MSc-thesis

Improving hydrological model performance using storm-dependent parameters

Yuhan Lin

TU Delft

Committee:
Dr.ir.Gerrit Schoups,
Dr.Markus Hrachowitz,
Dr.ir.Rob van Nes

MSc-thesis

by

Yuhan Lin

to obtain the degree of Master of Science
at the Delft University of Technology,

Student number:	4866703	
Project duration:	Feb 1, 2020 – Oct 10, 2020	
Thesis committee:	Dr.ir.Gerrit Schoups	TU Delft, supervisor
	Dr.Markus Hrachowitz	TU Delft
	Dr.ir.Rob van Nes	TU Delft

Abstract

Calibration and model prediction is always affected by uncertainty in the forcing data, response data and structural error in the model. The storm-dependent parameters are believed to be able to capture these errors and improve model prediction. The goal of this research will be to further investigate and develop the storm-based approach and compare it to more traditional approaches, including the use of static (time-invariant) parameters and the use of GLUE to capture model errors. The hypothesis in this paper is the random variation of storm-dependent parameters can capture the model error and improve the prediction. The storm-based method will apply a sensitivity analysis to identify the storm-dependent parameters that are most likely to vary by storms. The variation of storm-dependent parameters will give large changes in model performance which is measured by Nash–Sutcliffe efficiency. In the storm-based method, parameters will be calibrated storm by storm in the calibration period. Then in the validation period, streamflow will still be predicted storm by storm through picking each parameter set from the calibrated parameter sets. Besides, the effect of dryness and the optimal threshold for identifying the storm epochs on the model performance will also be explored in the storm-based method. The results obtained by the storm-based method will be compared with the method using static parameters and GLUE method. Six case studies calibrating a conceptual rainfall-runoff model with for parameters with daily data illustrate the improvement of prediction obtained by the storm-based method for dry basins. The extent of variation of storm-dependent parameters is very random in each case which indicates there is error in the model. Moreover, the extent of variation of storm-dependent parameters has no relation with initial water storage, rainfall characteristic and basin characteristics. Although the parameters cannot be predicted deterministically, they can be predicted probabilistically with the histogram or fitted distribution for the calibrated parameter sets in the future work. By making storm-dependent parameters vary with storms and other parameters constant, the storm-based method performs better for drier basins while worse for wetter basins compared to GLUE method and traditional method. The logscore value obtained in storm-based method(e.g. -0.68 for one of the dry basin A) is larger than those obtained in the traditional method (e.g. -1.23 for basin A) and GLUE method (e.g. -0.67 for basin A). Additionally, the RMSE values for total flow obtained in the storm-based method are all smaller than those obtained in the traditional method, and GLUE method for dry basins. This suggests the storm-based method is more applicable for dry basins and this method should be better developed for wet basins. What is more, the extent of variation of storm-dependent parameters has no relation with basin characteristics but the mean and variation of the storm-dependent parameters can be obtained. Hence the extent of variation of parameters can be described probabilistically.

Contents

1 Introduction

Hydrological models are important tools to simulate the hydrological processes and forecast streamflow given rainfall and evaporation. They play an essential role to help to (1) understand the hydrological processes in basins; (2) to predict water quantity for multiple uses (e.g. hydroelectric production, water supply for domestic, recreational, agricultural, tourism, and industrial activities); (3)to predict the likely flood risks that relate to social safety and property; and (4) to assess the effects of human activities on water resources (e.g., land cover and climate changes)(Bouda et al. [2012]). Conceptual rainfall runoff models(CRR) are one type of the critical hydrological models. These models simplify complicated hydrological processes and responses in the real world by using spatially lumped parametric relations to describe water flow processes in a watershed. However, parameters of most hydrological models cannot represent some measurable catchment characteristics directly and have to be calibrated. Model calibration is an indispensable part of hydrological analyses to simulate and forecast streamflow better. During calibration, the best parameter values are determined according to some predetermined criteria so that the simulations match as closely as possible one or several observed system outputs(Schaefli and Zehe [2009]). The calibrated parameters can be used to predict the discharge which is meaningful for designing canals, water management and planning, flood control, predicting soil erosion and so on. Hence, hydrologists continuously aim to improve the performance of hydrological models.

However, the calibration and prediction of CRR models have always been affected by model uncertainty(Kuczera et al. [2006]). This uncertainty has three sources in previous studies: input error, response error and model error. For example, in this research, daily rainfall, potential evaporation and parameters are used as input data, which can be affected by measurement error and sampling uncertainty arising from the randomly distributed field in spatial and temporal scale. The response data like discharge at different locations is subject to measurement as well as rating curve error. At last, due to the simplification of the real hydrological process, even with accurate input data and response data, CRR models can not give a correct response. This error is termed structural or model error. Figure(1.1) summarizes the current understanding of model uncertainty. There is also some other literature that illustrates the value of model uncertainty analysis(Beven [2006],Beven et al. [2007],Todini and Mantovan [2007]). It is always significant to consider the uncertainties in hydrological models which are most often associated with input data, model parameters, model structure(Saltelli et al. [1999]).

In CRR models, an assumption will lead to a structural error: we think the parameters are static during time series. However, this is just an ideal case that dynamic components of the catchment process are oversimplified. Assuming the parameters are time-invariant is inappropriate because a set of optimized parameters are only able to represent an average process during the analyzed time series(Lan et al. [2019]). Besides, forcing data like rainfall and potential evaporation are spatial and temporal averages from random fields. The number of spatially and temporal distributed fields yielding the same average discharge can be huge. Although these distinct fields produce the same average rainfall they can cause a different hydrological response. In models, the relation between runoff and soil wetness is typically conceptualized based on spatial heterogeneity in topography and soil properties. While in reality, the relation also depends on the small-scale(unresolved) spatial heterogeneity of climate controls like precipitation. For example, if the primary source of rainfall takes place over a saturated part of the catchment, the soil store will be recharged quickly, and there will be quickflow generated. However, if the primary source of rainfall takes place over an unsat-

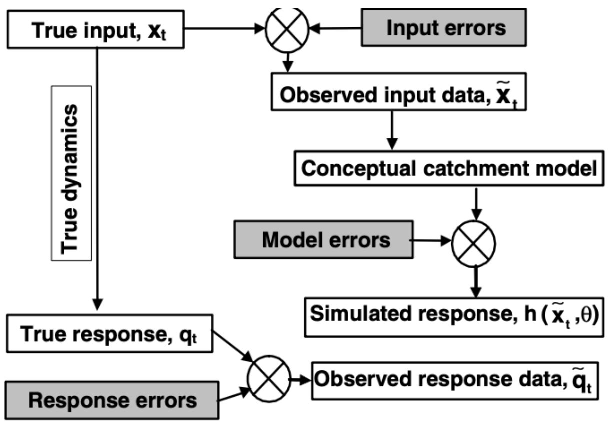

True input, x_t
Input errors
Observed input data, $\tilde{x}_t$
True dynamics
Conceptual catchment model
Model errors
Simulated response, $h(\tilde{x}_t, \theta)$
True response, q_t
Response errors
Observed response data, $\tilde{q}_t$

Figure 1.1: Schematic of error propagation in CRR models(sources of errors are shaded grey and 'input errors' indicates any errors in the model input, including forcing data and parameters)(taken from Kavetski et al. [2003])

urated part of the field, soil store would be firstly recharged, and there would be no quickflow generated. This yields multi-valued(dynamic) rather than single-valued (static) storage-flow relations like shown in Figure(1.2). Models with input data that are temporally and spatially averaged neglect this dynamic dimension and usually can not give correct responses(Kuczera et al. [2006]. This dynamic dimension should be considered to improve model calibration.This results in non-constant parameters that depend on the rainfall event, initial conditions, etc.

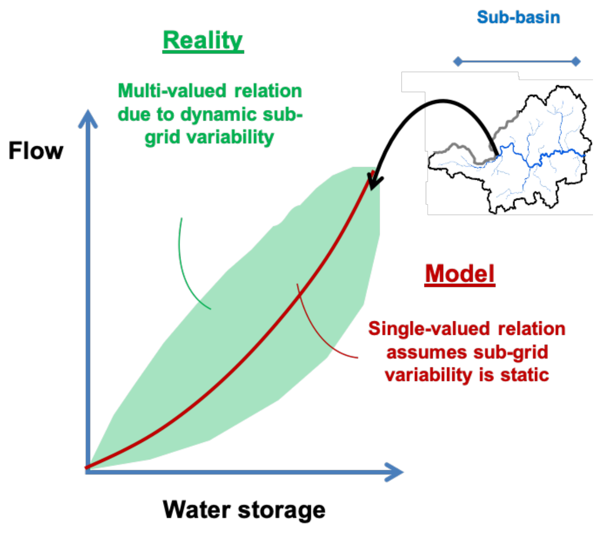

Reality
Multi-valued relation due to dynamic sub-grid variability
Sub-basin
Flow
Model
Single-valued relation assumes sub-grid variability is static
Water storage

Figure 1.2: The dynamic sub-grid variability. Averaging of rainfall is a significant source of model error because it gives a single-valued relation between storage and runoff. Multi-valued, non-unique storage-flow relations at the catchment scale are not considered in current models which would result in incorrect parameter estimations.

Using parameters that vary with time is believed to be able to significantly improve the accuracy and robustness of conventional models. In a previous study(Kuczera et al. [2006]), a hypothesis is demonstrated plausible that the input and model uncertainty can be adequately described by storm-dependent parameters that are randomly variable. Before Kuczera et al. doing this study, critical issue that needed to be addressed is the temporal variation of the random perturbations of the model fluxes. It is vital to use an appropriate time step for CRR models to do the computation because if the time step is significantly less than the response time of the store to receive the flux, each store would respond only to the average component of the input(Kuczera et al. [2006]). Kuczera et al. and Kuczera solved this issue by randomly perturbing the model parameters at the beginning of each storm. This method makes sense because the primary forcing(the most spatially and temporally heterogeneous) of the catchment water balance is coming from rainfall so that the flux variation

is expected to persist over storm event time series. Another reason that causes structure error is the approximation in determining parameters(Yang et al. [2008]). Parameter uncertainty is unavoidable but comparably easy to control by an appropriate calibration. There are several methods to determine the values of the parameters in the calibration process. Directly measuring the parameters is one of the ways to determine the parameter values, however,it is actually impossible to directly measure the parameters in the field. Besides, some conceptual parameters that obtained by empirical equations and literature references can also bring the uncertainty into the model(Gong et al. [2011],Shen et al. [2012],Xue et al. [2014]). In addition, the interactions and correlations between parameters can also cause uncertainties. Different parameter sets might give similar prediction results, and this is a phenomenon called equifinality that describes a kind of inherent property of inverse modelling(Beven and Binley [1992],Abbaspour et al. [2007],Abbaspour et al. [2011]). Sometimes, ignoring or underestimation of uncertainty may lead to unexpected losses and overestimation of uncertainty may cause a waste of resources(Shen et al. [2012]). Therefore, analysis of uncertainty is essential and needed to guarantee the relatively good performance of hydrological models(Beven and Binley [1992],Vrugt et al. [2003],Yang et al. [2007],Yang et al. [2007]).
Kuczera et al. [2006] introduced a robust framework named BATEA framework. In this framework, storm-dependent parameters are varying by storms, and this framework shows that it is able to describe hydrological process and characterise the inherent uncertainty. Kuczera et al. also discussed the differences between GLUE and BATEA framework. The generalized likelihood uncertainty estimation (GLUE) method is a method that widely applied and studied in researches about the uncertainty in hydrological modelling(e.g.Aronica et al. [2002], Cameron et al. [1999], Blazkova and Beven [2002], Freer et al. [1996],Heidari et al. [2006],Kuczera et al. [2006]). This method is usually utilized for investigating the uncertainties in water resources and environmental modelling(Beven and Binley [1992]). GLUE considers the equifinality phenomenon in hydrological modelling and produce the prediction limits for the future streamflow given a certain required certainty level and a set of previously identified behavioural parameter sets (Xiong and O'Connor [2008]) BATEA framework and GLUE reached a consensus that model error is vital and hard to characterise. However, they are basically different in their conceptual structures. For one thing, BATEA applies the error propagation and considers the input error, response error and model error, respectively. While in GLUE, all sources of errors are represented by nothing but parameter uncertainty. For another, they are different in using static or non-static parameters. GLUE is still based on the deterministic parameters from a series of behavioural parameter sets. In sharp contrast, parameters vary stochastically by storms in BATEA framework.
In summary, CRR models simplify complex rainfall-runoff process which may lead to poor performance of hydrological models. Storm-dependent parameters are believed to represent the dynamic process better and great emphasis will be placed on improving rainfall-runoff modeling by using storm-based parameters in this research. In previous studies, this method of using storm-dependent parameters is believed to be able to capture the model error in CRR models with great emphasis on the analysis of uncertainty. The goal of this research will be to further investigate and develop the storm-based approach, and compare it to more traditional approaches, including the use of static (time-invariant) parameters and the use of GLUE to capture model errors.
In this research, the focus is on analyzing the variation of storm-dependent parameters to capture model errors to find a systematic framework to predict these variations based on rainfall events and initial conditions in order to improve model performance. Specifically, four questions will be investigated:

1 To what extent do model parameters vary by storm/event? First of all, Nash-Sutcliffe sensitivity analysis will figure out which parameters are likely to vary by storm. Then a storm-based calibration needs to be done to actually answer this first research question. The hypothesis is that most model errors can be influenced by flux errors that arise from spatial and temporal averaging. We assume that these errors can be well described by randomly sampling each parameter from a distribution.

2 To what extent is this variation random and can it be predicted from rainfall event characteristics and initial conditions? The relation between the extent of variation and initial rainfall characteristics for each storm will be explored. Setting up a framework to predict parameters from the initial rainfall characteristics of each storm is possible.

3 To what extent does accounting for variation in model parameters capture model errors and improve rainfall-runoff prediction? The results obtained from three different methods will be compared for six case basins to illustrate the improvement of hydrological model performance. Several criteria will be applied to assess the overall performance and high-flow as well as low-flow performance.

4 Does the extent of variation of storm-dependent parameters depend on basin characteristics (e.g. dry vs humid, small vs large)? The regression relation between the extent of variation of storm-dependent parameters and basin characteristics will be explored for 392 basins to illustrate whether the sizes and dryness of basins will influence the extent of storm-dependent parameters variations.

This research is organized as follows: first of all, the methodology applied in this research is introduced in detail. In the methodology, firstly the applied hydrological model structure and function will be introduced. It is then using the storm-based method to do the calibration and validation. During the storm-based method, storm-dependent parameters will be firstly identified by a Nash-Sutcliffe sensitivity analysis. Then the hydrological model will be calibrated basing on storm events with storm-dependent parameters variable and other parameters constant. The extent of variations of parameters for each storm and the affecting factors are going to be explored. Next, comparing the results obtained by alternative methods(traditional method and GLUE method) with those obtained by the storm-based method. Six case studies with different characteristics illustrate how storm-dependent parameters improve hydrological model performance. Validation results obtained by different methods will be compared during case studies. Additionally, the calibrated storm-dependent parameters will show a variation for each storm epoch during the analyzed time series. The extent of variation of time-varying parameters will be analyzed as well as its relation with basin characteristics for 392 basins. Most of the results will be presented by taking one basin as an example, and other basins will be listed in the appendix. Future work about predicting the parameters, uncertainty in the forcing data and comparison between this research and other studies are then discussed. Moreover, finally are the summary of what has been done in this research and the main conclusions obtained from this research.

2 Methodology

2.1. Model

The conceptual rainfall-runoff model GR4J with four parameters was chosen because it was empirically developed to provide a relative satisfying performance across catchments with different ranges of climatic and hydrological regimes (Perrin et al., 2003) and has been tested in more than 240 Australian catchments(Wang et al. [2010]). GR4J model is proposed by Edijatno et al. [1999] and Nascimento [1995], which is a successfully improved version of GR3J. Figure(2.1) illustrates the structure of the GR4J model and the function of GR4J model is detailed introduced below.

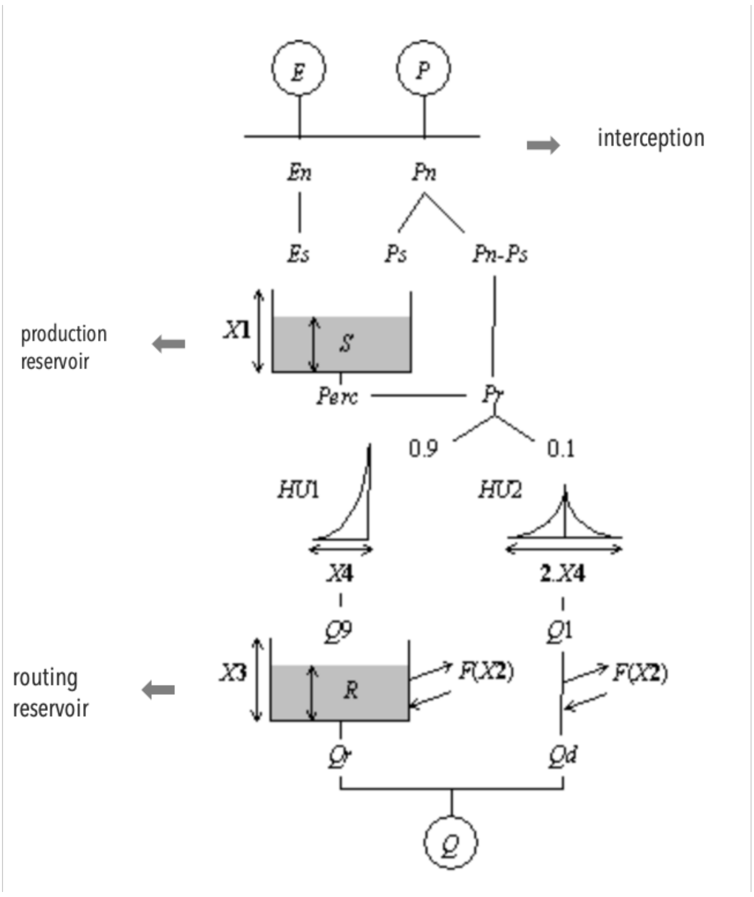


Figure 2.1: GR4J model
(https://webgr.inrae.fr/en/models/daily-hydrological-model-gr4j/)

Table 2.1: the definition of parameters and state variables in the GR4J rainfall-runoff model

	Property	Range	Range	Description
X1	Parameter	(20, 2000)	(mm)	maximum capacity of the production store
X2	Parameter	(-5, 3)	(mm/day)	the catchment water exchange coefficient
X3	Parameter	(20, 300)	(mm/day)	the maximum capacity of the routing reservoir
X4	Parameter	(1, 5)	(days)	time peak ordinate of unit hydrograph UH1
S	State variable		mm	production store level
R	State variable		mm	Routing store level

The definition of parameters and state variables are presented in Table(2.1). $X1$ is the maximum capacity of production store which can be affected by soil types in the river basins. With less porosity, the production store will be smaller because the soil can not hold too much water. $X2$ is the catchment exchange coefficient which quantifies groundwater exchange between catchments and can easily affect the routing store. If the catchment exchange coefficient is negative, it means groundwater enters into the deeper aquifer. If the coefficient is positive, it indicates more water enters into the routing store from deep aquifer (Harlan et al. [2010]). Routing store is easily affected by $X2$. $X3$ is the maximum capacity of routing reservoir in one day. This capacity is also affected by soil type and humidity. $X4$ is the time when the ordinate peak of flood hydrograph is created on GR4J modeling. The hydrograph ordinates are calculated from the S curves (the accumulation of the proportion of unit rainfall treated by the hydrograph in function of time). 90% runoff is slow flow and will reach to the routing reservoir. The slow flow that infiltrates into the ground will be routed by unit hydrograph HU1 and create ordinate of the unit hydrograph HU1. 10% runoff is fast flow and will flow on the ground surface(Harlan et al. [2010]). The fast flow will be routed by a unique unit hydrograph HU2.

P (mm/day) represents rainfall amount and E (mm/day) represents the potential evapotranspiration (PET). First of all, P is neutralized by E to determine the net rainfall P_n and net evapotranspiration E_n, calculated by:

If $P > E$ then $P_n = P - E$ and $E_n = 0$
If $P < E$ then $P_n = 0$ and $E_n = E\text{-}P$

If P_n is different from zero, a fraction of P_s of P_n goes into the production reservoir and is calculated by:

$$P_s = \frac{X1\left(1-\left(\frac{S}{X1}\right)^2\right)\cdot\tan\left(\frac{P_n}{X1}\right)}{1+\frac{S}{X1}\cdot\tanh\left(\frac{P_n}{X1}\right)} \tag{2.1}$$

Where X1 (mm) and S are, respectively, the maximum capacity and the production store level. Otherwise, when E_n is different from zero, a part of evaporation E_s is removed from the production store which is given by:

$$E_s = \frac{S\left(2-\frac{S}{X1}\right)\tanh\left(\frac{E_n}{X1}\right)}{1+\left(1-\frac{S}{X1}\right)\tanh\left(\frac{E_n}{X1}\right)} \tag{2.2}$$

Then the production store level is updated through: $S = S - E_s + P_s$. A percolation called Perc coming from the production store is then calculated:

$$Perc = S\cdot\left(1-\left[1+\left(\frac{4}{9}\frac{S}{X1}\right)^4\right]^{-\frac{1}{4}}\right) \tag{2.3}$$

The production store level is then again updated: S = S – Perc. The water quantity P_r that finally reaches the routing part of the model is:

$$P_r = Perc + (P_n – P_s) \tag{2.4}$$

Pr is divided into two flow components, 90% being routed by a unit hydrograph HU1 and a routing store, and 10% by a unique unit hydrograph HU2. The generation of flow from HU1 and HU2 is the routing process over time(2 times of X4). HU1 and HU2 depend on the same parameter X4 (the time when the ordinate peak of flood hydrograph is created). The hydrographs ordinates are calculated from the S curves (the accumulation of the proportion of unit rainfall treated by the hydrogram in function of time), respectively named SH1 and SH2. SH1 is defined in function of time by:

$$SH1_{(t)} = \begin{cases} 0 & \text{t = 0,} \\ \left(\frac{t}{X4}\right)^{\frac{5}{2}} & \text{0< t < X4,} \\ 1 & \text{t > X4.} \end{cases} \tag{2.5}$$

SH2 is defined in function of time by:

$$SH2_{(t)} = \begin{cases} 0 & \text{t = 0,} \\ \frac{1}{2}\left(\frac{t}{X4}\right)^{\frac{5}{2}} & \text{0< t < X4,} \\ 1-\frac{1}{2}\left(2-\frac{t}{X4}\right)^{\frac{5}{2}} & \text{X4 < t < 2X4} \end{cases} \tag{2.6}$$

The ordinates of HU1 and HU2 are then obtained from:

$$UH1_{(j)} = SH1_{(j)} - SH1(j-1) \tag{2.7}$$

$$UH2_{(j)} = SH2_{(j)} - SH2(j-1) \tag{2.8}$$

where j is an integer. For each time step i, the outputs Q9 and Q1 of the two hydrograms are calculated with:

$$Q9_{(i)} = 0.9 \cdot \sum_{k=1}^{l} UH1(k) \cdot \mathrm{P}_{\mathrm{r}(i-k+1)} \tag{2.9}$$

$$Q1_{(i)} = 0.1 \sum_{k=1}^{m} UH2(k) \cdot \mathrm{P}_{\mathrm{r}(i-k+1)} \tag{2.10}$$

with l = int(X4)+1 and m = int(2X4)+1. A groundwater exchange term (loss or gain) is calculated with:

$$F = X2\left(\frac{R}{X3}\right)^{7/2} \tag{2.11}$$

with R the routing store level, X3 the one-day maximal capacity of the store and X2 the water exchange coefficient, which is positive in case of a gain, and negative in case of a loss, or zero. The level in the routing store is updated by adding the Q9 output of the hydrogram HU1 and F: R = max (0 ; R + Q9 + F). Then, it empties in an output Qr given by:

$$Q_r = R \cdot \left\{ 1 - \left[1 + \left(\frac{R}{X3} \right)^4 \right]^{-\frac{1}{4}} \right\} \tag{2.12}$$

The level in the store becomes:

$$R = R – Q_r \tag{2.13}$$

The output Q1 of the hydrogram HU2 goes through the same exchanges to give the flow component Qd:

$$Qd = max(0; Q1 + F) \tag{2.14}$$

The total streamflow Q is finally given by:

$$Q = Q_r + Q_d \tag{2.15}$$

2.2. Data and Study area

2.2.1. Data

The MOPEX dataset (https://hydrology.nws.noaa.gov/pub/gcip/mopex/US_Data/) containing data from basins of United States covering a range in climate, land cover, soil type and topography are used in this research. Precipitation, potential evaporation and streamflow in the MOPEX data sets will be used as forcing data to model. The precipitation is processed in NWS Hydrology Laboratory. The Potential Evaporation is based on NOAA Evaporation Atlas (Farnsworth and Thompson [1982]). The streamflow is obtained from USGS National Water Information System (NWIS) (available athttp://water.usgs.gov/nwis). A spatial map of the MOPEX basins is shown below.

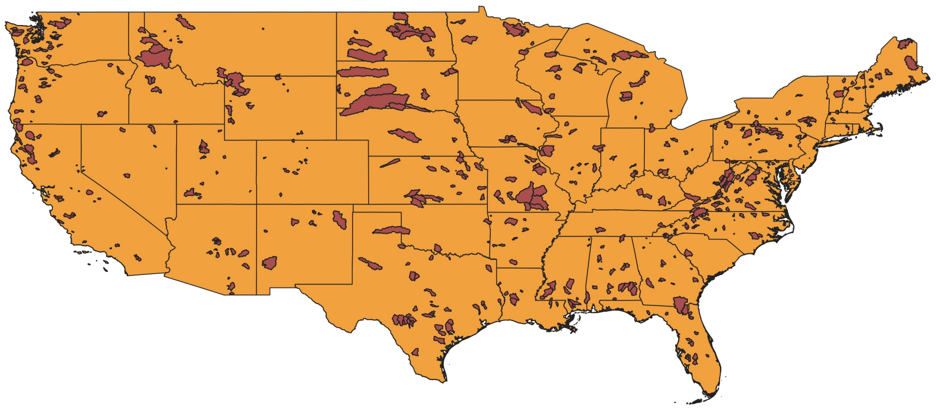


Figure 2.2: Locations of the MOPEX basins used in this study

Basins that are going to be explored should have their water balance closed for a long period (at least ten years). The water balance is checked by Budyko Framework. Budyko framework describes the long-term water and energy balances of catchments through a curvilinear relationship between Evaporative Index (Actual evaporation / Precipitation) and the Dryness Index (Potential evaporation/ Precipitation). The actual evaporation is calculated by Equation (2.16) and potential evaporation is given in the data sets.

$$\overline{E_A} = \overline{P} - \overline{Q} \tag{2.16}$$

$$\overline{E_A} < \overline{E_p} \tag{2.17}$$

$$\overline{P} - \overline{E_p} - \overline{Q} < 0 \tag{2.18}$$

$$\frac{E_A}{P} = \left[\frac{E_P}{P}\tan h\left(\frac{1}{\frac{E_P}{P}}\right)\left(1 - e^{-\frac{E_r}{P}}\right)\right]^{\frac{1}{2}} \tag{2.19}$$

where: $\overline{E_A}$ is actual evaporation, $\bar{P}$ is average precipitation, $\bar{Q}$ is average discharge, $\overline{E_p}$ is potential evaporation.

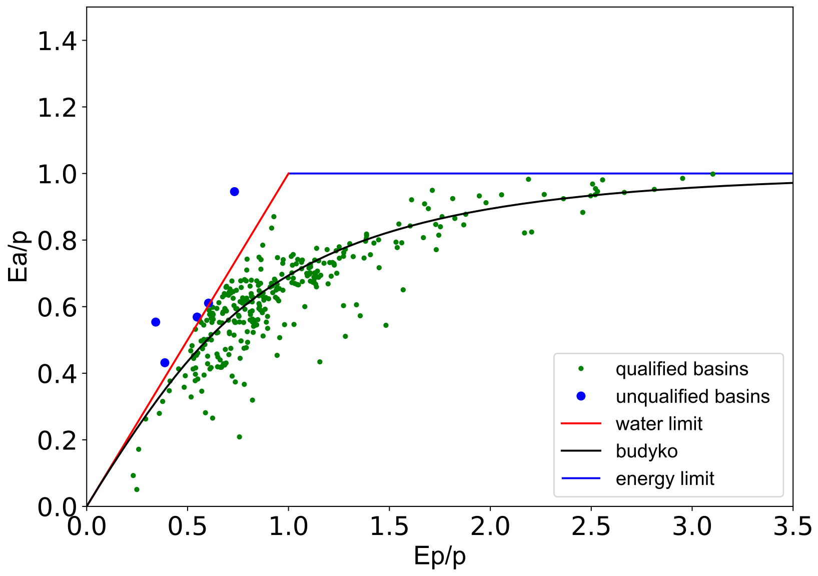


Figure 2.3: Budyko Framework for checking water balance

In the Figure(2.3), the red line represents the water limit which means the actual evaporation should always be less than the precipitation; the blue line represents the energy limit which means the actual evaporation should always be less than the potential evaporation. After checking the water balance, 392 basins (green dots under the red line and blue line) are selected out of 438 basins and qualify to be included in this study. 392 basins are included because they have their water balance closed for a long time while other basins are excluded for their water balance are not closed over the analysed period.

2.2.2. Selection of case study basins

Catchments with streamflow/rainfall ratios of about 0.2 or less are referred to as dry catchments (Gan et al. [1997]). Usually, the hydrological process of dry basins is more complex and variable than that of wet basins with relatively higher streamflow/ rainfall ratios. Six basins with different sizes and dryness are chosen to be case study basins to analyze the relationship with the variation of storm-dependent parameters. Necessary information about the selected areas is shown in the table below.

Table 2.2: Selected basins

Gauge ID	Basin	Name used in this study
7068000	Current river at Doniphan, MO	A
1664000	Rappahannock river at Remington,VA	B
3164000	South fork new fiver near Jefferson,NC	C
3161000	New river near Galax,VA	D
12027500	Chehalis river near grand mound, WA	E
11532500	Smith River near Crescent city, CA	F

Information is taken from https://waterwatch.usgs.gov

Table 2.3: Characteristics of selected basins

Basin	Q/P	Dryness index	Area (km^2)	Vegetation	Dominant soil extrue
A	0.35	0.87	2038	mixed forest closed shrublands	Silty clay loam
B	0.35	0.85	620	mixed forest, closed shrublands	Clay loam
C	0.49	0.64	1131	mixed forest, closed shrublands	Loam
D	0.50	0.57	205	mixed forest, closed shrublands	Loam
E	0.73	0.38	895	evergreen broadleaf forest	Clay loam
F	0.73	0.22	609	evergreen broadleaf forest	Clay loam

Information is taken from Mopex data base (https://hydrology.nws.noaa.gov/pub/gcip/mopex/US_Data/). In this table, Q/P means annual runoff/annual precipitation ratio.

2.3. Model calibration and validation

Automatic calibration via computer codes is an optimisation problem, that is to find the set of model parameters based on which the model output matches the observed system response with the highest degree of similarity. Objective functions are utilised as mathematical tools to represent the model performance, which aggregate model residuals into a single value. An optimisation algorithm will be applied to find the maximum value of the objective function. After calibration, model validation is used to evaluate the prediction performance of the model. In this research, for model validation, we applied the split sample test, that is to test model performance using the same parameter sets as the calibration process but over independent time series (Winsemius et al. [2009]). However, it is limited to only focus on the optimal parameter set, which will possibly misrepresent the real-world systematic dynamics. That can be reflected by the equifinality problem: different parameter sets yield equally good results. Hence it makes more sense to report uncertainty intervals, within which the parameter sets and model responses are relatively feasible and close to the reality. Uncertainty intervals will be computed using methods described in the following sections.

2.3.1. Storm-based method

Several assumptions are made before doing the research of storm-based method: because of spatial and temporal averaging, the flux errors primarily affect model performance. Besides, these errors can be described by sampling one or two parameters from a probability distribution at the beginning of each storm. The main characteristic of the storm-based method is that storm-based parameters are varying by storm instead of being kept constant during the analysed time series. For the storm-based method, the objective function is defined as Equation(2.20) to express the model performance. $Q_{o,i}$ is the time series of observed discharge, $Q_{s,i}$ is the time series of simulated discharge by model and n is the length of each storm period.

$$NSE = 1 - \frac{\sum_{i=1}^{n}\left(Q_{s,i} - Q_{o,i}\right)^2}{\sum_{i=1}^{n}\left(Q_{o,i} - \bar{Q}_{o,i}\right)^2} \tag{2.20}$$

The basic methodology of finding storm-dependent parameters can be illustrated as the following steps:

- Step 1: Dividing a time series of daily observations into distinct storm events. The storm events are defined by inter-storm dry spells of 1 or more days, followed by a day with rainfall exceeding a specific threshold
  Before calibration, storm-dependent parameters and storm epochs are needed to be identified from a rainfall time-series for a given basin. Before identifying the storm-dependent parameters, the storm epochs are defined by inter-storm dry spells of one or more days, followed by a day with rainfall exceeding a specific threshold.

- Step 2: Pre-screening model parameters that are likely to vary by storm (this is the sensitivity analysis)
  In step 2, a daily runoff time series Q_o and calibrated parameter set P_o will be obtained by fitting the

hydrological model to observed data assuming all the parameters are time-invariant. Another runoff time series Q_i will be generated by changing the selected i_{th} hydrological parameter stochastically and keeping the other parameters constant given the same observed data series. Each time only one parameter is selected to be stochastical. The selected parameter will be sampled from the assumed distribution at the beginning of each storm. Finally, the Nash-Sutcliffe statistic $NS_(i)$ will then be evaluated for the runoff time series Q_o (treated as traditional 'observed' data)and Q_i(treated as the 'simulated' time series). Then it can be observed that in which parameter the model predictions are sensitive to storm-dependent variation.

- Step 3: Estimating storm-dependent values for the model parameter(s) selected in step 2 (this is the calibration step)

- step 4: Testing model performance using the parameter sets calibrated in the calibration process over independent time series (this is the validation step)

During the calibration period, parameters are calibrated by storms with storm-dependent parameters varying by storms and other parameters constant. Values of the constant parameter are set as the same value from P_o obtained in the calibration period. Additionally, the end stage of the storm will be the initial condition(include water storage from conceptual production reservoir and routing reservoir in the model) of next storm. This means if the calibrated period is divided into m storms, there will be m sets of parameters, a series of initial water storage $PS_{(i)}$ (i=1, 2, ... $(m-1)$) from production reservoir and a series of initial water storage $RS_{(i)}$ (i=1, 2, ... $(m-1)$) from routing reservoir.
The basic methodology of calibration in storm-based method can be explained as following steps:

- Step 3a: Run the model for entire period without dividing it into storms to obtain the values for constant parameters $P_c(X_{c1}, X_{c2}, X_{c3}, X_{c4})$

- Step 3b: Set the values for constant parameters and bounds for storm-dependent parameters

- Step 3c: Calibrate the parameters for the first storm and save the water storage values at the end day of the storm from two reservoirs from the model

- Step 3d: Apply each water storage values at the end day of the former storm as initial conditions for the next storm until finishing the calibration of all the storms.

Here is a small example of around 200 days of time series to briefly illustrate the calibration process. During this period, 31 storms are obtained and the calibrated parameter set for theses storms are $P_1(X1_1, X2_1, X3_1, X4_1)$, $P_2(X1_2, X2_2, X3_2, X4_2)$, ... and $P_{30}(X1_{30}, X2_{30}, X3_{30}, X4_{30})$. The sub-periods between red dashed lines are every single storm epoch. The blue line is the simulated discharge Q_i from storm i computed by its corresponding parameter set P_i.

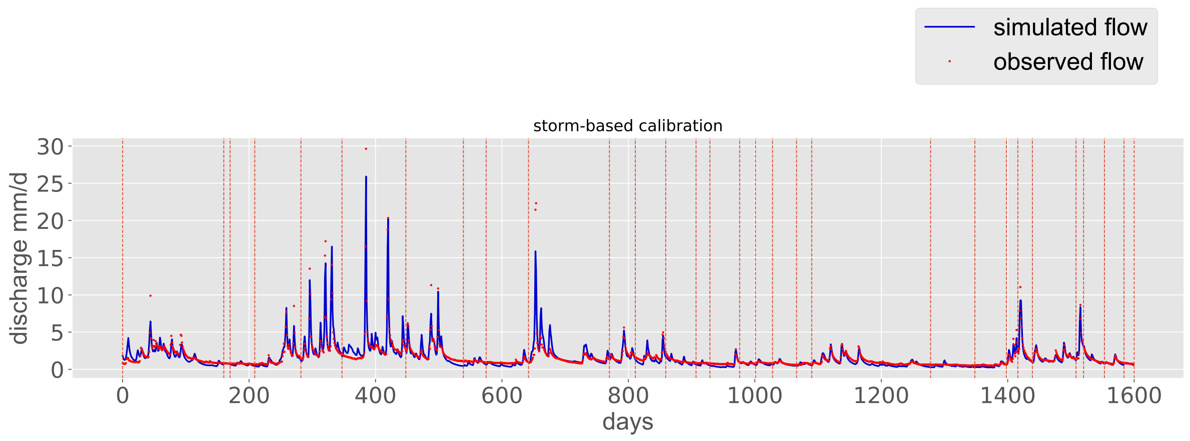


Figure 2.4: an example of storm-based calibration

During the step 4 (validation period), the analysed time series will be again divided into M storms(M may be different from m). The discharge will be predicted for each storm by randomly sampling a parameter set from m calibrated parameter sets. During the sampling, the water storage at the end of each storm will be used as the initial conditions of its next storm. After an entire run of sampling for all the storms in the entire validation period, this sampling process will be repeated for M times so that M series of predicted streamflow will be obtained.
For the sampling part, there are two methods to do the sampling. One is to sample the parameter set for each storm directly from the m parameter sets from the calibration period. The other one to obtain the parameter set for each storm is firstly fitting a distribution to calibrated storm-dependent parameters and then sampling from the fitted distributions. A comparison will be made between the performance obtained by these two sampling methods.
At each time step, mean predicted value and standard deviation of predicted values will be calculated so that 5% (Q_{5sim}) and 95% (Q_{95sim}) quantiles will be calculated for each time step. The area between Q_{5sim} series and Q_{95sim} series is the obtained 90% uncertainty band.
The basic methodology of validation in the storm-based method can be illustrated as the following steps:

- Step 4a: Divide the validation period into storm epochs like what is done in the calibration period. M storms will be obtained during this period.

- Step 4b: For the first storm, randomly pick a parameter set from the calibrated parameter sets and using the end state of the calibration period as the initial condition of the first storm.

- Step 4c: Move to the next storm. Each time using the end condition of the former storm as the initial condition for the next storm and randomly pick a parameter set from the calibrated results.

- Step 4d: After finish the predicting the discharge for all the storms, one run is finished.

- Step 4e: The run (step 4b, step 4c and step 4d) will be repeated for M times.

- Step 4f: m discharge values will be obtained at each time step. 5% and 95% quantile value will be computed as the boundary lines of the uncertainty band.

Here is a small example of around 120 days time series following the example calibration period mentioned before to illustrate the validation process briefly. A shorter length of a period is selected so that the number of storms identified in this period will be less and the illustration can be brief and plain. This period

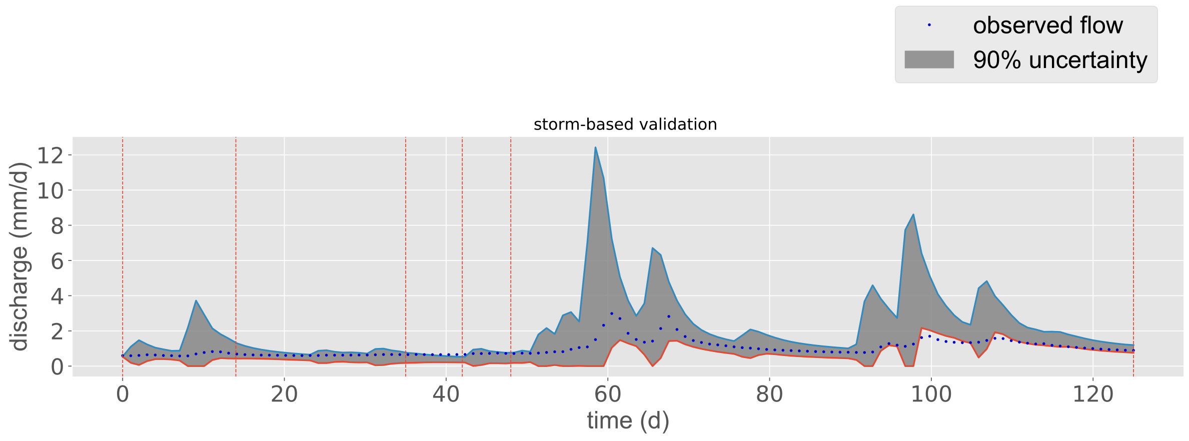


Figure 2.5: an example of storm-based validation

is a time series following the example period mentioned before. During this period, five storms are identified. For the first run, the first storm is predicted by using the end state of the example calibration period and randomly picked parameter set from $P_i(X1_i, X2_i, X3_i, X4_i)(i = 1,2,3,....31)$. When moving to the next storm, it is predicted by using the end state of the first storm and randomly picking a parameter set from $P_i(X1_i, X2_i, X3_i, X4_i)(i = 1,2,3,....31)$. For the third storm, it is predicted by using the end state of the second storm and again randomly picking a parameter set from $P_i(X1_i, X2_i, X3_i, X4_i)(i = 1,2,3,....31)$. For the fourth storm, it is predicted by using the end state of the second storm and again randomly picked parameter set from $P_i(X1_i, X2_i, X3_i, X4_i)(i = 1,2,3,....31)$. For the last storm, it is predicted by using the end state of the second storm and again randomly picked parameter set from $P_i(X1_i, X2_i, X3_i, X4_i)(i = 1,2,3,....31)$. Then one run is finished.This run will be repeated for as many times as possible. Usually we make the repeated times equal to the number of storms identified in the validation period (M). At each day, 5% and 95% percentile discharge will be calculated. In the example validation period below, the blue line represents the 95% percentile discharge, and the red line represents the 5% percentile discharge. It can be seen that the uncertainty band has a good match with observed discharge.

2.3.2. Existing methods

- **Traditional method with constant parameters**

  The traditional method assumes all the parameters are deterministic in the calibration and validation period. For the traditional method, Nash-Sutcliffe Efficiency or NSE is commonly used in conceptual rainfall-runoff modelling as an objective function (Equation (2.21)) to express model performance. $Q_{o,i}$ is the time series of observed discharge, $Q_{s,i}$ is the time series of simulated discharge by model and $\bar{Q}_{o,i}$ is the mean of observed discharge.

$$NSE = 1 - \frac{\sum_{i=1}^{n}\left(Q_{s,i} - Q_{o,i}\right)^2}{\sum_{i=1}^{n}\left(Q_{o,i} - \bar{Q}_{o,i}\right)^2} \tag{2.21}$$

  This Objective function with larger NSE values indicates a better fit with the data. In the computer codes of the objective function, a parameter set will be taken to compute the simulated discharge. Then the resulting simulated discharge will be used to compute the NSE values. After defining the objective function in computer codes, differential evolution available as a python function is chosen as an optimization algorithm which is a global optimization algorithm suitable for rainfall-runoff models. Global optimization algorithms will look for the global minimum of the objective function and the parameter set that give this minimum value. The objective function to be minimized (for the traditional method in this research, negative NSE) and bounds for the model parameters will be required as inputs for the differential evolution algorithm. Parameter bounds should make sure some parameters be positive, which make parameters physically reasonable and also consider past experience when researchers applied models in other basins.

Optimal parameter set $P_O(X1_O, X2_O, X3_O, X4_O)$ that minimizes the negative NSE obtained from the calibration period will be used to predict the discharge during the validation period. Variances between observed values and simulated values is expressed by σ (Equation(2.22)). By taking σ as standard deviations and predicted values Q_{sim} as mean values, 5% (Q_{5sim}) and 95% (Q_{95sim}) quantiles are calculated at each time step. The area between Q_{5sim} and Q_{95sim} is the obtained 90% uncertainty band.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{n}\left(Q_{s,i} - Q_{o,i}\right)^2} \tag{2.22}$$

The necessary steps of traditional method are:

– Step 1: Define objective function.

– Step 2: Choose an optimization algorithm for the objective function and obtain the optimized parameter set. calculate the standard deviation σ between simulated discharge and observed discharge at the same time.

– Step 3: Apply this optimized parameter set to predict the discharge in the validation period.

– Step 4: Use the σ obtained before and the predicted discharge to compute 5% and 95% value at each time step. The uncertainty band is computed by using 5% and 95% values as upper and lower limits.

For the traditional method, parameter $X1$, $X2$, $X3$ and $X4$ in parameter set P_O are constant during the entire calibration and validation period.

- **GLUE(Generalized Likelihood Uncertainty Estimation) method**

Unlike the traditional method, the GLUE method (Beven and Binley, 1992) rejects the idea of one single optimal solution and adopts the concept of equifinality of models and parameters. Before calibration, the objective function is also defined by Equation(2.23) but a threshold (>0.6 or 0.7) is defined for distinguishing good (behavioral) from bad (non-behavioral) models. $Q_{o,i}$ is the time series of observed discharge, $Q_{s,i}$ is the time series of simulated discharge by model and $\bar{Q}_{o,i}$ is the mean observed discharge.

$$N_{NSE} = 1 - \frac{\sum_{i=1}^{n}\left(Q_{s,i} - Q_{o,i}\right)^2}{\sum_{i=1}^{n}\left(Q_{o,i} - \bar{Q}_{o,i}\right)^2} > 0.6 \tag{2.23}$$

A large number of parameter sets are then generated by Monte Carlo sampling, and each parameter set that gives the value of the objective function larger than the threshold will be retained as behavioral models and all others will be discarded (like shown in Figure8). Behaved parameter sets $P_i(X1_i, X2_i, X3_i, X4_i)$(i=1,2,...n) will be used to predict the runoff during validation period. If n behaved models are obtained, n behaved parameter sets will be obtained. Theses behaved parameter sets will be used to compute n series predicted streamflow $Q_{m(i)}$ (i=1,2,...n) and rescaled to a cumulative sum of 1 in the validation period. For each time step, the cumulative distribution of simulated discharges is constructed using the rescaled weights. At each time step, mean predicted value and standard deviation of predicted values are calculated as well as 5% (Q_{5sim}) and 95% (Q_{95sim}) quantiles are calculated for each time step. The area between Q_{5sim} series and Q_{95sim} series is the obtained 90% uncertainty band. Each parameter is still constant during the analysed period in each behavioral model. However, in the GLUE method, multiple solutions are adopted.
The necessary steps of GLUE method are:

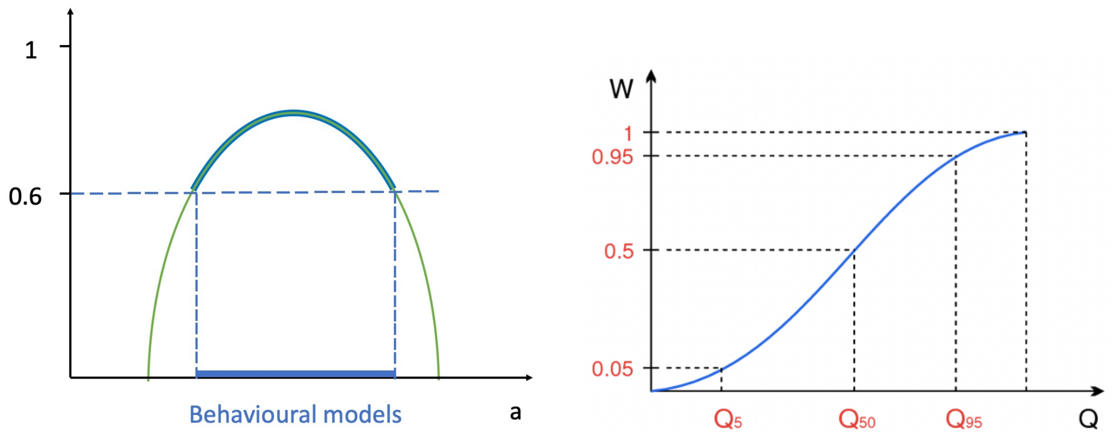


Figure 2.6: Behavioural models and rescaled cumulative distribution for each time step

- Step 1: Define the objective function and set the threshold to retain the behaviour models.
- Step 2: Using Monte Carlo sampling to randomly generate a large number of parameter sets between the parameter bounds.
- Step 3: Screen out the parameter sets that can simulate the discharge satisfying the function (Equation(2.23)).
- Step 4: Apply the screened out parameter sets to predict the discharge in the validation period. If n parameter sets are retained, there will be n discharge values at each time step.
- Step 5: Calculate the standard deviation and mean value of predicted discharge at each time step to compute 5% and 95% value. The uncertainty band is computed by using 5% and 95% values as upper and lower limits of the uncertainty band.

- **Brief summary**

Table(2.4) is a summary of the main differences or calibrated parameters between three different methods. The ways to compute the uncertainty band is also different in the three methods, as described above.

Table 2.4: Differences for calibrated parameters between three methods

Traditional method	Parameters are constant in time	$\mathrm{P}_O(X1_O, X2_O, X3_O, X4_O)$
GLUE method	Parameters are constant in time	$\mathrm{P}_i(X1_i, X2_i, X3_i, X4_i)(i = 1,2,...n)$ n is the number of behavioral models
Storm-based method	Parameters are variable in time	$\mathrm{P}_j(X1_j, X2_j, X3_j, X4_j)(j = 1,2,...m)$ m is the number of storm epochs

2.4. Assessment of predictive performance

There is a greater necessity for hydrologists to have realistic prediction intervals and other representations of uncertainty that describe the possible difference between actual flows and their forecasted values because hydrological models become more widely practised. Uncertainty analysis has now become a regular study in the application of conceptual rainfall-runoff modelsStedinger et al. [2008]. General uncertainty of prediction performance is evaluated by mean of $logscore_{(i)}$ and $RMSE$ (Root Mean Square Error). Additionally, $RMSE$ can also be applied for the evaluation of highflow and lowflow performance, respectively. RMSE is used to assess the general performance and highflow and lowflow performance. Logscore (Equation2.25) gives a good overall measure of how good the uncertainty band is. When we calculate the RMSE, we only look at the mean of the predicted discharge, so we do not take the uncertainty of the predicted discharge into account. While the logscore also looks into the distributions of the predicted discharge, which is a complete measurement of the model performance.

$$RMSE = \sqrt{\frac{1}{N}\sum\left(m_{obs(i)} - m_{predict(i)}\right)^2} \tag{2.24}$$

$$logscore_{(i)} = -\frac{1}{2}\log 2\pi - \frac{1}{2}\log\left(v_{obs(i)} + v_{Predict(i)}\right) - \frac{\left(m_{obs(i)} - m_{Predict(i)}\right)^2}{2\left(v_{obs(i)} + v_{predict(i)}\right)} \tag{2.25}$$

$$logscore = \frac{1}{N}\sum(logscore_{(i)}) \tag{2.26}$$

where $RMSE$ is the root mean square error for entire flow, highflow or lowflow, N is the number of days for entire flow, highflow or lowflow, $m_{obs(i)}$ is the mean of the observed discharge at each time step, $m_{predict(i)}$ is the mean of the predicted discharge at each time step, $v_{obs(i)}$ is the variation of observed discharge at each time, $v_{predict(i)}$ is the variation of predicted discharge at each time step, $logscore_{(i)}$ is the logscore value at each time step and $logscore$ is the mean of $logscore_{(i)}$ for entire flow. A good uncertainty band should be close to the observation (accuracy), and it should be narrow (precision). Nevertheless, these two are related: if the mean prediction is further away from the observation, then a wider uncertainty band that covers the observation is better.

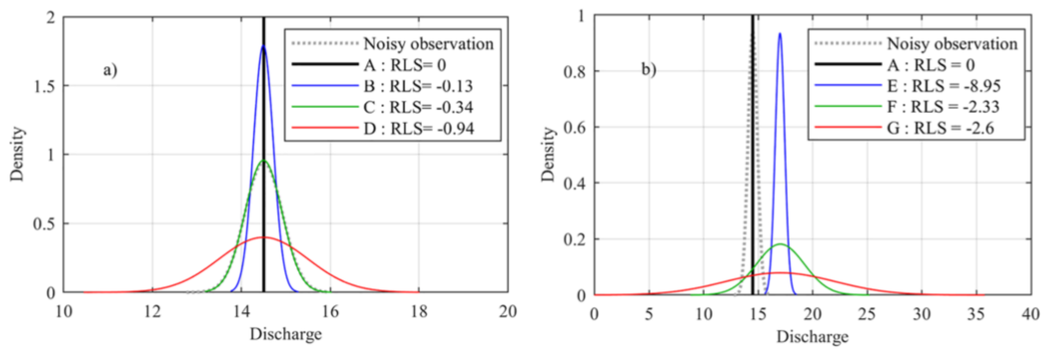


Figure 2.7: Comparison between different predictions with a noisy observation.(cite from Tajiki et al. [2020])

This figure (Tajiki et al. [2020]) helps understand this better. The RLS in the figure means the extent of a probabilistic prediction approaching to the best prediction which has the maximum logscore value. When the mean of the uncertainty band is the observed discharge(a) in the Figure(2.7)), the narrower the uncertainty band is, the better the prediction performance is. However, when the mean prediction is further away from the observed discharge(b) in the Figure(2.7)), a wider uncertainty band would be better.
Table(2.5) lists the values that are going to be used for measuring the model prediction performance by different methods. For GLUE method, if there are n sets of behavioural models, there are n predicted $Q_i$for each time step. For the storm-based method, if there are m storm epochs during the calibration period, there are m predicted $Q_i$for each time step. Calculation of RMSE for highflow and lowflow is similar to the calculation of RMSE for the entire series.

Table 2.5: Values used for evaluating model performance obtained by different methods

	For each time step (i)		
	Traditional method	GLUE method	Storm-based method
$m_{\text{obs(i)}}$	observed Q_i	observed Q_i	observed Q_i
$m_{\text{Predict(i)}}$	Predicted Q_i	mean of predicted Q_i	mean predicted Q_i
$v_{\text{obs(i)}}$	0	0	0
$v_{\text{predict(i)}}$	σ obtained in calibration period	Variance of predicted Q_i	Variance of predicted Q_i

3
Results

In this Chapter, the first two research questions(to what extent do model parameters vary by storm/ event & to what extent is this variation random and can it be predicted from rainfall event characteristic and initial conditions) will be answered in the section3.1.1 . The third question (to what extent does accounting for variation in model parameters capture model errors and improve rainfall-runoff prediction) will be answered in section 3.2. The last research question (whether the extent of variation depends on basin characteristics) will be answered in section 3.3.

3.1. Results obtained by storm-based method

Six basins with different dryness index(table(2.3)) will be researched by using the storm-based method. Basin A, which is driest, will be used as an example to illustrate the detailed procedure of calibration and validation based on the storm-dependent parameters. Besides, further explorations about thresholds for determining the storm-epoch will be proceeded. Additionally, the research on basin A will give some applicable methods (e.g.the choice of thresholds for identifying the storm epochs and sampling methods) for other basins. Moreover, the effect of basin dryness on the parameter sensitivity will be analysed to improve the storm-based method.

3.1.1. Case study of Basin A

Basin A with the highest dryness index in six selected case basins is chosen as an example to explore to what extent does storm-based parameters vary by storms (research question 1) and whether this extent can be predicted from event characteristics and conditions (research question 2). Results obtained by this method later will be compared with those obtained by the traditional method and GLUE method to answer the research question 3.

- **Storm epochs and sensitivity analysis**

  First of all, the analysed time series needs to be divided into distinct storm events because the calibration and validation will be researched by storms and how storm-dependent parameters vary by storms will be analysed. Storm epochs are defined by inter-storm with dry spells of one or more days, followed by a rainfall exceeding a specific threshold. The specific threshold here in basin A is related to the median value of runoff series. If the median value of the entire rainfall series is larger than 0.3 (mm/day), the threshold will be set as 0.1 (mm/day), and if the median value of the entire rainfall series is smaller than 0.15 (mm/day), the threshold will be set as 0.15 (mm/day). Otherwise, the threshold will be set as the median value of the rainfall series. For this basin, the median value of the rainfall series is larger than 0.15 (mm/day) and smaller than 0.3 (mm/day) so the threshold is set as the median value of the entire rainfall series. 99 and 103 storm epochs are identified respectively in five-year calibration and five-year validation period respectively. Figure(3.1) shows how many storm epochs are identified in the calibration period.

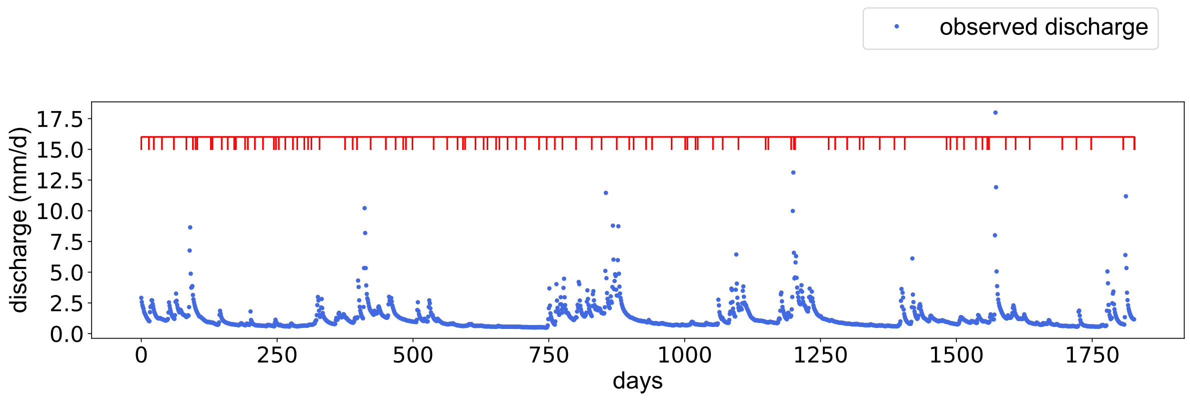


Figure 3.1: Storm epochs during analysed calibration period for basin A

After identifying the storms in the analysed period, the effects of storm-dependent parameter stochasticity were explored using a synthetic daily runoff time series Q_0 derived from five-year daily rainfall record for basin A. The series Q_0 is generated by the parameter set P_o when the model is calibrated for the entire 5-year period with all the parameters in-variant. Another runoff time series Q_{p_i} were generated with p_i (the i_{th} parameter) selected as be stochastic (its distribution is assumed to be normal distribution with the i_{th} parameter value from P_o that generates Q_0 and a given coefficient of variation CV), while keeping the remaining parameters constant (values are the same that generate Q_0). A new value of p_i was sampled from assumed normal distribution at the beginning of each storm. The Nash-Sutcliffe statistic $NS_{(i)}$ was then evaluated like shown in the equation Equation(3.1): the runoff series Q_0 and Q_i, where the Q_i was treated as 'simulated' time series Q_{p_i} and Q_0 as the traditional 'observed' time series.

$$NS_i = 1 - \frac{\sum_{i=1}^{n} \left(Q_{p_i,i} - Q_{0,i}\right)^2}{\sum_{i=1}^{n} \left(Q_{0,i} - \bar{Q}_{0,i}\right)^2} \tag{3.1}$$

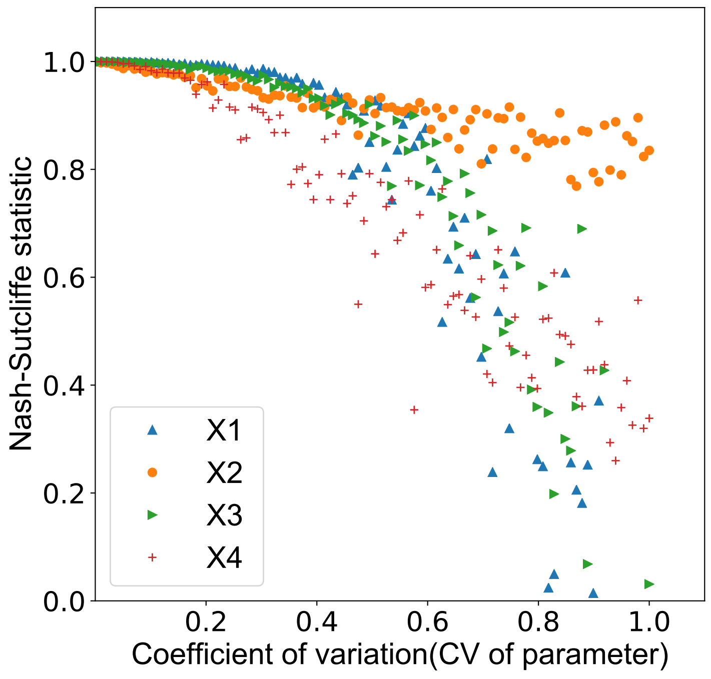


Figure 3.2: Nash-Sutcliffe efficiency sensitivity analysis

Figure(3,2) presents a plot of $NS_{(i)}$(i=1,2,3,4) for a range of CVs. Several important observations can be made from this plot:

1. The model predictions, and hence the NS statistic, are most sensitive to storm-dependent variation in the parameter X4(the HU1 unit hydrograph time base). This parameter represents how many days of the routing process. A CV of 50% reduces the NS statistic to values as low as 50-65%.
2. The second most sensitive parameter is X1(maximum capacity of production reservoir). This parameter relates to soil moisture and generating runoff. If X1 is small, there will be more discharge given the same forcing.

From these two observations, the parameter X1 and X4 will be chosen as storm-dependent parameters, and other parameters will be kept constant during the calibration and validation period. The values of X2 and X3 are constant and the same as those in P_o, which is obtained by running the model for the entire period without dividing the 5-year calibration period into storms. The calibration will be proceeded for each storm and apply the end condition of the former storm as the initial condition of the next storm. Figure(3.3) is the calibration results by making X1 and X4 varying by storms and the other two parameters constant using the objective function Equation(2.5) for each storm as introduced in section 2.3.1. 99 calibrated parameter sets respectively for 99 storms are obtained after this procedure, and we can see from the calibration results, most peak flows and low regression flows are captured well.

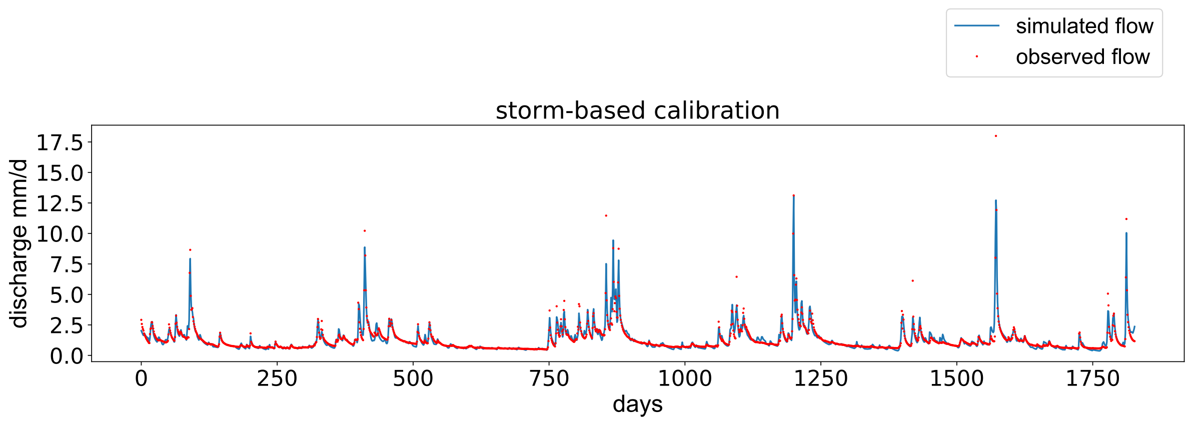


Figure 3.3: calibration result for storm-based method for Basin A

Figure(3.4) and Figure(3.5) are the plots of the calibrated values of storm-based parameters X1 and X4 from each storm. In this figure, the blue dots represent the X1 or X4 values from each storm, and the red lines are the X1 and X4 values in the P_o (obtained by running the model for the entire period without dividing the 5-year calibration period into storms). We can see there are some variations for both X1 and X4, which means there is structure error in the model. Parameter X1 and X4 represent the maximum capacity of the production reservoir and routing days respectively. Usually, the maximum capacity of the production reservoir should be constant during the analysed time series while parameter X1 shows variations by storms. Additionally, routing days is assumed in-variant in the model while parameter X4 also shows variations by storms. The coefficient of variation for X1 and and the X4 are 0.17 and 0.47 respectively. The extent of variation of X4 is more significant than x1, which agrees with the observations in the former subsection that X4 is more sensitive to the storms in the sensitivity analysis in Figure(3.2). It is worth noting that in the scatter plot of X1, the red line which represents the X1 value from P_o is totally below the scatters. This means when doing the calibration for the entire period without dividing the 5-year calibration period into storms, parameter X1 is underestimated. A larger calibrated X1 for each storm(maximum capacity of the production reservoir will give a larger water storage at the end of this storm. The next storm will apply this water storage at the end of the former storm as the initial water storage. When the initial water storage is larger, it is easier to generate the runoff. That is why the peaks are captured well in the storm-based calibration. These two scatter plots illustrate that using storm-dependent parameters can capture the model error and answered the research question 1.

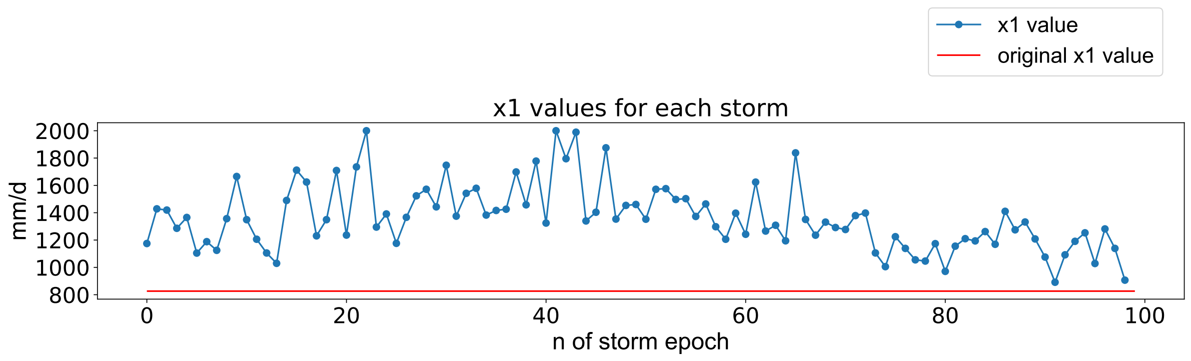


Figure 3.4: calibrated x1 values for storm-based method for Basin A

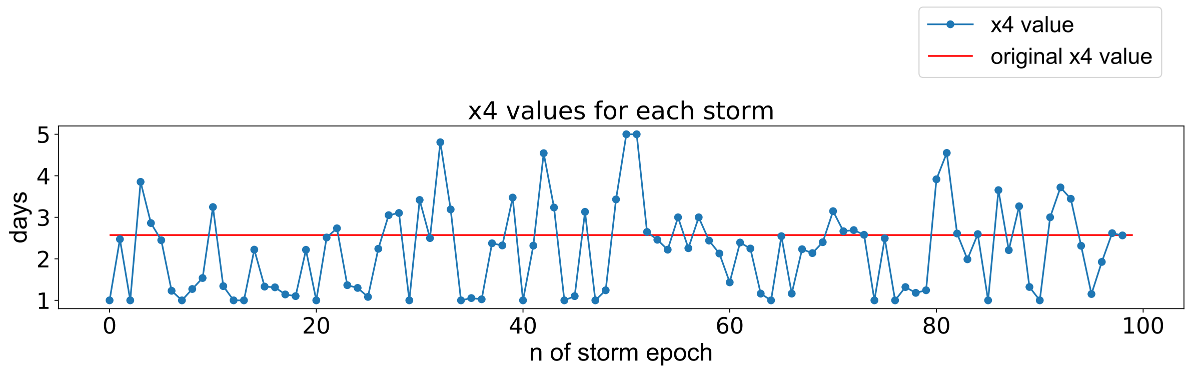


Figure 3.5: calibrated x4 values for storm-based method for Basin A

- **Validation results obtained by storm-based methods**

During the validation period, discharge is predicted by storms as well with 99 parameter sets obtained from the calibration period. When predicting the discharge for each storm, one parameter set will be randomly picked from the 99 parameter sets obtained in the calibration period. Two different sampling method and different combinations of variable parameters are explored for basin A. Prediction results obtained by different sampling methods will be compared to give the instructions of choosing sampling methods for other basins. Sampling methods that can give relative good performance will be identified and apply for basin B, C, D, E and F.
The sampling methods are generally divided into two kinds. The first one is sampling each parameter set from the 99 calibrated parameter sets for each storm directly. After distributing the parameter set randomly for each storm, an entire time series of predicted discharge for 5 years will be obtained. This process of randomly distributing the parameter set for each storm will be repeated for 99 times to make sure the process of distributing the parameter sets for each storm is random enough. The repeating process will give 99 predicted discharge at each time step so that the variances and mean value of the predicted discharge at each time step can be calculated. Hence, 5% (Q_{5sim}) and 95% (Q_{95sim}) quantiles of predicted discharge is computed at each time step. Average logscore value and root mean square error for the entire time series and the root mean square error for highflow and lowflow are computed respectively at the same time. Validation result obtained by randomly sampling the parameter set for each storm respectively is shown below in figure (3.5). The area between Q_{5sim} and Q_{95sim} is the obtained 90% uncertainty band.

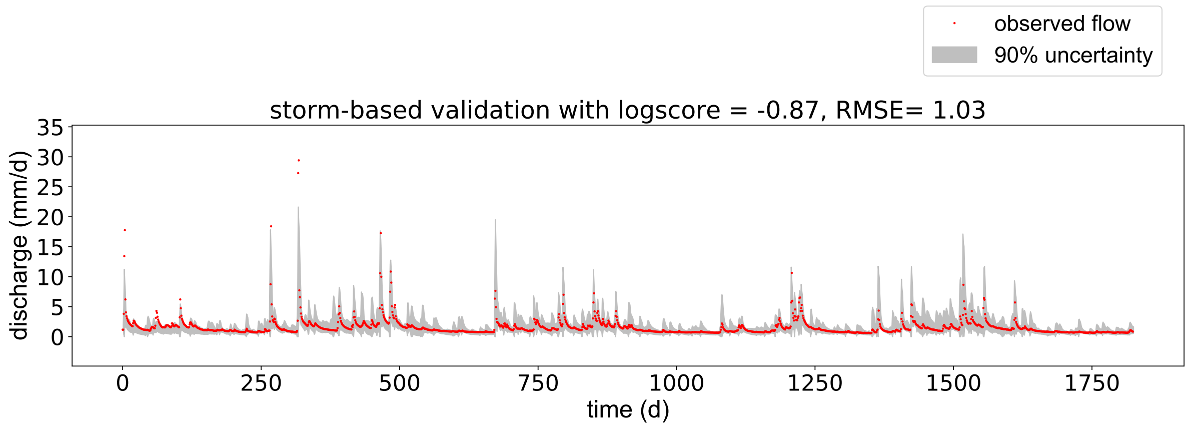


Figure 3.6: validation result by storm-based method for basin A

The second sampling method is firstly fitting a distribution to the calibrated parameter values for X1 or X4 and then sampling the parameter values of X1 and X4 from their distributions. This sampling method is mainly different from the first sampling method in including boundary values or excluding boundary values.

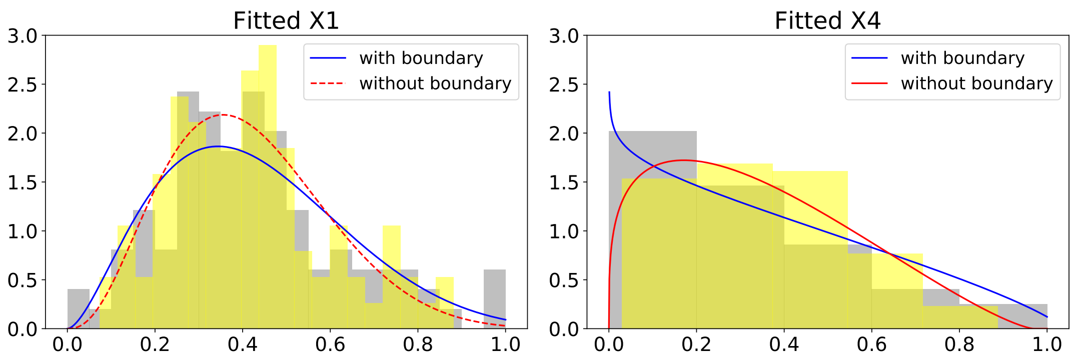


Figure 3.7: histograms and distributions to parameter X1 & X4

When fit the distributions to calibrated parameter X1 and X4, histograms for calibrated values of storm-dependent parameters X1 and X4 are plotted in Figure(3.7). Beta distributions are fitted to the histograms. To fit better distributions, all the X1 and X4 values are normalized and then boundary values are excluded. This is because boundary values usually give unreasonable beta fit as it can be seen from the difference between blue and red distributions. In Figure(3.7), the blue lines are the distributions to grey histograms which include the boundary values of storm-dependent parameters, and the red lines are the distributions to yellow histograms which exclude the boundary values.
Moreover, within the second sampling method, three different combinations are tried. Three trials are proceeded by sampling values from red distributions (exclude the boundary values) with both X1 and X4 variable, with only X1 variable and with only X4 variable respectively. The computation of the uncertainty band and the calculation of logscore and RMSE (Root Mean Square Error) are the same as before. Three figures below show the validation results by different combinations using the second sampling method. When X1 and X4 are both variable (Figure(3.8)), values of parameter X1 and X4 are sampled from the red beta distributions. It gives better general performance than that obtained by the first sampling method because it gives relative higher logscore value and lower Root Mean Square Error. When $X1$ is fixed (Figure(3.9))and only x4 variable, although the uncertainty band looks very narrow, it can not capture some peaks and lowflow in the regression part as good as the uncertainty band obtained in the first sampling method. When $X4$ is fixed (Figure(3.10)), and $X1$ is variable, although the uncertainty band captures more peaks and covers almost all the observed discharge, it looks too peaky because it usually overestimates some lowflow.

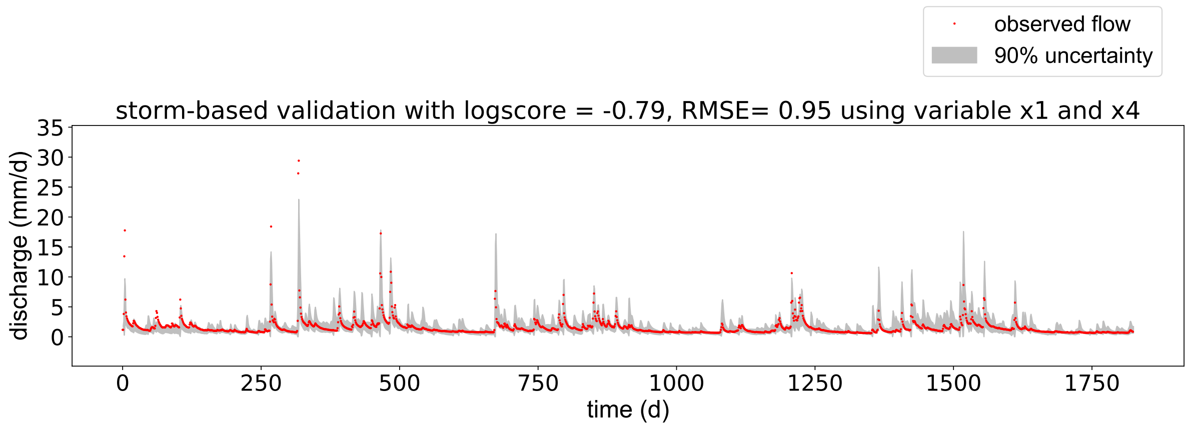


Figure 3.8: validation result by storm-based method for basin A with both x1, x4 variable

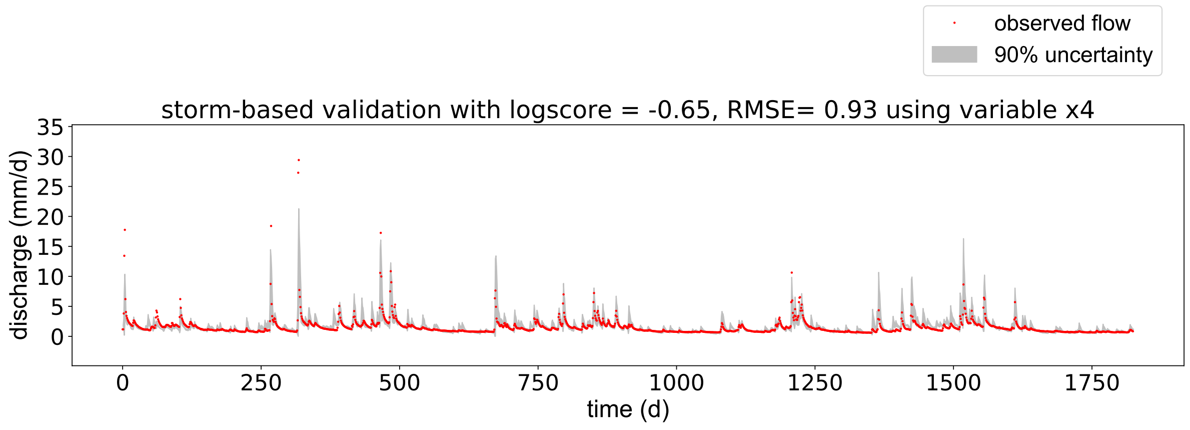


Figure 3.9: validation result by storm-based method for basin A with x1 fixed

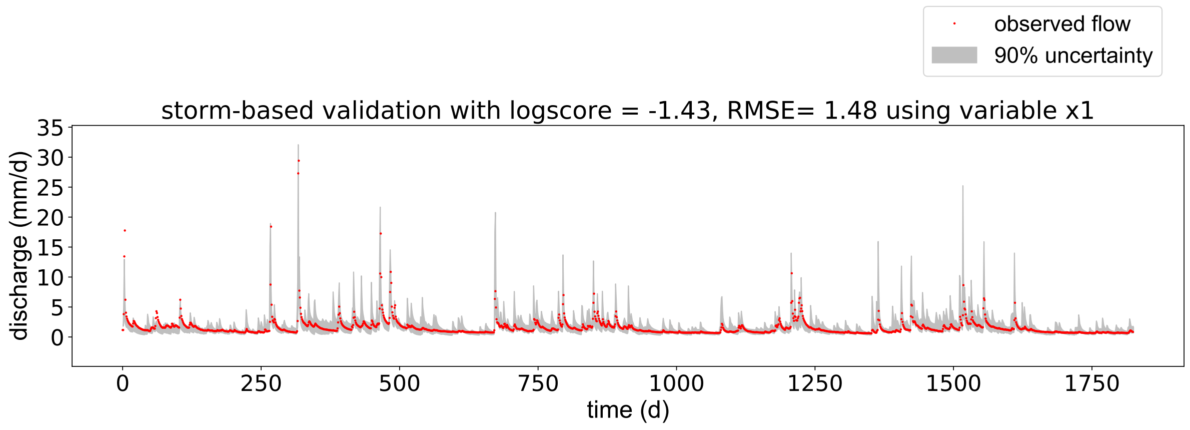


Figure 3.10: validation result by storm-based method for basin A with x4 fixed

Table(3.1) lists the RMSE (root mean square error) value for highflow, lowflow and entire series of predicted discharge and logscore value for entire time series. It compares the performance obtained by directly sampling the parameters from calibrated results and sampling the parameters from distributions. When $X1$ is fixed, although it gives the highest value of logscore and lower values for lowflow and entire flow, the uncertainty band is too narrow to cover the observed discharge, the uncertainty band of some parts missed the observed discharge as can be seen from the figure. When $X4$ is fixed, it gives the poorest general performance for it gives the smallest logscore value and largest RMSE value. The performance is worst for highflow and lowflow as well.

Table 3.1: Comparison between two sampling methods

		highfow	lowflow	total flow	
basin A		RMSE	RMSE	Logscore	RMSE
Directly sampling	Both X1 and X4	1.48	0.91	-0.87	1.03
Sampling from the fitted distributions	Both variable	1.23	0.86	-0.79	0.95
	X1 fitted	1.26	0.84	-0.65	0.93
	X4 fitted	2.66	1.09	-1.43	1.48

From this table and figures above, it is recommended that sampling both X1 and X4 randomly from the grey histograms (calibrated results) directly and red distributions (boundary values are excluded). They are considered as good trials of storm-based validation. These two sampling methods will be applied to the other five basins for the storm-based validation.

- **Relation between thresholds and performance**

When defining the storm epochs, the threshold is set as a value relative with the median value of rainfall series. However, the threshold is believed to be another factor that affects the model performance of storm-based method. The reason is the threshold determines the number of storms identified. In principle, if the threshold for defining the storm epochs is large enough, then there will be only one storm event identified which makes the storm-based method to traditional method with constant parameters during the entire analysed period. If the threshold is lower enough, there will be large amounts of storm events identified which means the analysed period will be calibrated almost day by day. The model should give the most precise result when it is calibrated day by day. Hence, a further research of the effect of thresholds for identifying storm epochs on performance is explored in basin A by changing thresholds to a series of values (Table(3.2)) to see whether a lower threshold gives better prediction performance and whether using the threshold relative with the median value of the entire rainfall series is reasonable.

		total flow		highfow	lowflow
threshold (mm/day)	storms	logscore	RMSE	RMSE	RMSE
6	16	-0.77	0.95	1.24	0.88
5	17	-0.79	0.91	1.06	0.88
4	22	-0.80	0.90	1.24	0.82
3	26	-0.68	0.85	0.97	0.83

2	35	-0.72	0.93	1.26	0.84
1	47	-0.82	0.96	1.20	0.90
0.9	50	-0.79	0.95	1.23	0.88
0.8	52	-0.82	0.98	1.33	0.90
0.7	59	-0.82	0.96	1.30	0.87
0.6	65	-0.83	0.98	1.31	0.89
0.5	72	-0.83	0.98	1.36	0.88
0.4	76	-0.87	1.01	1.39	0.91
0.3	88	-0.86	1.03	1.48	0.91
0.2	105	-0.86	1.06	1.58	0.92
0.1	134	-0.89	1.10	1.71	0.92
0.05	158	-0.90	1.12	1.79	0.93

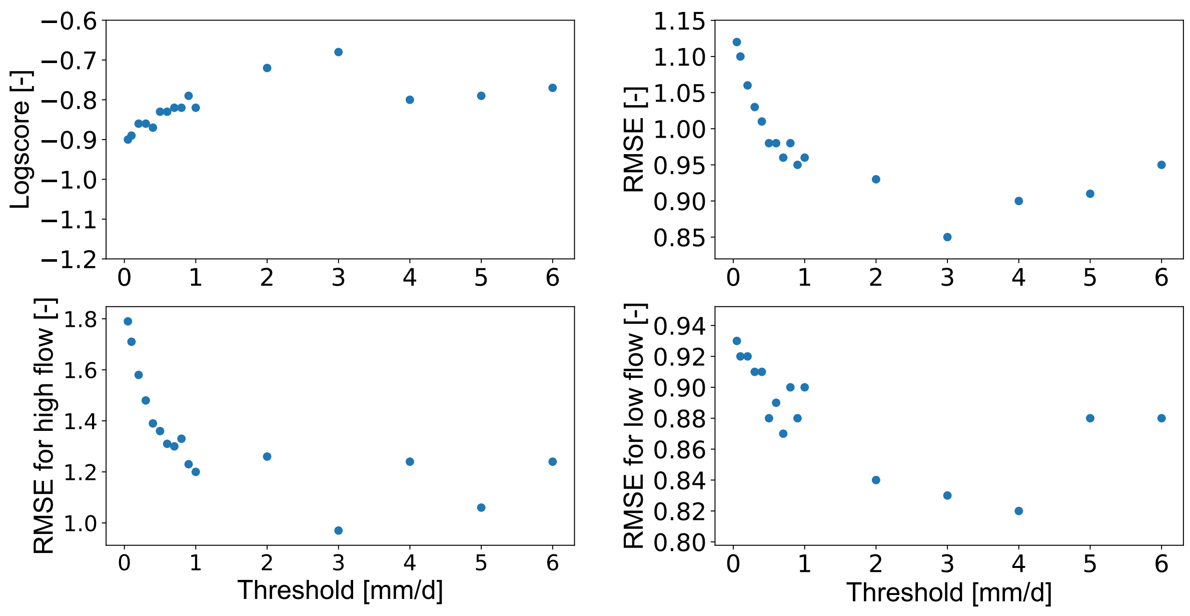


Figure 3.11: relation between thresholds and performance

From the plots in the Figure(3.11), when threshold is smaller than 3 mm/day, larger thresholds that give more storm events will result in relatively better performance in highflow, low flow and entire time series. What is more, the thresholds that is around 2 mm/day give higher value of logscore for entire series and lower value of RMSE (Root Mean Square Error) for highflow, lowflow and entire period. This discovery suggests that it is recommended that when setting the threshold for other basins, thresholds that are going to be used for identifying the storm epochs should be chosen manually. A series of threshold values should be tested so that the optimized threshold for identifying the storm epochs that gives the best performance can be recognized.

- **Relation between storm-dependent parameters with initial water storage and average rainfall during storm**

This subsection will keep on exploring basin A and answer the research question 2. The relation between storm-dependent parameters and rainfall event characteristics and initial conditions are explored. The threshold is set as 3 mm/day which is derived from the former subsection. When proceeding calibration for each storm, the water level in production store at the end of each storm will be exported at the same time to explore its regression relation with X1 values. Regression relations are hoped to be seen between the values of storm-dependent parameters and rainfall event characteristics or initial conditions so that a framework can be set up based on the regression relations and rainfall event characteristics or initial conditions. In this subsection, the regression relations are explored by utilizing the optimized threshold (3 mm/day) and sampling method(sampling X1 and X4 directly from the calibrated results) derived from the subsection before. Initial condition of each storm is plotted against the parameter X1 and X4, respectively. Additionally, the average rainfall of each storm is plotted against the parameter X1 and X4 respectively as well. However,from Figure(3.12), it can be seen that the scatters are very random which means there is no relation between storm-dependent parameter values and initial water storage as well as average rainfall during each storm. Hence the framework can not be set based on these plots to predict the parameter values according to the initial conditions and rainfall characteristics.

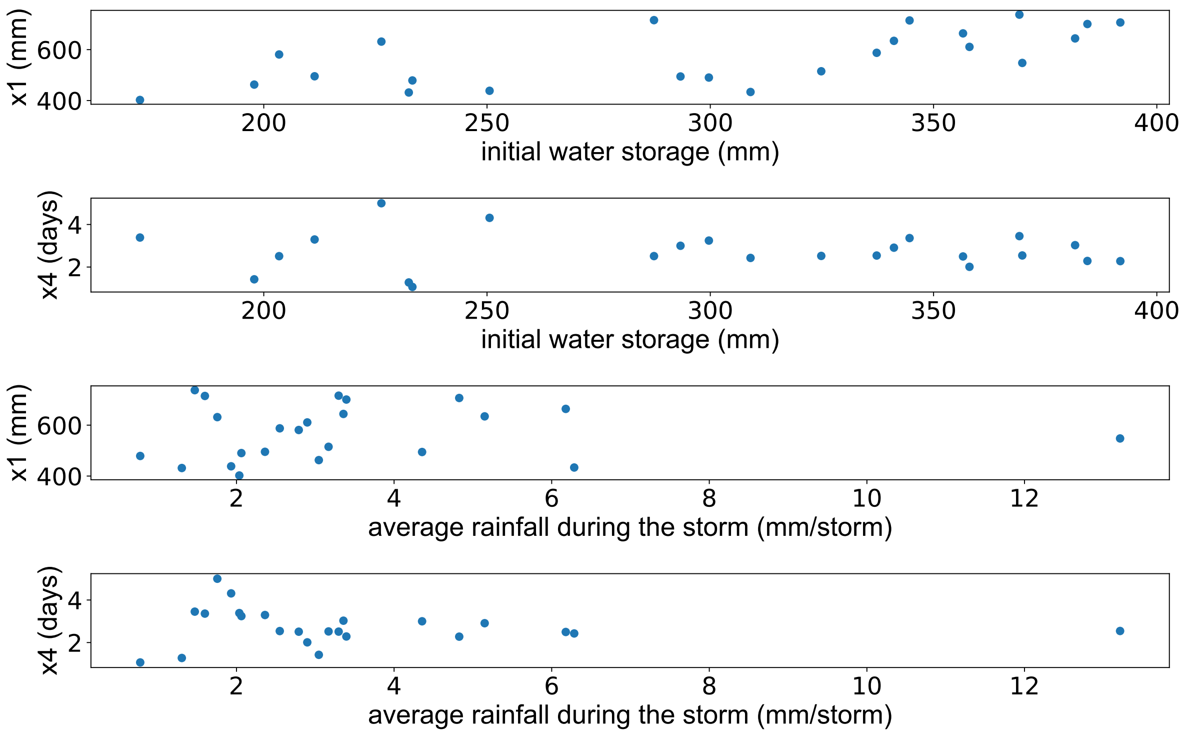


Figure 3.12: relation between storm-dependent parameters with initial water storage and average rainfall during storm (threshold=3mm/d)

3.1.2. Case study of other 5 basins

- **Effect of changing thresholds on improving prediction performance of storm-based method for 6 basins**

In the former part of section 3.1.1, when the effect of the threshold is analysed, we derived the conclusion that the optimized thresholds should be identified manually to obtain a relative better performance. Additionally, directly sampling $X1$ and $X4$ values from the calibrated results and sampling both $X1$ and $X4$ values from their distributions are recommended as relative good ways to sample the storm-dependent parameters for the validation period. The table below lists the assessment results by using two different thresholds. The second thresholds for each basin are defined by the way introduced in the methodology section: It relates with the median value of the entire rainfall series. If the median value is larger than 0.3 mm/day, the threshold will be set as 0.1 mm/day. If the median value is smaller than 0.15 mm/day, the threshold will be set as 0.15 mm/day. If the median value is smaller than 0.3 mm/day and larger than 0.15 mm/day, the threshold will be set as the median value. The first row of thresholds in yellow for each basin are set as its optimized thresholds identified manually. The optimal thresholds are picked by trying a series of thresholds and selecting the threshold that gives the largest logscore value and smallest RMSE value for total flow. Two storm-based method trials with two different thresholds will be done for each basin by directly sampling storm-dependent parameter X1 and X4 values from their calibrated results.

Table 3.3: performance improvement after increasing thresholds

				highflow	lowflow	total flow	
	Dryness index	Threshold	Number of storms	RMSE	RMSE	logscore	RMSE
A	0.87	3	26	0.97	0.83	-0.68	0.85
		0.21	99	1.48	0.91	-0.87	1.03
B	0.85	6	37	2.03	1.35	-1.17	1.50
		0.16	159	2.05	1.72	-2.61	1.79
C	0.64	5	22	1.82	0.63	-0.79	1.00
		0.1	115	2.21	0.80	-1.23	123
D	0.57	4	41	2.89	0.63	-0.93	1.44
		0.25	111	2.83	0.65	-1.03	1.42
E	0.38	1	21	3.61	2.31	-10.06	2.76
		0.1	65	4.15	2.65	-26.03	3.17
F	0.22	3	26	13.46	3.20	-12.28	7.55
		0.19	105	14.47	3.37	-37.31	8.10

It can be seen from the Table(3.3), the model performance for the entire time series is improved with the optimized thresholds for all basins and the performance of lowflow and highflow for almost all basins are also improved. This improvement indicates the threshold for identifying the storm epochs is another important parameter in storm-based method and it is necessary to pick the optimized threshold for identifying the storm epochs before calibration and validation. From this table, we can also observe that the drier the basin is, the better performance of the storm-based method gives. The reason behind this is, for wet basins, parameter X3(one-day maximum capacity of routing reservoir) and X4 are actually the storm-dependent parameters instead X1 and X4 in basin A. This observation will be further explored in next subsection.

- **Additional exploration about the wet basins**

  In the Table(3.3), the storm-based method for six basins are proceeded by making X1 and X4 variable by storms. However, different basins have different dryness characteristics, and this may affect the sensitivity of parameters to storms. For drier basins, the parameter X1 and parameter X4 are identified as storm-sensitive parameters which are reasonable. While for wetter basins, the parameter X3 and parameter X4 should be sensitive parameters because, for wetter basins, the capacity of the routing store is more significant according to the empirical finding and the definition of parameters. This argument corresponds to the sensitivity plots (Figure(3.13)) for wet basin E and F.

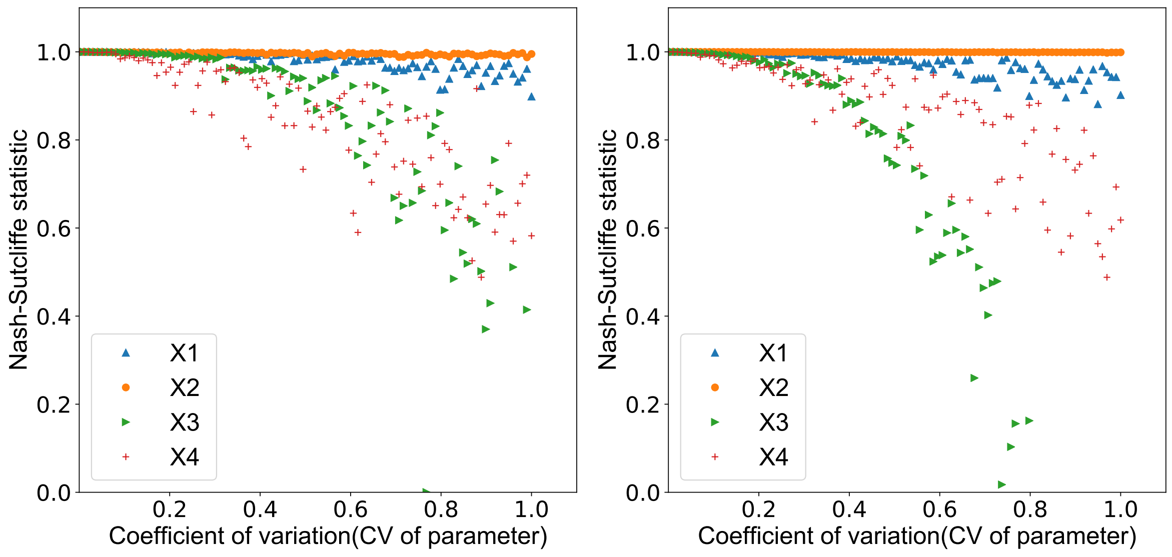


Figure 3.13: sensitivity plot for basin E and F

This means for wetter basins, if still set parameter X1 and X4 as storm-dependent parameters and set parameter X3 constant which is sensitive to storms, worse prediction performance will can be obtained. Hence, the exploration of wet basins by making parameter X3 and X4 variable by storms are proceeded to see whether the performance of the storm-based method for Basin E and F is improved. Here, a wet basin F will be taken as an example to show the performance improvement with the optimized threshold (3 mm/day) after making X3 and X4 variable instead of X1 and X4.

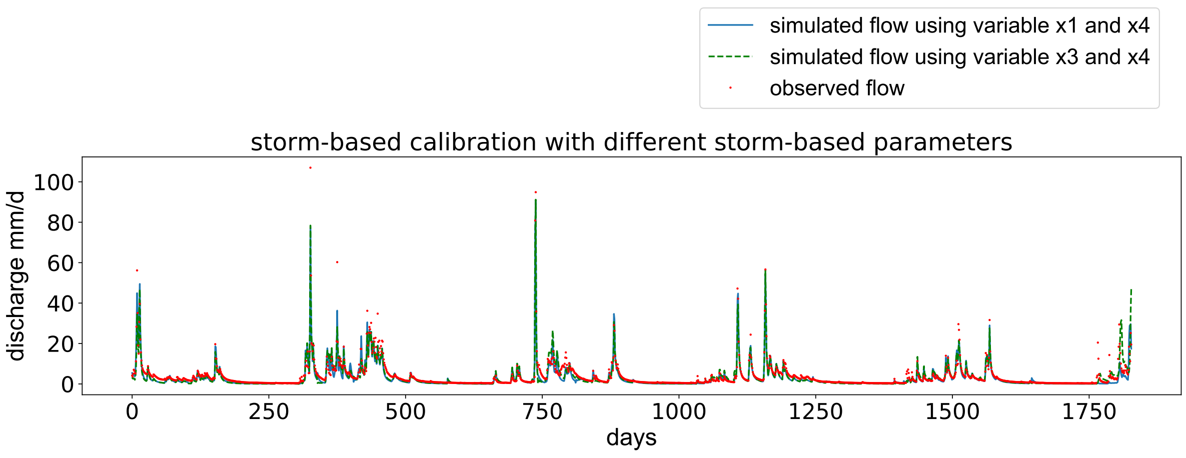


Figure 3.14: comparison between the calibration results by using different storm-dependent parameters for basin F

In the Figure(3.14), the green line represents the calibration result by using X3 and X4 as storm-dependent parameters and the blue line represents the calibration result by using X1 and X4 as storm-dependent

parameters. It can be seen that in the calibration period when X3 and X4 are variable with storms, the simulated flow can also capture the peaks well and performs better especially in the last storm.
In the Figure(3.15), the green band represents the validation result by using X3 and X4 as storm-dependent parameters and the grey band represents the validation result by using X1 and X4 as storm-dependent parameters. It is obvious that the green band(with X3 and X4 variable) covers the observed flow better especially around the date between 0 and 200 as well as the date between 1700 and 1828. And the uncertainty band is much narrower around the date between 300 and 500, 800 and 1000 and 1250 and 1400. Additionally, the logscore value of the green band is higher than that of the grey band, and the RMSE value for the entire flow of the green band is lower than that of the grey band. Furthermore, the RMSE value for the highflow and lowflow of the green band are both lower than those of the grey band. The improvement for the uncertainty band and evaluation indexes means by changing the storm-dependent parameter X1 to X3 improves the performance of the storm-based method for basin F.

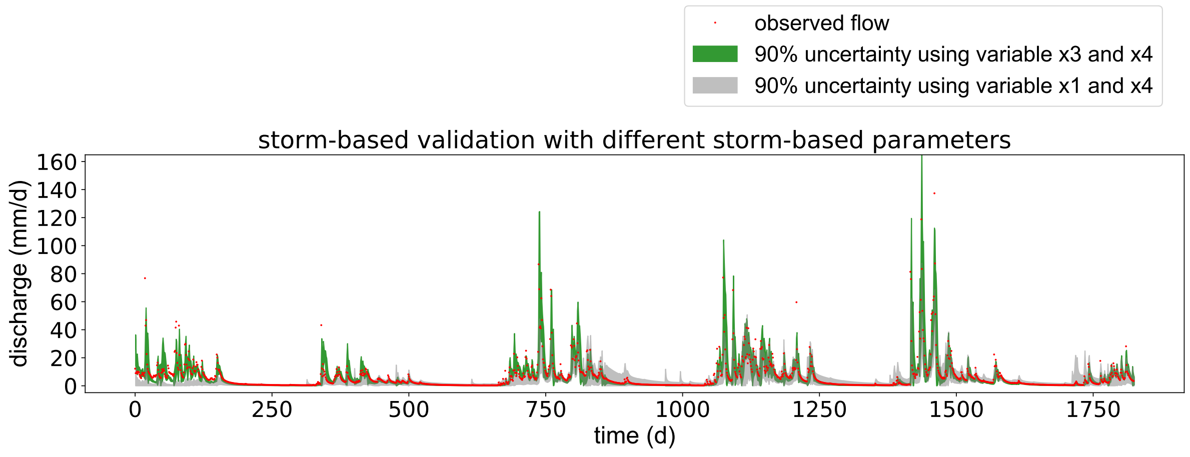


Figure 3.15: comparison between the validation results by making different storm-dependent parameters variable by storms for basin F

The similar improvement is obtained for basin E as well which is shown in the Figure(3.16) and Table(3.4). It is also obvious that the green band(X3 and X4) covers the observed flow better especially around the date between 0 and 200, 300 and 450 and 1700 and 1828.

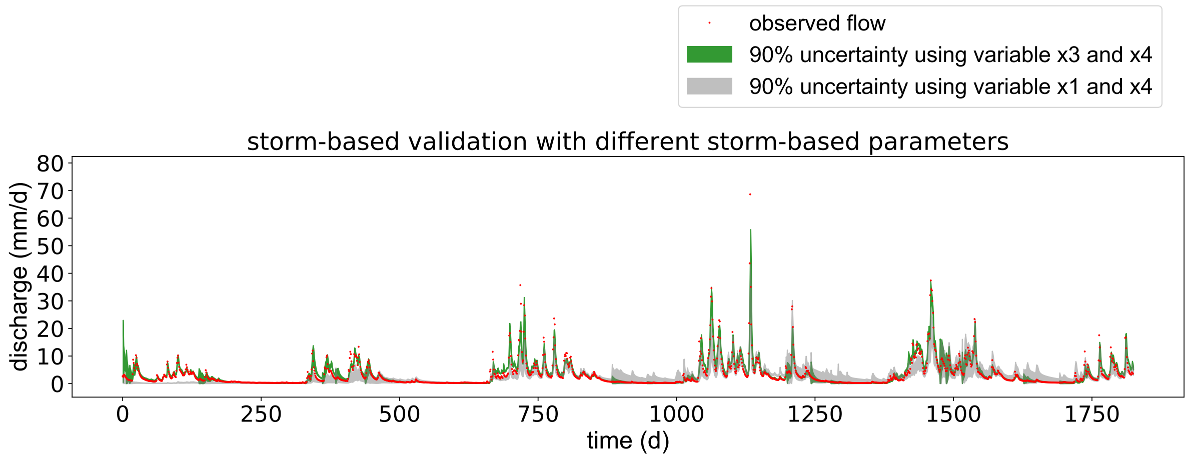


Figure 3.16: comparison between the validation results by making different storm-dependent parameters variable by storms for basin E

Additionally, the logscore value of the green band is also higher than that of the grey band, and the RMSE value for the entire flow, highflow and lowflow of the green band is lower than that of the grey band. This means by changing the storm-dependent parameter X1 to X3 improves the performance of the storm-based method for wet basin E as well. Table(3.4) lists the different assessment results for highflow, lowflow and entire flow obtained by using different storm-depedent parameters. It can be seen from the table, when using X3 and X4 as storm-dependent parameters, the performance improves a lot for highflow, lowflow and entire flow because they have a higher logscore value for entire flow and lower RMSE value foe highflow, lowflow and entire flow. This result indicates that storm-dependent parameters are different for the basins with different dryness characteristics. Besides, choosing parameter X3 and X4 as storm-dependent parameters for wet basins makes more sense according to the sensitivity plot and performance improvement after changing storm-dependent parameters from X1 and X4 to X3 and X4.

Table 3.4: performance improvement after changing the storm-dependent parameters for wet basin E and F

basin	Storm-dependent parameters	highflow	lowflow	Total flow	
		RMSE	RMSE	logscore	RMSE
E	X3 &X4	1.77	1.30	-1.96	1.46
	X1 &X4	3.61	2.31	-10.06	2.76
F	X3 &X4	7.47	1.87	-4.14	4.22
	X1 &X4	13.46	3.20	-12.28	7.55

3.2. Comparison with existing methods

Traditional method and GLUE method are proceeded as alternative methods to explore the how the model performance improves by using storm-dependent parameters. Results obtained by traditional methods will also be presented by taking basin A as an example. Results of traditional method and GLUE method for other basins will be listed in the table in subsection 3.2.3, which will answer the research question 3.

3.2.1. Results obtained by traditional method

In traditional method, model will be run for the entire period without dividing the 5-year calibration period into any periods. NSE is the objective function and the negative NSE is minimized by a global optimization algorithm named differential evolution. Except for the objective function, bounds for the model parameters are set the same as storm-based method and they are utilized as inputs for the differential evolution algorithm.

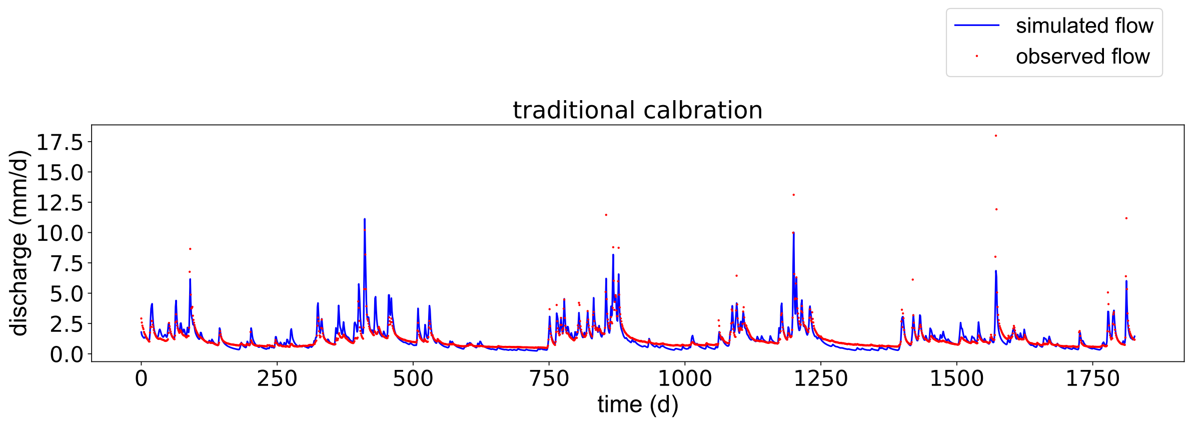


Figure 3.17: calibration result by traditional method for basin A

Figure(3.17) is the traditonal calibration results for basin A and the Table(3.5) lists the values of obtained optimal parameter set $P_o(X1_o, X2_o, X3_o, X4_o)$. The traditional method is actually one of the steps in storm-based method. From the calibrated results, it can be seen that some lowflow regression is not simulated well, and several peaks are not caught well. $P_o(X1_o, X2_o, X3_o, X4_o)$ is also used to predict the discharge during the validation period. Variances between observed values and simulated values are expressed by σ obtained by (Equation(2.22)).

Table 3.5: The obtained optimal parameter set by traditional method

Parameter	description	optimized value P_o
$X1_o$	the production store maximal capacity (mm)	824.36
$X2_o$	the catchment water exchange coefficient (mm/day)	1.24
$X3_o$	the one-day maximal capacity of the routing reservoir (mm)	36.27
$X4_o$	the HU1 unit hydrograph time base (days)	2.57

By taking σ as stand deviations and predicted values Q_{sim} as mean values, 5% (Q_{5sim}) and 95% (Q_{95sim}) quantiles is calculated at each time step. The grey area between Q_{5sim} and Q_{95sim} is the obtained 90% uncertainty band. Average logscore value and RMSE (root mean square error) for the entire time series and the RMSE (root mean square error) for highflow and lowflow are computed respectively at the same time. The uncertainty band missed most peaks and the covered area too wide as it can be seen from the Figure(3.18). For traditional method, parameter $X1$, $X2$, $X3$ and $X4$ in parameter set P_o are constant during the entire calibration and validation period.

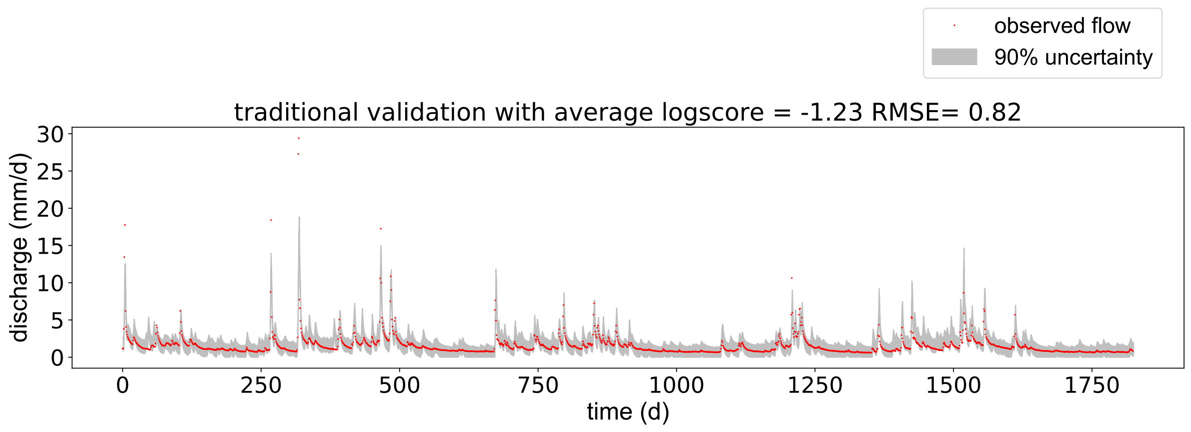


Figure 3.18: validation result by traditional method for basin A

3.2.2. Results obtained by GLUE method

In this method, the bounds for parameters are the same as those in the storm-based method and the traditional method. A Monte Carlo sampling is run for 10^5 times to generate 10^5 parameter sets randomly between the parameter bounds. When separating behavioural models from non-behavioural models with these parameter sets, the standard to define the behavioural models is the value of the objective function should be larger than 0.6. Figure(3.19) presents the calibration result obtained by the parameter set that has the largest value of NSE (which is the same in traditional method). From this figure, it can be seen that lots of peak vales are missed, and some regression parts are not simulated well like in the storm-based method.

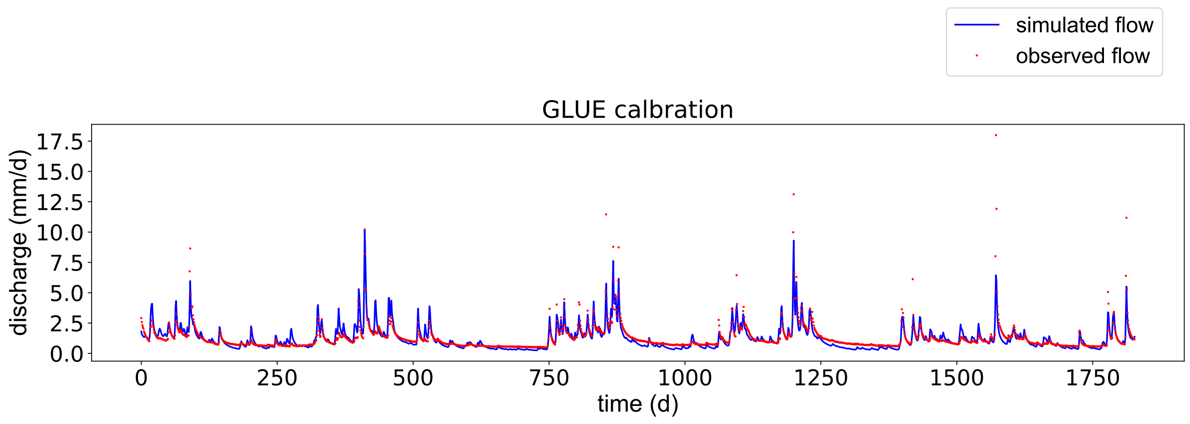


Figure 3.19: calibration result by GLUE method for basin A

Behaved parameter sets $P_i(X1_i, X2_i, X3_i, X4_i)$(i=1,2,...n) (with objective function that is larger than 0.6) retained from calibration period are used to predict the runoff during validation period. 199 behaved models are retained which means 199 behaved parameter sets are obtained from the calibration period and will be applied for streamflow prediction in validation period.
Theses behaved parameter sets are used to compute 199 time series of predicted streamflow $Q_{m(ij)}$ (i=j=1,2,...199, j=1,2,...last day of examined period). This means, there will be 199 predicted discharge at each time step after running each parameter set P_i. At each time step, mean predicted value and standard deviation of predicted values will be calculated so that 5% ($Q_{5sim(i)}$) and 95% ($Q_{95sim(i)}$) quantiles can be calculated for each time step. The area between Q_{5sim} series and Q_{95sim} series is the obtained 90% uncertainty band. Average logscore value and RMSE (root mean square error) for the entire time series and RMSE (the root mean square error) for highflow and lowflow are computed respectively at the same time. Figure(3.20) is the validation result obtained by GLUE method, it can be seen that although it gives a high logscore value and low RMSE, the uncertainty band totally missed some observed flows around the date between 50 and 1450.

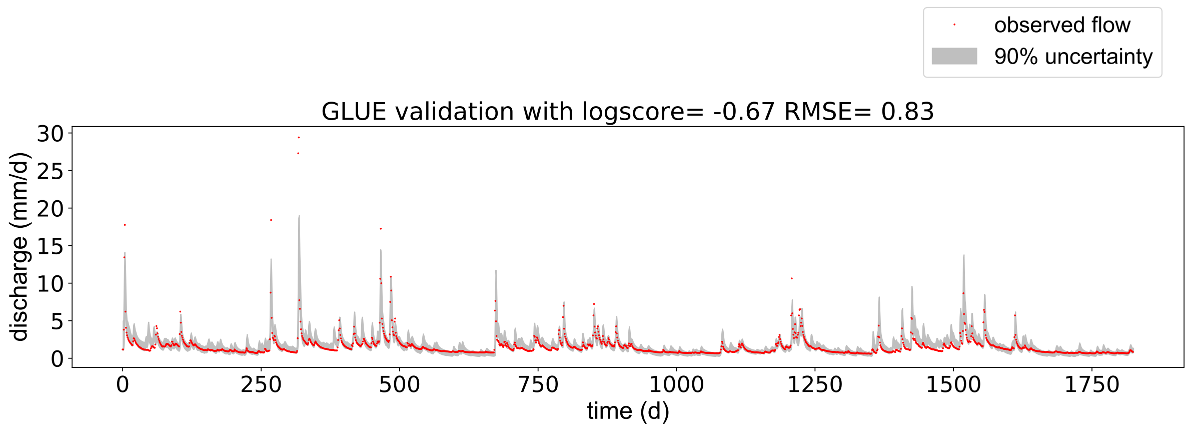


Figure 3.20: validation result by GLUE method for basin A

3.2.3. Assessment of prediction performance for six basins derived by 3 different methods

This section will answer the research question 3. Firstly, the function in the model that generates the runoff will be extracted out to illustrate the difference between the storm-based method and alternative methods(traditional method and GLUE method). Then the table that lists the evaluation results obtained by three methods for six basins will compare the performance of three methods. The effect of dryness on the storm-based method performance will also be discussed in this section.

- **Different responses by using fixed and storm-dependent parameters**
  The function Equation(3.2) represents the runoff and storage relation in the model. In Figure (3.21), the

left figure represents the difference between modelled response and actual response in the real world. In the model, the spatial and temporal heterogeneity is ignored; hence $X1$ is constant, which gives a single relation between runoff and water storage. However, when $X1$ is variable, the multi-valued relation will be found between runoff and water storage. This difference is proved by exploring the basin A with variable $X1$ and constant X1. As shown in the left picture in the Figure (3.21), the red line represents the single value relation between runoff and storage when $X1$ is fixed. However, when storm-dependent parameter $X1$ is variable, it gives multi-valued relations (grey lines) between runoff and storage. In the traditional and GLUE method, parameters are constant so the runoff- storage relation is a single value relation. However, parameter X1 is variable by storms, so the runoff-storage relation is a multi-valued relation which agrees with the reality more. Figure (3.21) suggests storm-dependent parameter fits the reality better, which means using storm-dependent parameters are reasonable and necessary.

$$runoff = P_s - P_n = \frac{R1\left(1-\left(\frac{S}{X1}\right)^2\right)\tan\left(\frac{Pn}{X1}\right)}{1+\frac{S}{X1}\cdot\tanh\left(\frac{Pn}{X1}\right)} \tag{3.2}$$

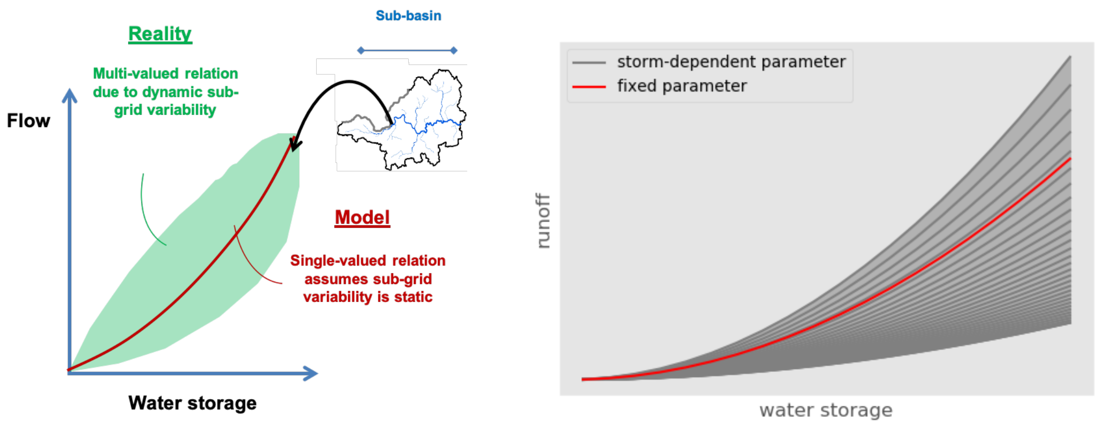


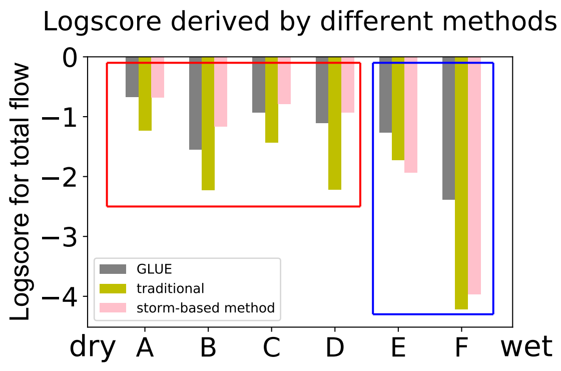
Figure 3.21: runoff versus water storage

- **Evaluation of the performance obtained by three methods**

Performance of highflow and lowflow and general performance obtained by different methods are listed in Table(3.6).

Table 3.6: different performance obtained by three methods for 7 basins

	Traditional method				GLUE method				Storm-based method			
	highfow	lowflow	total flow		highfow	lowflow	total flow		highfow	lowflow	total flow	
	RMSE	RMSE	logscore	RMSE	RMSE	RMSE	logscore	RMSE	RMSE	RMSE	logscore	RMSE
A	0.94	0.80	-1.23	0.82	1.08	0.77	-0.67	0.83	0.97	0.83	-0.68	0.85
B	1.96	1.34	-2.23	1.47	1.80	1.36	-1.55	1.45	2.04	1.35	-1.17	1.50
C	1.77	0.64	-1.43	0.98	2.12	0.66	-0.93	1.13	1.82	0.63	-0.79	1.00
D	3.02	0.58	-2.22	1.48	3.53	0.58	-1.11	1.70	2.89	0.63	-0.93	1.44
E	1.61	1.19	-1.73	1.32	2.08	1.45	-1.27	1.66	1.77	1.30	-1.96	1.46
F	6.85	2.32	-4.22	4.09	6.91	1.59	-2.39	3.87	7.47	1.87	-4.14	4.22



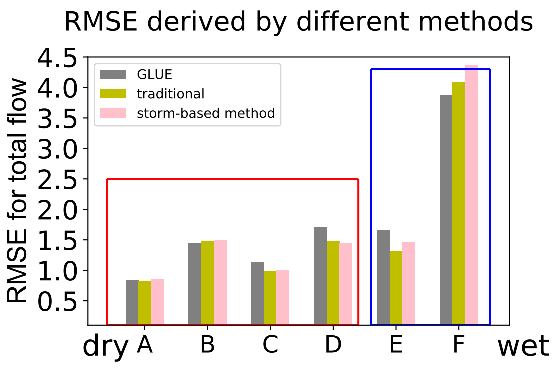


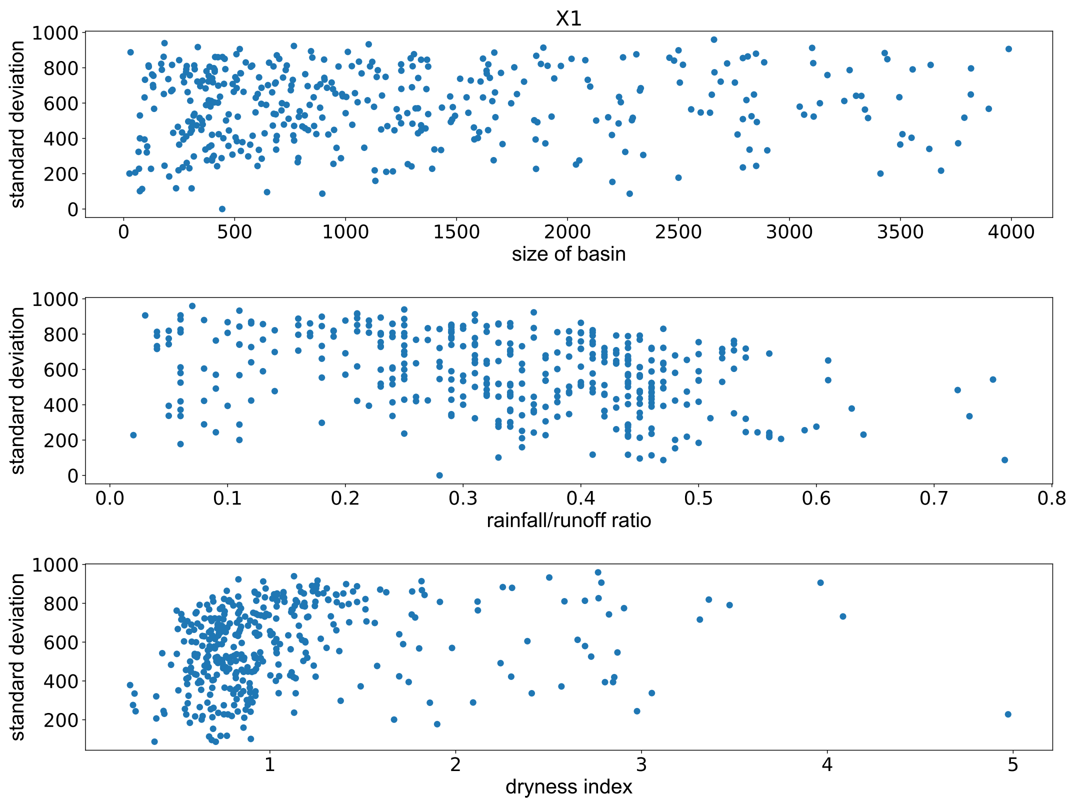
Figure 3.22: histograms of logscore and RMSE value obtained by 3 different methods for six basins

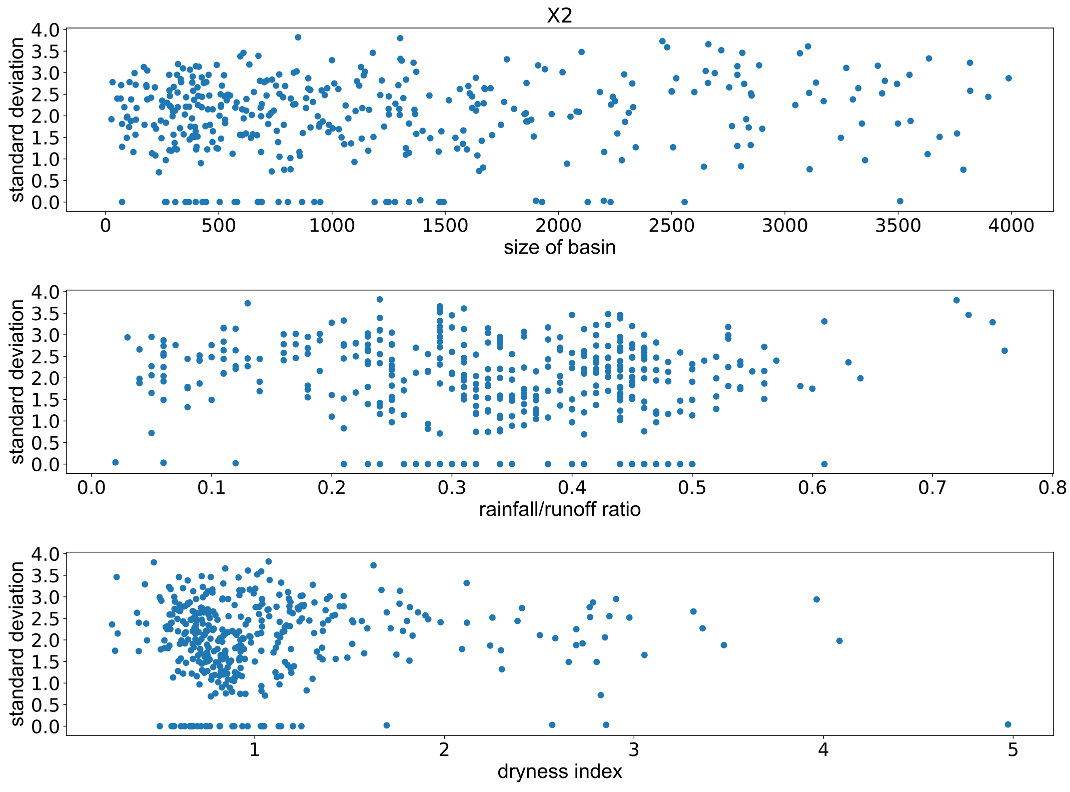
Two histograms following the table illustrate the difference more obvious. The storm-based method gives lower logscore vale and higher RMSE (Root Mean Square Error) value when the basin is wetter (e.g. basin E and F) and gives higher logscore vale and lower Root Mean Square Error value when the basin is drier (e.g. basin A, B, C and D). This means the storm-based method performs better for drier basins while worse for wetter basins. GLUE method gives better performance for wetter basins than storm-based method and traditional method gives, which means GLUE method is more applicable for wetter basins than the storm-based method and traditional method. Dry basins are usually hard to predict while storm-based method gives relative better performance than other two methods give. This section answered the research question 3: To what extent does accounting for variation in model parameters capture model errors and improve rainfall-runoff prediction? Storm-based method captured the model error and improved the rainfall-runoff prediction for drier basins.

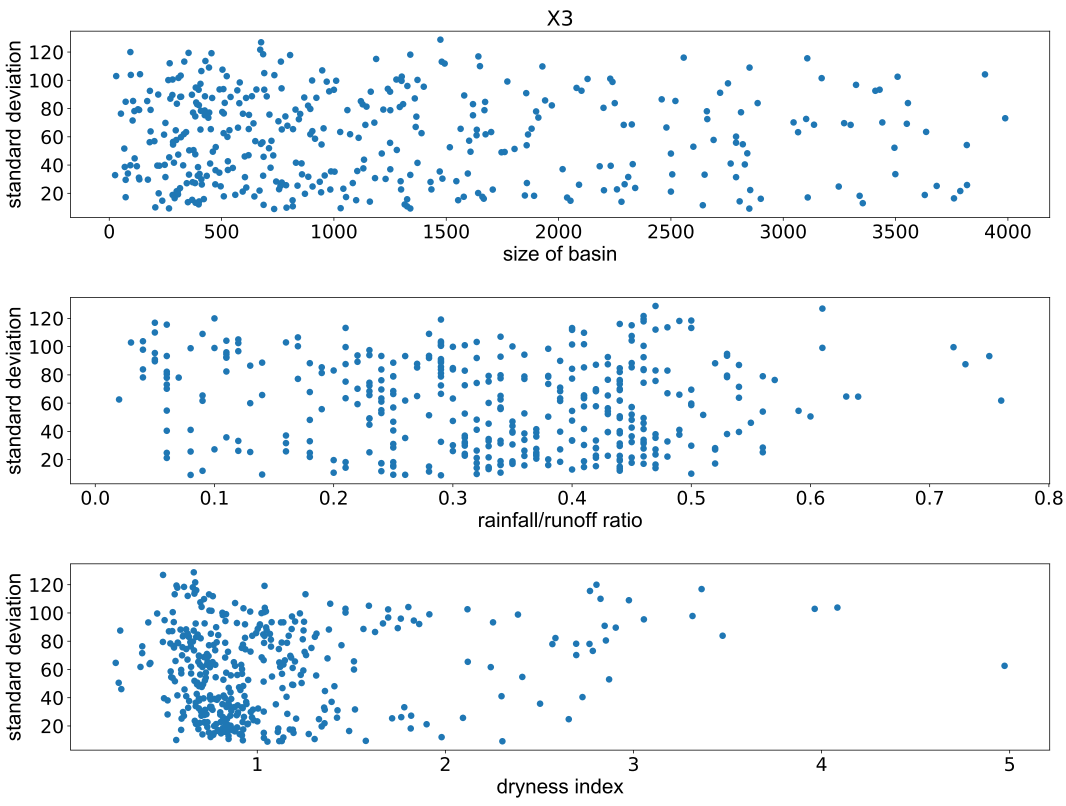
3.3. Results for 392 basins

3.3.1. Extent of variation and relation between the extent of variation and basin characteristics

For this part, the number of basins is extended to 392 to explore the extent of variation and relation between the extent of variation of storm-dependent parameters and basin characteristics in order to answer the research question 4. For each basin, they are calibrated by making one of the four parameters varying by storms respectively and the other three parameters constant. In this section, each parameter has one chance to be the storm-dependent parameter for one basin. The reason behind this is it is not clear that which parameters should be the storm-sensitive parameters. Doing a sensitivity analysis for each basin and selecting the storm-dependent parameters manually for each basin is too time-consuming, which is unrealistic. Hence, each parameter will be considered as the storm-dependent parameter, and the other three parameters are set constant for one time. Storms are identified by the threshold relative with their median value of the rainfall series like introduced in the methodology part. The standard deviation of the calibrated parameter from each basin is calculated. And the standard deviation of each basin is plotted against their size, rainfall/ runoff ratio and dryness index. Additionally, if the number of storms for one basin is under 20, it means there are not enough sample size of calibrated storm-dependent parameters. Hence the calculated result for this basin can be discarded for its result is not reliable. In the Figure(3.23), each scatter represent a basin (with the number of storms larger than 20) with different size, dryness index, or rainfall/runoff ratio. From the scatters, it can be seen that the scatters are also very random here, which means the extent of variation does not depend on basin characteristics like dryness or sizes. In the plots below, the scatters are very random, and no patterns can be found, which means the extent of parameters variation does not depend on the basin size and basin dryness.


X1
standard deviation
size of basin
rainfall/runoff ratio
dryness index


X2
standard deviation
size of basin
rainfall/runoff ratio
dryness index



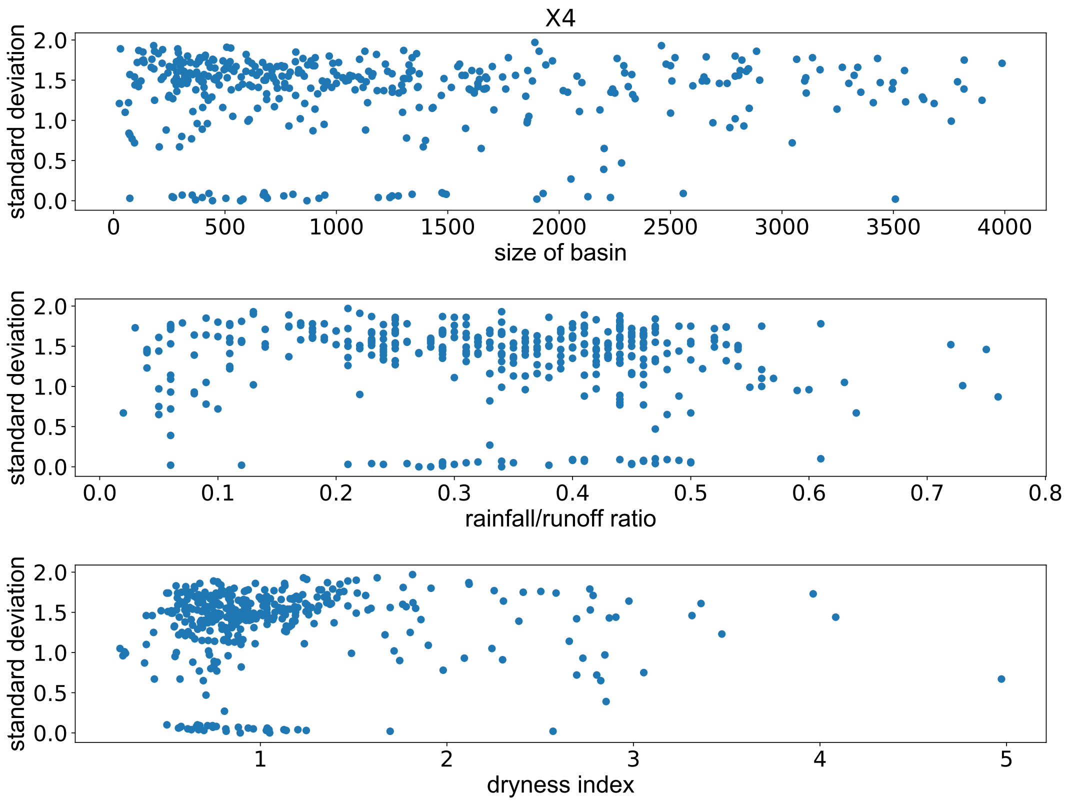


Figure 3.23: standard deviation of x3 versus dryness index and sizes basins

For six case basins, they all have parameter X4 as storm-dependent parameters. In average, basin A has a standard deviation of 0.89 for X4; basin B has a standard deviation of 1.59 for X4; basin C has a standard deviation of 1.47 for X4; basin D has a standard deviation of 1.13 for X4; basin E has a standard deviation of 2.63 for X4; basin F has a standard deviation of 3.46 for X4. With the dryness index decreases, the standard deviation increases, which means the wetter basin (i.e. basin F) has a larger standard deviation of storm-dependent parameters. The time series(Figure(3.24), (3.25), (3.26), (3.27), (3.28), (3.29) of the storm-dependent parameter X4 in six basins are present below to show how storm-dependent parameter X4 is varying with time within each basin.

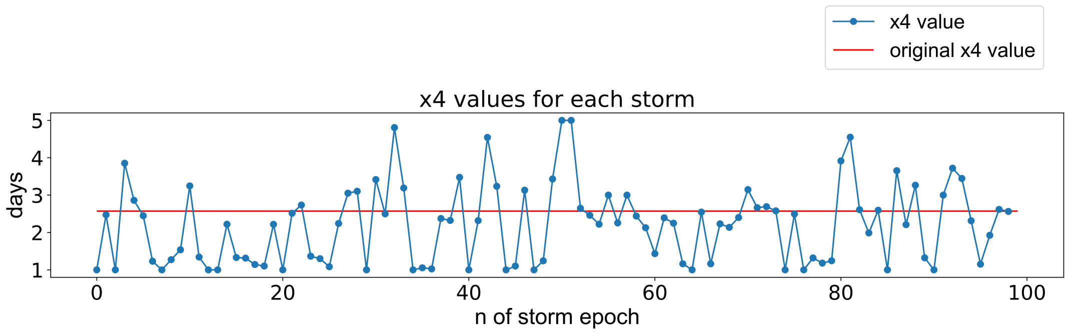


Figure 3.24: time series of storm-dependent parameter X4 for basin A

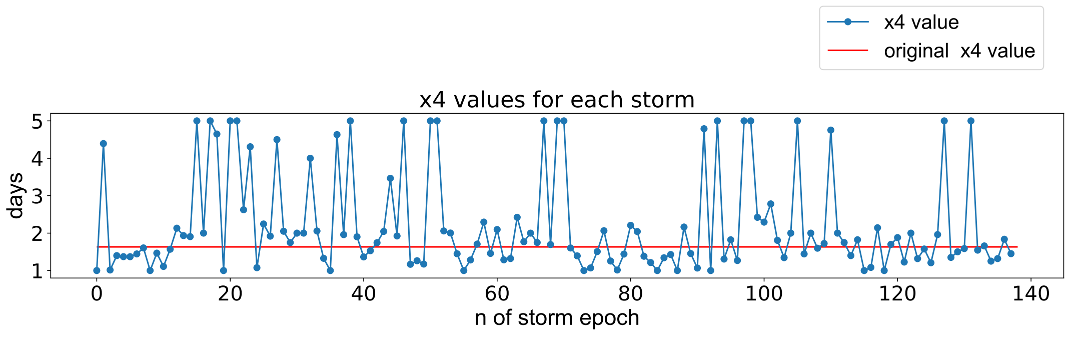


Figure 3.25: time series of storm-dependent parameter X4 for basin B

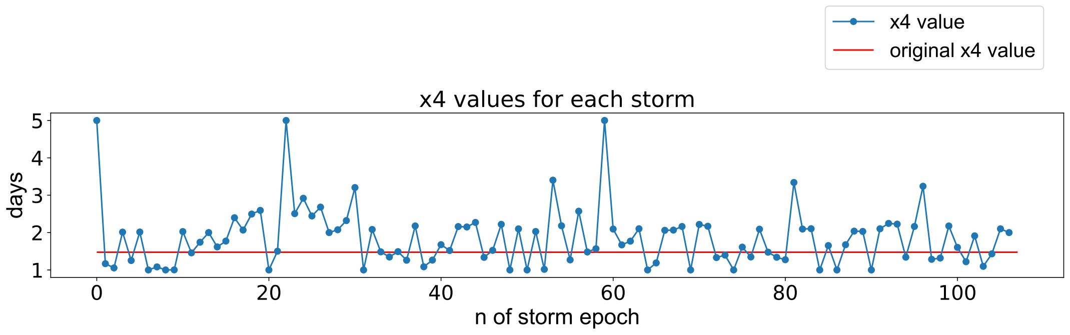


Figure 3.26: time series of storm-dependent parameter X4 for basin C

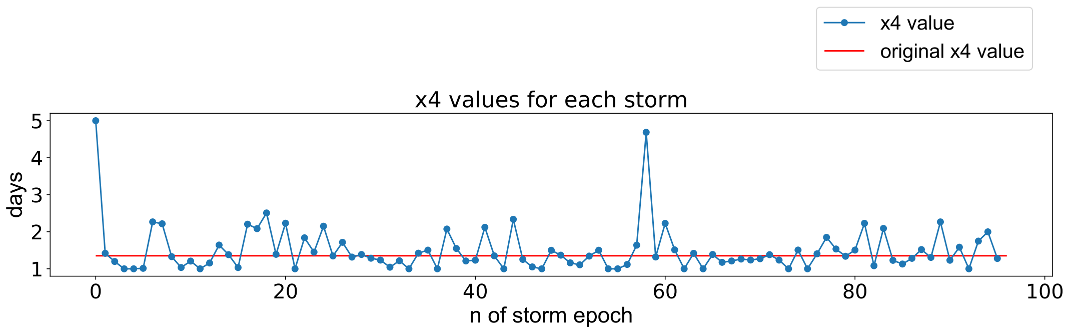


Figure 3.27: time series of storm-dependent parameter X4 for basin D

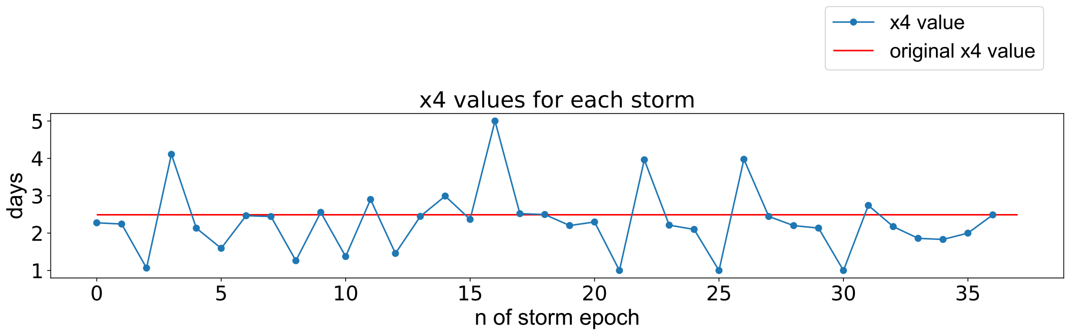


Figure 3.28: time series of storm-dependent parameter X4 for basin E

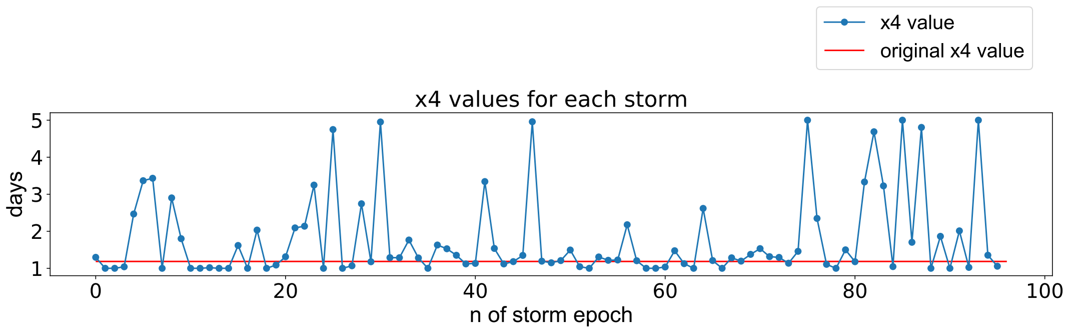


Figure 3.29: time series of storm-dependent parameter X4 for basin F

4
Discussion

- **The effect of thresholds for identifying the storm epochs in the storm-based method**

  It was found that the thresholds for identifying the storm epochs have a considerable effect on the prediction performance. The threshold to identify the first rainfall in a storm determines how many storms will be obtained during the analysed time series. In principle, the threshold that gives more storms will give better prediction performance because the calibration and validation are more precise, given a shorter period. However, it turns out the prediction performances are relatively better when the thresholds is optimized manually. The effect of the threshold is only explored in six case basins and all the prediction performance are all improved with an optimized threshold. Hence in a subsequent study, the threshold for identifying the storm epochs should be considered as an essential parameter in the storm-based method.

- **The effect of basin dryness on identifying the storm-sensitive parameters**

  It is found that the storm-sensitive parameters are different for the basins with different dryness index. For the wet basins, parameter X3(one-day maximum capacity of the routing reservoir) and X4 (routing days) are more sensitive to the storms instead of X1(maximum capacity of the production store) and X4. The reason for this could be in the dry basins, the water storage is low and dry basins are heterogeneous, so the parameter X1 which controls runoff generation is more sensitive compared to the parameter X3. While in the wet basins, water storage in the upper layer is stable and wet basins are more homogeneous, so the capacity of the routing reservoir affects more on the discharge. In future studies, the effect of dryness on identifying the storms-dependent parameters should be researched based on a larger number of basins.

- **No patterns in the plots of storm-dependent parameter values versus rainfall characteristics and initial condition**

  When exploring the relations between the storm-dependent parameters and rainfall characteristics and initial conditions, a regression relation was hoped to be identified so that the value of the storm-dependent parameters can be predicted from the rainfall characteristics or initial conditions. However, there are no patterns found in the plots which means rainfall characteristics or initial conditions can not be utilized to set up a framework to predict the values of storm-dependent parameters. Even though the parameters cannot be predicted deterministically, they can be predicted probabilistically, using the histogram or fitted distribution for the calibrated parameter sets. The parameters sampled from the histogram or fitted distribution are then can be used for predicting discharge in validation period. Additionally, in the future, other factors can be explored to identify whether there is a relation between the variation of storm-based parameters and these factors.

- **Assumptions**

  In this research, several assumptions were made in storm-based method: because of spatial and temporal averaging, the flux errors primarily affect model performance. Besides, these errors can be described by sampling one or two parameters from a probability distribution at the beginning of each storm. These two assumptions are proved to be reasonable because the storm-based methods which based on these two assumptions show a better performance than traditional method and GLUE method. Based on these two assumption, the entire time series are divided into storms in the storm-based method to capture the model error. Besides, after finishing the case study of Basin A, the choosing of storm-dependent parameters is assumed to be same for other basins. This assumption leads to poor performance for some basins like basin E and basin F. This unreasonable assumption indicates the importance of doing sensitivity analysis for each basin individually.
  However, the assumptions made in GLUE method differ from those in storm-based method. The GLUE method assumes that all parameter sets have an equal likelihood of being acceptable. All the accepted parameters are retained in the GLUE method. The assumption in this method does not consider the input data error and capture the structure error like that in the storm-based method. The reason is that although multiple parameter sets are accepted in the GLUE method, all the parameters are still constant in time.

- **Comparing with other literature**

  There is no literature on the difference between the prediction performance obtained by storm-based method, traditional method and GLUE method. In this research, results are compared obtained by three different methods to explore whether the storm-based method gives better prediction performance considering the model error. Storm-based method has better performance than other two methods for drier basins. The reason of good performance of storm-based method in dry basins may be drier basins are more heterogeneous so that they will lead to more model errors while storm-dependent parameters accounts for these errors better than other two methods. In future work, more comparison and explorations should be done to research why storm-based methods perform better than GLUE method in drier basins while perform worse than GLUE method in wet basins.

- **Uncertainty in input data**

  In this research, storm-dependent parameters are used to capture the structure error while there are some other uncertainties in the model, e.g. input error. Input error is due to the measurement error and averaging in temporal and spatial scale. Using storm-dependent parameters can compensate for the error from the temporally and spatially averaging while the measurement error can not be corrected. Hence in the subsequent researches, a robust framework needs to be set up to account for the data uncertainty.

- **Distinguish the error captured in the research**

  The variation of storm-dependent parameters can be found during storm events, which means there are errors in the model. However, the error can be structure and input error. An oversimplification of the hydrological process in the real world can lead to this error. Additionally, the measurement error of data could also lead to the variation of storm-dependent parameters because the input data is not correct. In future work, a framework can be set up to distinguish which error it is and deal with these errors, respectively.

- **Future work about the relation between variation of storm-dependent parameters and basin size and characteristics**

  According to the results in section 3.3, the scatters in the plots of sigma of storm-dependent parameters against basin dryness and sizes and rainfall/runoff ratio are very random. However, the variation and mean of the mean and standard deviation of storm-dependent parameters from 392 basins can be derived so that distributions can be fitted to the mean and sigma of storm-dependent parameters from 392 basins. In future research, the distributions for the extent of variation of storm-dependent parameters can be explored. The extent of variation of parameters can not be predicted deterministically, but it can be described probabilistically. When predict the discharge for a new basin, e.g. an ungauged basin, the extent of variation of parameters can be picked from the distributions for extent of variation of parameters from observed basins.

- **Future work about variation of storm-dependent parameters in the spatial pattern**

  In the storm-based method of this research, the model errors are influenced by the flux errors arising from spatial and temporal averaging. These errors are assumed can be captured by making storm-dependent parameters vary with time. However, in future work, the errors arising from the spatial averaging can be described by dividing the catchment into different parts and randomly sampling each parameter for the subbasins. Besides, the variation of parameters in spatial patterns could play a role in finding the relation between storm-dependent parameters and rainfall characteristics and initial condition.

- **Future work about using storm type or categories**

  The definition of storm epoch in the storm-based method of this research is inter-storm dry spells of one or more days, followed by a day with rainfall exceeding a specific threshold. The threshold is determined by finding an optimum threshold that gives the best model performance. At the same time, it is also recommended to use different storm type or categories to divide the storm epochs. For example, a small storm epoch can be defined by an inter-storm dry spell of one or more days, followed by one or more days of rainfall that have an average rainfall less than 2mm/day. A median storm epoch can be defined by an inter-storm dry spell of one or more days, followed by one or more days of rainfall that have an average rainfall less than 10 mm/day and larger than 2 mm/day. A large storm epoch can be defined by an inter-storm dry spell of one or more days, followed by one or more days of rainfall that have an average rainfall that is larger than 10 mm/day. Different types of storms will also influence the model calibration. In the subsequent research, the analysed time series can be divided into different types of storm epochs to develop the storm-based method better.

- **Future work about the selection of case basin**

  In this research, six representative basins with different basin size and dryness index are selected as case basins. In future work, it is recommended to use a systematic approach to select basins, e.g. three categories for dryness and two for size. A basin with the streamflow/rainfall ratio less than 0.2 can be defined as dry basins and larger than 0.5 can be identified as wet basins. Besides, a basin with the size smaller than 700 km^2 can be identified as a small basin and larger than 1500 km^2 is a large basin. Other basins with the size larger than 700 km^2 and smaller than 1500 km^2 can be identified as a median basin. The relation between basin size and dryness and the extent of variation of storm-dependent parameters is likely to be better explored.

- **The implication of the finding that GLUE performs better than storm based for wet basins**

  For wet basins, GLUE method performs better than the storm-based method indicates that the storm-based method needs to be improved to better applicable for wetter basins. Several approaches can be investigated to develop the storm-based method. Firstly, the definition of storm epoch. It can be noticed that the numbers of identified storm epochs for wet basin E and F are smaller than those identified for dry basins. Secondly, a small number of calibrated parameter sets can not give very reasonable distributions. Hence, a different definition of storm epoch can be utilised to divide the storms, e.g. the storm type or categories. Thirdly, the effect of spatial heterogeneity can be analysed in the wet basins. Besides, the change in land cover can also be another reason. Due to the urbanism, some land cover like forest and grass can be changed into a residential area for ten years.

5 Conclusions

The goal of this research is to further investigate and develop the storm-based approach, and compare it to traditional approaches using static (time-invariant) parameters and GLUE (method) to capture model errors. In this research, 392 basins from the MOPEX data set are applied for research after checking the long-time water balance. Rainfall-runoff model GR4J is applied to simulate the discharge with the input data from the MOPEX data set. Firstly, steps of storm-based methods, as well as alternative methods(traditional method and GLUE method) and the differences between these methods, are introduced.
Six typical basins with different dryness index and rainfall-runoff ratio are researched as case basins to answer the first three research questions. The first two questions(to what extent do model parameters vary by storm/event, to what extent is this variation random and can it be predicted from rainfall event characteristics and initial conditions) are answered by using storm-based methods. Then storm-based methods proceed as the steps introduced before. Basin A is researched as an example to find an applicable standard of setting threshold for storm epochs identifying and sampling methods of picking parameter sets for other basins. After the calibration in the storm-based method, storm-dependent parameters showed variations by storms. Besides, the initial water storage and average rainfall during each storm are plotted against the storm-dependent parameters X1 and X4(time peak ordinate of unit hydrograph UH1) to explore how randomness this variation is and the relation between the variation and rainfall characteristics or initial conditions. However, the scatter plots are very random, and no pattern can be found in the plots. Hence, although the calibrated storm-dependent parameters show variations by storms, the random of the variations have no relation with the initial conditions and rainfall characteristics. Before validation, the two sampling methods are used to sample the parameter sets. One method is directly sampling the storm-dependent parameters from the calibrated results, and another one is sampling the storm-dependent parameters from the fitted distributions. The validation results obtained by two sampling methods are compared to derive the relative better sampling methods that can be applied for other basins. According to the comparison, directly sampling from the calibrated results and sampling both X1(maximum capacity of the production store) and X4 from the fitted distributions are considered as good sampling methods. After determining the sampling methods in the validation period of the storm-based method, the effects of the threshold for identifying the storm epochs on the performance are explored. According to the evaluations for total flow, highflow and lowflow of 6 basins by trying different threshold values, it can be derived that the thresholds that can identify 20 to 60 storms give relative good validation results and the optimized threshold should be chosen manually. The calibration and validation processes proceed again for the basin A, B, C, D, E and F using the picked optimized thresholds. It can be seen that the performances for six basins are all improved after using the optimized thresholds picked manually for they all have a higher logscore value for total flow and lower RMSE for total flow, highflow and lowflow. This exploration indicates the set of the threshold for identifying the storm epochs matters a lot, and it can be seen as another parameter of the storm-based method.

Except for the storm-based method, two other methods are also applied to do the model calibration and validation. Comparisons are proceeded between the different results obtained by three methods so that the research question 3 (to what extent does accounting for variation in model parameters capture model errors and improve rainfall-runoff prediction) can be answered. By making storm-sensitive parameters varying by storms in storm-based method, the model performance for total flow, lowflow and highflow are all improved compared with that of traditional method with constant parameters during the analysed time series. Additionally, although the total flow, highflow and lowflow performances of the GLUE method is better than that of storm based method, the GLUE method can miss some lowflow regression part and can not capture lots of peak flows. GLUE method also assumes the parameters are constant in time like the traditional method so that it can not capture the structure error compared with the storm-based method. What's more, we found that if the basin is wetter, the storm-based method performance will be poorer than GLUE method. Therefore, another exploration of the wet basins are proceeded. A new sensitivity analysis is done for basin F, and it turns out the parameter X3(the maximum capacity of the routing reservoir) and X4 are storm-sensitive parameters instead of X1 and X4 that identified in basin A. Then new calibration and validation are performed for wet basins E and F with X3 and X4 as storm-dependent parameters variable by storms. And we found the performance for wet basins are improved especially for the RMSE of total flow, highflow and lowflow.

The last research question (does the extent of variation of storm-dependent parameters depend on basin characteristics (e.g. dry vs humid, small vs large))is answered by exploring the 392 basins with each basin calibrated by making one of the parameters variable by storms each time. From the results, it can be derived that there is no relation between the extent of variation and basin characteristics.

Main conclusions obtained for each research questions are listed below:

1 To what extent do model parameters vary by storm/event?

   The scatter plots of storm-dependent parameters showed the variations by storms. These variations indicate there is the structure error or input error in the model.

2 To what extent is this variation random and can it be predicted from rainfall event characteristics and initial conditions?

   The scatters in the plots of storm-dependent parameters values against rainfall characteristics and initial conditions are very random which means the extent of variations of storm-dependent parameters can not be predicted from rainfall event characteristics and initial conditions. However, even though the parameters cannot be predicted deterministically, they can be predicted probabilistically with the histogram or fitted distribution for the calibrated parameter sets.

3 To what extent does accounting for variation in model parameters capture model errors and improve rainfall-runoff prediction?

   Storm-based method performs better for drier basins while worse for wetter basins compared to GLUE method and traditional method. The logscore value obtained in storm-based method(e.g. -0.68 for basin A, -1.17 for basin B, -0.79 for basin C, -0.93 for basin D) is larger than those obtained in the traditional method (e.g. -1.23 for basin A, -2.23 for basin B, -1.43 for basin C, -2.22 for basin D) and GLUE method (e.g. -0.67 for basin A, -1.55 for basin B, -0.93 for basin C, -1.11 for basin D) respectively for each basin. Additionally, the RMSE values for total flow obtained in the storm-based method are all smaller than those obtained in the traditional method and GLUE method for these dry basins. GLUE method gives better performance for wetter basins than storm-based method and traditional method gives, which means GLUE method is more applicable for wetter basins than the storm-based method and traditional method. Dry basins are usually hard to predict while storm-based method gives relative better performance than other two methods give.

4 Does the extent of variation of storm-dependent parameters depend on basin characteristics (e.g. dry vs humid, small vs large)?

   The extent of variation of storm-dependent parameters has no relation with basin characteristics but the mean and variation of the storm-dependent parameters can be obtained. Hence the extent of variation of parameters can be described probabilistically.

Bibliography

[1] K. C. Abbaspour, J. Yang, I. Maximov, R. Siber, K. Bogner, J. Mieleitner, J. Zobrist, and R. Srinivasan. Modelling hydrology and water quality in the pre-alpine/alpine thur watershed using swat. *Journal of hydrology*, 333(2-4):413–430, 2007.

[2] K. C. Abbaspour et al. Swat-cup4: Swat calibration and uncertainty programs–a user manual. *Swiss Federal Institute of Aquatic Science and Technology, Eawag*, 106, 2011.

[3] G. Aronica, P. Bates, and M. Horritt. Assessing the uncertainty in distributed model predictions using observed binary pattern information within glue. *Hydrological processes*, 16(10):2001–2016, 2002.

[4] K. Beven. On undermining the science? *Hydrological Processes: An International Journal*, 20(14):3141–3146, 2006.

[5] K. Beven and A. Binley. The future of distributed models: model calibration and uncertainty prediction. *Hydrological processes*, 6(3):279–298, 1992.

[6] K. Beven, P. Smith, and J. Freer. Comment on "hydrological forecasting uncertainty assessment: Incoherence of the glue methodology" by pietro mantovan and ezio todini. *Journal of Hydrology*, 338(3-4): 315–318, 2007.

[7] S. Blazkova and K. Beven. Flood frequency estimation by continuous simulation for a catchment treated as ungauged (with uncertainty). *Water Resources Research*, 38(8):14–1, 2002.

[8] M. Bouda, A. N. Rousseau, B. Konan, P. Gagnon, and S. J. Gumiere. Bayesian uncertainty analysis of the distributed hydrological model hydrotel. *Journal of Hydrologic Engineering*, 17(9):1021–1032, 2012.

[9] D. Cameron, K. J. Beven, J. Tawn, S. Blazkova, and P. Naden. Flood frequency estimation by continuous simulation for a gauged upland catchment (with uncertainty). *Journal of Hydrology*, 219(3-4):169–187, 1999.

[10] Edijatno, N. DE OLIVEIRA NASCIMENTO, X. YANG, Z. MAKHLOUF, and C. MICHEL. Gr3j: a daily watershed model with three free parameters. *Hydrological sciences journal*, 44(2):263–277, 1999.

[11] R. K. Farnsworth and E. S. Thompson. *Mean monthly, seasonal, and annual pan evaporation for the United States.* National Oceanic and Atmospheric Administration, National Weather Service, 1982.

[12] J. Freer, K. Beven, and B. Ambroise. Bayesian estimation of uncertainty in runoff prediction and the value of data: An application of the glue approach. *Water Resources Research*, 32(7):2161–2173, 1996.

[13] T. Y. Gan, E. M. Dlamini, and G. F. Biftu. Effects of model complexity and structure, data quality, and objective functions on hydrologic modeling. *Journal of Hydrology*, 192(1-4):81–103, 1997.

[14] Y. Gong, Z. Shen, Q. Hong, R. Liu, and Q. Liao. Parameter uncertainty analysis in watershed total phosphorus modeling using the glue methodology. *Agriculture, ecosystems & environment*, 142(3-4):246–255, 2011.

[15] D. Harlan, M. Wangsadipura, and C. M. Munajat. Rainfall-runoff modeling of citarum hulu river basin by using gr4j. In *Proceedings of the world congress on engineering*, volume 2, pages 4–8, 2010.

[16] A. Heidari, B. Saghafian, and R. Maknoon. Assessment of flood forecasting lead time based on generalized likelihood uncertainty estimation approach. *Stochastic Environmental Research and Risk Assessment*, 20(5):363–380, 2006.

[17] D. Kavetski, S. W. Franks, G. Kuczera, et al. Confronting input uncertainty in environmental modelling. *Calibration of watershed models*, 6:49–68, 2003.

[18] G. Kuczera. Estimation of runoff-routing model parameters using incompatible storm data. *Journal of Hydrology*, 114(1-2):47–60, 1990.

[19] G. Kuczera, D. Kavetski, S. Franks, and M. Thyer. Towards a bayesian total error analysis of conceptual rainfall-runoff models: Characterising model error using storm-dependent parameters. *Journal of Hydrology*, 331(1-2):161–177, 2006.

[20] T. Lan, K. Lin, C.-Y. Xu, X. Tan, and X. Chen. Dynamics of hydrological model parameters: mechanisms, problems, and solution. *Hydrology and Earth System Sciences Discussions*, pages 1–27, 2019.

[21] N. D. O. Nascimento. Appréciation à l'aide d'un modèle empirique des effets d'actions anthropiques sur la relation pluie-débit à l'échelle d'un bassin versant. 1995.

[22] A. Saltelli, S. Tarantola, and K.-S. Chan. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics*, 41(1):39–56, 1999.

[23] B. Schaefli and E. Zehe. Hydrological model performance and parameter estimation in the wavelet-domain. *Hydrology and Earth System Sciences*, 13(ARTICLE):1921–1936, 2009.

[24] Z. Shen, L. Chen, T. Chen, and G. Di Baldassarre. Analysis of parameter uncertainty in hydrological and sediment modeling using glue method: a case study of swat model applied to three gorges reservoir region, china. *Hydrology & Earth System Sciences*, 16(1), 2012.

[25] J. R. Stedinger, R. M. Vogel, S. U. Lee, and R. Batchelder. Appraisal of the generalized likelihood uncertainty estimation (glue) method. *Water resources research*, 44(12), 2008.

[26] M. Tajiki, G. Schoups, H. Hendricks Franssen, A. Najafinejad, and A. Bahremand. Recursive bayesian estimation of conceptual rainfall-runoff model errors in real-time prediction of streamflow. *Water Resources Research*, 56(2):e2019WR025237, 2020.

[27] E. Todini and P. Mantovan. Comment on:'on undermining the science?'by keith beven. *Hydrological Processes: An International Journal*, 21(12):1633–1638, 2007.

[28] J. A. Vrugt, H. V. Gupta, W. Bouten, and S. Sorooshian. A shuffled complex evolution metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. *Water resources research*, 39(8), 2003.

[29] Q. Wang, P. Hapuarachchi, and T. Pagano. Continuous rainfall-runoff model comparison and short-term daily streamflow forecast skill evaluation. 2010. doi: 10.4225/08/58542C672DD2C. URL https://publications.csiro.au/publications/#publication/PIcsiro:EP103545.

[30] H. Winsemius, B. Schaefli, A. Montanari, and H. Savenije. On the calibration of hydrological models in ungauged basins: A framework for integrating hard and soft hydrological information. *Water Resources Research*, 45(12), 2009.

[31] L. Xiong and K. M. O'Connor. An empirical method to improve the prediction limits of the glue methodology in rainfall–runoff modeling. *Journal of Hydrology*, 349(1-2):115–124, 2008.

[32] C. Xue, B. Chen, and H. Wu. Parameter uncertainty analysis of surface flow and sediment yield in the huolin basin, china. *Journal of Hydrologic Engineering*, 19(6):1224–1236, 2014.

[33] J. Yang, P. Reichert, and K. C. Abbaspour. Bayesian uncertainty analysis in distributed hydrologic modeling: A case study in the thur river basin (switzerland). *Water resources research*, 43(10), 2007.

[34] J. Yang, P. Reichert, K. C. Abbaspour, and H. Yang. Hydrological modelling of the chaohe basin in china: Statistical model formulation and bayesian inference. *Journal of Hydrology*, 340(3-4):167–182, 2007.

[35] J. Yang, P. Reichert, K. C. Abbaspour, J. Xia, and H. Yang. Comparing uncertainty analysis techniques for a swat application to the chaohe basin in china. *Journal of Hydrology*, 358(1-2):1–23, 2008.

6 Appendix

storm-based calibration and validation for basin A

with the threshold relative with the median value of rainfall series (in this case, threshod=0.21) to divide the storm epochs

In [1]:

```
#%% 0
import numpy as np
import matplotlib.pyplot as plt
from gr4j import gr4j#the rainfall-runoff model
from scipy.optimize import differential_evolution#the calibratio2an algorithm
import pandas as pd
import scipy.stats as st
from scipy.stats import beta
import warnings
warnings.filterwarnings('ignore')
A='07068000.dly'
def forcingdata(A): #read forcing data
    data = np.loadtxt(A,dtype=str,delimiter='\t')
    data = [data[ii][0:4] + ' ' + data[ii][4:6] + ' '  + data[ii][6:8] + data[ii][8:
            for ii in np.arange(len(data))]
    data = [data[ii].split() for ii in np.arange(len(data))]
    data = pd.DataFrame(data,columns =['y','m','d','r','ep','s','t1','t2'])
    discharge=np.array(data['s'][14610:16438],dtype=float)
    rainfall=np.array(data['r'][14610:16438],dtype=float)
    ep=np.array(data['ep'][14610:16438],dtype=float)
    data.head()
    return discharge,rainfall,ep
d_o=forcingdata(A)[0]
r_o=forcingdata(A)[1]
p_o=forcingdata(A)[2]
def G(r,ep,p):   #define the G function for first storm  (storm 0)
    params={ 'X1': p[0], 'X2': p[1], 'X3': p[2], 'X4': p[3] }
    states= { 'production_store': 0.60 * params['X1'], 'routing_store': 0.70 * param
    Qm= gr4j(r, ep, params, states)
    s=gr4j(r, ep, params, states,return_state=True)[1] # return the states at the en
    return Qm,s

def obj(B):  #run for the whole period to find the value of constant parameter X2,X3
        Qm=np.array(G(r_o,p_o,B))[0]
        Qo=d_o
        QM=Qm
        # RMS=np.sum(((Qo-QM)**2) )/len(Qo)

        ErrUp=np.sum((QM-Qo)**2)
        ErrDo = np.sum((Qo-np.mean(Qo))**2)
        NSE = 1- ErrUp / ErrDo
        return -NSE
Bounds= [(20 ,2000),(-5,3),(20,300),(1,5)]
result= differential_evolution(obj, Bounds)
if np.median(r_o)>0.3:
    threshold_o=0.1
elif np.median(r_o)<0.15:
    threshold_o=0.15
else:
    threshold_o=np.median(r_o)
```

In [2]:

```
#%% 1
def stormepoch():  # define storm epoch
    ind_o=[]
    ind_o.append(0)
    for i in range(1,len(r_o)-1):
        if (r_o[i]>threshold_o)&(r_o[i-1]==0):
            ind_o.append(i)
    ind_o.append(len(r_o))
    storm_o=[]

    for i in range(len(ind_o)-1):
        storm_o.append((ind_o[i],ind_o[i+1]-1))
    return ind_o,storm_o
ind_o=stormepoch()[0]
storm_o=stormepoch()[1]

def obj1(B): #run for the first storm (storm 0), use the initial condition: 0.6*X1,
        i=ind_o[0] #index of fisrt day of storm 0
        j=ind_o[1] #index of end day of storm 0
        Qm=np.array(G(r_o[i:j],p_o[i:j],B)[0])
        Qo=d_o[i:j]
        QM=Qm
        ErrUp=np.sum((QM-Qo)**2)
        ErrDo = np.sum((Qo-np.mean(Qo))**2)
        NSE = 1- ErrUp / ErrDo
        return -NSE
Bounds= [(20 ,2000),(result.x[1],result.x[1]),(result.x[2],result.x[2]),(1,5)]
result1= differential_evolution(obj1, Bounds)

m=ind_o[0] # index of fisrt day of storm 0
M=ind_o[1] # index of end day of storm 0
q=[]       # a list for simulated discharge
par=[]     # a list for calibrated parameter sets

q=q+G(r_o[m:M],p_o[m:M],result1.x)[0]
par.append(result1.x)
states= np.array(G(r_o[m:M],p_o[m:M],result1.x))[1]
production_store = states['production_store']  # production_store at the end of stor
routing_store= states['routing_store']          # routing_store at the end of storm 0
```

In [3]:

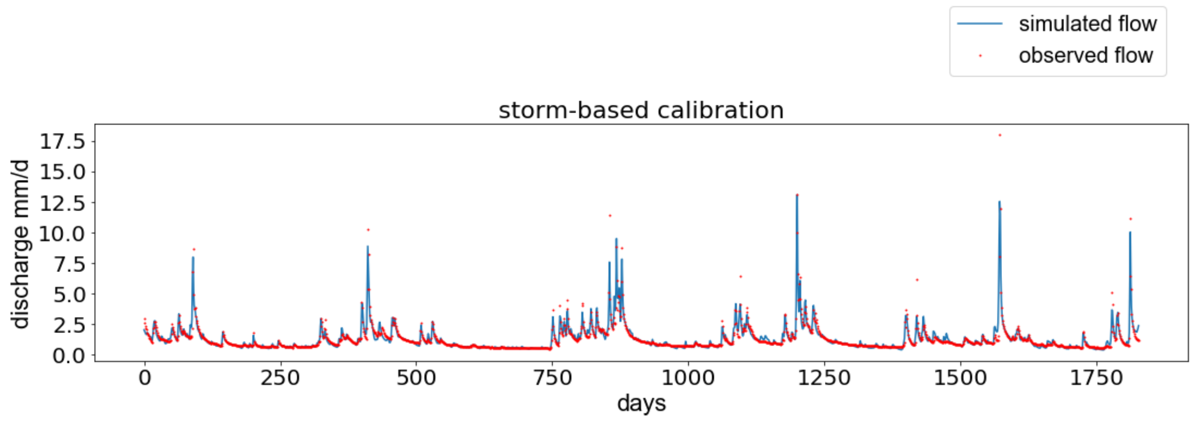
```
#%% 2 change initial state from second storm
p_s=np.zeros(len(storm_o)) # an array for all the END production store for each stor
r_s=np.zeros(len(storm_o)) # an array for all the END routing store for each storm
p_s[0]=production_store
r_s[0]=routing_store
```

In [4]:

```
#%% 3
for t in range(1,len(storm_o)):
    i=ind_o[t]    #index of first day of each storm
    j=ind_o[t+1]  #index of end day of each storm

    def G2(r,ep,p):   # other storms use the the state at end of former storm as the

        params={ 'X1': p[0], 'X2': p[1], 'X3': p[2], 'X4': p[3] }
        states= { 'production_store': p_s[t-1], 'routing_store':r_s[t-1] }


        Qm,s= gr4j(r, ep, params, states, return_state=True)
        return Qm,s

    def obj2(B): #calculate the objective value

        Qm=np.array(G2(r_o[i:j],p_o[i:j],B)[0])
        Qo=d_o[i:j]
        QM=Qm
        ErrUp=np.sum((QM-Qo)**2)
        ErrDo = np.sum((Qo-np.mean(Qo))**2)
        NSE = 1- ErrUp / ErrDo
        return -NSE

    Bounds2=[(20 ,2000),(result1.x[1],result1.x[1]),(result1.x[2],result1.x[2]),(1,5
    resultn=differential_evolution(obj2, Bounds2)
    statesn= G2(r_o[i:j],p_o[i:j],resultn.x)[1]
    p_s[t]=statesn['production_store']
    r_s[t]= statesn['routing_store']
    q=q+G2(r_o[i:j],p_o[i:j],resultn.x)[0]
    par.append(resultn.x)
font1 = {'family' : 'Arial',
    'weight' : 'normal',
    'size' : 20,}
font2 = {'family' : 'Arial',
    'weight' : 'normal',
    'size' : 22,}
num1=1.02
num2=1.02
num3=4
num4=2
```

In [5]:

```
#%%4 plt calibration result
plt.figure(figsize=(18,4))
plt.title('storm-based calibration',fontsize=22)
plt.xlabel('days',font2)
plt.ylabel('discharge mm/d',font2)
plt.plot(q,'-',markersize=2,label='simulated flow')
l=len(q)
plt.plot(d_o[0:l],'r.',markersize=2,label='observed flow')

plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.legend(bbox_to_anchor=(num1, num2), loc=num3, borderaxespad=num4,prop=font1)
plt.savefig('A_s_c.pdf', bbox_inches='tight')
np.savetxt('par',par)
```



In [6]:

```
#%%5
def validationdata(A):
        data = np.loadtxt(A,dtype=str,delimiter='\t')
        data = [data[ii][0:4] + ' ' + data[ii][4:6] + ' '+ data[ii][6:8]+data[ii][8:
                for ii in np.arange(len(data))]
        data = [data[ii].split() for ii in np.arange(len(data))]
        data = pd.DataFrame(data,columns =['y','m','d','r','ep','s','t1','t2'])
        discharge=np.array(data['s'][16438:18263],dtype=float)
        rainfall=np.array(data['r'][16438:18263],dtype=float)
        ep=np.array(data['ep'][16438:18263],dtype=float)
        data.head()
        return discharge,rainfall,ep
discharge=validationdata(A)[0]
rainfall=validationdata(A)[1]
potential_evap=validationdata(A)[2]
if np.median(rainfall)>0.3:
    threshold=0.1
elif np.median(rainfall)<0.15:
    threshold=0.15
else:
    threshold=np.median(rainfall)
```

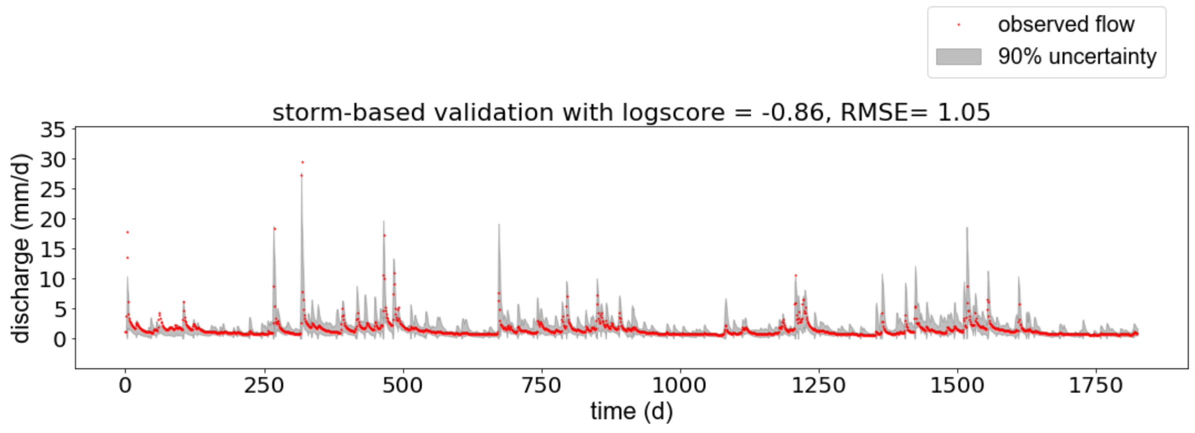
In [7]:

```
#%%6
def stormevent():
    ind=[]
    ind.append(0)
    for i in range(1,len(rainfall)-1):
        if (rainfall[i]>threshold)&(rainfall[i-1]==0):
            ind.append(i)
    ind.append(len(rainfall))
    storm=[]
    for i in range(len(ind)-1):
        storm.append((ind[i],ind[i+1]-1))
    return ind,storm
ind=stormevent()[0]
storm=stormevent()[1]
QS=np.zeros( (len(storm),len(discharge)))
QS_o=np.zeros((len(discharge)))
q=np.array(q)
p_s_v=np.zeros(len(storm))
r_s_v=np.zeros(len(storm))
for k in range (len(storm)):
    for n in range(len(storm)):
        if n==0:
            i=ind[0]
            j=ind[1]
            m=int(np.random.choice(np.linspace(1,len(par)-1,len(par)-1),1)) #randoml
            if p_s[-1]>par[m][0]:
                    par[m][0]=p_s[-1]/0.6
            # if r_s[-1]>par[m][2]:
            #          par[m][2]=r_s[-1]/0.7
            def G3(r,ep,p):
                    params={ 'X1': p[0], 'X2': p[1], 'X3': p[2], 'X4': p[3] }
                    states= { 'production_store': p_s[-1], 'routing_store':r_s[-1] }
                    Qm_v,s_v= gr4j(r, ep, params, states, return_state=True)
                    return Qm_v,s_v
            QS[k,i:j]=G3(rainfall[i:j], potential_evap[i:j], par[m])[0]
            states_v=G3(rainfall[i:j], potential_evap[i:j], par[m])[1]
            p_s_v[n]=states_v['production_store']
            r_s_v[n]= states_v['routing_store']

        else:
            for n in range(1,len(storm)):
                i=ind[n]
                j=ind[n+1]
                m=int(np.random.choice(np.linspace(1,len(par)-1,len(par)-1),1)) #rar
                if p_s_v[n-1]>par[m][0]:
                    par[m][0]=p_s_v[n-1]/0.6
                # if r_s_v[n-1]>par[m][2]:
                #     par[m][2]=r_s_v[n-1]/0.7
                def G4(r,ep,p):
                    params={ 'X1': p[0], 'X2': p[1], 'X3': p[2], 'X4': p[3] }
                    states= { 'production_store': p_s_v[n-1], 'routing_store':r_s_v[
                    Qm_v,s_v= gr4j(r, ep, params, states, return_state=True)
                    return Qm_v,s_v
                QS[k,i:j]=G4(rainfall[i:j], potential_evap[i:j], par[m])[0]
                states_v=G4(rainfall[i:j], potential_evap[i:j], par[m])[1]
                p_s_v[n]=states_v['production_store']
                r_s_v[n]= states_v['routing_store']
#     print('validation process:',k+1,'/',len(storm))
```

directly sampling X1 and X4 from the calibrated results

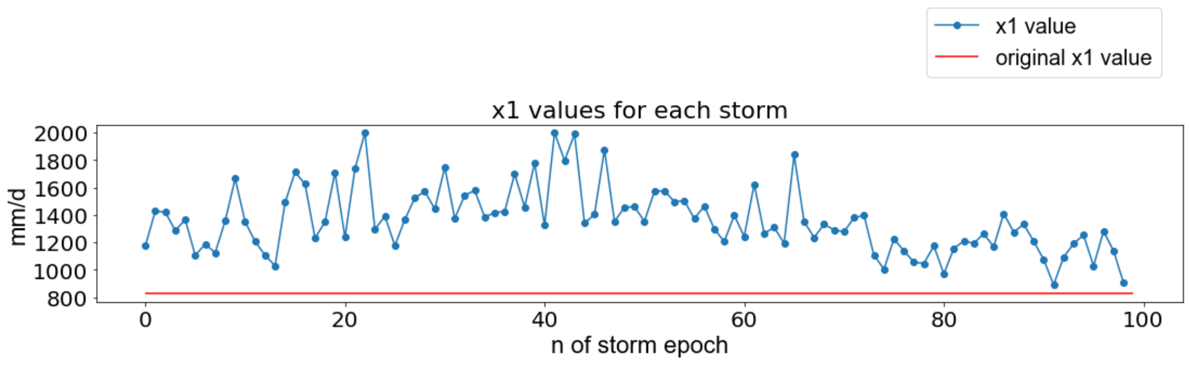
In [8]:

```
#%% 7
Qm_5_k=np.zeros(len(discharge))
Qm_95_k=np.zeros(len(discharge))
logscore1=np.zeros(len(discharge))
RMSE1=np.zeros(len(discharge))
for i in range(len(discharge)):
    Vobs=0
    Mobs=discharge[i]
    Mpredict=np.mean(QS[:,i])
    Vpredict=(np.sum(  (QS[:,i]-Mpredict) **2   ) / len(storm) ) **0.5
    RMSE1[i]= (Mobs-Mpredict)**2
    logscore1[i]=-1/2*np.log(2*np.pi)-1/2*np.log(Vobs+Vpredict)-(Mobs-Mpredict)**2/(
    Qm_5_k[i] = norm.ppf(0.05,Mpredict,Vpredict) # use the  simulated discharge as m
    Qm_95_k[i] =norm.ppf(0.95,Mpredict,Vpredict) #
    Qm_5_k[Qm_5_k<0]=0
logscore_av1=(np.mean(logscore1))
RMSE_av1=(np.sum(RMSE1)/len(discharge))**0.5
plt.figure(figsize=(18,4))
plt.fill_between(np.linspace(1,len(discharge),len(discharge)),Qm_5_k,Qm_95_k,alpha=0
plt.plot(discharge,'r.',markersize=2,label='observed flow')
plt.title('storm-based validation with logscore = ' +str(format(logscore_av1, '.2f')
plt.ylim(-4.9,np.max(discharge)*1.2)
plt.xlabel('time (d)',font2)
plt.ylabel('discharge (mm/d)',font2)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.legend(bbox_to_anchor=(num1, num2), loc=num3, borderaxespad=num4,prop=font1)
plt.savefig('A_s_v.pdf', bbox_inches='tight')
```

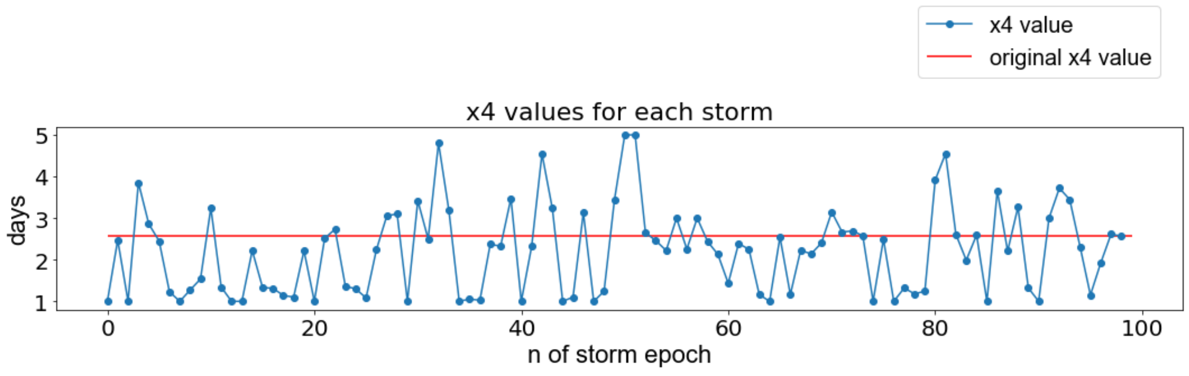


In [9]:

```
#%%8_1 plot the scatters
#%% 7.2
hflow=[]
hindex=[]
lflow=[]
lindex=[]
for m in range(len(rainfall)):
    if rainfall[m]<5:
        lindex.append(m)
        lflow.append(discharge[m])
    else:
        hindex.append(m)
        hflow.append(discharge[m])
hflow=np.array(hflow)
lflow=np.array(lflow)
hindex=np.array(hindex)
lindex=np.array(lindex)
hrmse=np.zeros(len(hindex))
lrmse=np.zeros(len(lindex))
for i in range(len(hindex)):
    n=hindex[i]
    hrmse[i]=(hflow[i]-np.mean(QS[:,n]))**2
Hrmse1=(np.sum(hrmse)/len(hflow) )**(1/2)
for i in range(len(lindex)):
    n=lindex[i]
    lrmse[i]=(lflow[i]-np.mean(QS[:,n]))**2
Lrmse1=(np.sum(lrmse)/len(lflow) )**(1/2)
par=np.array(np.loadtxt('par'))
p1=par[:,0]
p4=par[:,3]
plt.figure(figsize=(18,3))
plt.title('x1 values for each storm',fontsize=22)
plt.plot(p1,'o-',label='x1 value')
plt.hlines(result.x[0],0,len(par[:,0]),'r',label='original x1 value')
plt.ylabel('mm/d',font2)
plt.xlabel('n of storm epoch',font2)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.legend(bbox_to_anchor=(num1, num2), loc=num3, borderaxespad=num4,prop=font1)
plt.savefig('A_s_x1.pdf', bbox_inches='tight')

plt.figure(figsize=(18,3))
plt.title('x4 values for each storm',fontsize=22)
plt.plot(p4,'o-',label='x4 value')
plt.xlabel('n of storm epoch',font2)
plt.ylabel('days',font2)
plt.hlines(result.x[3],0,len(par[:,3]),'r',label='original x4 value')
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.legend(bbox_to_anchor=(num1, num2), loc=num3, borderaxespad=num4,prop=font1)
plt.savefig('A_s_x4.pdf', bbox_inches='tight')
```



x1 value
original x1 value
x1 values for each storm
mm/d
n of storm epoch

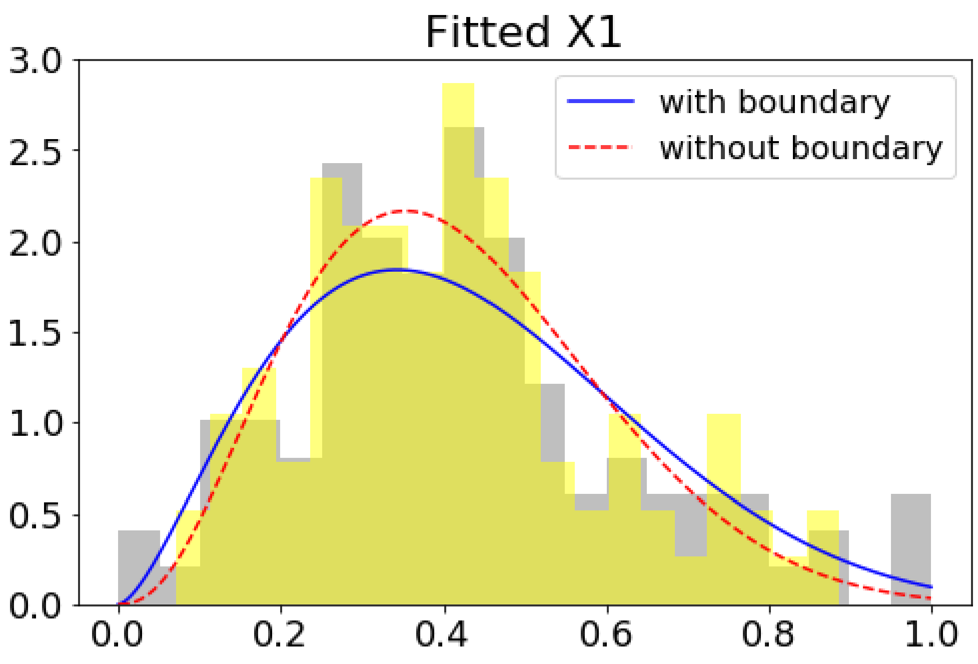


x4 value
original x4 value
x4 values for each storm
days
n of storm epoch

Sampling x1 and x4 from their distributions

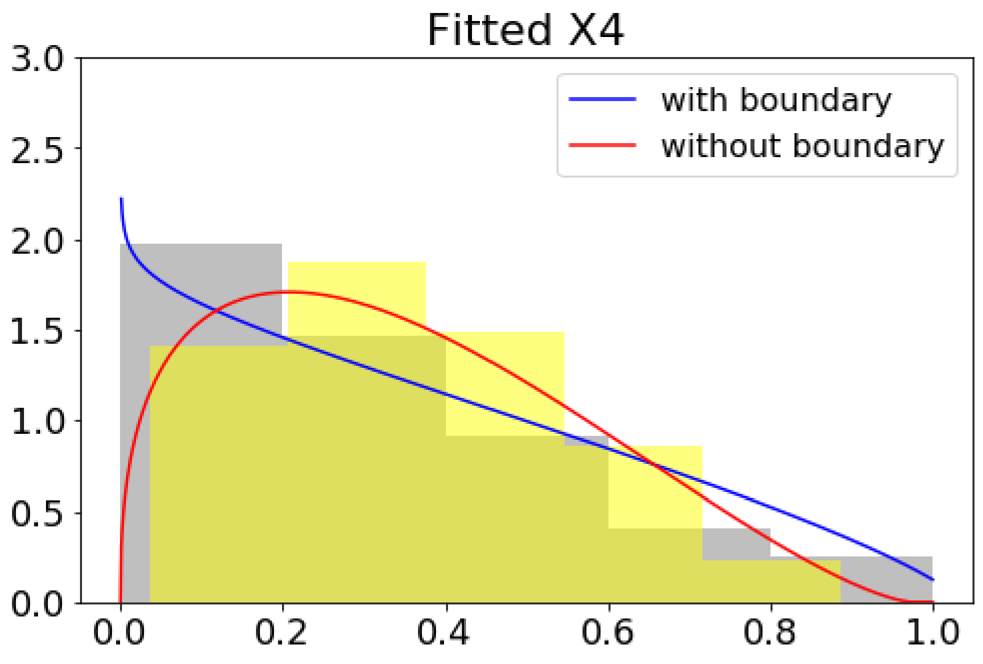
In [10]:

```
#%% 8 #fit distribution
p1=np.loadtxt('par')[:,0]
p4=np.loadtxt('par')[:,3]
plt.figure(figsize=(8,5))
dividedby1=np.max(p1)-np.min(p1)
num_bins1 = 20
pp1=(p1-np.min(p1))/dividedby1
n1, bins1, patches1 = plt.hist(pp1, num_bins1,density=1, facecolor='grey', alpha=0.5
# x1=np.linspace(0.025,0.975,1000)
x1 = np.linspace(np.min(p1-np.min(p1))/dividedby1,np.max(p1-np.min(p1))/dividedby1,1
a,b,c,d= beta.fit(pp1, floc=0)
d1=st.beta(a,b,c,d).rvs(1000)
y1 = beta.pdf(x1, a,b,c,d)
plt.title('Fitted X1',fontsize=22)
plt.plot(x1, y1,'blue',label='with boundary')
dividedby1=np.max(p1)-np.min(p1)
n1, bins1, patches1 = plt.hist(pp1[(pp1<0.95)&(pp1>0.05)], num_bins1,density=1, face
x1 = np.linspace(np.min(p1-np.min(p1))/dividedby1,np.max(p1-np.min(p1))/dividedby1,1
a,b,c,d= beta.fit(pp1[(pp1<0.95)&(pp1>0.05)], floc=0)
d1=st.beta(a,b,c,d).rvs(1000)
y1 = beta.pdf(x1, a,b,c,d)
plt.plot(x1, y1,'r--',label='without boundary')
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
plt.legend(fontsize=16)
plt.ylim(0,3)
plt.savefig('A_s_x1_beta.pdf', bbox_inches='tight')
```



In [11]:

```
#%%9
plt.figure(figsize=(8,5))
dividedby4=np.max(p4)-np.min(p4)
num_bins4 = 5
pp4=(p4-np.min(p4))/dividedby4

n4, bins4, patches4 = plt.hist(pp4, num_bins4,density=1, facecolor='grey', alpha=0.5
# x4=np.linspace(0.05,0.95,1000)
x4 = np.linspace(np.min(p4-np.min(p4))/dividedby4,np.max(p4-np.min(p4))/dividedby4,1

a,b,c,d= beta.fit(pp4, floc=0)
d4=st.beta(a,b,c,d).rvs(1000)
y4 = beta.pdf(x4, a,b,c,d)
plt.title('Fitted X4')
plt.plot(x4, y4,'blue',label='with boundary')
plt.ylim(0,3)
plt.legend()

dividedby4=np.max(p4)-np.min(p4)
num_bins4 = 5
pp4=(p4-np.min(p4))/dividedby4

n4, bins4, patches4 = plt.hist(pp4[(pp4<0.95)&(pp4>0.03)], num_bins4,density=1, face
# x4=np.linspace(0.05,0.95,1000)
x4 = np.linspace(np.min(p4-np.min(p4))/dividedby4,np.max(p4-np.min(p4))/dividedby4,1
a,b,c,d= beta.fit(pp4[(pp4<0.95)&(pp4>0.03)], floc=0)
d4=st.beta(a,b,c,d).rvs(1000)
y4 = beta.pdf(x4, a,b,c,d)
plt.title('Fitted X4',fontsize=22)
plt.plot(x4, y4,'red',label='without boundary')
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
plt.legend(fontsize=16)
plt.savefig('A_s_x4_beta.pdf', bbox_inches='tight')
```

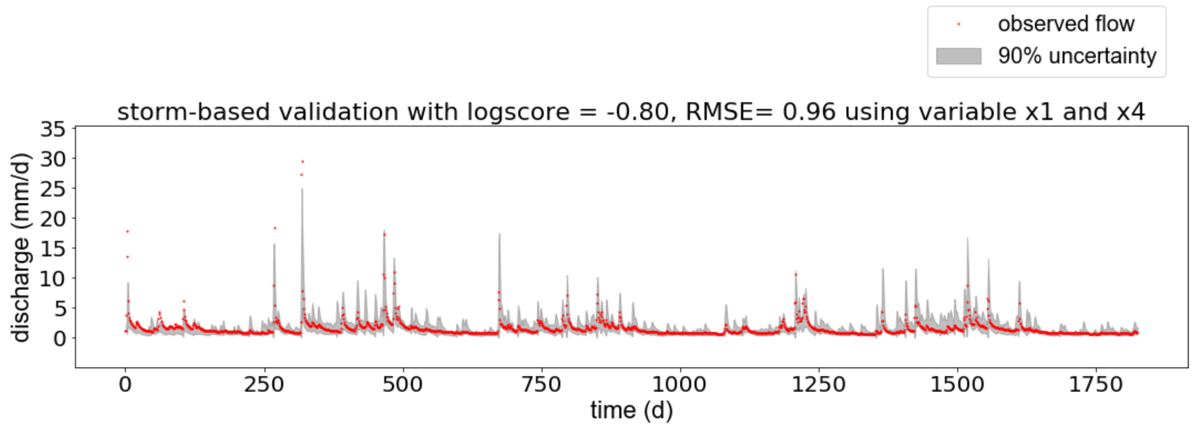


In [12]:

```
#%%10
import math
QS=np.zeros( (len(storm),len(discharge)))
QS_o=np.zeros((len(discharge)))
q=np.array(q)
p_s_v=np.zeros(len(storm))
r_s_v=np.zeros(len(storm))
for i in range(len(storm_o)):
    par[i][0]=dividedby1*np.random.choice(d1)+np.min(p1)    #result1.x[0]#
    par[i][3]=np.random.choice(d4)*dividedby4+np.min(p4)    #result1.x[3]   #
for k in range (len(storm)):
    for n in range(len(storm)):
        if n==0:
            i=ind[0]
            j=ind[1]
            m=int(np.random.choice(np.linspace(1,len(par)-1,len(par)-1),1)) #randoml
            if p_s[-1]>par[m][0]:
                    par[m][0]=p_s[-1]/0.6

            def G3(r,ep,p):
                    params={ 'X1': p[0], 'X2': p[1], 'X3': p[2], 'X4': p[3] }
                    states= { 'production_store': p_s[-1], 'routing_store':r_s[-1] }
                    Qm_v,s_v= gr4j(r, ep, params, states, return_state=True)
                    return Qm_v,s_v
            QS[k,i:j]=G3(rainfall[i:j], potential_evap[i:j], par[m])[0]
            states_v=G3(rainfall[i:j], potential_evap[i:j], par[m])[1]
            p_s_v[n]=states_v['production_store']
            r_s_v[n]= states_v['routing_store']
        else:
            for n in range(1,len(storm)):
                i=ind[n]
                j=ind[n+1]
                m=int(np.random.choice(np.linspace(1,len(par)-1,len(par)-1),1)) #ran
                if p_s_v[n-1]>par[m][0]:
                    par[m][0]=p_s_v[n-1]/0.6
                def G4(r,ep,p):
                    params={ 'X1': p[0], 'X2': p[1], 'X3': p[2], 'X4': p[3] }
                    states= { 'production_store': p_s_v[n-1], 'routing_store':r_s_v[
                    Qm_v,s_v= gr4j(r, ep, params, states, return_state=True)
                    return Qm_v,s_v

                QS[k,i:j]=G4(rainfall[i:j], potential_evap[i:j], par[m])[0]
                states_v=G4(rainfall[i:j], potential_evap[i:j], par[m])[1]
                p_s_v[n]=states_v['production_store']
                r_s_v[n]= states_v['routing_store']
#     print('validation process using variable x1 and x4: ',k+1,'/',len(storm))
```

In [13]:

```
#%%  11
Qm_5_k=np.zeros(len(discharge))
Qm_95_k=np.zeros(len(discharge))
logscore2=np.zeros(len(discharge))
RMSE2= np.zeros(len(discharge))
for i in range(len(discharge)):
    Vobs=0
    Mobs=discharge[i]
    Mpredict=np.mean(QS[:,i])
    Vpredict=(np.sum(  (QS[:,i]-Mpredict) **2  ) / len(storm) ) **0.5
    RMSE2[i]=(Mobs-Mpredict)**2
    logscore2[i]=-1/2*np.log(2*np.pi)-1/2*np.log(Vobs+Vpredict)-(Mobs-Mpredict)**2/(
    Qm_5_k[i] = norm.ppf(0.05,Mpredict,Vpredict) # use the  simulated discharge as m
    Qm_95_k[i] =norm.ppf(0.95,Mpredict,Vpredict) #
    Qm_5_k[Qm_5_k<0]=0
logscore_av2=(np.mean(logscore2))
RMSE_av2=(np.sum(RMSE2)/len(discharge))**0.5
plt.figure(figsize=(18,4))
plt.fill_between(np.linspace(1,len(discharge),len(discharge)),Qm_5_k,Qm_95_k,alpha=0
plt.plot(discharge,'r.',markersize=2,label='observed flow')
plt.title('storm-based validation with logscore = ' +str(format(logscore_av2, '.2f')
plt.ylim(-4.9,np.max(discharge)*1.2)
plt.xlabel('time (d)',font2)
plt.ylabel('discharge (mm/d)',font2)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.legend(bbox_to_anchor=(num1, num2), loc=num3, borderaxespad=num4,prop=font1)
plt.savefig('A_s_v_x1x4variable.pdf', bbox_inches='tight')
```



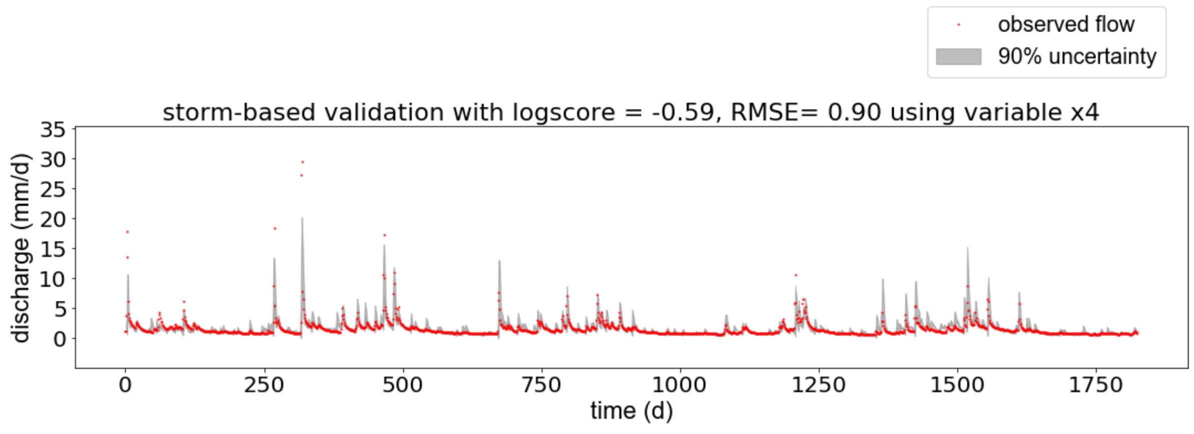
make x1 fixed and sample x4 from its distribution

In [17]:

```
#%%10
par=np.array(np.loadtxt('par'))
p4=par[:,3]
dividedby4=np.max(p4)-np.min(p4)
pp4=(p4-np.min(p4))/dividedby4
x4 = np.linspace(np.min(p4-np.min(p4))/dividedby4,np.max(p4-np.min(p4))/dividedby4,1
a,b,c,d= beta.fit(pp4[(pp4<0.95)&(pp4>0.03)], floc=0)
d4=st.beta(a,b,c,d).rvs(1000)
QS=np.zeros( (len(storm),len(discharge)))
QS_o=np.zeros((len(discharge)))
q=np.array(q)
p_s_v=np.zeros(len(storm))
r_s_v=np.zeros(len(storm))
for i in range(len(storm_o)):
    par[i][0]=result1.x[0]#dividedby1*np.random.choice(d1)+np.min(p1)   #
    par[i][3]=np.random.choice(d4)*dividedby4+np.min(p4)   #result1.x[3]   #
for k in range (len(storm)):
    for n in range(len(storm)):
        if n==0:
            i=ind[0]
            j=ind[1]
            m=int(np.random.choice(np.linspace(1,len(par)-1,len(par)-1),1)) #randoml
            if p_s[-1]>par[m][0]:
                    par[m][0]=p_s[-1]/0.6

            def G3(r,ep,p):
                    params={ 'X1': p[0], 'X2': p[1], 'X3': p[2], 'X4': p[3] }
                    states= { 'production_store': p_s[-1], 'routing_store':r_s[-1] }
                    Qm_v,s_v= gr4j(r, ep, params, states, return_state=True)
                    return Qm_v,s_v
            QS[k,i:j]=G3(rainfall[i:j], potential_evap[i:j], par[m])[0]
            states_v=G3(rainfall[i:j], potential_evap[i:j], par[m])[1]
            p_s_v[n]=states_v['production_store']
            r_s_v[n]= states_v['routing_store']
        else:
            for n in range(1,len(storm)):
                i=ind[n]
                j=ind[n+1]
                m=int(np.random.choice(np.linspace(1,len(par)-1,len(par)-1),1)) #ran
                if p_s_v[n-1]>par[m][0]:
                    par[m][0]=p_s_v[n-1]/0.6
                def G4(r,ep,p):
                    params={ 'X1': p[0], 'X2': p[1], 'X3': p[2], 'X4': p[3] }
                    states= { 'production_store': p_s_v[n-1], 'routing_store':r_s_v[
                    Qm_v,s_v= gr4j(r, ep, params, states, return_state=True)
                    return Qm_v,s_v

                QS[k,i:j]=G4(rainfall[i:j], potential_evap[i:j], par[m])[0]
                states_v=G4(rainfall[i:j], potential_evap[i:j], par[m])[1]
                p_s_v[n]=states_v['production_store']
                r_s_v[n]= states_v['routing_store']
#     print('validation process using fitted x1 and variable x4: ',k+1,'/',len(storm
```

In [18]:

```
Qm_5_k=np.zeros(len(discharge))
Qm_95_k=np.zeros(len(discharge))
logscore3=np.zeros(len(discharge))
RMSE3= np.zeros(len(discharge))
for i in range(len(discharge)):
    Vobs=0
    Mobs=discharge[i]
    Mpredict=np.mean(QS[:,i])
    Vpredict=(np.sum(  (QS[:,i]-Mpredict) **2  ) / len(storm) ) **0.5
    RMSE3[i]=(Mobs-Mpredict)**2
    logscore3[i]=-1/2*np.log(2*np.pi)-1/2*np.log(Vobs+Vpredict)-(Mobs-Mpredict)**2/(
    Qm_5_k[i] = norm.ppf(0.05,Mpredict,Vpredict) # use the  simulated discharge as m
    Qm_95_k[i] =norm.ppf(0.95,Mpredict,Vpredict) #
    Qm_5_k[Qm_5_k<0]=0
logscore_av3=(np.mean(logscore3))
RMSE_av3=(np.sum(RMSE3)/len(discharge))**0.5
plt.figure(figsize=(18,4))
plt.fill_between(np.linspace(1,len(discharge),len(discharge)),Qm_5_k,Qm_95_k,alpha=0
plt.plot(discharge,'r.',markersize=2,label='observed flow')
plt.title('storm-based validation with logscore = ' +str(format(logscore_av3, '.2f')
plt.ylim(-4.9,np.max(discharge)*1.2)
plt.xlabel('time (d)',font2)
plt.ylabel('discharge (mm/d)',font2)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.legend(bbox_to_anchor=(num1, num2), loc=num3, borderaxespad=num4,prop=font1)
plt.savefig('A_s_v_x1fixedx4variable.pdf', bbox_inches='tight')
```



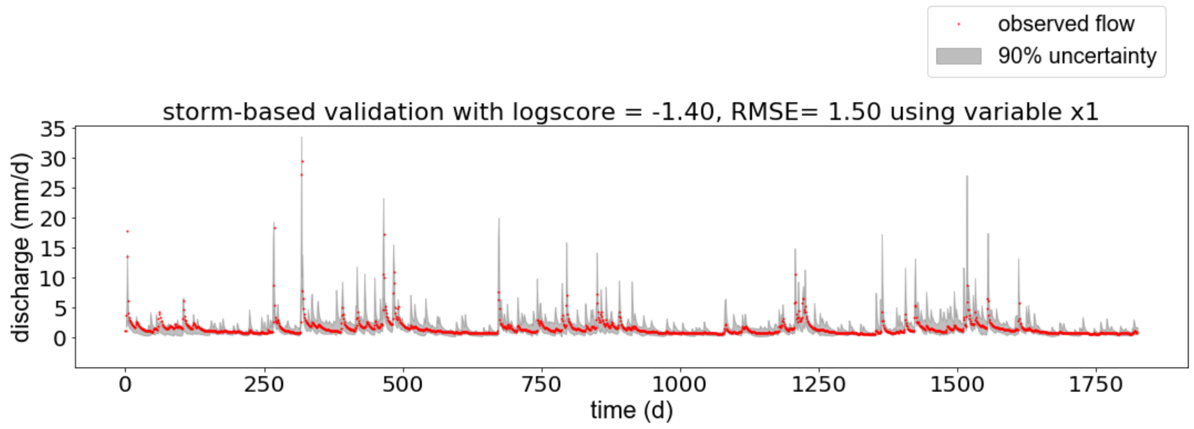
sample x1 from its distribution and make x4 fixed

In [22]:

```
#%% 8 #fit distribution
par=np.array(np.loadtxt('par'))
p1=par[:,0]
p4=par[:,3]
dividedby1=np.max(p1)-np.min(p1)
pp1=(p1-np.min(p1))/dividedby1
x1 = np.linspace(np.min(p1-np.min(p1))/dividedby1,np.max(p1-np.min(p1))/dividedby1,1
a,b,c,d= beta.fit(pp1[(pp1<0.95)&(pp1>0.05)], floc=0)
d1=st.beta(a,b,c,d).rvs(1000)
#%%10
QS=np.zeros( (len(storm),len(discharge)))
QS_o=np.zeros((len(discharge)))
q=np.array(q)
p_s_v=np.zeros(len(storm))
r_s_v=np.zeros(len(storm))
for i in range(len(storm_o)):
    par[i][0]=dividedby1*np.random.choice(d1)+np.min(p1)   #result1.x[0]#
    par[i][3]=result1.x[3]   #np.random.choice(d4)*dividedby4+np.min(p4)   #
for k in range (len(storm)):
    for n in range(len(storm)):
        if n==0:
            i=ind[0]
            j=ind[1]
            m=int(np.random.choice(np.linspace(1,len(par)-1,len(par)-1),1)) #randomi
            if p_s[-1]>par[m][0]:
                    par[m][0]=p_s[-1]/0.6

            def G3(r,ep,p):
                    params={ 'X1': p[0], 'X2': p[1], 'X3': p[2], 'X4': p[3] }
                    states= { 'production_store': p_s[-1], 'routing_store':r_s[-1] }
                    Qm_v,s_v= gr4j(r, ep, params, states, return_state=True)
                    return Qm_v,s_v
            QS[k,i:j]=G3(rainfall[i:j], potential_evap[i:j], par[m])[0]
            states_v=G3(rainfall[i:j], potential_evap[i:j], par[m])[1]
            p_s_v[n]=states_v['production_store']
            r_s_v[n]= states_v['routing_store']
        else:
            for n in range(1,len(storm)):
                i=ind[n]
                j=ind[n+1]
                m=int(np.random.choice(np.linspace(1,len(par)-1,len(par)-1),1)) #ran
                if p_s_v[n-1]>par[m][0]:
                    par[m][0]=p_s_v[n-1]/0.6
                def G4(r,ep,p):
                    params={ 'X1': p[0], 'X2': p[1], 'X3': p[2], 'X4': p[3] }
                    states= { 'production_store': p_s_v[n-1], 'routing_store':r_s_v[
                    Qm_v,s_v= gr4j(r, ep, params, states, return_state=True)
                    return Qm_v,s_v

                QS[k,i:j]=G4(rainfall[i:j], potential_evap[i:j], par[m])[0]
                states_v=G4(rainfall[i:j], potential_evap[i:j], par[m])[1]
                p_s_v[n]=states_v['production_store']
                r_s_v[n]= states_v['routing_store']
#     print('validation process using variable x1 and fixed x4: ',k+1,'/',len(storm)
```

In [23]:

```
#%%  11
Qm_5_k=np.zeros(len(discharge))
Qm_95_k=np.zeros(len(discharge))
logscore4=np.zeros(len(discharge))
RMSE4= np.zeros(len(discharge))
for i in range(len(discharge)):
    Vobs=0
    Mobs=discharge[i]
    Mpredict=np.mean(QS[:,i])
    Vpredict=(np.sum(  (QS[:,i]-Mpredict) **2  ) / len(storm) ) **0.5
    RMSE4[i]=(Mobs-Mpredict)**2
    logscore4[i]=-1/2*np.log(2*np.pi)-1/2*np.log(Vobs+Vpredict)-(Mobs-Mpredict)**2/(
    Qm_5_k[i] = norm.ppf(0.05,Mpredict,Vpredict) # use the  simulated discharge as m
    Qm_95_k[i] =norm.ppf(0.95,Mpredict,Vpredict) #
    Qm_5_k[Qm_5_k<0]=0
logscore_av4=(np.mean(logscore4))
RMSE_av4=(np.sum(RMSE4)/len(discharge))**0.5
plt.figure(figsize=(18,4))
plt.fill_between(np.linspace(1,len(discharge),len(discharge)),Qm_5_k,Qm_95_k,alpha=0
plt.plot(discharge,'r.',markersize=2,label='observed flow')
plt.title('storm-based validation with logscore = ' +str(format(logscore_av4, '.2f')
plt.ylim(-4.9,np.max(discharge)*1.2)
plt.xlabel('time (d)',font2)
plt.ylabel('discharge (mm/d)',font2)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.legend(bbox_to_anchor=(num1, num2), loc=num3, borderaxespad=num4,prop=font1)
plt.savefig('A_s_v_x1variablex4fixed.pdf', bbox_inches='tight')
```



In [24]:

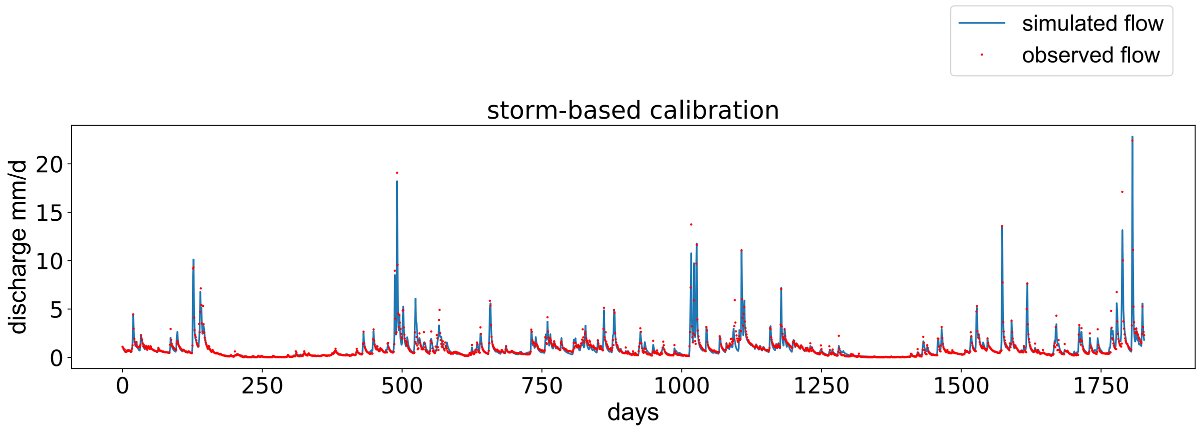
```python
#%%  12
hflow=[]
hindex=[]
lflow=[]
lindex=[]
for m in range(len(rainfall)):
    if rainfall[m]<5:
        lindex.append(m)
        lflow.append(discharge[m])
    else:
        hindex.append(m)
        hflow.append(discharge[m])

hflow=np.array(hflow)
lflow=np.array(lflow)
hindex=np.array(hindex)
lindex=np.array(lindex)
hrmse4=np.zeros(len(hindex))
lrmse4=np.zeros(len(lindex))
for i in range(len(hindex)):
    n=hindex[i]
    hrmse4[i]=(hflow[i]-np.mean(QS[:,n]))**2
Hrmse4=(np.sum(hrmse4)/len(hflow) )**(1/2)
for i in range(len(lindex)):
    n=lindex[i]
    lrmse4[i]=(lflow[i]-np.mean(QS[:,n]))**2
Lrmse4=(np.sum(lrmse4)/len(lflow) )**(1/2)
print('threshold: ',threshold)
print('n of storms: ', len(storm))
print('')
print('VALIDATION results:')
print('RMSE for highflow= ',format(Hrmse1, '.2f'))
print('RMSE for lowflow= ',format(Lrmse1, '.2f'))
print('Logscore for totalflow= ',format(logscore_av1, '.2f'))
print('RMSE for totalflow= ',format(RMSE_av1, '.2f'))
print('')
print('VALIDATION results with variable x1 and x4:')
print('RMSE for highflow= ',format(Hrmse2, '.2f'))
print('RMSE for lowflow= ',format(Lrmse2, '.2f'))
print('Logscore for totalflow= ',format(logscore_av2, '.2f'))
print('RMSE for totalflow= ',format(RMSE_av2, '.2f'))
print('')
print('VALIDATION results with fitted x1 and variable x4:')
print('RMSE for highflow= ',format(Hrmse3, '.2f'))
print('RMSE for lowflow= ',format(Lrmse3, '.2f'))
print('Logscore for totalflow= ',format(logscore_av3, '.2f'))
print('RMSE for totalflow= ',format(RMSE_av3, '.2f'))
print('')
print('VALIDATION results with variable x1 and fixed x4:')
print('RMSE for highflow= ',format(Hrmse4, '.2f'))
print('RMSE for lowflow= ',format(Lrmse4, '.2f'))
print('Logscore for totalflow= ',format(logscore_av4, '.2f'))
print('RMSE for totalflow= ',format(RMSE_av4, '.2f'))
```

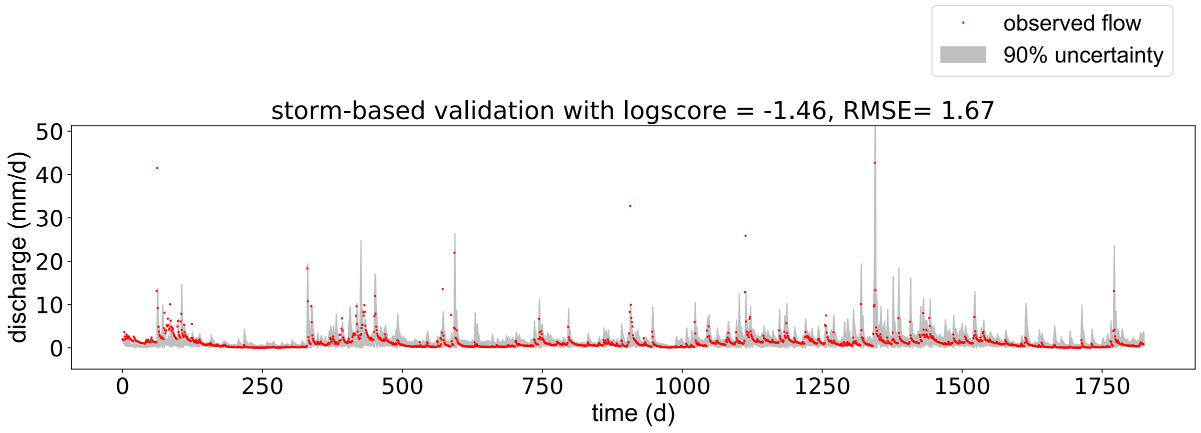
```
threshold:  0.21
n of storms:  103

VALIDATION results:
```

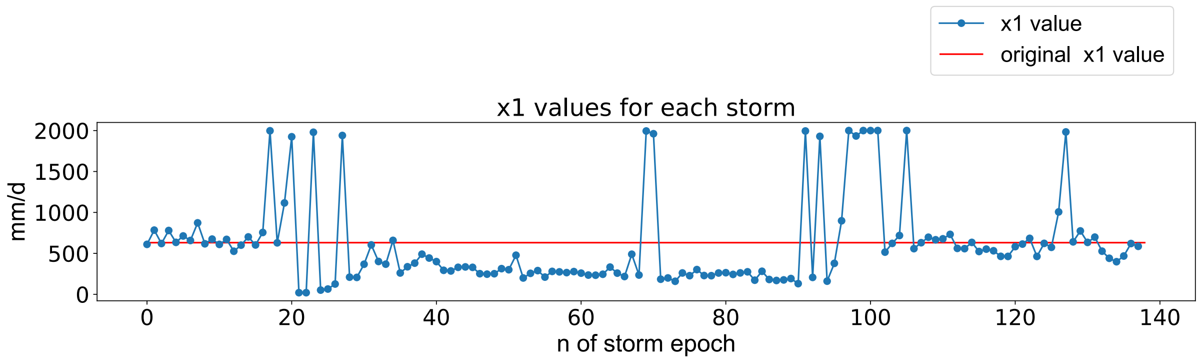
```
RMSE for highflow=  1.54
RMSE for lowflow=  0.92
Logscore for totalflow=  -0.86
RMSE for totalflow=  1.05

VALIDATION results with variable x1 and x4:
RMSE for highflow=  1.30
RMSE for lowflow=  0.87
Logscore for totalflow=  -0.80
RMSE for totalflow=  0.96

VALIDATION results with fitted x1 and variable x4:
RMSE for highflow=  1.17
RMSE for lowflow=  0.83
Logscore for totalflow=  -0.59
RMSE for totalflow=  0.90

VALIDATION results with variable x1 and fixed x4:
RMSE for highflow=  2.73
RMSE for lowflow=  1.08
Logscore for totalflow=  -1.40
RMSE for totalflow=  1.50
```

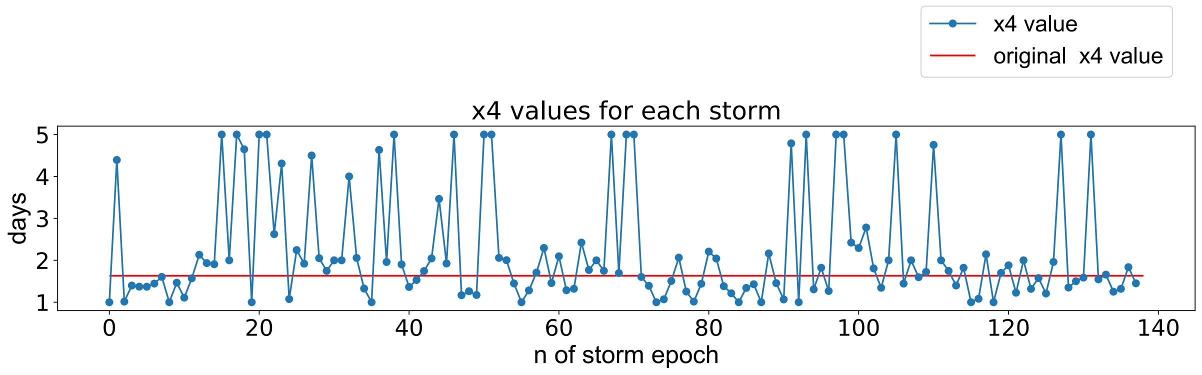
storm-based calibration & validation results for basin B

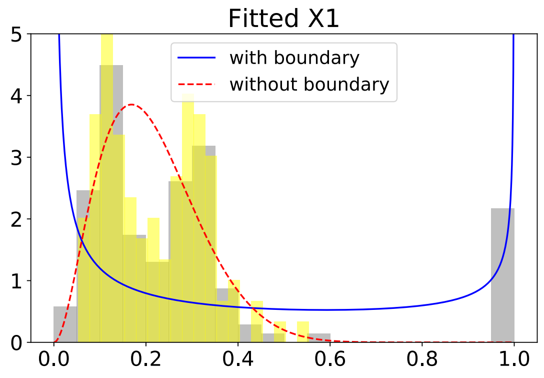
- with the threshold relative with the median value of rainfall series (in this case, threshod=0.16) to divide the storm epochs
- with variable X1 and X4

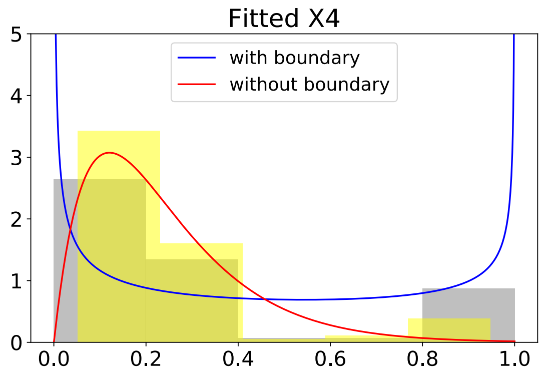


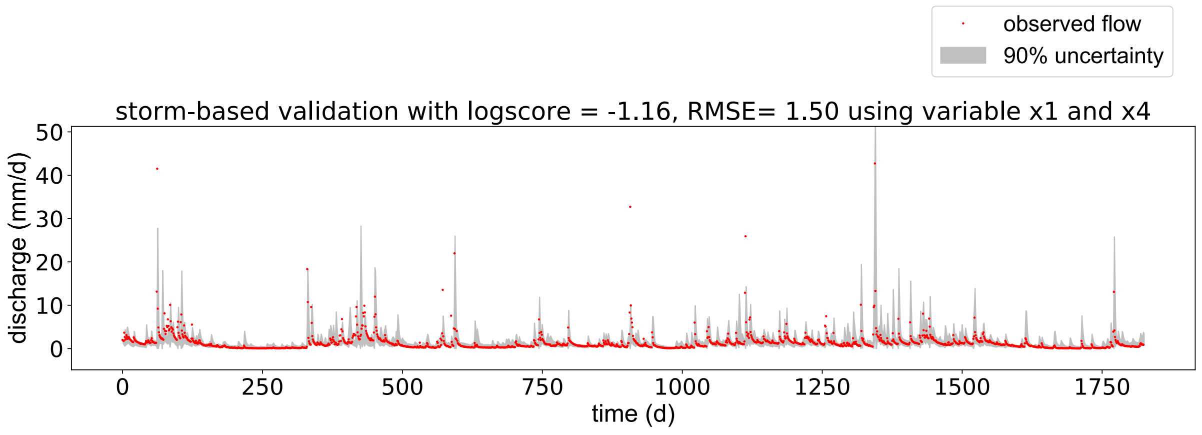





x4 value
original x4 value
x4 values for each storm
days
n of storm epoch

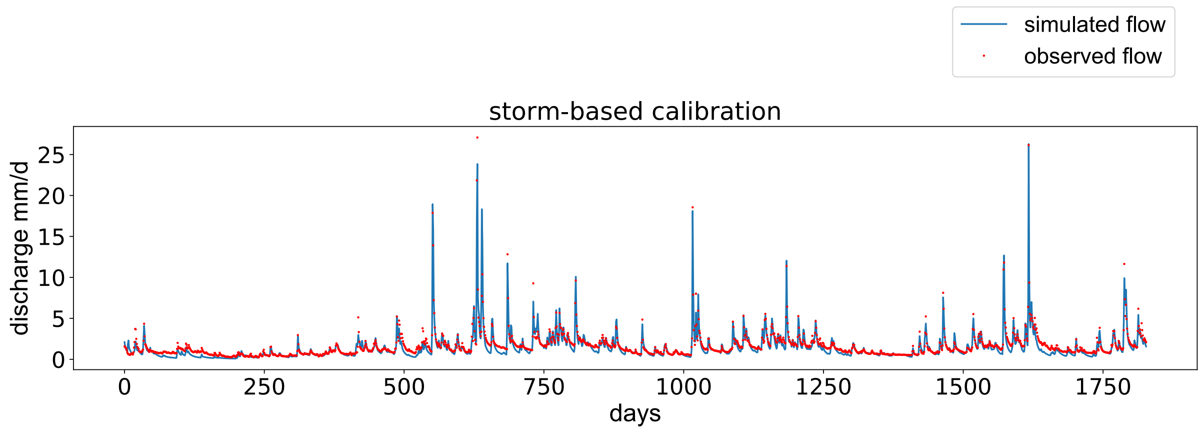

Fitted X1
with boundary
without boundary

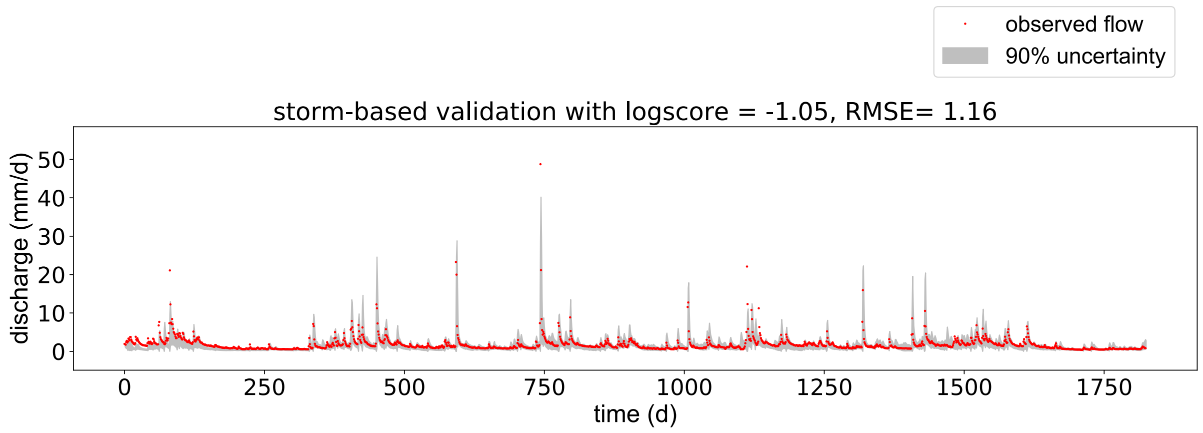

Fitted X4
with boundary
without boundary

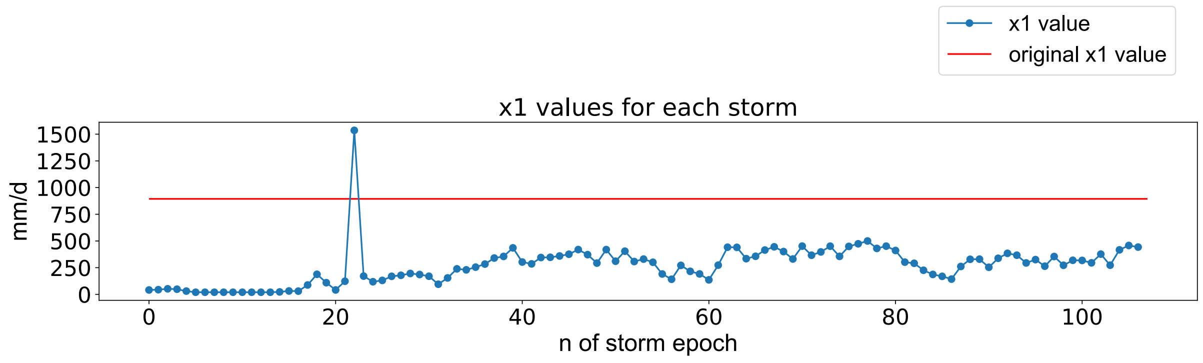

observed flow
90% uncertainty
storm-based validation with logscore = -1.16, RMSE= 1.50 using variable x1 and x4
discharge (mm/d)
time (d)

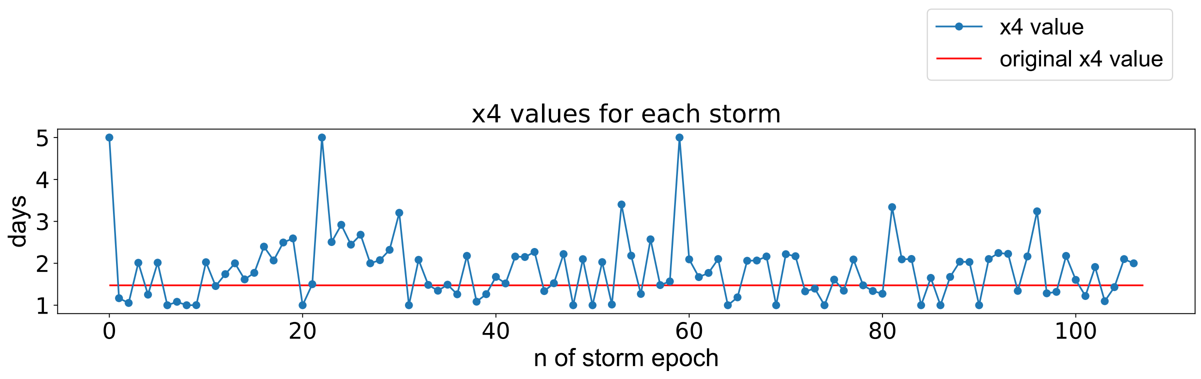
storm-based calibration & validation results for basin C

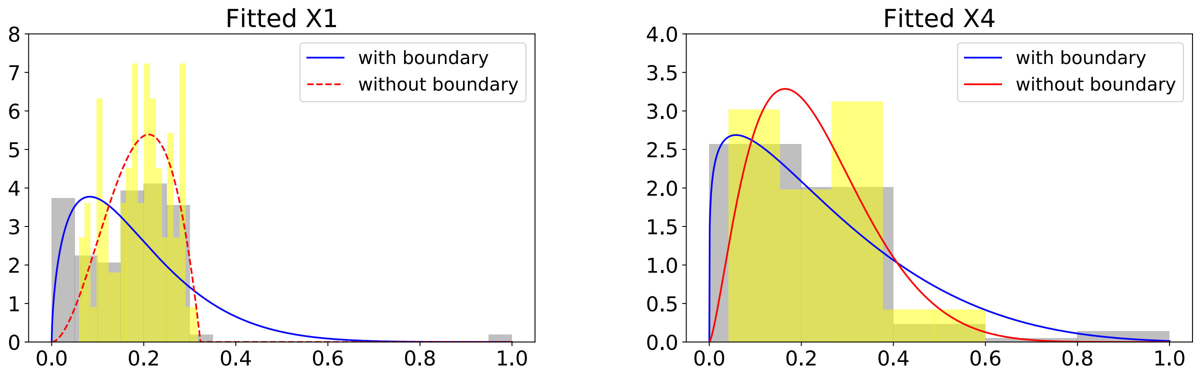
- with the threshold relative with the median value of rainfall series (in this case, threshod=0.1) to divide the storm epochs
- with variable X1 and X4

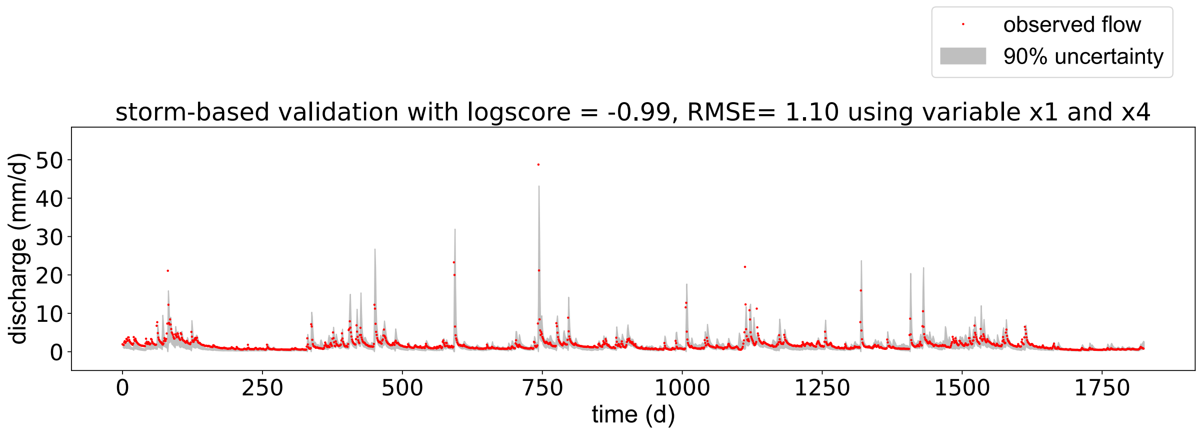






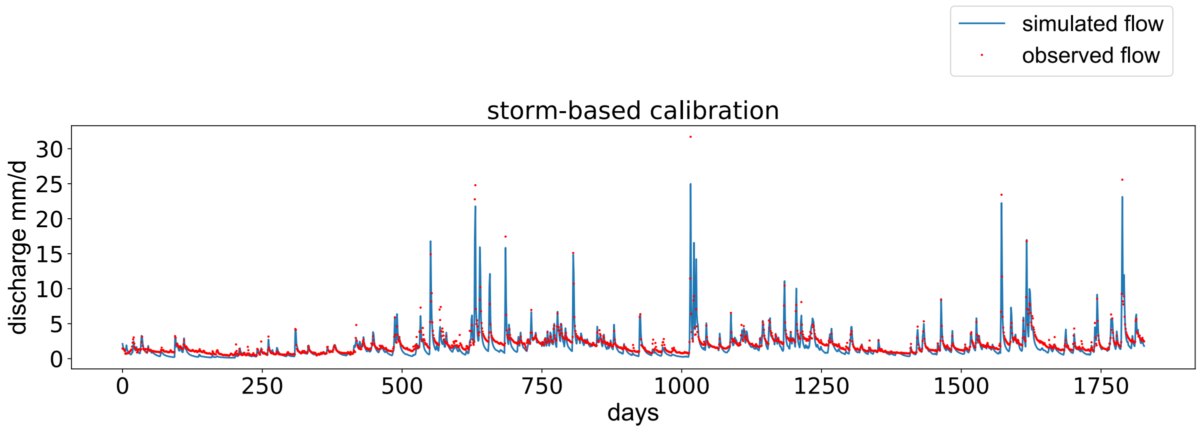

x4 value
original x4 value
x4 values for each storm
days
n of storm epoch

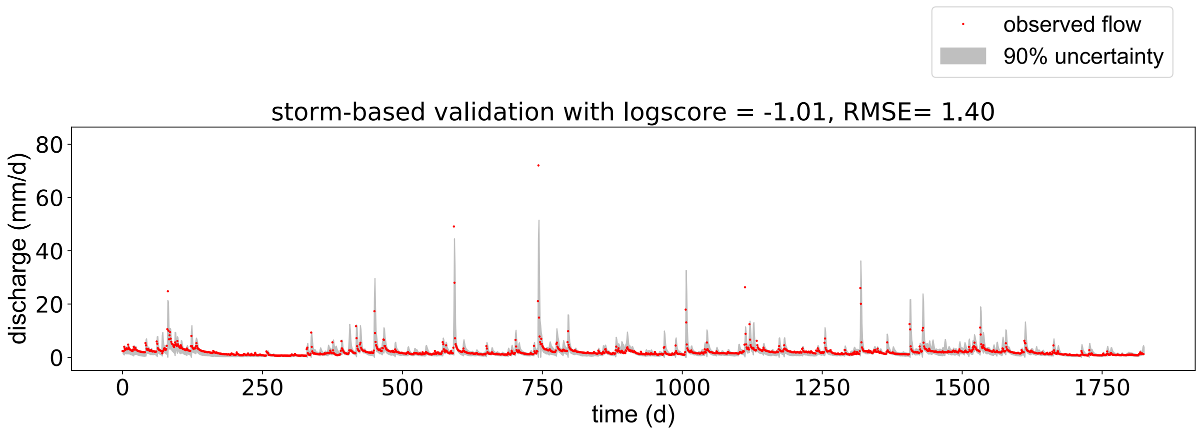

Fitted X1
with boundary
without boundary
Fitted X4
with boundary
without boundary

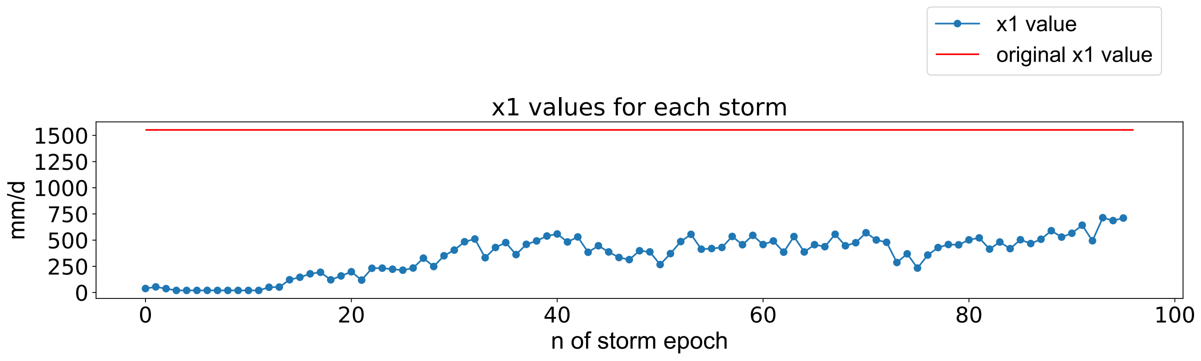

observed flow
90% uncertainty
storm-based validation with logscore = -0.99, RMSE= 1.10 using variable x1 and x4
discharge (mm/d)
time (d)

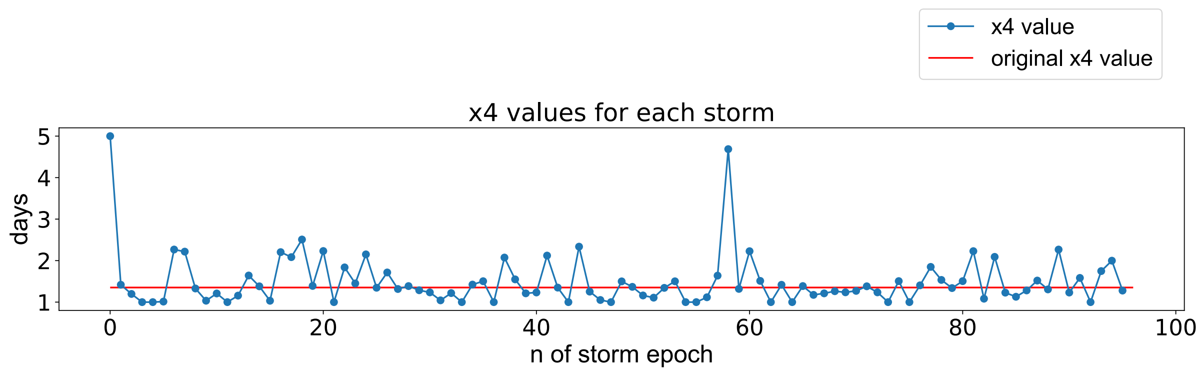
storm-based calibration & validation results for basin D

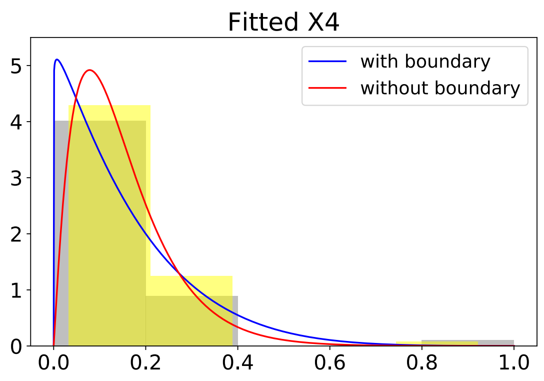
- with the threshold relative with the median value of rainfall series (in this case, threshod=0.25) to divide the storm epochs
- with variable X1 and X4

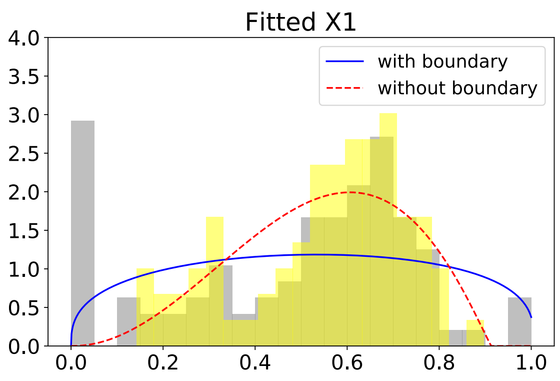


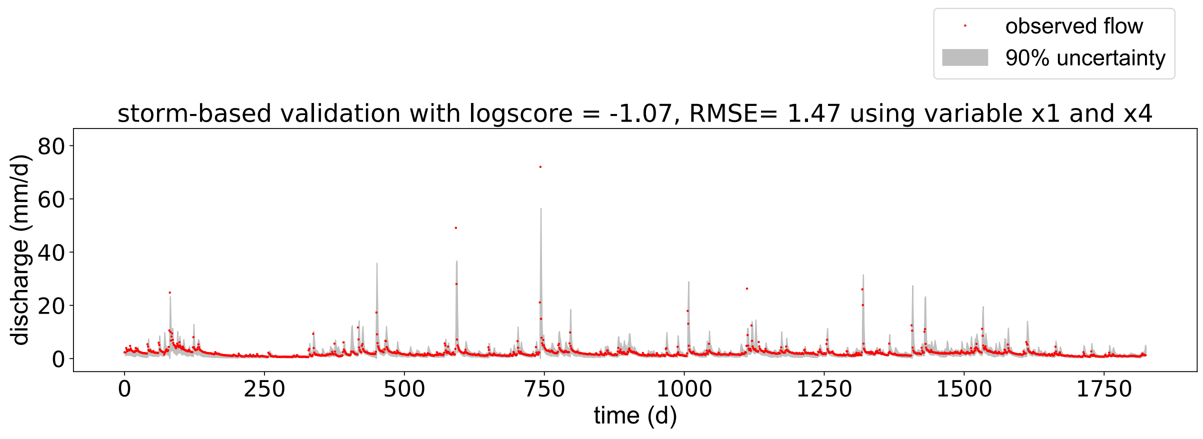





x4 value
original x4 value
x4 values for each storm
days
n of storm epoch
0
20
40
60
80
100
1
2
3
4
5

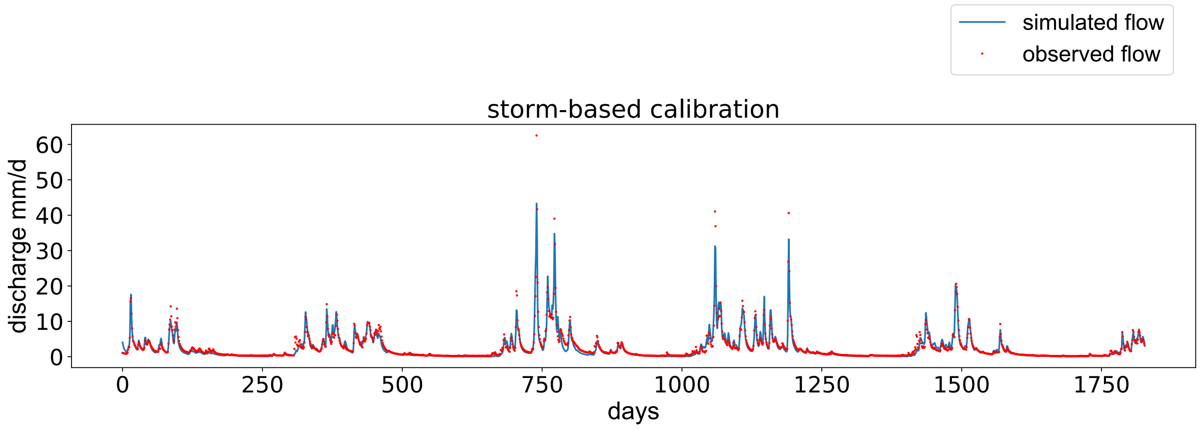

Fitted X4
with boundary
without boundary
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5

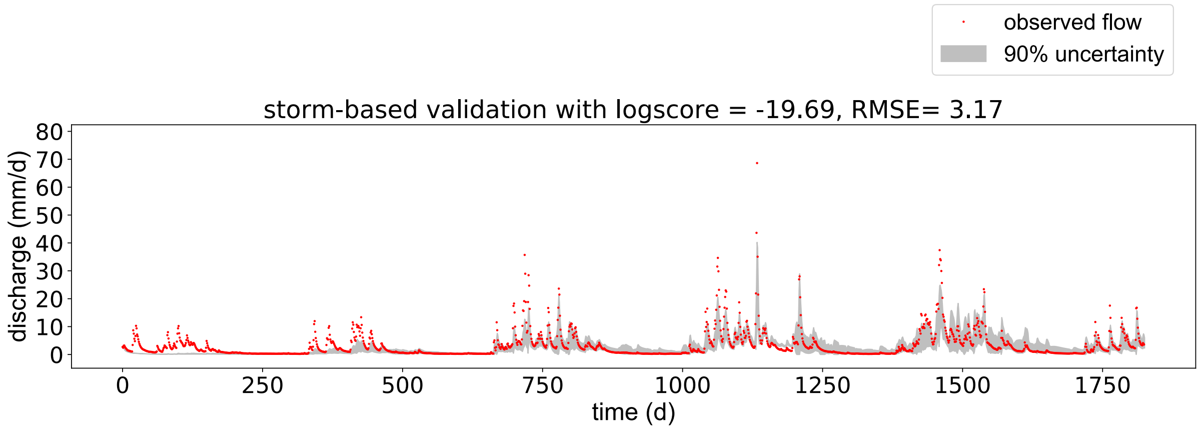

Fitted X1
with boundary
without boundary
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

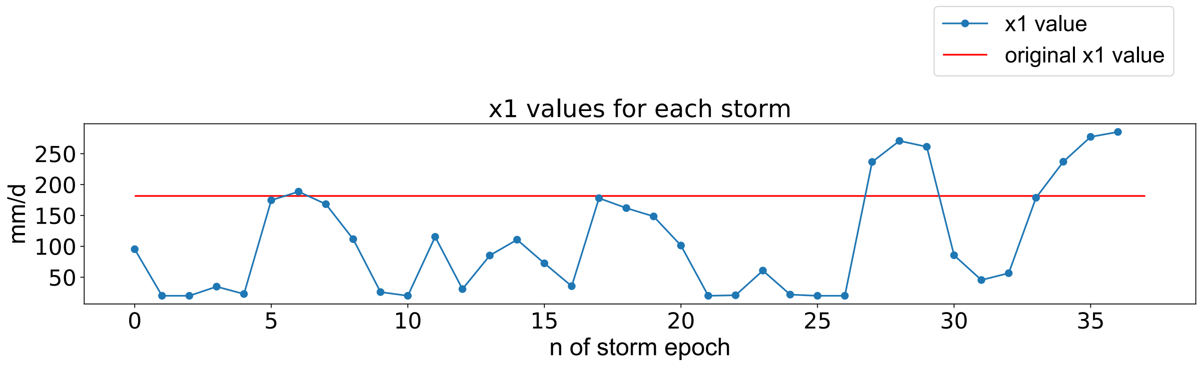

observed flow
90% uncertainty
storm-based validation with logscore = -1.07, RMSE= 1.47 using variable x1 and x4
discharge (mm/d)
time (d)
0
250
500
750
1000
1250
1500
1750
0
20
40
60
80

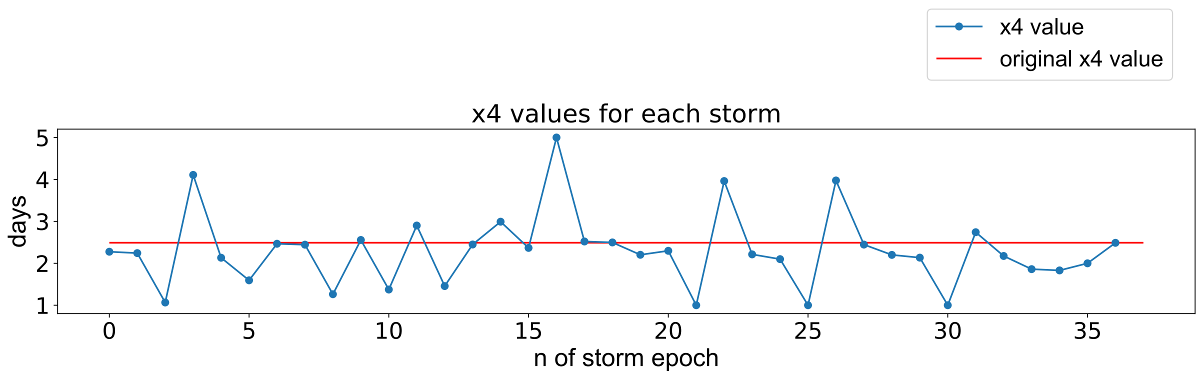
storm-based calibration & validation results for basin E

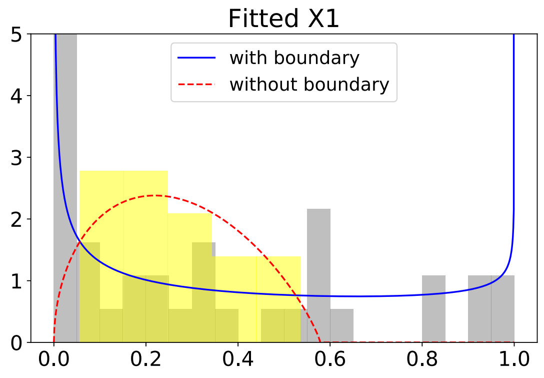
- with the threshold relative with the median value of rainfall series (in this case, threshod=0.1) to divide the storm epochs
- with variable X1 and X4

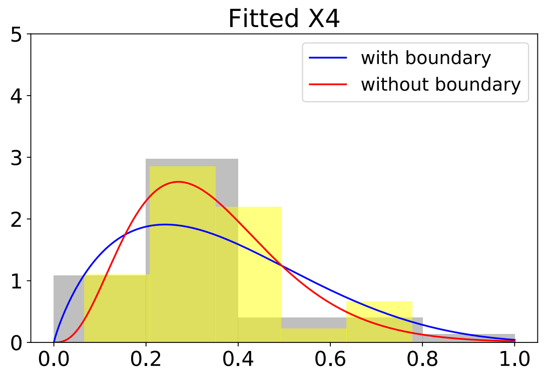


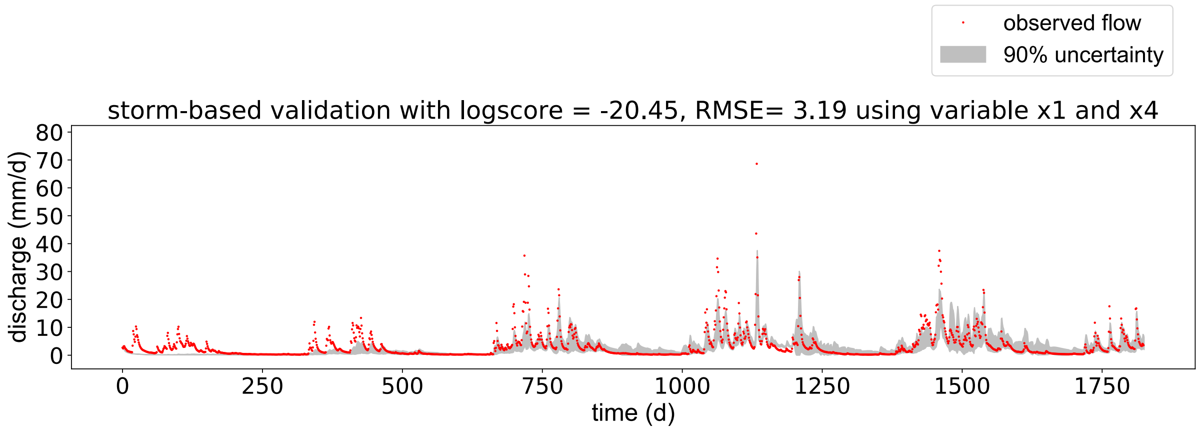





x4 value
original x4 value
x4 values for each storm
days
n of storm epoch

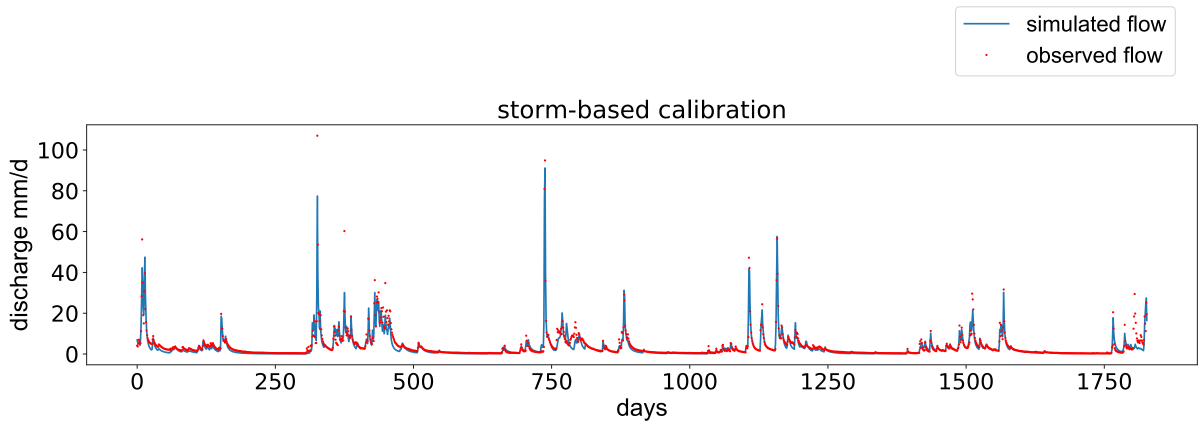

Fitted X1
with boundary
without boundary

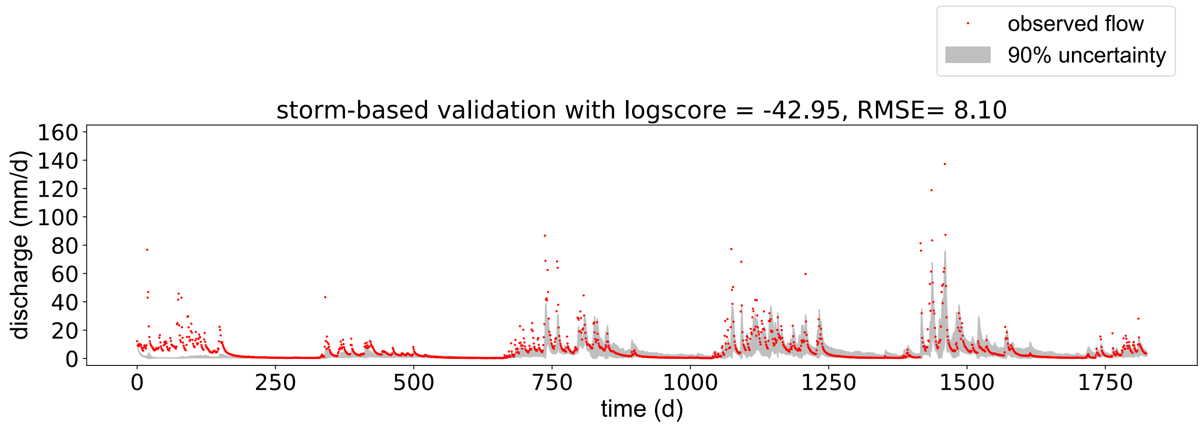

Fitted X4
with boundary
without boundary

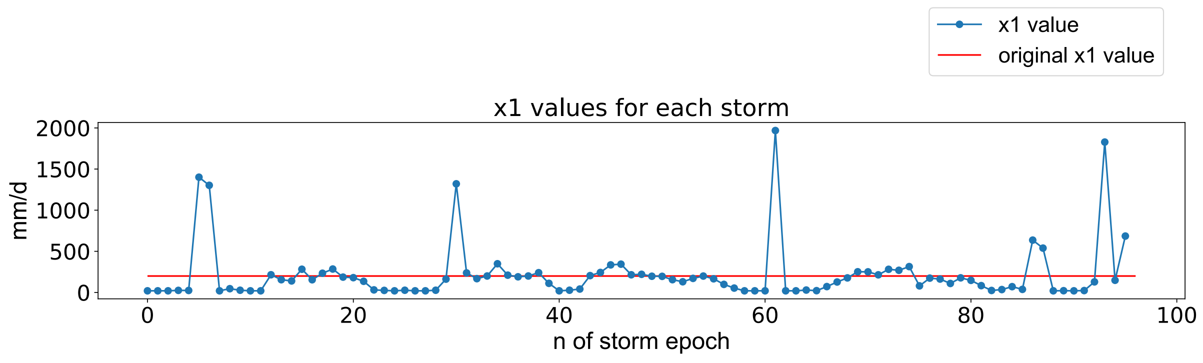

observed flow
90% uncertainty
storm-based validation with logscore = -20.45, RMSE= 3.19 using variable x1 and x4
discharge (mm/d)
time (d)

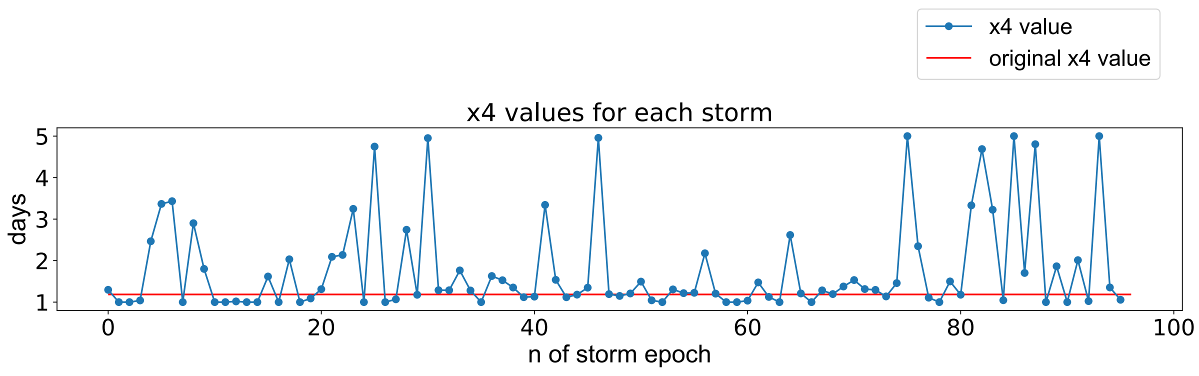
storm-based calibration & validation results for basin F

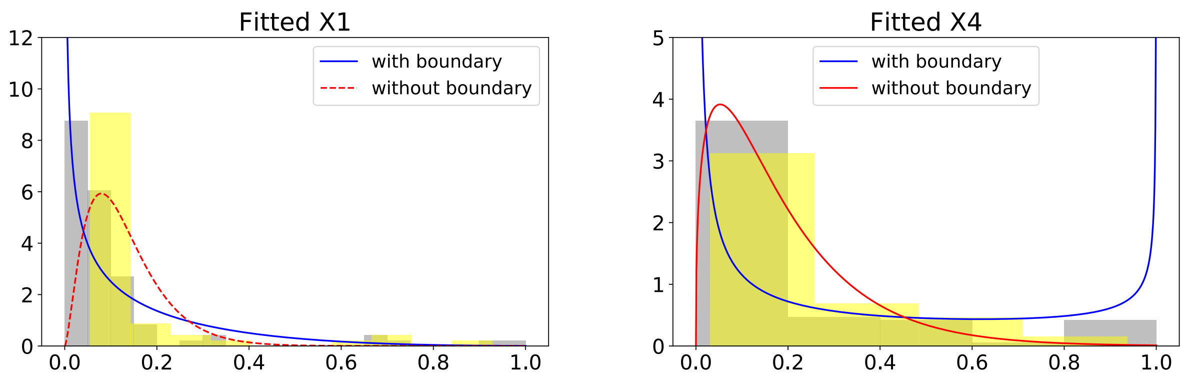
- with the threshold relative with the median value of rainfall series (in this case, threshod=0.19) to divide the storm epochs
- with variable X1 and X4

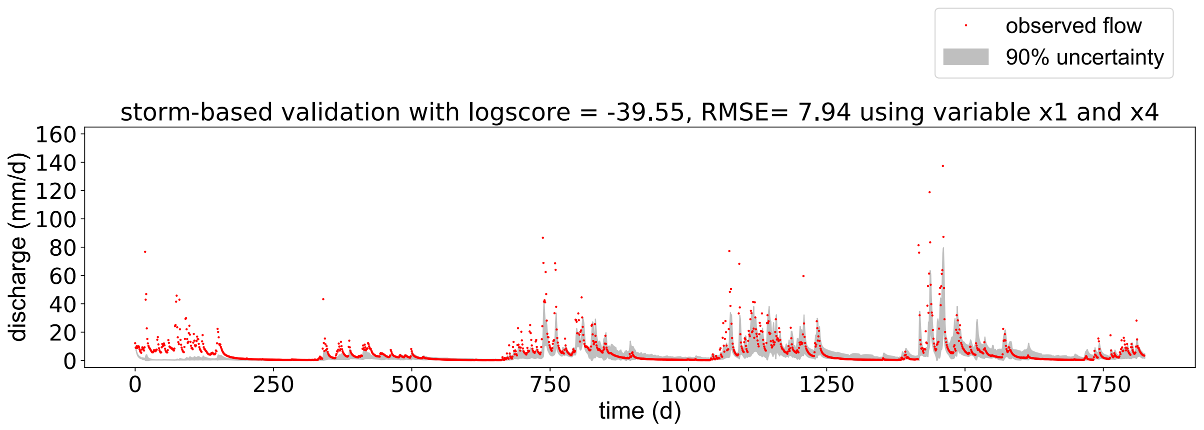







x4 value
original x4 value
x4 values for each storm
days
n of storm epoch


Fitted X1
with boundary
without boundary
Fitted X4
with boundary
without boundary


observed flow
90% uncertainty
storm-based validation with logscore = -39.55, RMSE= 7.94 using variable x1 and x4
discharge (mm/d)
time (d)

Calibration and validation resutls for GLUE method

Codes below are for basin A

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
from gr4j import gr4j#the rainfall-runoff model
from scipy.optimize import differential_evolution#the calibratio2an algorithm
from scipy.stats import norm#the normal distribution
import pandas as pd
def G(r,ep,p):
    params={ 'X1': p[0], 'X2': p[1], 'X3': p[2], 'X4': p[3] }
    states= { 'production_store': 0.60 * params['X1'], 'routing_store': 0.70 * param
    Qm = gr4j(r, ep, params,states)
    return Qm
font1 = {'family' : 'Arial',
        'weight' : 'normal',
        'size' : 20,}
font2 = {'family' : 'Arial',
        'weight' : 'normal',
        'size' : 22,}
num1=1.02
num2=1.02
num3=4
num4=2
```

In [ ]:

```
```

In [2]:

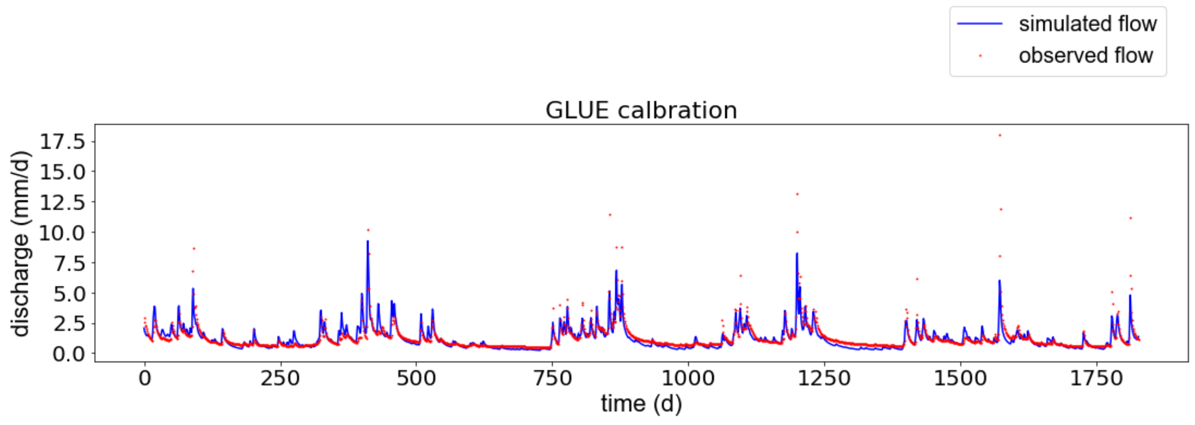
```
def calibration_validation_MC(N):
    A=N
    def forcingdata(A):
        data = np.loadtxt(A,dtype=str,delimiter='\t')
        data = [data[ii][0:4] + ' ' + data[ii][4:6] + ' '  + data[ii][6:8] + data[ii
                for ii in np.arange(len(data))]
        data = [data[ii].split() for ii in np.arange(len(data))]
        data = pd.DataFrame(data,columns =['y','m','d','r','ep','s','t1','t2'])
        discharge=np.array(data['s'][14610:16438],dtype=float)
        rainfall=np.array(data['r'][14610:16438],dtype=float)
        ep=np.array(data['ep'][14610:16438],dtype=float)
        data.head()
        return discharge,rainfall,ep
    d_o=forcingdata(A)[0]
    r_o=forcingdata(A)[1]
    p_o=forcingdata(A)[2]
    def validationdata(A):
        data = np.loadtxt(A,dtype=str,delimiter='\t')
        data = [data[ii][0:4] + ' ' + data[ii][4:6] + ' '  + data[ii][6:8] + data[ii
                for ii in np.arange(len(data))]
        data = [data[ii].split() for ii in np.arange(len(data))]
        data = pd.DataFrame(data,columns =['y','m','d','r','ep','s','t1','t2'])
        discharge=np.array(data['s'][16438:18263],dtype=float)
        rainfall=np.array(data['r'][16438:18263],dtype=float)
        ep=np.array(data['ep'][16438:18263],dtype=float)
        data.head()
        return discharge,rainfall,ep
    discharge=validationdata(A)[0]
    rainfall=validationdata(A)[1]
    potential_evap=validationdata(A)[2]
    Par_min=np.array([  20, -5 , 20 , 1 ])
    Par_max=np.array([  2000, 3 , 300 , 5])
    def MC(Par_min,Par_max):
        nmax=10000
        A=np.zeros((nmax,5))
        n_feasible=0
        for i in range(1,nmax):
            Rnum=np.random.rand(4)
            Par =Par_min +(Par_max-Par_min)*Rnum
            Qm=np.array(G(r_o, p_o,Par))
            if np.isreal(Qm.all()):
                Qe=d_o
                QeAv=np.mean(Qe)
                ErrUp=np.sum((Qm-Qe)**2)
                ErrDo = np.sum((Qe-QeAv)**2)
                Obj = 1- ErrUp / ErrDo
            if Obj>0.6:
                A[n_feasible,0:4]=Par
                A[n_feasible,4]=Obj
                n_feasible=n_feasible+1
        return n_feasible,A
    n_feasible,A=MC(Par_min,Par_max)
    ind=np.argmax(A[:,4])
    Qsim=np.array(G(r_o, p_o, A[ind,0:4]))
    #%%
    plt.figure(figsize=(18,4))
    plt.plot(Qsim,'b-',markersize=2,label='simulated flow')
    plt.plot(d_o,'r.',markersize=2,label='observed flow')
    plt.title('GLUE calbration',fontsize=22)
    plt.xlabel('time (d)',font2)
    plt.ylabel('discharge (mm/d)',font2)
    plt.xticks(fontsize=20)
    plt.yticks(fontsize=20)
    plt.legend(bbox_to_anchor=(num1, num2), loc=num3, borderaxespad=num4,prop=font1)
    plt.savefig('A_G_c.pdf', bbox_inches='tight')
    B=np.zeros((n_feasible,5))
    QM=np.zeros((len(B),len(discharge)))
    sigma=np.zeros(len(discharge))
    QM_5=np.zeros(len(discharge))
    QM_95=np.zeros(len(discharge))
    logscore1=np.zeros(len((discharge)))
    RMSE1=np.zeros(len(discharge))
    miu=np.zeros(len(discharge))
    sig=np.zeros(len(discharge))
    for i in range(n_feasible):
        B[i,:]=A[i,:]
    for i in range(n_feasible):
        QM[i,:]=np.array(G(rainfall, potential_evap,B[i,0:4]))
    for j in range(len(discharge)):
        sigma[j]=( np.sum( ( QM[:,j]-np.mean(QM[:,j])) **2)/len(B) ) **(1/2)
    for i in range(len(discharge)):
        miu[i],sig[i]=norm.fit(QM[:,i])
        QM_5[i] = norm.ppf(0.05,miu[i],sig[i])
        QM_95[i] =norm.ppf(0.95,miu[i],sig[i])
    QM_5[QM_5<0]=0
    for i in range(len(discharge)):
        Vobs=0
        Vpredict1= sig[i]
        Mobs1=discharge[i]
        Mpredict1=np.mean(QM[:,i])
        RMSE1[i]= (Mobs1-Mpredict1)**2
        logscore1[i]=-1/2*np.log(2*np.pi)-1/2*np.log(Vobs+Vpredict1) -(Mobs1-Mpredic
    logscore1_av=(np.mean(logscore1))
    RMSE_av1=(np.sum(RMSE1)/len(discharge))**0.5
    plt.figure(figsize=(18,4))
    plt.fill_between(np.linspace(1,len(discharge),len(discharge)),QM_5,QM_95,alpha=0
    plt.plot(discharge,'r.',markersize=1.3,label='observed flow')
    plt.title('GLUE validation with logscore= '+str(format(logscore1_av,'.2f'))+' RM
    plt.xlabel('time (d)',font2)
    plt.ylabel('discharge (mm/d)',font2)
    plt.xticks(fontsize=20)
    plt.yticks(fontsize=20)
    plt.legend(bbox_to_anchor=(num1, num2), loc=num3, borderaxespad=num4,prop=font1)
    plt.savefig('A_G_v.pdf', bbox_inches='tight')
    plt.ylim(-4,np.max(discharge)*1.2)
    hflow=[]
    hindex=[]
    lflow=[]
    lindex=[]
    for m in range(len(rainfall)):
        if rainfall[m]<5:
            lindex.append(m)
            lflow.append(discharge[m])
        else:
            hindex.append(m)
            hflow.append(discharge[m])
    hflow=np.array(hflow)
    lflow=np.array(lflow)
    hindex=np.array(hindex)
    lindex=np.array(lindex)
```

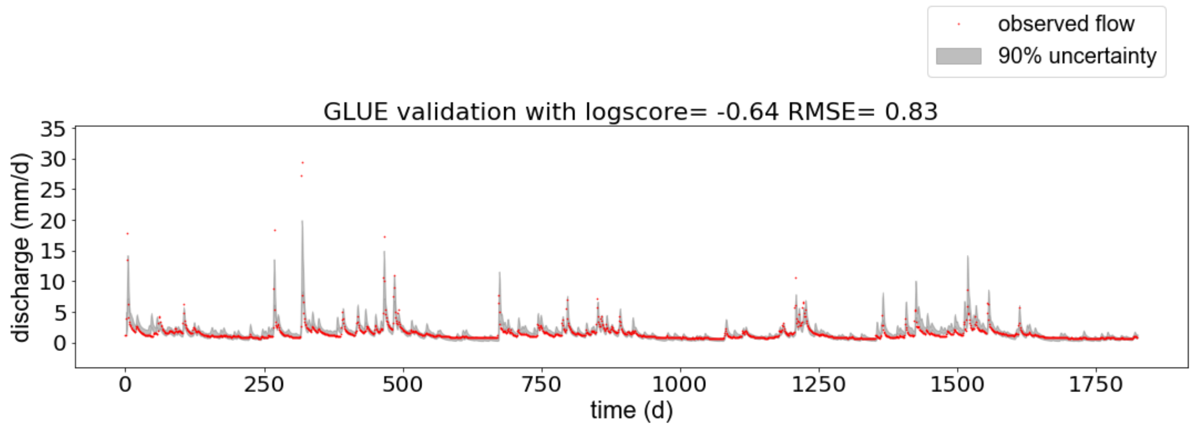
```
    hrmse=np.zeros(len(hindex))
    lrmse=np.zeros(len(lindex))
    for i in range(len(hindex)):
        n=hindex[i]
        hrmse[i]=(hflow[i]-np.mean(QM[:,n]))**2
    Hrmse1=(np.sum(hrmse)/len(hflow) )**(1/2)
    for i in range(len(lindex)):
        n=lindex[i]
        lrmse[i]=(lflow[i]-np.mean(QM[:,n]))**2
    Lrmse1=(np.sum(lrmse)/len(lflow) )**(1/2)
    print('for basin A:',Hrmse1,Lrmse1)
    return
```

In [3]:

```
calibration_validation_MC('07068000.dly') #A
```

```
for basin A: 1.0755484333691538 0.7658080626628302
```





In [4]:

```
#calibration_validation_MC('1664000.dly') #B
```

In [5]:

```
#calibration_validation_MC('3164000.dly') #C
```

In [6]:

```
#calibration_validation_MC('3161000.dly') #D
```

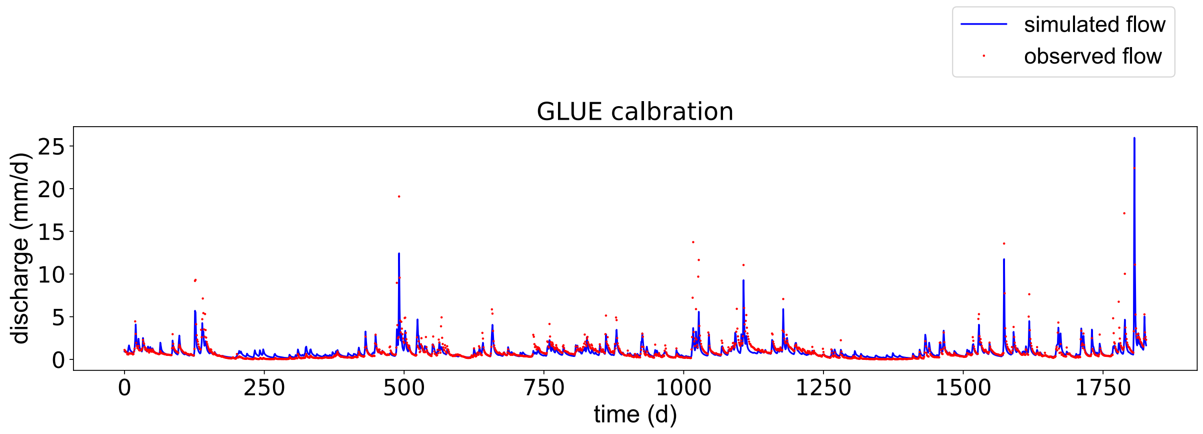
In [7]:

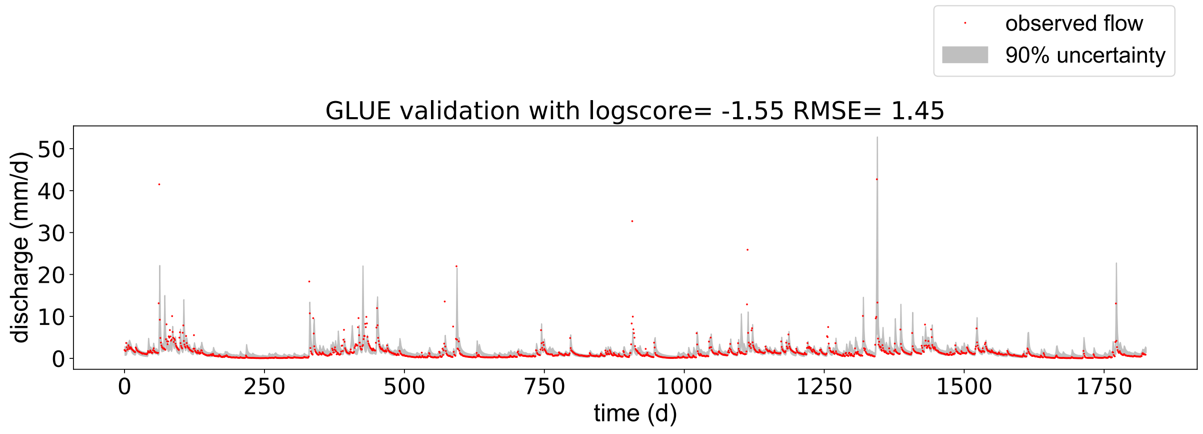
```
#calibration_validation_MC('12027500.dly') #E
```

In [8]:

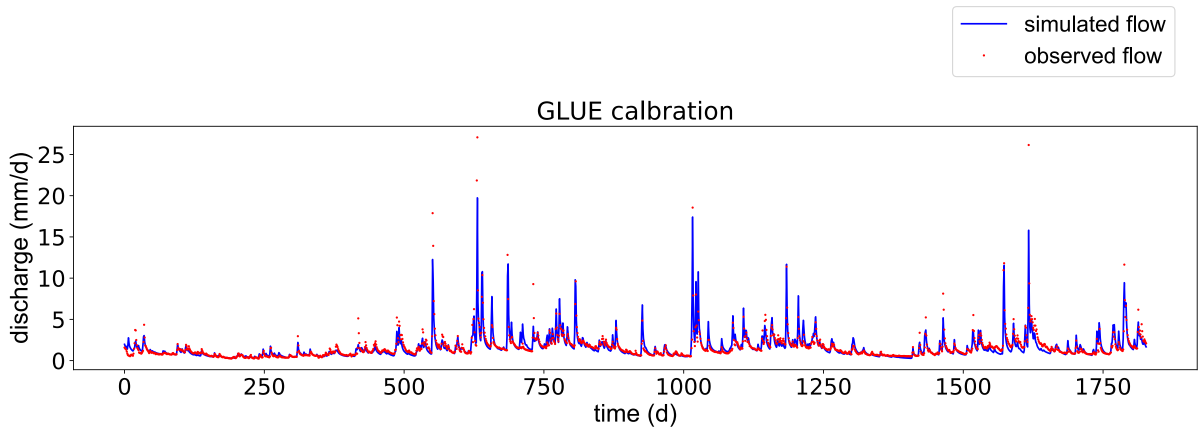
```
#calibration_validation_MC('11532500.dly') #F
```

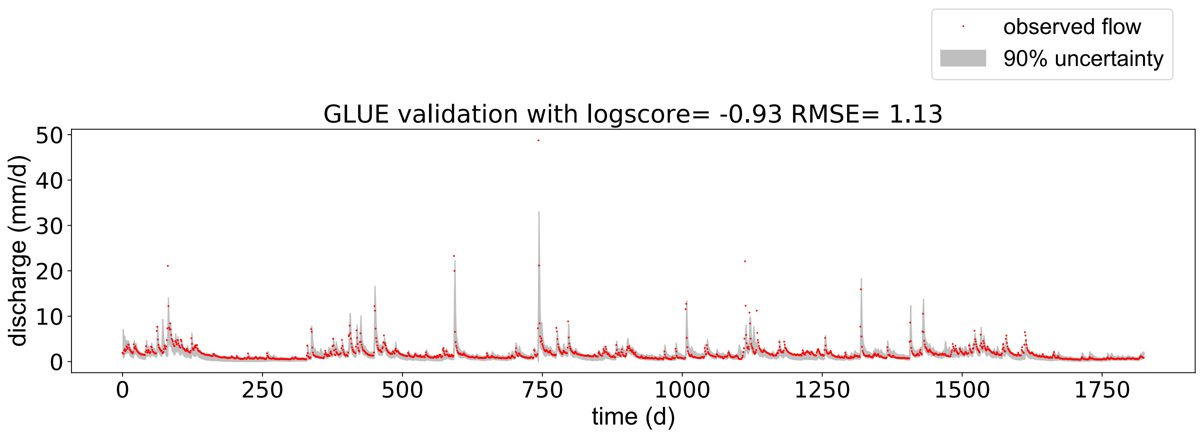
GLUE calibration & validation results for basin B



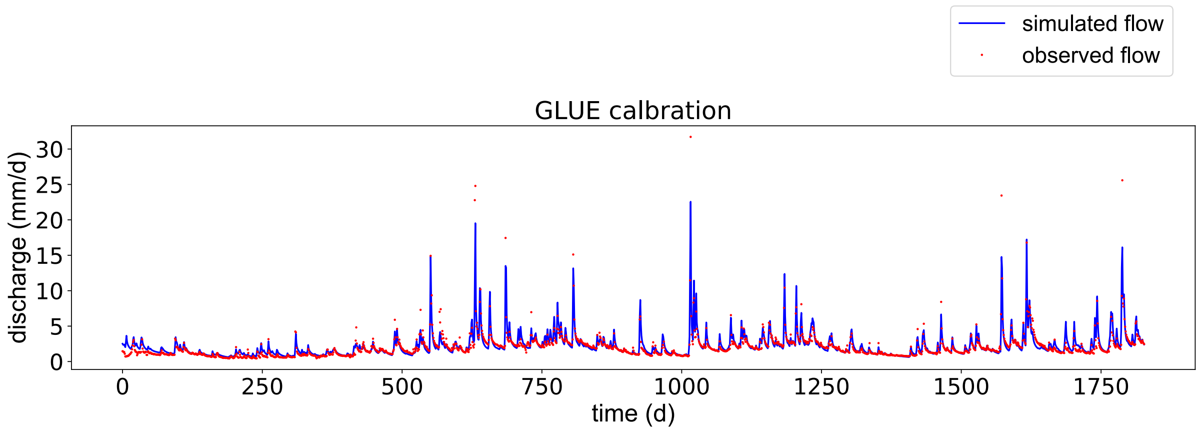


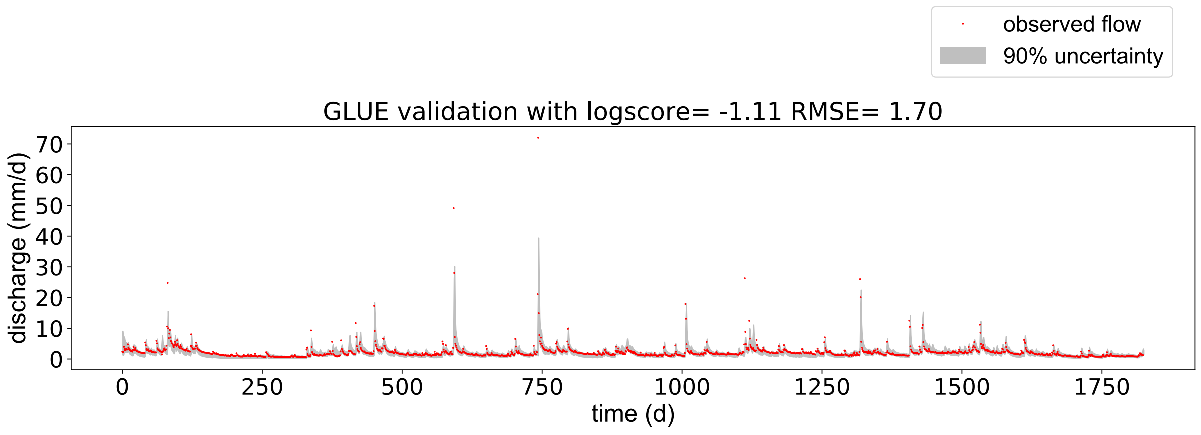
GLUE calibration & validation results for basin C



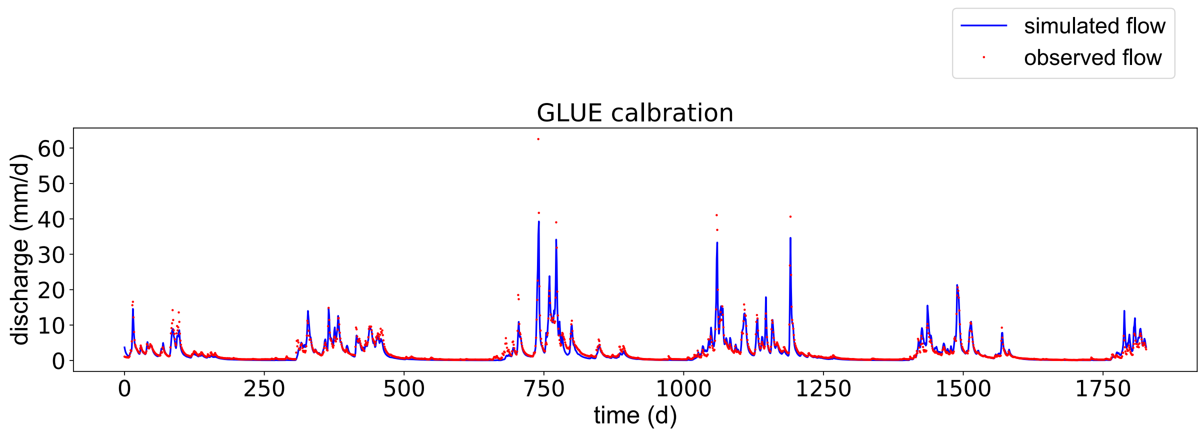


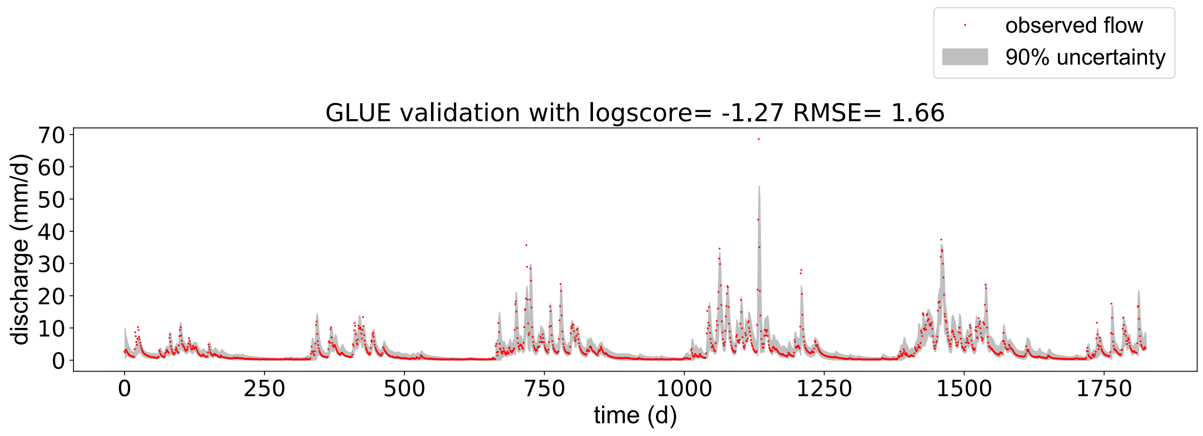
GLUE calibration & validation results for basin D



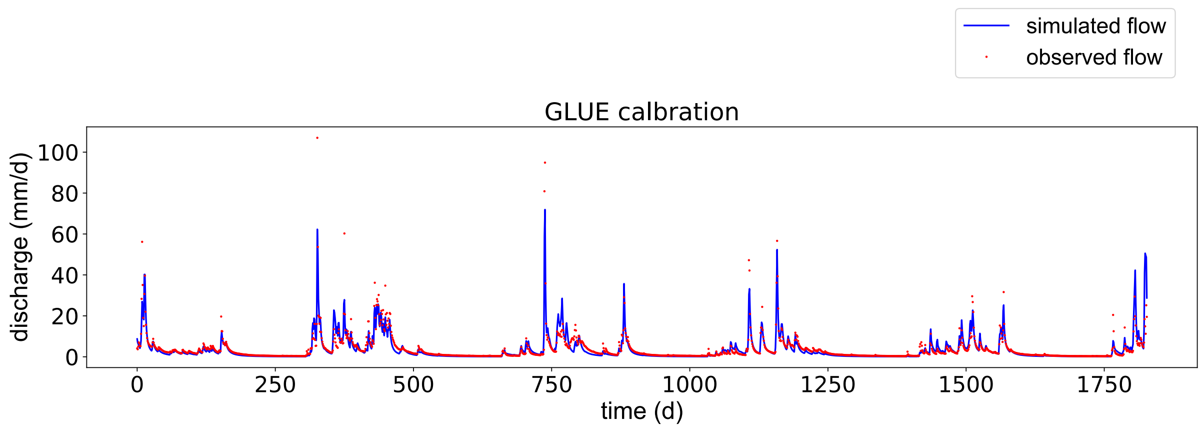


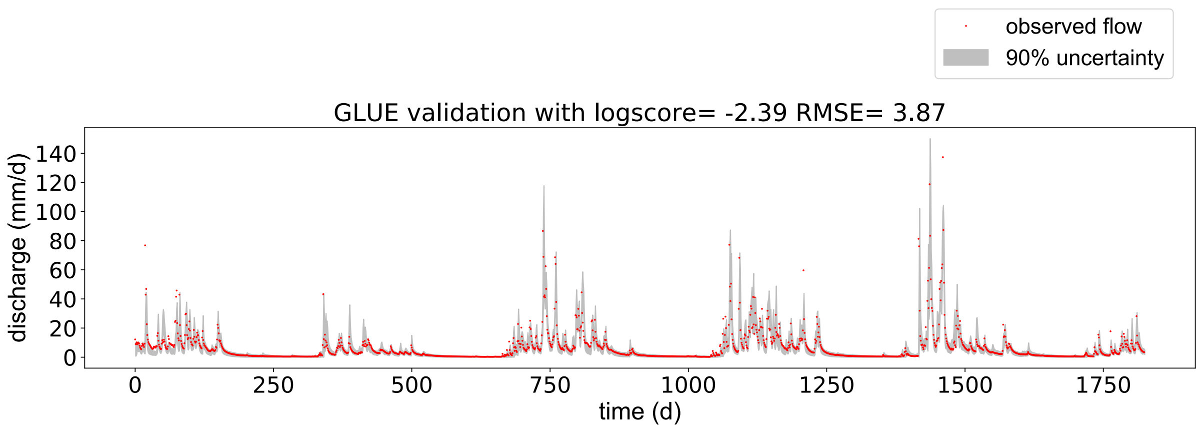
GLUE calibration & validation results for basin E





GLUE calibration & validation results for basin F




observed flow
90% uncertainty
GLUE validation with logscore= -2.39 RMSE= 3.87
discharge (mm/d)
time (d)
0
20
40
60
80
100
120
140
0
250
500
750
1000
1250
1500
1750

Calibration and validation results obtained by traditiional method.

the codes beblow are for basin A

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
from gr4j import gr4j#the rainfall-runoff model
from scipy.optimize import differential_evolution#the calibratio2an algorithm
from scipy.stats import norm#the normal distribution
import pandas as pd
font1 = {'family' : 'Arial',
    'weight' : 'normal',
    'size' : 20,}
font2 = {'family' : 'Arial',
    'weight' : 'normal',
    'size' : 22,}
num1=1.02
num2=1.02
num3=4
num4=2
```

In [ ]:

```

```

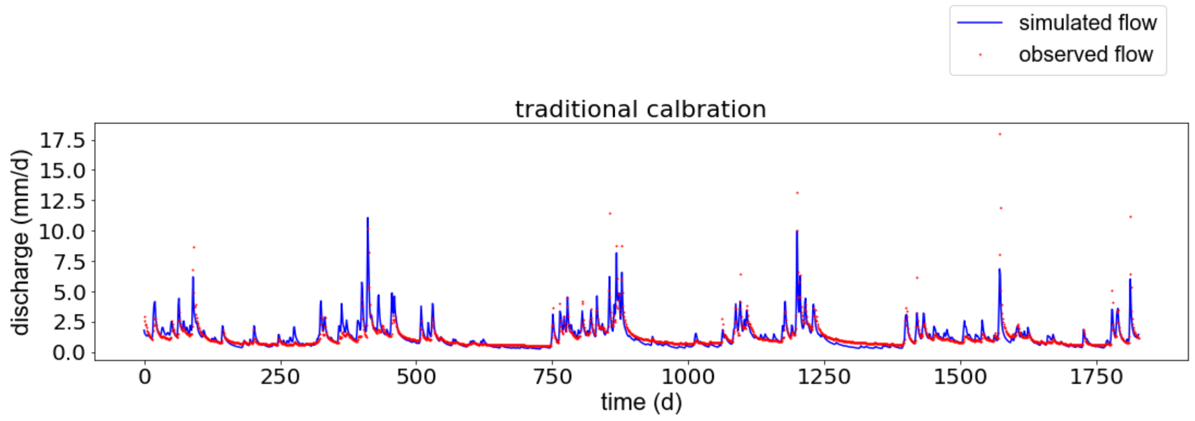
In [2]:

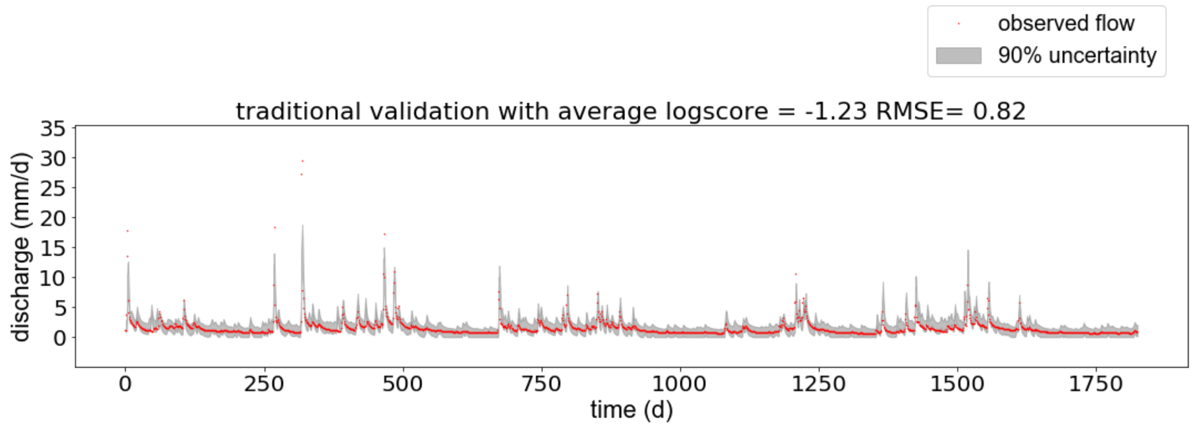
```
def tra_method(N):
    A=N
    def G(r,ep,p):
        params={ 'X1': p[0], 'X2': p[1], 'X3': p[2], 'X4': p[3] }
        states= { 'production_store': 0.60 * params['X1'], 'routing_store': 0.70 * p
        Qm = gr4j(r, ep, params,states)
        return Qm
    def forcingdata(A):
        data = np.loadtxt(A,dtype=str,delimiter='\t')
        data = [data[ii][0:4] + ' ' + data[ii][4:6] + ' '  + data[ii][6:8] + data[ii
                for ii in np.arange(len(data))]
        data = [data[ii].split() for ii in np.arange(len(data))]
        data = pd.DataFrame(data,columns =['y','m','d','r','ep','s','t1','t2'])
        discharge=np.array(data['s'][14610:16438],dtype=float)
        rainfall=np.array(data['r'][14610:16438],dtype=float)
        ep=np.array(data['ep'][14610:16438],dtype=float)
        data.head()
        return discharge,rainfall,ep
    d_o=forcingdata(A)[0]
    r_o=forcingdata(A)[1]
    p_o=forcingdata(A)[2]
    def validationdata(A):
        data = np.loadtxt(A,dtype=str,delimiter='\t')
        data = [data[ii][0:4] + ' ' + data[ii][4:6] + ' '  + data[ii][6:8] + data[ii
                for ii in np.arange(len(data))]
        data = [data[ii].split() for ii in np.arange(len(data))]
        data = pd.DataFrame(data,columns =['y','m','d','r','ep','s','t1','t2'])
        discharge=np.array(data['s'][16438:18263],dtype=float)
        rainfall=np.array(data['r'][16438:18263],dtype=float)
        ep=np.array(data['ep'][16438:18263],dtype=float)
        data.head()
        return discharge,rainfall,ep
    discharge=validationdata(A)[0]
    rainfall=validationdata(A)[1]
    potential_evap=validationdata(A)[2]
    def obj(B):  #run for the whole period to find the value of constant parameter X
        Qsim=np.array(G(r_o,p_o,B))
        Qobs=d_o
        QM=Qsim
        ErrUp=np.sum((QM-Qobs)**2)
        ErrDo = np.sum((Qobs-np.mean(Qobs))**2)
        NSE = 1- ErrUp / ErrDo
        return -NSE
    Bounds= [(20 ,2000),(-5,3),(20,300),(1,5)]
    result1= differential_evolution(obj, Bounds)
    Qm=np.array(G(r_o, p_o, result1.x[0:4]))
    plt.figure(figsize=(18,4))
    plt.plot(Qm,'b-',markersize=2,label='simulated flow')
    plt.plot(d_o,'r.',markersize=2,label='observed flow')
    plt.title('traditional calbration',fontsize=22)
    plt.xlabel('time (d)',font2)
    plt.ylabel('discharge (mm/d)',font2)
    plt.xticks(fontsize=20)
    plt.yticks(fontsize=20)
    plt.legend(bbox_to_anchor=(num1, num2), loc=num3, borderaxespad=num4,prop=font1)
    plt.savefig('A_t_c.pdf', bbox_inches='tight')
    hflow=[]
    hindex=[]
    lflow=[]
    lindex=[]
    for m in range(len(rainfall)):
        if rainfall[m]<5:
            lindex.append(m)
            lflow.append(discharge[m])
        else:
            hindex.append(m)
            hflow.append(discharge[m])
    hflow=np.array(hflow)
    lflow=np.array(lflow)
    hindex=np.array(hindex)
    lindex=np.array(lindex)
    hrmse=np.zeros(len(hindex))
    lrmse=np.zeros(len(lindex))
    Qsim=np.array(G(rainfall, potential_evap, result1.x[0:4]))
    sigma=(np.sum((Qm-d_o)**2)/len(d_o))**(1/2)
    Qm1_5=np.zeros(len(discharge))
    Qm1_95=np.zeros(len(discharge))
    logscore1=np.zeros(len((discharge)))
    RMSE=np.zeros(len(discharge))
    for i in range(len(discharge)):
        Qm1_5[i] = norm.ppf(0.05,Qsim[i],sigma)
        Qm1_95[i] =norm.ppf(0.95,Qsim[i],sigma)
    Qm1_5[Qm1_5<0]=0
    Qm1_95[Qm1_95<0]=0
    for i in range(len(discharge)):
        Vobs=0
        Vpredict1= sigma
        Mobs1=discharge[i]
        Mpredict1=Qsim[i]
        RMSE[i]= (Mobs1-Mpredict1)**2
        logscore1[i]=-1/2*np.log(2*np.pi)-1/2*np.log(Vobs+Vpredict1) -(Mobs1-Mpredic
    logscore1_av=(np.mean(logscore1))
    RMSE_av1=(np.sum(RMSE)/len(discharge))**0.5
    for i in range(len(hindex)):
        n=hindex[i]
        hrmse[i]=(hflow[i]-Qsim[n])**2
    Hrmse1=(np.sum(hrmse)/len(hflow) )**(1/2)
    for i in range(len(lindex)):
        n=lindex[i]
        lrmse[i]=(lflow[i]-Qsim[n])**2
    Lrmse1=(np.sum(lrmse)/len(lflow) )**(1/2)
    print('for basin A',Hrmse1,Lrmse1)
    plt.figure(figsize=(18,4))
    plt.fill_between(np.linspace(1,len(discharge),len(discharge)),Qm1_5,Qm1_95,alpha
    plt.plot(discharge,'r.',markersize=1,label='observed flow')
    plt.title('traditional validation with average logscore = '+str(format(logscore1
    plt.xlabel('time (d)',font2)
    plt.ylabel('discharge (mm/d)',font2)
    plt.xticks(fontsize=20)
    plt.yticks(fontsize=20)
    plt.legend(bbox_to_anchor=(num1, num2), loc=num3, borderaxespad=num4,prop=font1)
    plt.savefig('A_t_v.pdf', bbox_inches='tight')
    plt.ylim(-4.9,np.max(discharge)*1.2)
    return
```

In [3]:

```
tra_method('07068000.dly') #basin A
```

```
for basin A 0.9394200940416538 0.7978684692326611
```





In [13]:

```
#tra_method('1664000.dly') #basin B
```

In [12]:

```
#tra_method('3164000.dly') #basin C
```

In [11]:

```
#tra_method('3161000.dly') #basin D
```

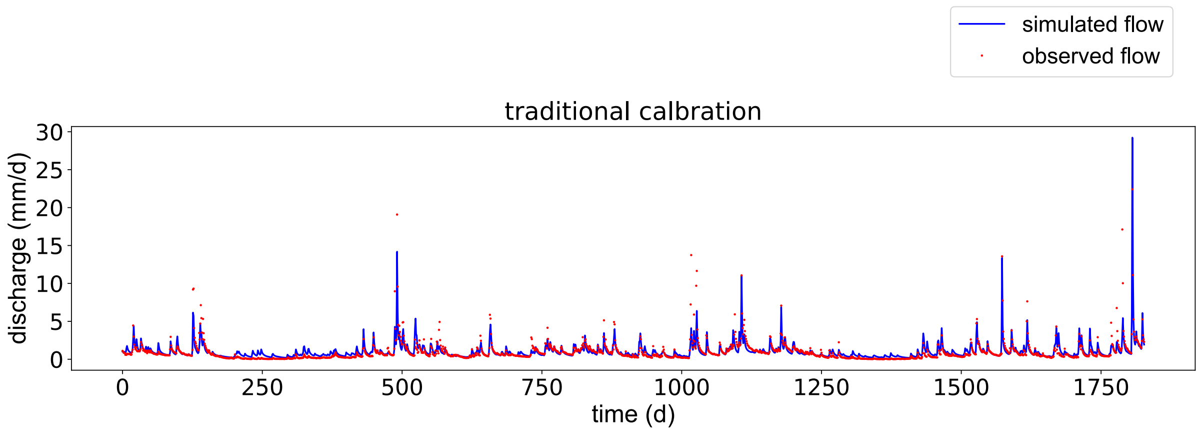
In [10]:

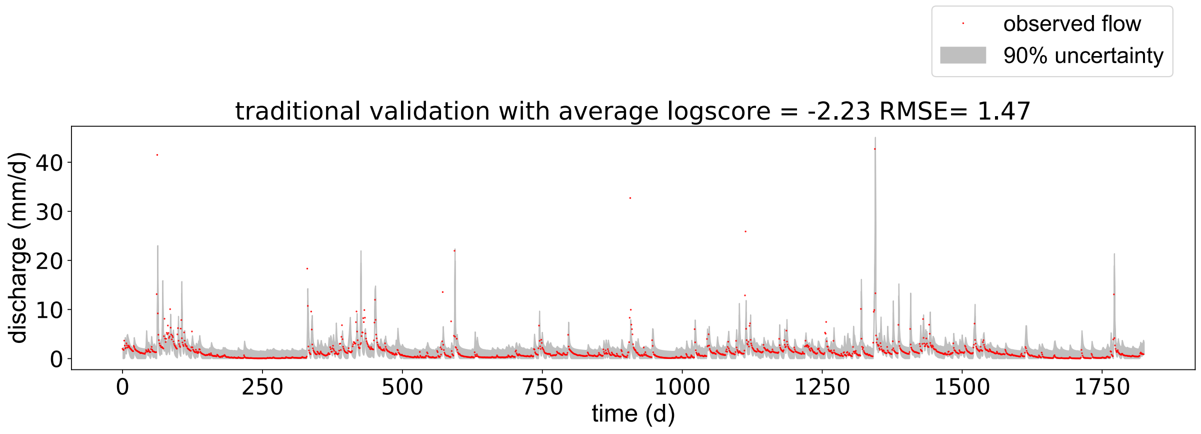
```
#tra_method('12027500.dly') #basin E
```

In [9]:

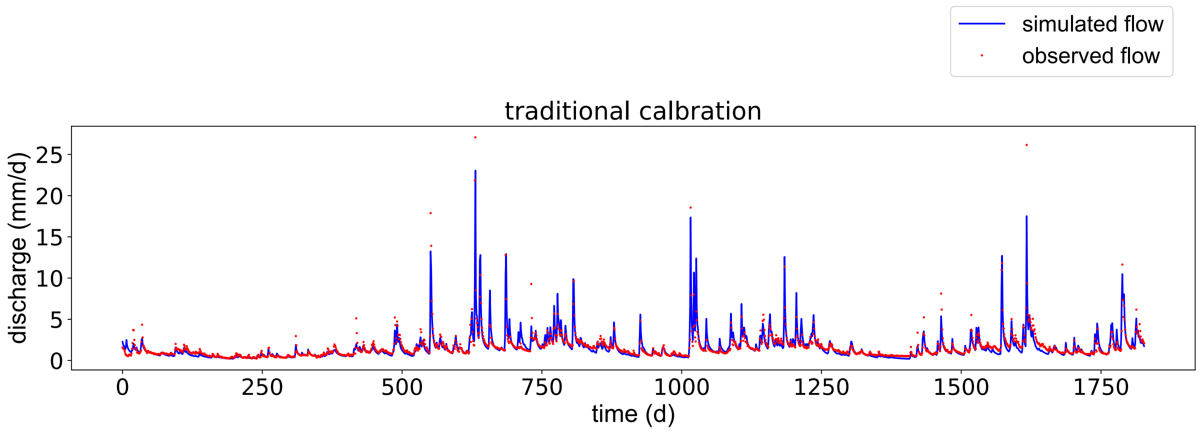
```
#tra_method('11532500.dly') #basin F
```

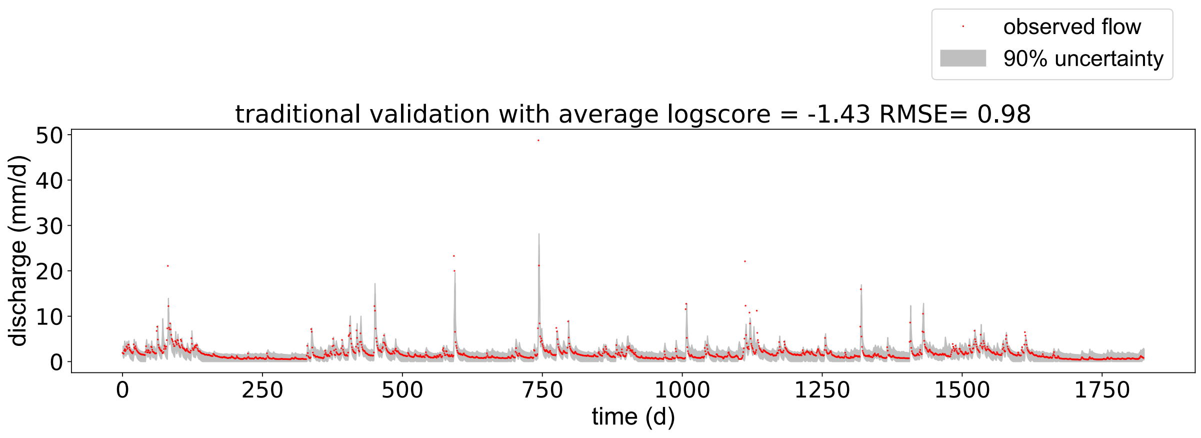
traditional calibration & validation results for basin B



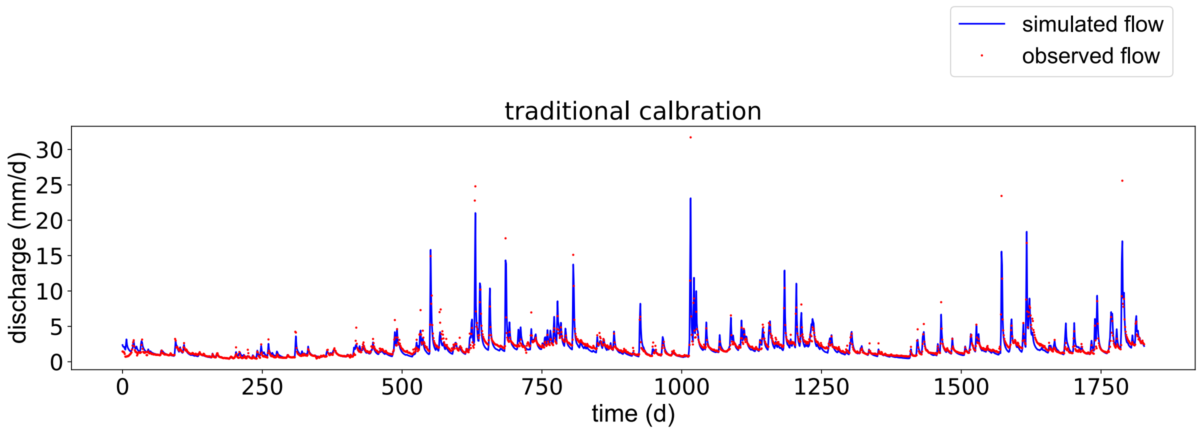


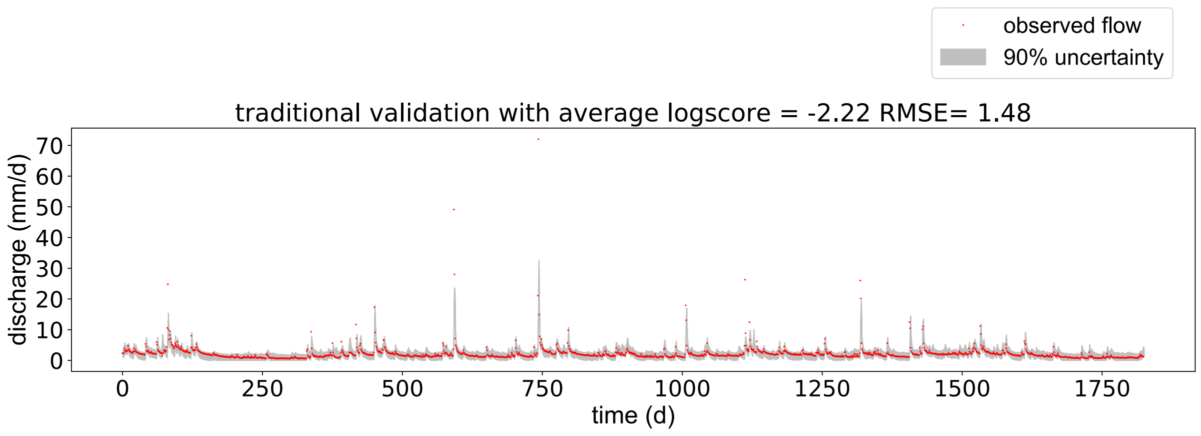
traditional calibration & validation results for basin C



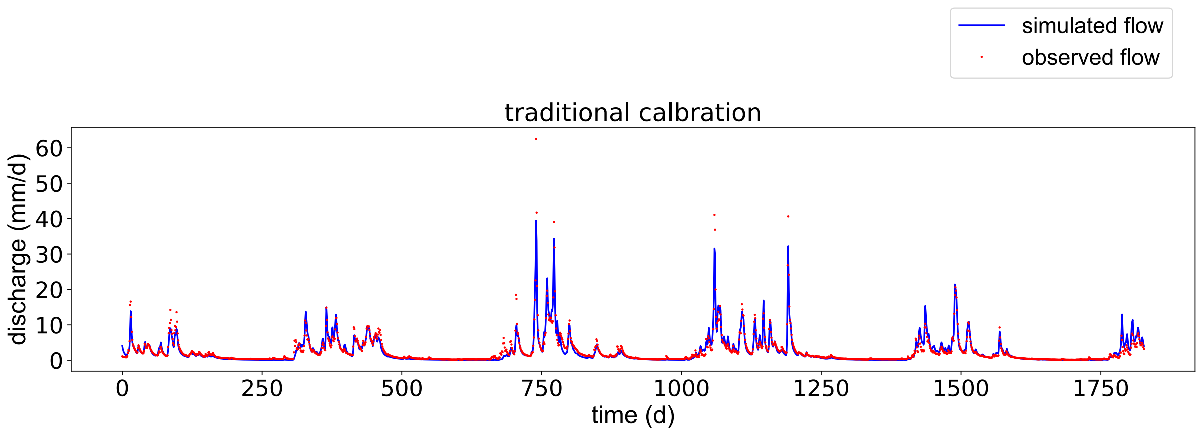


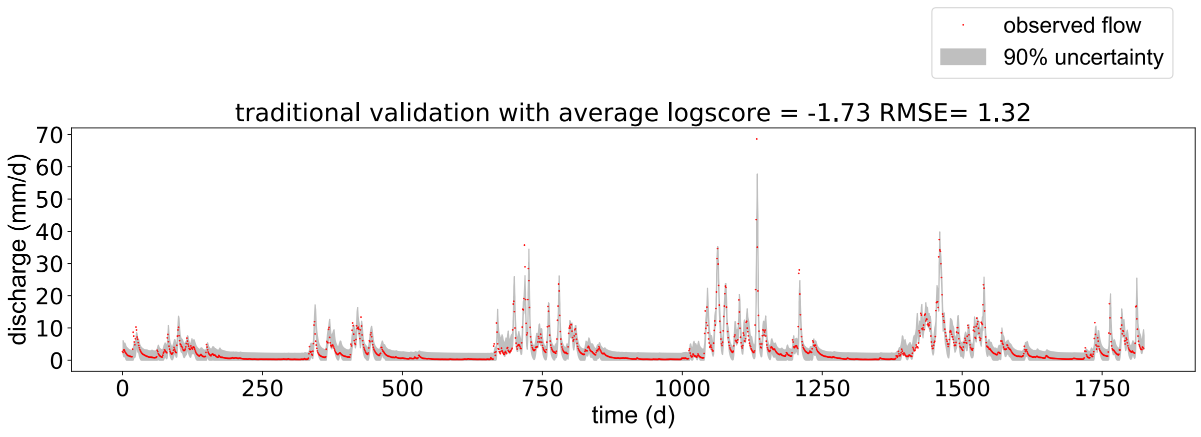
traditional calibration & validation results for basin D



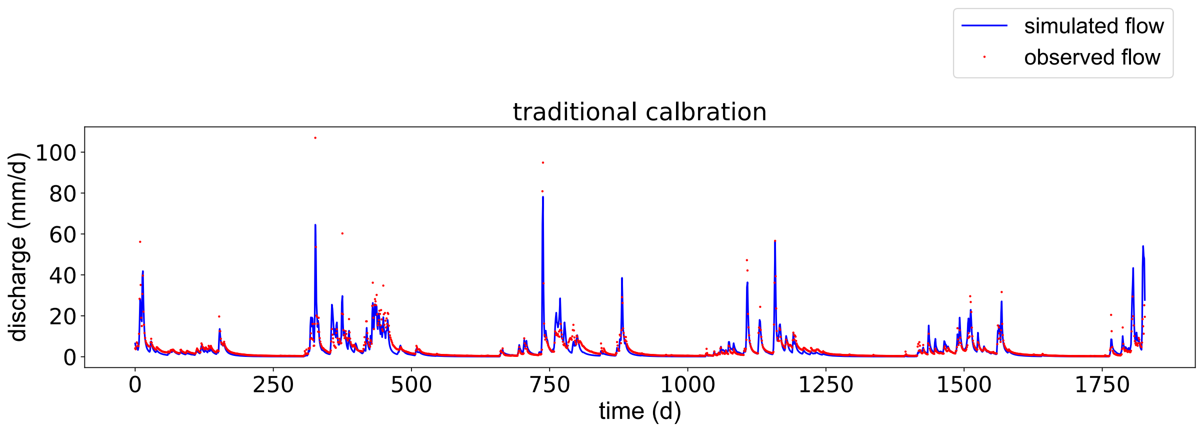


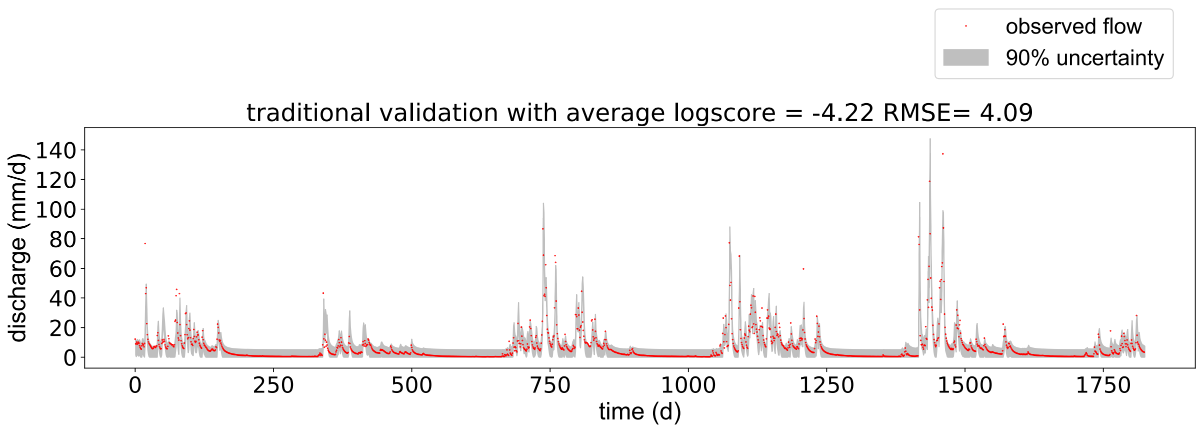
traditional calibration & validation results for basin E





traditional calibration & validation results for basin F




observed flow
90% uncertainty
traditional validation with average logscore = -4.22 RMSE= 4.09
discharge (mm/d)
0
20
40
60
80
100
120
140
0
250
500
750
1000
1250
1500
1750
time (d)