solomatine, D. P. & Shrestha, D. L. 2009 A novel method to estimate model uncertainty using machine learning techniques. *Water Resources Research* **45**, W00B11.

Solomatine, D. P., Maskey, M. & Shrestha, D. L. 2008 Instance-based learning compared to other data-driven methods in hydrological forecasting. *Hydrological Processes* **22**, 275–287.

Stedinger, J. R., Vogel, R. M., Lee, S. U. & Batchelder, R. 2008 Appraisal of the generalized likelihood uncertainty estimation (GLUE) method. *Water Resources Research* **44**, W00B06.

Thiemann, M., Trosset, M., Gupta, H. V. & Sorooshian, S. 2001 Bayesian recursive parameter estimation for hydrologic models. *Water Resources Research* **37**, 2521–2535.

Tung, Y.-K. 1996 Uncertainty and reliability analysis. In: *Water Resources Handbook* (L. W. Mays, ed.). McGraw-Hill, New York, 7.1–7.65.

Todini, E. 2008 A model conditional processor to assess predictive uncertainty in flood forecasting. *Journal of River Basin Management* **6**, 123–137.

Vrugt, J. A., ter Braak, C., Gupta, H. & Robinson, B. 2009 Equifinality of formal (DREAM) and informal (GLUE) Bayesian approaches in hydrologic modeling? *Stochastic Environmental Research and Risk Assessment* **23**, 1011–1026.

Vrugt, J. A., Diks, C., Gupta, H. V., Bouten, W. & Verstraten, J. M. 2005 Improved treatment of uncertainty in hydrologic modeling: Combining the strengths of global optimization and data assimilation. *Water Resources Research* **41**, W01017.

Wagener, T. & Gupta, H. V. 2005 Model identification for hydrological forecasting under uncertainty. *Stochastic Environmental Research and Risk Assessment* **19**, 378–387.

Witten, I. H. & Frank, E. 2000 *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers, San Francisco, USA, 371 pp.

Xiong, L. & O'Connor, K. 2008 An empirical method to improve the prediction limits of the GLUE methodology in rainfall-runoff modeling. *Journal of Hydrology* **349**, 115–124.

Yapo, P., Gupta, H. V. & Sorooshian, S. 1996 Automatic calibration of conceptual rainfall-runoff models: sensitivity to calibration data. *Journal of Hydrology* **181**, 23–48.

First received 20 December 2012; accepted in revised form 3 June 2013. Available online 25 July 2013