Surface and sub-surface flow estimation at high temporal resolution using deep neural networks

Ather Abbas, Sangsoo Baek, Minjeong Kim, Mayzonee Ligaray, Olivier Ribolzi, Norbert Silvera, Joong-Hyuk Min, Laurie Boithias, Kyung Hwa Cho

Surface and sub-surface flow estimation at high temporal resolution using deep neural networks

Ather Abbas [1*], *Sangsoo Baek* [1*], *Minjeong Kim* [1], *Mayzonee Ligaray* [1], *Olivier Ribolzi* [2], *Norbert Silvera* [3], *Joong-Hyuk Min*[4], *Laurie Boithias* [2†], *Kyung Hwa Cho* [1†]

[1] School of Urban and Environmental Engineering, Ulsan National Institute of Science and Technology, Ulsan, 689-798, Republic of Korea

[2] Geosciences Environnement Toulouse, Université de Toulouse, CNRS, IRD, UPS, 31400 Toulouse, France

[3] iEES-Paris, UMR 7618 (IRD, CNRS, UPMC), Centre IRD d'Ile de France – 32, avenue Henri Varagnat, 93143 Bondy cedex, France

[4] Water Quality Assessment Research Division, National Institute of Environmental Research, Environmental Research Complex, Hwangyeong-ro 42, Seo-gu, Incheon 22689, Korea

[*] *Co-first authors:* Ather Abbas, Sangsoo Baek

[†] *Corresponding author: Kyung Hwa Cho (*khcho@unist.ac.kr*), Laurie Boithias (*laurie.boithias@get.omp.eu*)*

Manuscript for

Journal of Hydrology

Abstract

Recent intensification in climate change have resulted in the rise of hydrological extreme events. This demands modeling of hydrological processes at high temporal resolution to better understand flow patterns in catchments. To model surface and sub-surface flows in a catchment we utilized a physically based model called Hydrological Simulated Program-FORTRAN and two deep learning-based models. One deep learning model consisted of only one long short-term memory (simple LSTM), whereas the other model simulated processes in each hydrological response unit (HRU) by defining one separate LSTM for each HRU (HRU-based LSTM). The models use environmental time-series data and two-dimensional spatial data to predict surface and sub-surface flows at 6-minute time step simultaneously. We tested our models in a tropical humid headwater catchment in northern Lao PDR and compared their performances. Our results showed that the simple LSTM model outperformed the other models on surface runoff prediction with the lowest MSE (7.4e-5 m^3s^{-1}), whereas HRU-based LSTM model better predicted patterns and slopes in sub-surface flow in comparison with the other models by having the smallest MSE value (3.2e-4 m^3s^{-1}). This study demonstrated the performance of a deep learning model when simulating hydrological cycle with high temporal resolution.

KEYWORDS: Deep learning model, Long short-term memory (LSTM), Sub-surface flow, Surface runoff, Hydrological Simulated Program-FORTRAN

1 Introduction

Recently, increase in precipitation extreme events has been witnessed at global scale (Papalexiou and Montanari, 2019). Roxy et al., (2017) documented threefold increase in rainfall events from 1950 to 2017 in central India. The International Disaster Database noted global annual loss of over $30 billion as a result of floods in last decade (Roxy et al., 2017). To understand the hydrological complexity of flash floods, we need to model hydrological phenomena at sub-daily time-step (Jodar-Abellan et al., 2019). Reynolds et al., (2017) has also stressed the need for streamflow prediction at a sub-daily time-step for flood forecasting in medium-sized (10-1000 Km^2) catchments due to small concentration time. The identification of water components contributing to total streamflow plays a key role in understanding bio-geochemical cycles and transport processes at catchment scale (Burns and Kendall, 2002). Accurate estimation of sub-surface water flow is critical in time-continuous models in comparison with event-based models due to the dominance of groundwater in continuous rainfall-runoff models (Huang et al., 2016; Guse et al., 2014).

Several research groups have developed catchment-scale modeling tools, such as the Soil & Water Assessment Tool (SWAT) (Arnold and Fohrer, 2005), the Stormwater Management Model (SWMM) (Rossman, 2010), and the Hydrological Simulation Program-FORTRAN (HSPF) (Bicknell et al., 2001). However, most of these hydrological models are used for a single rainfall event or daily simulation because sub-hourly simulation is complex and time consuming (Bennett et al., 2016). Jeong et al., (2010) improved the SWAT model to simulate stream discharge at a sub-hourly time-step; however, some processes in their improved model, such as baseflow and lateral flow, are still modeled with daily time-step. The output of these processes modeled at daily time-step is then distributed equally at the sub-hourly time-step. This

62 improved SWAT model has been used to perform sub-daily rainfall-runoff simulations (Boithias
63 et al., 2017). Ficchì et al., (2016) modeled streamflow using a four-parameter lumped GR4
64 (modified from GR4J, which stands for modèle du *G*énie *R*ural à *4* paramètres *J*ournalier) model
65 at different time-steps ranging from 6 min to 1 day, and studied the impact of frequency on
66 model performance. However, their study did not model continuous streamflow, rather they
67 simulated individual storm events. Their study found a mixed model response, i.e., both increase
68 (0.7 to 0.8) and decrease (0.8 to 0.69) in Kling Gupta Efficiency (Gupta et al., 2009) for model
69 with increases in temporal resolution. Temporal resolution of input data also affects simulation
70 results in a catchment model. Numerous studies have shown that the use of input data with
71 higher temporal frequency improves model accuracy (Huang et al., 2019; Wang et al., 2009;
72 Pang et al., 2018). On the other hand, simulating model output at a shorter time-step causes a
73 reduction in model accuracy (Gassman et al., 2007; Boithias et al., 2017; Bressiani et al., 2015).
74 A data-driven model is considered as an alternative approach to overcome these
75 complexities, and such models have higher predictive accuracy (Pascual et al., 2013; Park et al.,
76 2019). Numerous studies have been conducted to simulate hydrological processes using neural
77 networks (NNs), which is one of the popular types of data-driven models (Ilunga and Stephenson,
78 2005; Ogwueleka and Ogwueleka, 2009). Several review papers (Besaw et al., 2010; Yaseen et
79 al., 2015; Mosavi et al., 2018) have shown that most data-driven rainfall-runoff models have
80 used daily time-steps and very few studies have been conducted using an hourly time-step. No
81 study mentioned in these reviews showed any data-driven model using sub-hourly time-step for
82 continuous streamflow prediction or estimating both surface and sub-surface outflows
83 simultaneously. Granata et al., (2016) also employed a machine learning methodology for sub-

84 hourly streamflow simulation. However, this study focused only on individual hydrographs and
85 not on continuous streamflow prediction.

86 Long short-term memory (LSTM) cell constitute a special type of NN, which has also
87 been used for streamflow simulation (Kratzert et al., 2018; Le et al., 2019; Yan et al., 2019;
88 Campos et al., 2019). The special feature of LSTM is its ability to learn time-dependent features
89 using its "memory" thereby it is regarded as an adequate NN for modeling hydrological cycles
90 (Shen, 2018; Greff et al., 2016; Zhang et al., 2018). To the best of our knowledge, no studies
91 have been published so far that discuss the use of LSTMs to simulate surface and sub-surface
92 flow simultaneously at a sub-hourly time-step. However, such a study is required because
93 modeling hydrological response of catchments at minutes-scale temporal resolution can enhance
94 the capacity of hydrological models to simulate contaminants whose concentrations vary at
95 logarithmic scale within short span of time such as fecal bacteria. In this study, we developed
96 two different data-driven models based on LSTM neural networks and compared their
97 performance with a physical model (HSPF) using sub-hourly data collected at the outlet of a
98 highly responsive headwater catchment in northern Laos, where land use dramatically changed
99 over the past 20 year: annual crops was replaced by teak tree plantations managed without
100 understorey. As different landuses have different flow patterns, change in landuse can affect soil
101 loss or stream water quality (Ribolzi et al., 2017). The specific objectives of this study were (1)
102 to evaluate the models' ability to perform simulations of surface runoff and sub-surface flow at
103 high temporal resolution of 6 minutes, (2) to conduct sensitivity analysis of the models
104 developed in this study, and (3) to predict HRU level surface and sub-surface responses from
105 LSTM-based models and to analyze whether HRU-level discretization in LSTM-based model
106 can yield better result as compared to non-HRU based LSTM model.

2 Materials and Methods

2.1 Study area

This study was conducted in a 0.6 km² Houay Pano headwater catchment, located 10 km south of Luang Prabang city in Northern Lao P.D.R. It is a sub-basin of the Houay Xon river basin, which is a tributary of the Mekong River (**Figure 1**). The experimental site is part of the critical zone observatories' network named Multiscale TROPIcal CatchmentS (M-TROPICS), which belongs to the French Research Infrastructure OZCAR (Gaillardet et al., 2018). This catchment can be considered as being representative of the montane agro-ecosystems of South-East Asia. The bedrock is made up of siltstone and fine-grained sandstones of Permian to upper carboniferous age. Soils are Entisol, Ultisol, and Alfisol covering 20, 30, and 50% of the catchment, respectively (Chaplot et al., 2005). The slope in the area varies from 0% to 171% with an average slope of 54% (Ribolzi et al., 2016). The climate is sub-tropical humid. The mean annual rainfall is 1500 mm per year; however, the rainfall pattern is highly seasonal as the monsoon season, which runs from mid-May to mid-October, constitutes 77% of the rainfall. The average monthly temperature at the site lies between 12 and 35 °C with the highest temperature in April, just at the beginning of the wet season. The humidity level varies from 17% to 100%. The catchment experiences several storms during the rainy season, with heavy rainfall (up to 100 mm h^{-1}), a phenomenon which is characteristic of tropical regions (Ribolzi et al., 2016). We measured hourly temperature (Celsius), relative humidity (%), wind speed (m/s), and solar radiation (J/m^2). The precipitation, electrical conductivity of stream water, surface and sub-surface flow were measured at 6-minute time-step at the study site. Moreover, a survey is

conducted every year to ascertain landuse changes in study area. More details on data acquisition are given in Supplementary Information (**Text S1**).

The area consists of four major land use types consisting of annual crops, teak, fallow, and forests (**Figure S1**). Recently, the area has undergone major land use changes with an increasing number of teak tree plantations (Ribolzi et al., 2017). Several studies have been conducted to understand the impact of land use changes on hydrological responses in the area (Ribolzi et al., 2018; Patin et al., 2018; Kim et al., 2018). We observed that the increase in teak tree plantations from 2002 onwards has resulted in a decrease of infiltration and consequently an increase in overland flow and sediment yield (Ribolzi et al., 2017).

2.2 Hydrological Simulation Program-FORTRAN Model setup

Hydrological Simulation Program-FORTRAN (HSPF) is a lumped model for water quality and quantity modeling at catchment scale developed during the 1970s (Johanson and Davis, 1980). The steps used to build the HSPF model are summarized in **Figure 2(a)**. We used BASINS 4.1 (Kinerson et al., 2009) software to prepare the input file for HSPF. This software pre-processes land use shapefiles, digital elevation models (DEM), and timeseries data of environmental variables such as precipitation, evapotranspiration, and temperature to prepare an input file for HSPF. Post processing steps were carried out on the input file and input data to incorporate the impact of changing land use, details of which are given in the Supplementary Information (**Text S2).** Furthermore, we performed Morris OAT (Morris, 1991) sensitivity analysis to find out the most sensitive parameters for the model. We chose 13 parameters for each of the four land uses present in the study area for sensitivity analysis (**Table S1**). Thus, the total number of parameters chosen for sensitivity analysis were 52. These 52 parameters are used

150 in equations which control the movement of water on pervious land segments in HSPF. Details
151 on the implementation of the sensitivity analysis can be found in the Supplementary Information
152 (**Text S3**). After sensitivity analysis, we calibrated the model by reducing the mean square error
153 (MSE) between observed and predicted outflows, which was calculated using the following
154 equation:

155 $$MSE = \frac{[\sum_{i=1}^{n}(o_i - p_i)^2]}{n} \quad (1)$$

156 where p_i and o_i are simulated and observed data, respectively, and n represents the number of
157 points in the data set. We used a truncated Newton algorithm (Nash, 1984) provided by the
158 Python library 'scipy' (Jones et al., 2001) to minimize the loss function.

159 **2.3 Long Short-Term Memory (LSTM)**

160 Neural networks (NN) are a group of algorithms which work similar to how the human
161 neural system is thought to work and are used to recognize patterns (Fukushima, 1980). They
162 consist of a stack of layers of neurons where each neuron is associated with weights and
163 activations. NNs can be calibrated to learn a non-linear function through backpropagation, in
164 which the weights and biases in each layer of the NN are optimized by reducing the error/loss
165 between observed and predicted output from the network (Rumelhart et al., 1988). The calibrated
166 network, which is commonly known as a trained network, is then used to predict output from
167 unseen data during validation (Rumelhart et al., 1988). Such models based on NNs are also
168 considered as black-box models, where we only deal with input and output while the model itself
169 finds relationships between the input and output data (Benítez et al., 1997). Recurrent Neural
170 Networks (RNN) are a special kind of neural networks which are designed to work with time-

series data because of their ability to capture long term temporal dependencies in data (Rumelhart et al., 1988). Simple RNNs suffer from the problem of vanishing gradient in deep neural networks where they fail to capture long range dependencies (Hochreiter, 1998).

One solution to this problem is known as the Gated Recurrent Unit (GRU) and was proposed by Cho et al., (2014). They introduced the concept of gates which control the flow of information within the recurrent unit. Another solution to the problem of vanishing gradient was proposed by Hochreiter and Schmidhuber, (1997) which is known as long short-term memory (LSTM). LSTM uses three gates namely forget, update and output gates to control the flow of information. The current candidate cell state $C_c^{<t>}$ depends on previous activation. The current cell state $C^{<t>}$ is calculated using update and forget gate and then the output gate is used to decide the activations at current time step.

$$C_c^{<t>} = tanh(W_c[\, a^{<t-1>}, x^{<t>}] + \, b_c) \tag{2}$$

$$\Gamma_f = \sigma(W_f[c^{<t-1>}, x^{<t>}] + b_{f)} \tag{3}$$

$$\Gamma_o = \sigma(W_o[c^{<t-1>}, x^{<t>}] + b_{o)} \tag{4}$$

$$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_{u)} \tag{5}$$

$$c^{<t>} = \Gamma_u * C_c^{<t>} \, + \, \Gamma_f) \, * \, c^{<t-1>} \tag{6}$$

$$a^{<t>} = \Gamma_o \, * \, tanh\, c^{<t>} \tag{7}$$

In these equations, W and b are weights and biases whose values are calibrated when the model is trained. The model is fed with inputs x at time-step t and the activation at time-step t which is calculated using (Eq. 7) is taken as the output. In rainfall-runoff modeling, model output has strong time dependence. There is time delay in the stream response to precipitation (Talei and Chua, 2012) and this lag time depends on catchment features (Singh, 1988). LSTMs have the

188 ability to learn this behavior because of their ability to learn patterns in time-series data (lin Hsu
189 et al., 1997). In this study, we developed two NN-based models. The steps to develop a simple
190 LSTM model are briefly described in **Figure 2(b)** while the steps for a detailed, more complex,
191 HRU-based LSTM model are summarized in **Figure 2(c)**.
192
193 **2.3.1 Setup for simple LSTM model**
194 The simple LSTM model (**Figure 2(b)**) consists of only a single LSTM layer. We chose
195 the parameters to build this NN based on a trial and error procedure. This simple LSTM model is
196 considered our baseline model and was used to compare the performance of the HRU-based
197 LSTM model. The LSTM layer takes all input data and is calibrated to produce two outputs
198 which are considered representative of surface and sub-surface flow (**Figure 3**). Although the
199 same LSTM layer generates two outputs, it does not know which one is surface and which one is
200 sub-surface flow. However, this distinction is made by comparing one (first) output of the model
201 with observed surface flow and the second output of model with sub-surface flow. This
202 comparison is done by calculating the MSE and then the model adjusts the weights and biases of
203 the LSTM layer by back propagation in such a way that it tries to reduce the value of the MSE.
204 In this way, the model learns implicitly that its first output must correspond to surface flow and
205 the second output has to correspond to sub-surface flow.
206 **2.3.2 Setup for HRU-based LSTM model**
207 The second structure of LSTM model is the HRU-based structure consisting of 36
208 parallel NNs (**Figure 4**). Each of these parallel networks is similar to a simple LSTM structure,
209 however, it represents one hydrological response unit (HRU) instead of the whole catchment.

Each parallel NN consists of layered LSTM cells with 128 hidden units. We chose the number of hidden units based upon hyper-parameter optimization results. The input to LSTM at each time step consists of a 2-dimensional array of shape (sequence length, input features). The sequence length and input features are given in **Table 1**. The output from LSTM at each time step is equal to the number of hidden units. A fully connected layer was used to convert the output from LSTM into the specific dimension. However, the multiple fully connected layer can increase computation time as well as the model complexity. In order to avoid this, we used a single fully connected layer instead of a deep neural network after LSTM. Several studies also showed a single fully connected layers for developing hydrological models (Kratzert et al., 2018; Zhang et al., 2018; Kratzert et al., 2019; Kratzert et al., 2019b; Li et al., 2020). The fully connected layer in this study generates two outputs: surface runoff and sub-surface flow (**Figure 3**) within an HRU. The number of layers in each of the parallel NN and the number of hidden units were decided by hyper-parameter optimization and are given in **Table 1**.

An HRU is defined here as a distinguished land use in a distinguished sub-basin. This means land uses present in different sub-basins were considered as separate HRUs. This is different from HSPF where similar land uses in different sub-basis are considered are merged into one HRU. Thus, if a land use type, e.g. grass, is present in a catchment with 9 sub-basins and all the sub-basins have this type of land use, we will have 9 HRUs for this land use and nine separate NNs are assigned to simulate processes in these HRUs. Similarly, if a 'forest' land use is present in 5 sub-basins in the catchment, this will result in 5 other HRUs in the model. This is further illustrated in **Figure S2** where a catchment consists of 5 sub-basins (A, B, C, D, E) and four land use types (i, ii, iii, iv) and 15 HRUs.

Each of the parallel NNs in **Figure 4** shares HRU-invariant input data such as solar radiation, air temperature, etc. The HRU-specific input data, such as precipitation received by

each HRU, was also prepared and was fed only to the corresponding NNs. The environmental data, such as air temperature and solar radiation, is experienced equally by the whole area of the catchment. Thus, all NNs share this environmental data. On the other hand, parameters which depend on HRU characteristics, such as its area and distance to the outlet, were calculated for each HRU. These HRU-specific parameters were only fed to the NNs representing the corresponding HRUs in the model. Precipitation data was measured for the whole catchment, and then precipitation received by each HRU was calculated based on the area of that HRU. This HRU-specific data can be viewed in **Figure S7** and **Figure S8**.

As the land use in the study area varies with time, this implies that the locations of HRUs also vary with time. Thus, all the HRU-specific data, i.e., distance to outlet, area, and volume of precipitation received, was also calculated accordingly. This time-varying HRU-specific data can be viewed in **Figure S7** and **Figure S8**. One implication of this is that we considered all possible HRUs in the study area. For example, if the land use 'grass' is not present in sub-basin 1 in year 2011, it may appear in 2012 and then disappear again. Thus, we considered the HRU 'grass in sub-basin 1' for the whole simulation period, though the input values for year 2011 and 2013 will be zeros in this case.

The catchment area consisted of 9 sub-basins and 4 types of land uses, thus implying 36 possible HRUs. As the land use inside a sub-basin is distributed and no specific distance from an HRU to the catchment outlet can be measured, we used the distance of the sub-basin from the outlet as representative of all land uses inside that sub-basin. This means all the land uses/HRUs inside a sub-basin were considered to have the same distance from the outlet as that sub-basin.

This generalization results in certain HRUs having the same distance from the outlet, however, it still maintains HRU-specific information.

Each of the parallel NNs in **Figure 4** produces two outputs which are considered as representative of HRU-specific surface and sub-surface flow. Corresponding values of surface and sub-surface flow are added in cumulative order. This order of addition is the same as the stream filling order in the catchment. This cumulative and ordered filling of outputs is similar to realistic stream routing. The final values of surface and sub-surface flow are considered as model outputs and are compared with observed values to calculate the mean square error using Equation 1.

This structure allows us to build a detailed model of the catchment, where outflow from each land use is simulated separately and instead of one value of total streamflow, the surface and subsurface outflow from the catchment are simulated. More details on the implementation of this NN in the computer program are given in **Text S4** of the Supplementary Information. The motivation for HRU-based LSTM model was drawn from physically based models such as SWAT or HSPF where the study area is discretized into smaller HRUs. All processes are modeled at HRU level in these models. In order to compare results of LSTM-based model with HSPF, the study area was discretized into HRUs. Another purpose of discretizing study area for HRU-based LSTM was to assess the impact of increase in spatial resolution on model performance.

2.3.3 Hyper-parameter optimization (HPO) based on window size

The performance of a NN model is mainly governed by a set of parameters, which are used to build the NN, such as length of input data fed to it at each time-step, and number of

nodes in a layer (Hutter et al., 2015). These parameters are called hyper-parameters and their description and possible ranges are presented in **Table S3**. To achieve the best performance with calibration, we need to optimize these parameters, because a slight change in any of these parameters can worsen or improve the performance of the model. Different derivative free optimization algorithms are those which aim to solve problems given as black box (Lakhmiri et al., 2019). Direct search and Bayesian are one of such methods. Direct search is an 'a priori' method where the decision maker articulates preferences before optimization. Bayesian optimization is one of 'a posteriori' methods which aims to generate a representative set of pareto optimal solutions and then the best among them is chosen (Chen and Li, 2018). We selected Bayesian optimization approach because of it being a popular approach to optimize hyper-parameters in machine learning models (Shahriari et al., 2015; Snoek et al., 2015; Frazier, 2018). The details about implementation of Bayesian optimization are given in **Text S5**.

Several open source libraries are available for implementing Bayesian optimization method in Python programming language such as Hyperopt (Bergstra et al., 2015) and scikit-optimize (Kumar and Head, 2017). We used scikit-optimize library because it allows the use of Gaussian Processes as surrogate function. Implementation of Bayesian in Hyperopt can be done by making use of Tree Parzen Estimator (Bergstra et al., 2011). The advantage of Gaussian Processes is that it can consider the interaction between hyper-parameters during the optimization (Dewancker et al., 2015). The surrogate function is the probability model of objection function and it calculates the probability of loss with respect of input values. In Bayesian optimization method, this surrogate function is optimized instead of actual objective function. The methodology of selecting the new parameter from parameter space was 'Expected Improvement' to the surrogate function. The expected improvement algorithm (Mockus,1975;

299 Jones et al., 1998) considers the size of the improvement. We used default parameters used by
300 scikit-optimize (Kumar and Head, 2017) library in using Gaussian Processes. These parameters
301 include 'optimizer', 'kaapa' and 'xi' and are described in following sentences. The 'optimizer'
302 minimizes the acquisition function. We used Limited-Memory Broyden-Fletcher-Goldfarb-
303 Shanno (Andrew and Gao, 2002) as optimizer. The parameter 'kappa' controls variance in
304 predicted values and was set to 1.96. The parameter 'xi', determines how much improvement
305 should be taken into consideration over previous best values and was set to 0.01.

306 The computation time of HPO depends upon the number of epochs to train a single model and the
307 number of iterations used for optimization. Furthermore, it also depends upon the complexity of model
308 being built at a specific iteration. The computation time, in general, increased by increasing the sequence
309 length. We performed 50 iterations for each sequence length (**Figure S5**). The model was trained for 100
310 epochs during each of these iterations. The computation time taken for optimizing hyper-parameters was
311 15, 19, 25, 40 and 45 hours for sequence lengths of 20, 30, 40, 50 and 60, respectively. Although HPO
312 needs considerable computation power, this method can improve the model performance by finding the
313 optimal hyperparameter set. The model performance depending on the iteration is showed in **Figure S5**.

314 **2.4 Performance metrics and data splitting**

315 The performance of each model was evaluated using Nash-Sutcliffe Efficiency (NSE),
316 mean squared error (MSE) and percentage bias (PBIAS). The MSE was calculated according to
317 equation 1 and the equation used to calculate NSE and PBIAS are:

318 $$\mathrm{NSE} = 1 - \frac{\sum(p_i - o_i)^2}{\sum(o_i - \bar{o})^2} \quad (8)$$

319 $$\mathrm{PBIAS} = \frac{\sum_{i=1}^{n} O_i - p_i}{\sum_{i=1}^{n} O_i} \, x \, 100 \quad (9)$$

where p_i is simulated data, o_i is observed data, and n is the number of points in the data set. Values of MSE indicate how closely the predictions follow the observed values. In model training, 70% of the input data corresponding from January 2011 to January 2013 was applied. 15% of the data were used as validation and test, respectively. All three models produced two outputs—surface and sub-surface flow—and during optimization of parameters, an average error of the surface and sub-surface flow was calculated. This averaged error was then considered as the objective function which each model tries to minimize. Predicted total discharge is thus the sum of surface and sub-surface flows in each case.

In order to avoid overfitting in LSTM based models, we used a mild regularization technique, namely early stopping (Goodfellow et al., 2016). We checked the performance of the models after each epoch on validation data. The model had not processed this data during calibration. We stopped calibrating the model when validation loss reached a plateau even if calibration loss kept on decreasing.

3. Results

3.1 HSPF Results

3.1.1 Sensitivity Analysis

We carried out sensitivity analysis for sub-surface flow and total discharge separately. Based on the results of sensitivity analysis, we selected 12 parameters for baseflow and for total discharge which had the strongest impact on these outflows for calibration. Not all parameters for baseflow and total discharge are different, rather there are some common parameters which had a strong impact on both sub-surface flow and total discharge (**Table 2**). Both sub-surface

flow and total discharge were highly sensitive to INFILT, which describes infiltration capacity (Bicknell et al., 2001), while only sub-surface flow was sensitive to AGWETP, BASETP, and DEEPFR.

The parameters AGWETP and BASETP control the amount of evapotranspiration that can be taken from active groundwater storage and baseflow, respectively (Bicknell et al., 2001). As the definitions of these two variables suggest, they are closely associated with baseflow. Thus, sensitivity analysis showed these parameters to be more important for sub-surface flow, which is the sum of groundwater flow and interflow. These results are consistent with the similar studies (Xie and Lian, 2013; Baek et al., 2017; Diaz-Ramirez et al., 2013), which also demonstrated that INTFW, IFILT, DEEPFR, and AGWETP are among the most important parameters for streamflow.

3.1.2 Estimation of surface and sub-surface flow using HSPF

Figure 5 and **Figure 6** shows predictions from the HSPF model for the calibration, validation and test periods. We evaluated the performance of the model by measuring the MSE and NSE for the calibration and test periods separately. Values of these errors for surface, sub-surface, and total flow are given in **Table 3.** The predicted sub-surface flow in **Figure 5** and **Figure 6** is often underestimated, while the surface flow simulated by HSPF is mostly overestimated. This is the reason there is large positive PBIAS for surface flow and large negative PBIAS for sub-surface flow (**Table 3**). Although the predicted sub-surface flow is very low, the predicted total discharge is higher. The predicted total discharge is higher because of the large amount of input in the form of surface runoff. Overestimation of surface flow and under-estimation of sub-surface flow was also observed by Hoang et al., (2014) after the application of

363 the SWAT model to a watershed in Denmark. This research attributed this behavior to the
364 inadequacy of the model structure for simulating these processes.

365 UZSN represents the storage capacity of the upper zone in the HSPF model (Bicknell et
366 al., 2001), which is a soil zone (**Table 2**). A higher value of UZSN means that the soil has a
367 higher storage capacity and thus more water will be retained in the upper zone which becomes
368 available for evapotranspiration (Bicknell et al., 2001). Our calibrated UZSN values are closer to
369 the upper limits, which means more water is being retained in the soil zone, eventually leading to
370 higher evapotranspiration. In other words, this means more water is available for
371 evapotranspiration from upper zone.

372 We observed that predicted sub-surface flow was higher in 2013 as compared to 2011,
373 while it was the lowest in 2012. We observed this trend because of changes in land use during
374 these years (**Figure S9**). The rise of sub-surface flow in 2013 can be attributed to an increase in
375 fallow land use and a decrease in annual crop land use in 2013. A recent study by Ribolzi et al.,
376 (2017) found a correlation between higher sub-surface flow and the increase in teak plantations
377 in this catchments. Although teak and annual crop land use result in higher surface flow, the joint
378 contribution of teak and annual crop decreased in 2013 and the contribution of fallow land use
379 increased. The smaller sub-surface flow observed during 2012 could be due to the relatively
380 higher teak and annual crop land use during this year as compared to other years.

381 **3.2 Estimation of surface and sub-surface flow using deep learning**

382 **3.2.1 Simple LSTM**

383 The simple LSTM model consisted of a single LSTM layer and was built using the
384 hyper-parameters given in **Table 1**. We then used the model calibrated with these hyper-

parameters for evaluation during the test period. The performance of the model during calibration and test for surface, sub-surface, and total flow can be seen in **Figure 7, Figure 8** and **Table 3**.

The NSE values for surface runoff prediction during calibration and test were 0.43 and 0.64, respectively. These NSE values categorize the model performance 'unsatisfactory' and 'satisfactory' according to Moriasi et al., (2015). However, PBIAS, which measures average tendency of model to predict flow larger or smaller than observed is mostly between 'very good' (-3.2) and 'good (-5.6) **(Table 3)**. Despite this, the model captured most of the peaks in surface runoff during both the calibration and test periods. However, the predicted peaks are mostly lower than the observed peaks. This discrepancy can be attributed to the use of MSE for model calibration because MSE focused to reduce the average error between observation and simulation. The inability of model to capture peaks is also evident from flow duration curves for surface, sub-surface and total discharge (**Figure S13**). The percentage exceedance of predicted flows is below the observed in areas of high flows **(Figure S13)**. For sub-surface flow, this model predicted almost all the peaks yet failed to follow the trend of rising and falling limbs, which resulted in lower NSE values. Another important aspect of simple LSTM model is its ability to perform better for surface runoff as compared to sub-surface runoff. This is evident from Table 3, which shows all performance metrics for surface runoff better than those of sub-surface flow.

3.2.2 HRU-based LSTM model

We built the HRU-based LSTM model using information obtained from HPO. We used information about the activation function, normalization, and loss calculation methods and cell type from HPO. The choice of activation function affects the kind of non-linearity applied. Options for the loss calculation method were 'normal' or 'weighted'. In the weighted loss

408 calculation method, the loss value is more sensitive to peak flows. The choices for cell type, in
409 order to build NN, were GRU and LSTM. The HPO was then allowed to decide which of these
410 two cells perform best. The HPO algorithm varied the values of these hyper-parameters during
411 the optimization process until it found the best combination of hyper-parameters. This
412 optimization was performed for five different sequence lengths. **Figure S5** shows the results of
413 optimization, where the plot for each sequence length indicates how the loss value was reduced.
414 It shows how the optimization algorithm attempted to obtain the best hyper-parameters for a
415 specific sequence length. **Table 4** enlists configuration of models which resulted in maximum
416 reduction in loss value for each sequence length. It can be seen that the best HPO results were
417 mostly obtained using a rectified linear unit (ReLu), performing normalization of input data
418 before using it, using an LSTM cell instead of a GRU cell, and using the weighted loss
419 calculation method (**Table 4**). The values of sequence length and batch size were not optimized
420 using HPO because increasing them is equivalent to increasing the amount of input data being
421 fed to the NN. This exponentially increases the amount of computation, which requires greater
422 processing and memory resources. In this regard, we used a trial and error method to obtain
423 better optimum values for other hyper-parameters such as sequence length, batch size, etc. The
424 optimized set of hyper-parameters which were used to build the HRU-based LSTM model are
425 given in **Table 1**.

426 Plots for surface, sub-surface, and total flow for calibration, validation and test data are
427 shown in **Figure 9** and **Figure 10** respectively. The performance metrics obtained for this model
428 are given in **Table 3**. We observed underestimation of surface runoff, which is similar to what
429 we observed in predicted surface runoff from the simple LSTM model. However, in this case
430 there was more under estimation as compared to the simple LSTM. This is the reason that the

MSE value, which is the average error for the whole simulation range, was higher during calibration and test as compared to the MSE value of the simple LSTM model. The NSE values, which measure the accuracy of prediction, were 0.66 and 0.63 for calibration and test of sub-surface flow, respectively, which makes the model performance 'satisfactory' (Moriasi et al., 2015).

4. Discussion

4.1 Overall comparison of three models

By comparing the MSE and NSE values of all three models from **Table 3**, we can conclude that the simple model performed better for surface flow prediction. The simple LSTM model showed an NSE value of 0.64 and MSE value of 8.3-5 m^3s^{-1} for surface runoff prediction, which are the best values obtained among all three models. Although the simple LSTM model was meant to serve as a baseline model, the more complex HRU-based LSTM model could not perform better for surface flow prediction. It is interesting to note that in this case, increasing model complexity has not resulted in improved model performance. It has already been reported that adding complexity to an NN does not necessarily imply that it will outperform its simpler counterpart (Makridakis et al., 2018).

Overall performance of all models range from satisfactory to not-satisfactory as per criteria set by Moriasi et al., (2015). One of reasons for this lower accuracy can be attributed to a finer time-step of simulation. Indeed, several studies have reported deterioration in model performance for streamflow estimation with increase in simulation time-step (Stern et al., 2016; Gassman et al., 2007), especially in smaller catchments (Spruill et al., 2000). In an extensive review of over 100 SWAT applications in Brazil. Bressiani et al., (2015) found that only 6% of

studies with monthly simulations resulted in NSE values of less than 0.5. On the other hand, when daily time-step was used, 25% of studies rendered NSE value below 0.5. Boithias et al., (2017) reported degradation of validation NSE from 0.66 to 0.49 when streamflow was first simulated at a daily time-step and then at an hourly time-step. The reason for lower accuracy at shorter time-steps can be the use of sparsely distributed rainfall gauges which are unable to capture the spatial details of rainfall inputs (Gassman et al., 2007). Similarly, another explanation for higher model accuracy when using longer time-steps is that longer time-steps integrate the variability at smaller time-steps (Boithias et al., 2017).

For sub-surface flow, although the HSPF model performed best in terms of NSE, the predicted sub-surface flow was much lower than the observed sub-surface flow. The sub-surface flow predicted by HSPF in **Figure 5** and **Figure 6** , is much lower than the observed flow. This is the reason we see large negative values of PBIAS from the flow duration curves for predicted sub-surface flow is much below the observed **(Figure S12)**. The simple LSTM model was able to predict most of the peaks for sub-surface flow, however the value of NSE is lower as compared to that of the HRU-based LSTM model. The better values of NSE for sub-surface flow from the HRU-based LSTM model can be attributed to the better prediction of recession in peaks. The slopes in the falling limbs of predicted peaks from the HRU-based LSTM model in **Figure S10**, which are absent in the peaks predicted by the simple LSTM model. If we consider MSE values, the HRU-based LSTM model outperformed HSPF for both surface flow as well as sub-surface flow during calibration period.

In all models, the predicted flow peaks were lower than the observed peaks for both surface and sub-surface flow, except the predicted surface flow from HSPF. However, in HSPF,

the number of predicted storm events were much more than the observed, which resulted in a negative NSE value. In the case of the HRU-based LSTM model, although it predicted most of the storm events, it still under-predicted surface flow.

The discrepancy between train and test MSE was caused by the difference in data distribution during these periods. The train data set had larger standard deviation than the validation and test data set (Table S4). The dataset with larger variance can have large MSE (Munna et al., 2015; Grams et al., 2002). If the training performance of a model is significantly larger than that of test data, this indicates overfitting. On the other hand, better performance for test data set in our case indicates different distribution of training and test data sets.

4.2 Advantages and Limitations of HSPF

As the HSPF is a process driven model, thus simulations resulting from it give insights about behavior and condition of catchment. The higher values of a variable such as UZSN translate into large storage potential in upper zone of soil. However, simulation results from HSPF are greatly influenced by calculated potential evapotranspiration which itself can vary based on the method of evapotranspiration calculation used. Our HSPF results showed that major portion of the rainfall is evapotranspirated. In the HSPF model, the amount of actual evapotranspiration is increased until the requirement created by the potential evapotranspiration is satisfied. If potential evapotranspiration is very high, the model allows more available water from storage to return to the environment as actual evapotranspiration. In our simulations, lower predicted flows from the HSPF model may also be due to the overestimation of potential evapotranspiration (**Table S2**) as has been the case in the studies of Yeh, (2017) and

Prudhomme and Williamson, (2013). In this study, we calculated daily evapotranspiration and then distributed to 6 min by calculating daily sunshine hours. This interpolation of daily evapotranspiration to 6-min time-step can also be a reason of poor HSPF performance. There have been several studies showing that using evapotranspiration values calculated at a higher temporal resolution results in better model performance (Debele et al., 2009)). The conceptual physically-based models can render poor performance if one of the model variable is incorrectly calculated. Ouellet-Proulx et al., (2019) compared the performance of five different ET and evaporation models for rainfall-runoff modeling and showed that the choice of ET model affects streamflow by 3 to 24 percent. On the other hand, the deep learning models are less likely to suffer from these errors because they are not explicitly process driven.

4.3 Advantages of LSTM model

One of the key characteristics in surface runoff simulations is the lag time between rainfall and surface runoff (Talei and Chua, 2012). The lag time between observed surface runoff and incoming rainfall for two storm events of September 2013 can be seen in **Figure S11**. This figure also compares the results of lag time for all three models. These storm events are in the test period; thus this figure is a good representation of the ability of models to simulate lag time. It can be observed in the figure that HSPF predicts surface runoff as soon as there is a rainfall event while both our NN models show a lag time. Although the peaks predicted by the HRU-based LSTM model are lower than those predicted by the other models, the model showed responses to both storm events with lag time.

The simple LSTM model takes much less computing time as compared to the HRU-based LSTM model. This is self-evident because the HRU-based LSTM model has 36 times more

519 parameters to calibrate as compared to the simple model. The number of calibration parameters
520 for each NN model can be calculated from the hyper-parameters given in **Table 1.** For the simple
521 LSTM model there were 258 parameters to calibrate (128 weights and biases for an LSTM with
522 128 hidden units and one weight and bias for the fully connected layer). Similarly, the number of
523 parameters calibrated by the HRU-based LSTM model were 9,288. The computation time of
524 HRU-based LSTM model did not scale with number of parameters. We observed an average
525 time of 4 minutes per epoch for HRU-based LSTM while 0.5 minutes per epoch for simple
526 LSTM model. We used Intel® Core™ i7-8700 processor with graphic card of NVIDIA GeForce
527 GTX 1060 having 6 Gigabytes of dedicated GPU memory along with 32 Gigabytes of Random-
528 Access Memory. The parallel computing power of Tensorflow (Abadi et al., 2016) prevented the
529 scaling of training time with model parameters (Adie et al. (2018)). The total training time
530 however depends upon the number of training epochs used. The HRU-based LSTM model took
531 262 minutes for 66 epochs while it took approximately 100 minutes for simple LSTM to train for
532 214 epochs.

533 The results of the LSTM-based models show that total streamflow was mainly governed
534 by sub-surface flow while the surface flow only contributed to flood peaks during rainfall events.
535 It can be observed that the predicted flow patterns for sub-surface flow and total discharge are
536 the same, except for peak heights (**Figure 7** to **Figure 10**). The peaks are higher in total
537 discharge, which means that surface flow only contributes to increases in peak heights. This can
538 also be seen in flow duration curves for all three models **(Figure S12, S13 and S14)** where the
539 predicted flow duration curve is always below observed in high flow regions. This result is
540 consistent with a previous study within Houay Pano catchment that showed the larger
541 contribution of baseflow to streamflow during floods (Ribolzi et al., 2018). In our study, the total

surface and sub-surface flow from simple LSTM model was 78 and 2,160 mm. Similarly, the HRU-based LSTM model also showed dominance of sub-surface flow in the catchment. The total simulated surface and sub-surface flow for three years using HRU-based LSTM model was 32 and 1,913 mm respectively. The large gap between two values is because of absence of surface runoff during most of the days in year when there is no rainfall.

Under-estimation of peak flows using lumped, physically-based models is a frequent drawback, which has been observed in several studies (Boithias et al., 2014; Bieger et al., 2014; Fohrer et al., 2014; Loukas and Vasiliades, 2014). We observed a similar trend in sub-surface flow predicted by HSPF. However, in our LSTM-based models, this problem is partially solved. We can observe large gap between predicted and observed flow duration curve in Figure S12 while this gap is smaller for LSTM-based models (**Figure S13, Figure S14**). The simple LSTM model captured peaks in surface runoff more accurately as compared to those predicted by the HRU-based LSTM model. This trend was also found in the total estimated discharge from each model. When peaks in surface runoff are underestimated, peaks in total discharge are also underestimated (**Figure 9** and **Figure 10**), and when peaks in surface runoff are better estimated, peaks in total discharge are also better estimated (**Figure 7** and **Figure 8**). This means that the predicted surface runoff mainly contributes to peaks in streamflow while the sub-surface flow makes up the baseflow portion. This makes our LSTM-based models closer to real observations.

4.4 Challenges and limitations of LSTM models

Calibrating NN for surface flow is extremely challenging. The reason for this is that 96% of the surface flow data consists of zeros because discernable surface runoff only happens during rainfall events. This problem is similar to anomaly detection or rare event detection where less

than 5–10% of the total data is positively labeled. We observed during HPO iterations that total surface flow did not change once it became zero upon further calibration of model. It can be argued that as 96% of surface flow has one unique value, i.e., zero, the network learns to predict zeros. The matching of the surface flow curve became more challenging because surface flow does not always coincide with rainfall events. Indeed, during the hot and dry season (generally from January to April), the soil is dry and evapotranspiration is high. Indeed, the potential evapotranspiration is higher than rainfall during this period (**Table S2**). For small rainfall events during this hot and dry season, rainfall is either evapotranspirated or infiltrated, however, it is not transferred to the stream by surface runoff. Precipitation absorbed by the soil may later become part of sub-surface flow. Calibrating a NN to learn this behavior is the most difficult part, and if hyper-parameters are not chosen appropriately, the model fails to generalize the surface flow patterns.

Surface and sub-surface flows obtained for each of the 36 HRUs using HRU-based LSTM model are plotted in Figure S6. By comparing these HRU-specific surface and sub-surface responses with the HRU-specific input data (**Figure S7** and **S8**), we cannot draw a one to one correspondence between HRU-specific input data and the corresponding output. In certain cases, an LSTM produces no outflow even when it receives precipitation; thus, all LSTMs in our HRU-specific model are not necessarily representative of HRU-specific inputs. This can be because NNs act as black-box models, and the inner workings of these networks are random (Karpatne et al., 2017). Thus we cannot draw a simple link between the weights of a NN and the function being approximated. Another reason for these unpredictable HRU-specific outputs could be that we used separate NN for each HRU. This means that each of these NNs have separate weights and biases; thus, if one network gets higher input values of curve number or

precipitation, it is completely independent of what the other networks receive as input. Thus, each network forms its own 'context' when it calibrates its weights and biases by looking at the total output of the whole model.

The interest of model interpretability has increased in the field of machine learning (Samek, W. 2019). Several studies have proposed the ways to incorporate scientific knowledge into deep learning (Karpatne et al., 2017; Karpatne et al., 2017b; Wang et al., 2020). The suggested method in our study, the neural network down into sub-models for each HRU, would be a way to introduce more interpretability into a data-driven models in that this approach can analyze sub-models in the neural network.

5. Conclusions

In this study, we modeled the surface and sub-surface flow using three models: one lumped model called HSPF and two deep learning models. One deep learning model consisted of a single LSTM representing the whole catchment, whereas the second model consisted of LSTMs representing each HRU. All three models predicted total flow, surface and sub-surface flow separately. The following conclusions were then derived from the results:

- By replacing the constant values of the area factors in the HSPF model with time series values, we were able to model land use changes in a catchment.
- Although HSPF was able to estimate surface and sub-surface flow simultaneously, it over-estimated surface runoff and under-estimated sub-surface flow. Contrary to this, our deep learning models were more consistent in predicting surface and sub-surface flow. Therefore, deep learning models are more suitable when prediction of both surface and sub-surface flow is required simultaneously.

- The simple LSTM model, where one LSTM layer is used to represent the whole catchment, performed best for surface runoff prediction during both calibration and test period.
- The HRU-based LSTM model performed better than simple LSTM model for sub-surface flow prediction during both test and calibration period. It has the best performance for total streamflow simulation during test period.

Understanding the combined impact of climate change and land use changes on the catchment by modeling surface and sub-surface flows at a very high temporal resolution of 6 min can help assessing extreme low and extreme high discharge, and improve water resource management. This would allow more accurate modeling of pollutants (e.g. fecal bacteria) whose concentrations vary exponentially with time. This study presents a methodology for incorporating land use changes into hydrological models of surface and sub-surface flow at catchment scale. This study also demonstrates that deep learning can be an alternative to physically based or conceptual models by taking in account model complexity at spatial and temporal scales.

5. Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2018K1A3A1A21041779). The authors sincerely thank the Lao Department of Agricultural Land Management (DALaM) for its support, including granting the permission for field access, and the M-TROPICS Critical Zone Observatory (https://mtropics.obs-mip.fr/), which belongs to the French Research Infrastructure OZCAR (http://www.ozcar-ri.org/), for data access.

6. References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. 2016. Tensorflow: A system for large-scale machine learning. In 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16) 265-283.

Andrew, G., & Gao, J. 2007. Scalable training of L 1-regularized log-linear models. In Proceedings of the 24th international conference on Machine learning, 33-40. https://doi.org/10.1145/1273496.1273501

Arnold, J.G., Fohrer, N., 2005. SWAT2000: current capabilities and research opportunities in applied watershed modelling. Hydrological Processes: An International Journal, 19(3), 563-572. https://doi.org/10.1002/hyp.5611

Adie, H. T. R., & Pradana, I. A. 2018. Parallel computing accelerated image inpainting using GPU CUDA, Theano, and Tensorflow. In *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE),* 621-625. IEEE. https://doi.org/10.1109/ICITEED.2018.8534858

Baek, S.-S. et al., 2017. Developing a hydrological simulation tool to design bioretention in a watershed. Environmental Modelling & Software. https://doi.org/10.1016/j.envsoft.2017.11.006

Benítez, J.M., Castro, J.L., Requena, I., 1997. Are artificial neural networks black boxes? IEEE Transactions on neural networks, 8(5), 1156-1164. https://doi.org/10.1109/72.623216

Bennett, J.C., Robertson, D.E., Ward, P.G., Hapuarachchi, H.P., Wang, Q., 2016. Calibrating hourly rainfall-runoff models with daily forcings for streamflow forecasting applications in meso-scale catchments. Environmental Modelling & Software, 76, 20-36. https://doi.org/10.1016/j.envsoft.2015.11.006

Bergstra, J. S., Bardenet, R., Bengio, Y., & Kégl, B. 2011. Algorithms for hyper-parameter optimization. In Advances in neural information processing systems, 2546-2554.

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., & Cox, D. D. 2015. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, *8*(1), 014008.

Besaw, L.E., Rizzo, D.M., Bierman, P.R., Hackett, W.R., 2010. Advances in ungauged streamflow prediction using artificial neural networks. Journal of Hydrology, 386(1-4), 27-37. https://doi.org/10.1016/j.jhydrol.2010.02.037

Bicknell, B.R. et al., 2001. Hydrological simulation program-Fortran: HSPF version 12 user's manual. AQUA TERRA Consultants, Mountain View, California, 845.

Bieger, K., Hörmann, G., Fohrer, N., 2014. Simulation of streamflow and sediment with the soil and water assessment tool in a data scarce catchment in the three Gorges region, China. Journal of environmental quality, 43(1), 37-45. http://doi.org/10.2134/jeq2011.0383

Boithias, L. et al., 2017. Simulating flash floods at hourly time-step using the SWAT model. Water, 9(12), 929. http://doi.org/10.3390/w9120929

Boithias, L., Srinivasan, R., Sauvage, S., Macary, F., Sánchez-Pérez, J.M., 2014. Daily nitrate losses: implication on long-term river quality in an intensive agricultural catchment of southwestern France. Journal of environmental quality, 43(1), 46-54. http://doi.org/10.2134/jeq2011.0367

Bressiani, D.d.A. et al., 2015. Review of soil and water assessment tool (SWAT) applications in Brazil: Challenges and prospects. International Journal of Agricultural and Biological Engineering, 8(3), 9-35. http://doi.org/10.3965/j.ijabe.20150803.1765

Burns, D.A., Kendall, C., 2002. Analysis of δ15N and δ18O to differentiate NO3− sources in runoff at two watersheds in the Catskill Mountains of New York. Water Resources Research, 38(5), 9-1-9-11. https://doi.org/10.1029/2001WR000292

675 Campos, L.C.D. et al., 2019. Short-Term Streamflow Forecasting for Paraíba do Sul River Using Deep
676 Learning, EPIA Conference on Artificial Intelligence. Springer. 507-518.
677 Chaplot, V.A., Rumpel, C., Valentin, C., 2005. Water erosion impact on soil and carbon redistributions
678 within uplands of Mekong River. Global biogeochemical cycles, 19(4).
679 https://doi.org/10.1029/2005GB002493
680 Chen, Y., & Li, Y. 2018. Computational intelligence assisted design: in industrial revolution 4.0. CRC Press.
681 Cho, K. et al., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine
682 translation. arXiv preprint arXiv:1406.1078.
683 Debele, B., Srinivasan, R., Parlange, J.-Y., 2009. Hourly analyses of hydrological and water quality
684 simulations using the ESWAT model. Water resources management, 23(2), 303-324.
685 Dewancker, I., McCourt, M., & Clark, S. 2015. Bayesian optimization primer.
686 Diaz-Ramirez, J., Johnson, B., McAnally, W., Martin, J., Alarcon, V., 2013. Estimation and Propagation of
687 Parameter Uncertainty in Lumped Hydrological Models: A Case Study of HSPF Model Applied to
688 Luxapallila Creek Watershed in Southeast USA. J Hydrogeol Hydrol Eng 2: 1. of, 9, 2.
689 http://dx.doi.org/10.4172/2325-9647.1000105
690 Frazier, P. I. 2018. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*.
691 Ficchì, A., Perrin, C., Andréassian, V., 2016. Impact of temporal resolution of inputs on hydrological
692 model performance: An analysis based on 2400 flood events. Journal of Hydrology, 538, 454-470.
693 https://doi.org/10.1016/j.jhydrol.2016.04.016
694 Fohrer, N., Dietrich, A., Kolychalow, O., Ulrich, U., 2014. Assessment of the environmental fate of the
695 herbicides flufenacet and metazachlor with the SWAT model. Journal of environmental quality,
696 43(1), 75-85. http://doi.org/10.2134/jeq2011.0382
697 Fukushima, K., 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern
698 recognition unaffected by shift in position. Biological cybernetics, 36(4), 193-202.
699 https://doi.org/10.1007/BF00344251
700 Gaillardet, J. et al., 2018. OZCAR: the French network of critical zone observatories. Vadose Zone Journal,
701 17(1).
702 Gassman, P.W., Reyes, M.R., Green, C.H., Arnold, J.G., 2007. The soil and water assessment tool:
703 historical development, applications, and future research directions. Transactions of the ASABE,
704 50(4), 1211-1250. https://doi.org/10.13031/2013.23637
705 Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. MIT press.
706 Grams, J. S., Gallus Jr, W. A., Koch, S. E., Wharton, L. S., Loughe, A., & Ebert, E. E. 2006. The use of a
707 modified Ebert–McBride technique to evaluate mesoscale model QPF as a function of
708 convective system morphology during IHOP 2002. *Weather and forecasting*, *21*(3), 288-306.
709 https://doi.org/10.1175/WAF918.1
710 Granata, F., Gargano, R., de Marinis, G., 2016. Support vector regression for rainfall-runoff modeling in
711 urban drainage: A comparison with the EPA's storm water management model. Water, 8(3), 69.
712 Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J., 2016. LSTM: A search space
713 odyssey. IEEE transactions on neural networks and learning systems, 28(10), 2222-2232.
714 http://doi.org/10.1109/TNNLS.2016.2582924
715 Gupta, H.V., Kling, H., Yilmaz, K.K., Martinez, G.F., 2009. Decomposition of the mean squared error and
716 NSE performance criteria: Implications for improving hydrological modelling. Journal of
717 hydrology, 377(1-2), 80-91. https://doi.org/10.1016/j.jhydrol.2009.08.003
718 Guse, B., Reusser, D.E., Fohrer, N., 2014. How to improve the representation of hydrological processes in
719 SWAT for a lowland catchment–temporal analysis of parameter sensitivity and model
720 performance. Hydrological processes, 28(4), 2651-2670. https://doi.org/10.1002/hyp.9777

Hoang, L. et al., 2014. Comparison and evaluation of model structures for the simulation of pollution fluxes in a tile-drained river basin. Journal of environmental quality, 43(1), 86-99. https://doi.org/10.2134/jeq2011.0398

Hochreiter, S., 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(02), 107-116.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), 1735-1780.

Huang, P., Li, Z., Yao, C., Li, Q., Yan, M., 2016. Spatial combination modeling framework of saturation-excess and infiltration-excess runoff for semihumid watersheds. Advances in Meteorology, 2016. http://dx.doi.org/10.1155/2016/5173984

Huang, Y., Bárdossy, A., Zhang, K., 2019. Sensitivity of hydrological models to temporal and spatial resolutions of rainfall data. Hydrology and Earth System Sciences, 23(6), 2647-2663. https://doi.org/10.5194/hess-23-2647-2019

Hutter, F., Lücke, J., Schmidt-Thieme, L., 2015. Beyond manual tuning of hyperparameters. KI-Künstliche Intelligenz, 29(4), 329-337. https://doi.org/10.1007/s13218-015-0381-0

Ilunga, M., Stephenson, D., 2005. Infilling streamflow data using feed-forward back-propagation (BP) artificial neural networks: application of standard BP and Pseudo Mac Laurin power series BP techniques. Water SA, 31(2), 171-176. http://dx.doi.org/10.4314/wsa.v31i2.5199

Jeong, J. et al., 2010. Development and integration of sub-hourly rainfall–runoff modeling capability within a watershed model. Water Resources Management, 24(15), 4505-4527. http://doi.org/10.1007/s11269-010-9670-4

Jodar-Abellan, A., Valdes-Abellan, J., Pla, C., Gomariz-Castillo, F., 2019. Impact of land use changes on flash flood prediction using a sub-daily SWAT model in five Mediterranean ungauged watersheds (SE Spain). Science of the Total Environment, 657, 1578-1591. https://doi.org/10.1016/j.scitotenv.2018.12.034

Johanson, R.C., Davis, H.H., 1980. Users manual for hydrological simulation program-Fortran (HSPF), 80. Environmental Research Laboratory, Office of Research and Development, US ….

Jones, D. R., Schonlau, M., & Welch, W. J. 1998. Efficient global optimization of expensive black-box functions. Journal of Global optimization, 13(4), 455-492. https://doi.org/10.1023/A:1008306431147

Jones, E., Oliphant, T., Peterson, P., 2001. SciPy: Open source scientific tools for Python.

Karpatne, A. et al., 2017. Theory-guided data science: A new paradigm for scientific discovery from data. IEEE Transactions on Knowledge and Data Engineering, 29(10), 2318-2331. https://doi.org/10.1109/TKDE.2017.2720168

Karpatne, A., Watkins, W., Read, J., & Kumar, V. 2017b. Physics-guided neural networks (pgnn): An application in lake temperature modeling. arXiv preprint arXiv:1710.11431.

Kim, M., Boithias, L., Cho, K.H., Sengtaheuanghoung, O., Ribolzi, O., 2018. Modeling the impact of land use change on basin-scale transfer of fecal indicator bacteria: SWAT model performance. Journal of environmental quality, 47(5), 1115-1122. http://doi.org/10.2134/jeq2017.11.0456

Kinerson, R.S., Kittle, J.L., Duda, P.B., 2009. BASINS: Better assessment science integrating point and nonpoint sources, Decision Support Systems for Risk-Based Management of Contaminated Sites. Springer, pp. 1-24. http://doi.org/10.1007/978-0-387-09722-0

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., Herrnegger, M., 2018. Rainfall–runoff modelling using long short-term memory (LSTM) networks. Hydrology and Earth System Sciences, 22(11), 6005-6022. https://doi.org/10.5194/hess-22-6005-2018

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., & Nearing, G. 2019. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. arXiv preprint arXiv:1907.08456. https://doi.org/10.5194/hess-23-5089-2019

Kratzert, F., Herrnegger, M., Klotz, D., Hochreiter, S., & Klambauer, G. 2019b. NeuralHydrology–Interpreting LSTMs in Hydrology. In Explainable AI: Interpreting, Explaining and Visualizing Deep Learning (pp. 347-362). Springer, Cham. https://doi.org/10.1007/978-3-030-28954-6_19

Kumar, G.M., Head, T., 2017. Scikit-optimize. Tim Head and contributors

Lakhmiri, D., Digabel, S. L., & Tribes, C. 2019. HyperNOMAD: Hyperparameter optimization of deep neural networks using mesh adaptive direct search. arXiv preprint arXiv:1907.01698.

Le, X.-H., Ho, H.V., Lee, G., Jung, S., 2019. Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. Water, 11(7), 1387.

Li, W., Kiaghadi, A., & Dawson, C. N. 2020. High Temporal Resolution Rainfall Runoff Modelling Using Long-Short-Term-Memory (LSTM) Networks. *arXiv preprint arXiv:2002.02568.* https://doi.org/10.1007/s00521-020-05010-6

lin Hsu, K., Gupta, H.V., Sorooshian, S., 1997. Application of a recurrent neural network to rainfall-runoff modeling, Proceedings of the 1997 24th Annual Water Resources Planning and Management Conference. ASCE. 68-73.

Loukas, A., Vasiliades, L., 2014. Streamflow simulation methods for ungauged and poorly gauged watersheds. Natural Hazards and Earth System Sciences, 14(7), 1641-1661. https://doi.org/10.4296/cwrj2804633

Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2018. Statistical and Machine Learning forecasting methods: Concerns and ways forward. PloS one, 13(3), e0194889. https://doi.org/10.1371/journal.pone.0194889

Mockus, J., Tiesis, V., & Zilinskas, A. 1978. The application of Bayesian methods for seeking the extremum. Towards global optimization, 2(117-129), 2.

Moriasi, D.N., Gitau, M.W., Pai, N., Daggupati, P., 2015. Hydrologic and water quality models: Performance measures and evaluation criteria. Transactions of the ASABE, 58(6), 1763-1785. https://doi.org/10.13031/trans.58.10715

Mosavi, A., Ozturk, P., Chau, K.-w., 2018. Flood prediction using machine learning models: Literature review. Water, 10(11), 1536.

Munna, G. M., Kibriya, N. A., Nury, A. H., Islam, S., & Rahman, H. 2015. Spatial distribution analysis and mapping of groundwater quality parameters for the Sylhet City Corporation (SCC) area using GIS. Hydrology, 3(1), 1-10. https://doi.org/10.11648/j.hyd.20150301.11

Nash, S.G., 1984. Newton-type minimization via the Lanczos method. SIAM Journal on Numerical Analysis, 21(4), 770-788.

Ogwueleka, T.C., Ogwueleka, F.N., 2009. Feed-forward neural networks for precipitation and river level prediction. Adv. Natl. Appl. Sci, 3, 350-356.

Ouellet-Proulx, S., St-Hilaire, A., Boucher, M.A., 2019. Implication of evaporative loss estimation methods in discharge and water temperature modelling in cool temperate climates. Hydrological Processes. https://doi.org/10.1002/hyp.13534

Pang, S., Wang, X., Ma, W., 2018. Research of Parameter Uncertainty for the HSPF Model Under Different Temporal Scales. Huan jing ke xue= Huanjing kexue, 39(5), 2030-2038. http://doi.org/10.13227/j.hjkx.201710070

Papalexiou, S.M., Montanari, A., 2019. Global and Regional Increase of Precipitation Extremes under Global Warming. Water Resources Research.

Park, S. et al., 2019. Deep neural networks for modeling fouling growth and flux decline during NF/RO membrane filtration. Journal of Membrane Science.

Pascual, X. et al., 2013. Data-driven models of steady state and transient operations of spiral-wound RO plant. Desalination, 316, 154-161.
Patin, J. et al., 2018. Effect of land use on interrill erosion in a montane catchment of Northern Laos: An analysis based on a pluri-annual runoff and soil loss database. Journal of hydrology, 563, 480-494.
Prudhomme, C., Williamson, J., 2013. Derivation of RCM-driven potential evapotranspiration for hydrological climate change impact analysis in Great Britain: a comparison of methods and associated uncertainty in future projections. Hydrology and Earth System Sciences, 17(4), 1365-1377.
Reynolds, J., Halldin, S., Xu, C.-Y., Seibert, J., Kauffeldt, A., 2017. Sub-daily runoff predictions using parameters calibrated on the basis of data with a daily temporal resolution. Journal of hydrology, 550, 399-411.
Ribolzi, O. et al., 2017. From shifting cultivation to teak plantation: effect on overland flow and sediment yield in a montane tropical catchment. Scientific reports, 7(1), 3987. https://doi.org/10.1038/s41598-017-04385-2
Ribolzi, O. et al., 2016. Use of fallout radionuclides (7 Be, 210 Pb) to estimate resuspension of Escherichia coli from streambed sediments during floods in a tropical montane catchment. Environmental Science and Pollution Research, 23(4), 3427-3435.
Ribolzi, O. et al., 2018. Interacting land use and soil surface dynamics control groundwater outflow in a montane catchment of the lower Mekong basin. Agriculture, ecosystems & environment, 268, 90-102. https://doi.org/10.1016/j.agee.2018.09.005
Rossman, L.A., 2010. Storm water management model user's manual, version 5.0. National Risk Management Research Laboratory, Office of Research and ….
Roxy, M.K. et al., 2017. A threefold rise in widespread extreme rain events over central India. Nature communications, 8(1), 708.
Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1988. Learning representations by back-propagating errors. Cognitive modeling, 5(3), 1. https://doi.org/10.1038/323533a0
Samek, W. (2019). Explainable AI: interpreting, explaining and visualizing deep learning (Vol. 11700). Springer Nature.
Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. 2015. Taking the human out of the loop: A review of Bayesian optimization. Proceedings of the IEEE, 104(1), 148-175.
Shen, C., 2018. A transdisciplinary review of deep learning research and its relevance for water resources scientists. Water Resources Research, 54(11), 8558-8593. https://doi.org/10.1029/2018WR022643
Singh, V.P., 1988. Hydrologic systems: Rainfall-runoff modeling.
Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., ... & Adams, R. 2015. Scalable bayesian optimization using deep neural networks. In International conference on machine learning. 2171-2180.
Spruill, C.A., Workman, S.R., Taraba, J.L., 2000. Simulation of daily and monthly stream discharge from small watersheds using the SWAT model. Transactions of the ASAE, 43(6), 1431. https://doi.org/10.13031/2013.3041
Stern, M., Flint, L., Minear, J., Flint, A., Wright, S., 2016. Characterizing changes in streamflow and sediment supply in the Sacramento River Basin, California, using hydrological simulation program—FORTRAN (HSPF). Water, 8(10), 432. https://doi.org/10.3390/w8100432
Talei, A., Chua, L.H., 2012. Influence of lag time on event-based rainfall–runoff modeling using the data driven approach. Journal of Hydrology, 438, 223-233.

Wang, N., Zhang, D., Chang, H., & Li, H. 2020. Deep learning of subsurface flow via theory-guided neural network. Journal of Hydrology, 584, 124700. https://doi.org/10.1016/j.jhydrol.2020.124700

Wang, Y., He, B., Takase, K., 2009. Effects of temporal resolution on hydrological model parameters and its impact on prediction of river discharge/Effets de la résolution temporelle sur les paramètres d'un modèle hydrologique et impact sur la prévision de l'écoulement en rivière. Hydrological sciences journal, 54(5), 886-898.

Xie, H., Lian, Y., 2013. Uncertainty-based evaluation and comparison of SWAT and HSPF applications to the Illinois River Basin. Journal of Hydrology, 481, 119-131. https://doi.org/10.1016/j.jhydrol.2012.12.027

Yan, L., Feng, J., Hang, T., 2019. Small Watershed Stream-Flow Forecasting Based on LSTM, International Conference on Ubiquitous Information Management and Communication. Springer. 1006-1014.

Yaseen, Z.M., El-Shafie, A., Jaafar, O., Afan, H.A., Sayl, K.N., 2015. Artificial intelligence based models for stream-flow forecasting: 2000–2015. Journal of Hydrology, 530, 829-844. https://doi.org/10.1016/j.jhydrol.2015.10.038

Yeh, H.-F., 2017. Comparison of evapotranspiration methods under limited data. In: Bucur, D. (Ed.), Current perspective to predict actual evapotranspiration. https://doi.org/10.5772/intechopen.68495

Zhang, J., Zhu, Y., Zhang, X., Ye, M., Yang, J., 2018. Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas. Journal of hydrology, 561, 918-929. https://doi.org/10.1016/j.jhydrol.2018.04.065

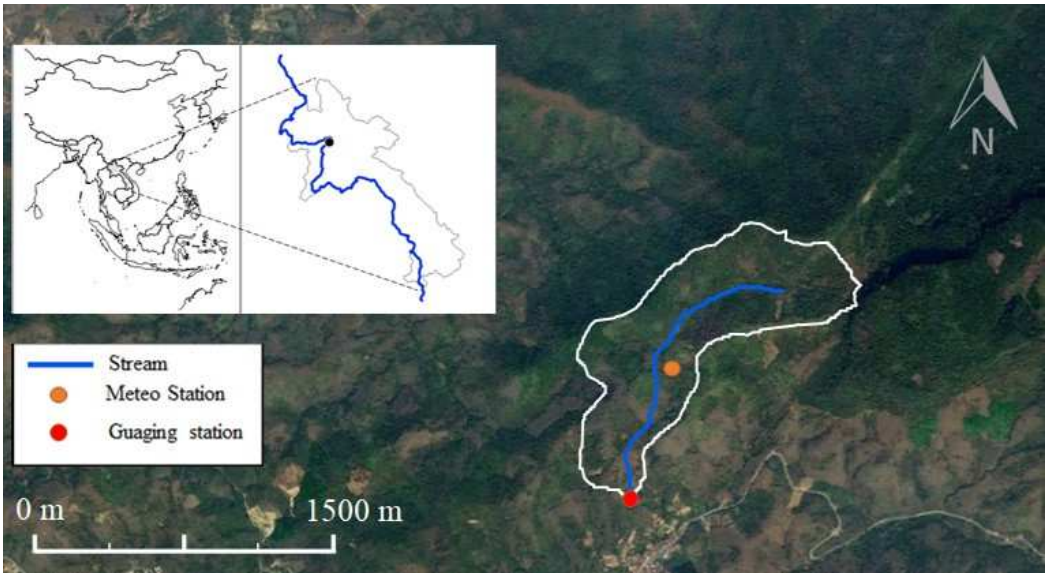

Figure 1. Location of the Houay Pano catchment in northern Lao P.D.R. and the observation stations. Topographic map is taken from Google Earth (19°51′29.49″ N, 102°10′21.51″ E) (Google, 2014).

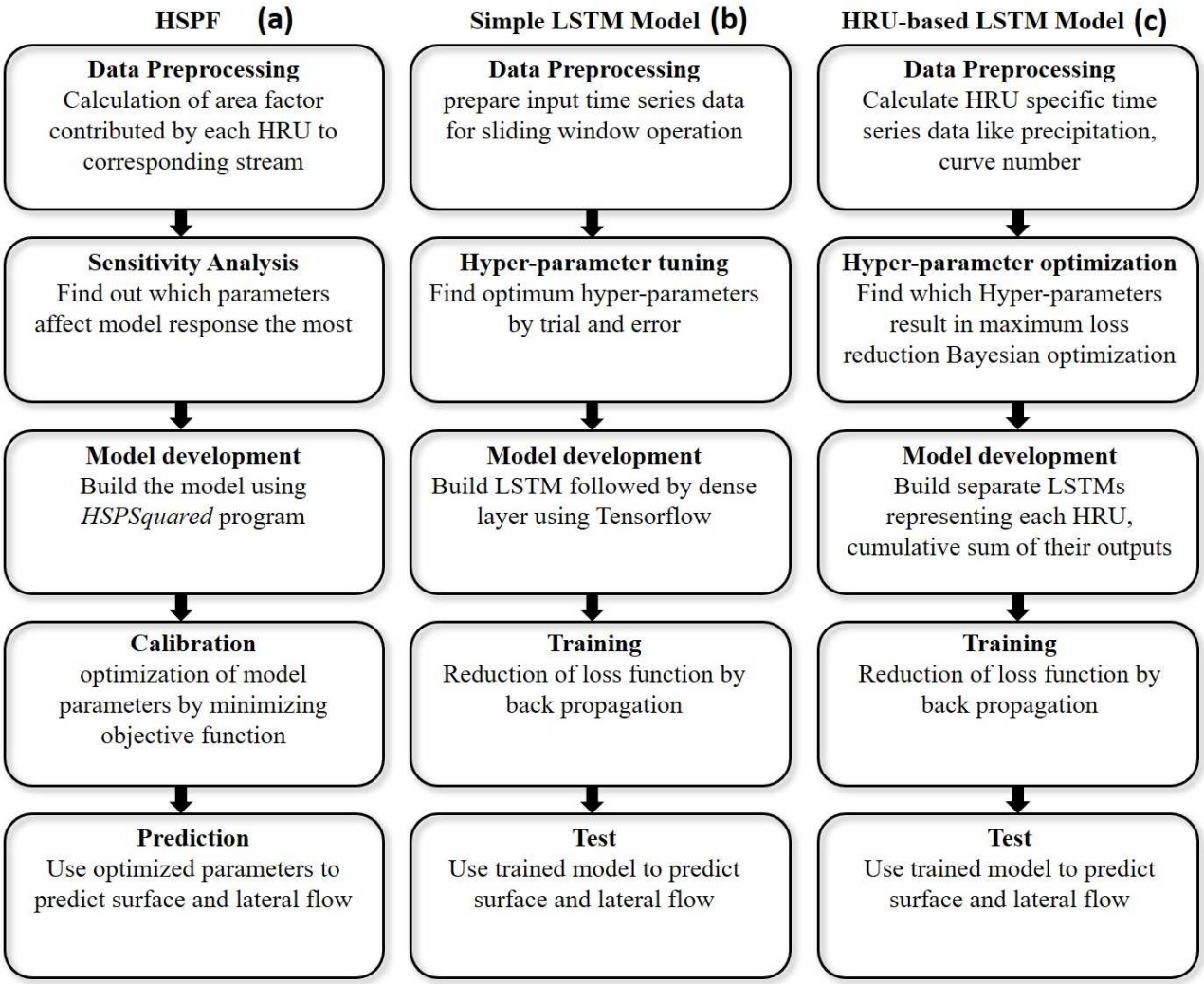

Figure 2. Framework for developing (a) HSPF, (b) Simple LSTM Model, and (c) HRU-based LSTM Model. The simple LSTM model consists of one Long Short-Term Memory (LSTM) layer representing the whole catchment, while the HRU-based LSTM model consists of separate LSTMs for each hydrological response unit (HRU). LSTMs are a special kind of neural network specialized for learning patterns in time dependent data. An HRU in the HRU-based model is defined as 'a distinct land use in a distinct sub-basin'. The output from each of these LSTMs is added in a cumulative manner.

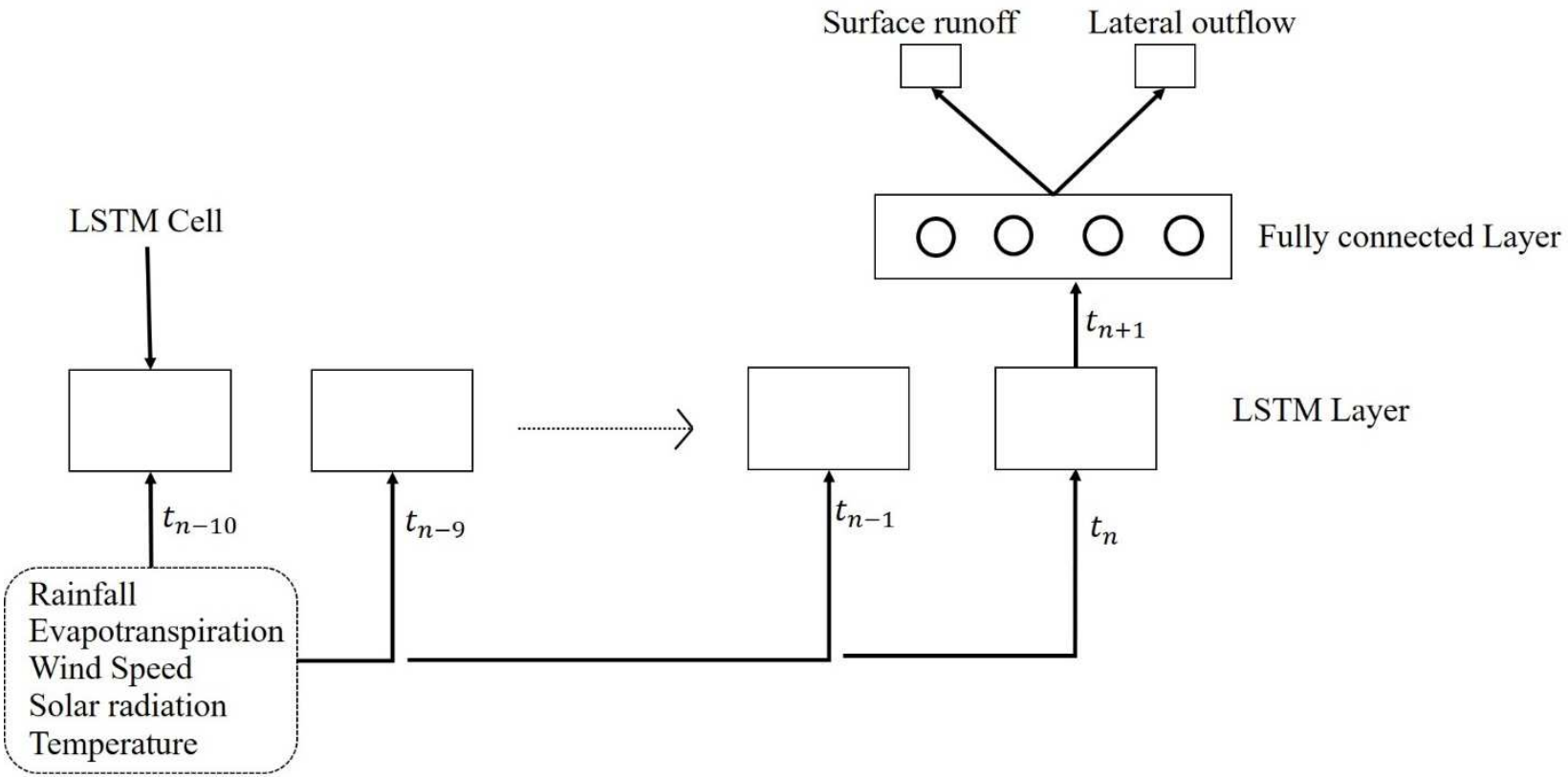

Figure 3. Structure of a simple LSTM consisting of an LSTM layer followed by a fully connected layer which generates two outputs. The two outputs are supposed to be representative of surface and sub-surface flow. The LSTM layer consists of LSTM cells, each of which take input at a particular time-step. The diagram represents the working of an NN at time-step 't_n' to produce output at t_{n+1} using a window size of 10.

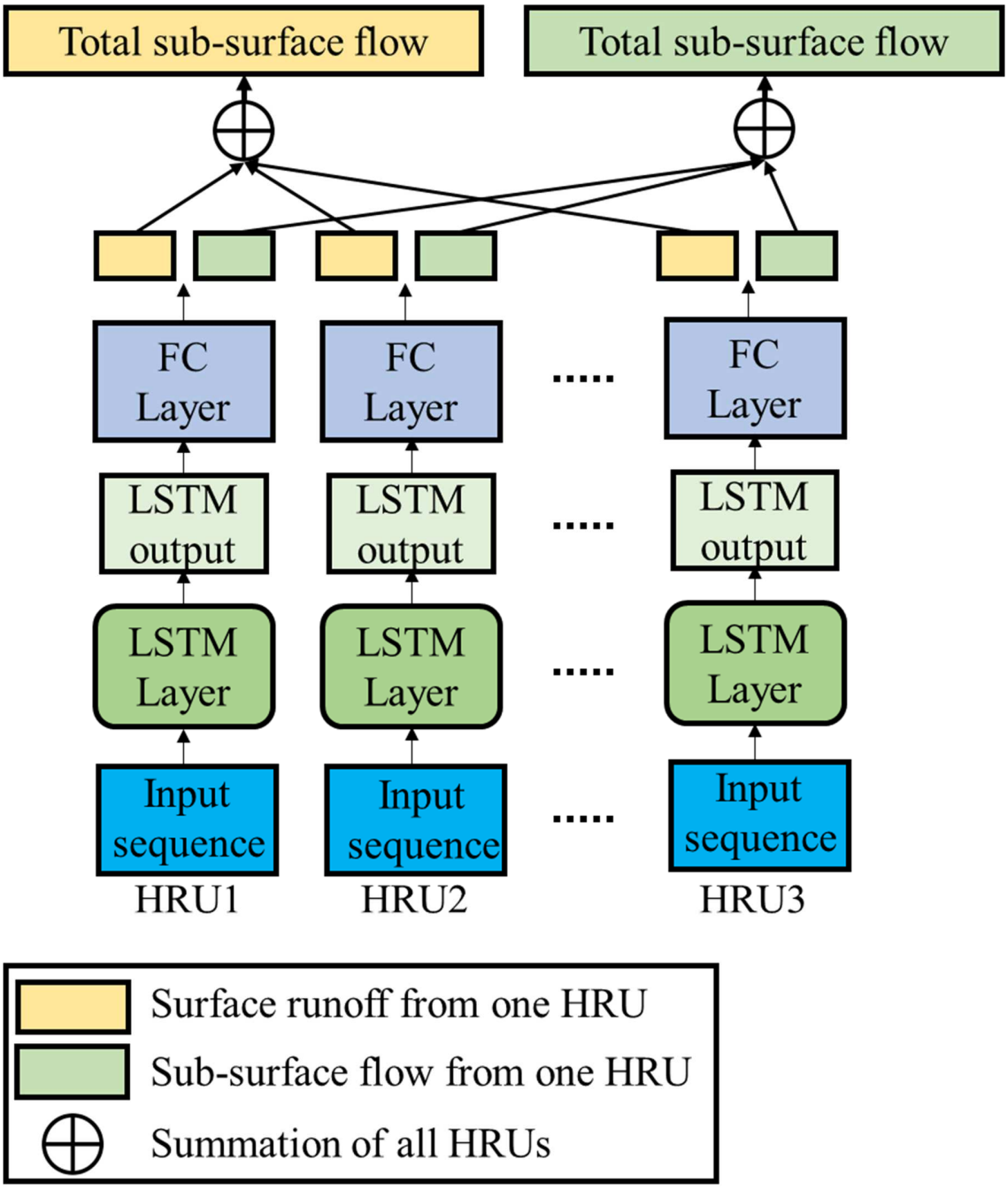

Figure 4. Structure of HRU-based LSTM model consisting of 36 parallel layers. Each layer represents one HRU in the catchment. Every LSTM is followed by a dense layer which produces two outputs. These two outputs represent surface and sub-surface flow from one HRU. Finally, surface and sub-surface flow are summed together to get total surface runoff and sub-surface flow.

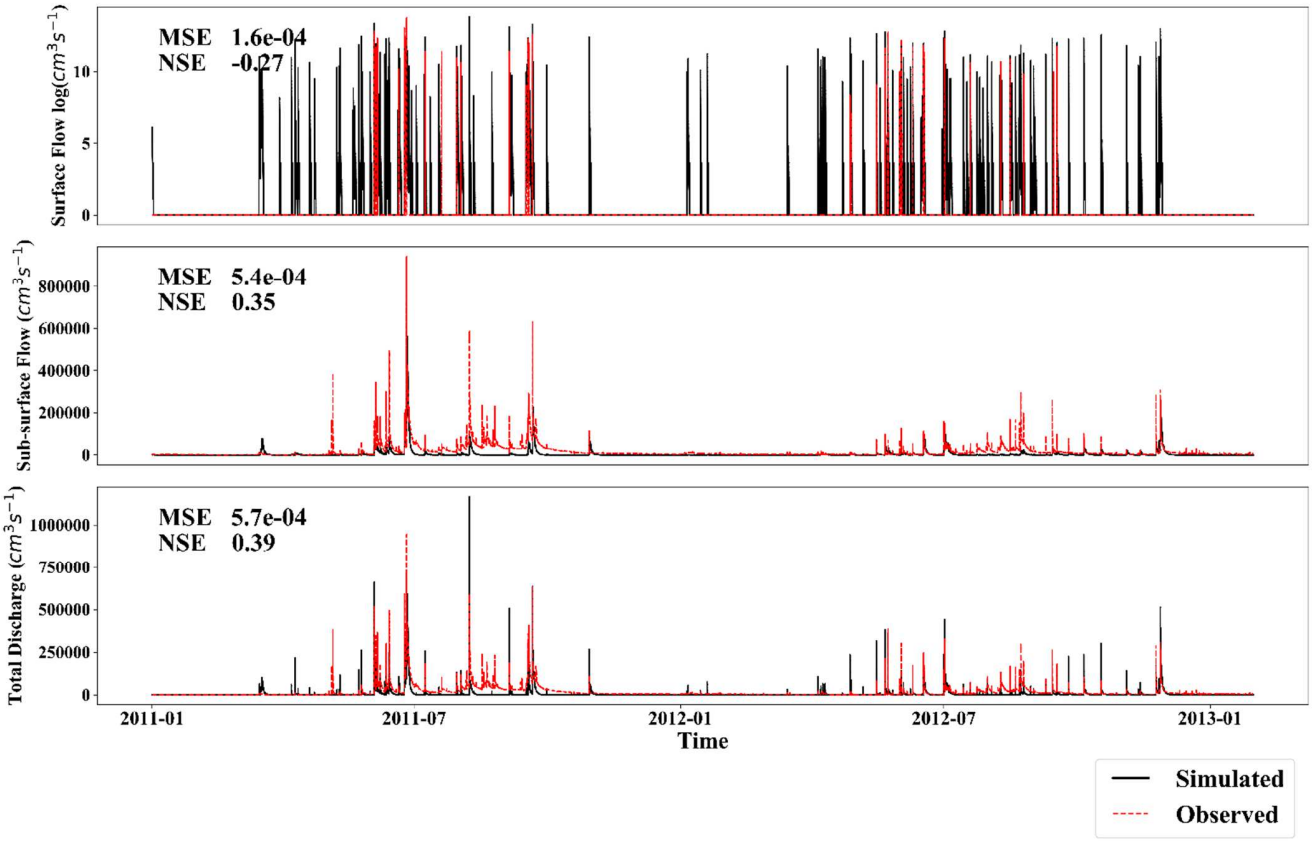

Figure 5. Calibration of surface runoff (m^3s^{-1}), sub-surface flow (m^3s^{-1}), and total discharge (m^3s^{-1}) using HSPF model.

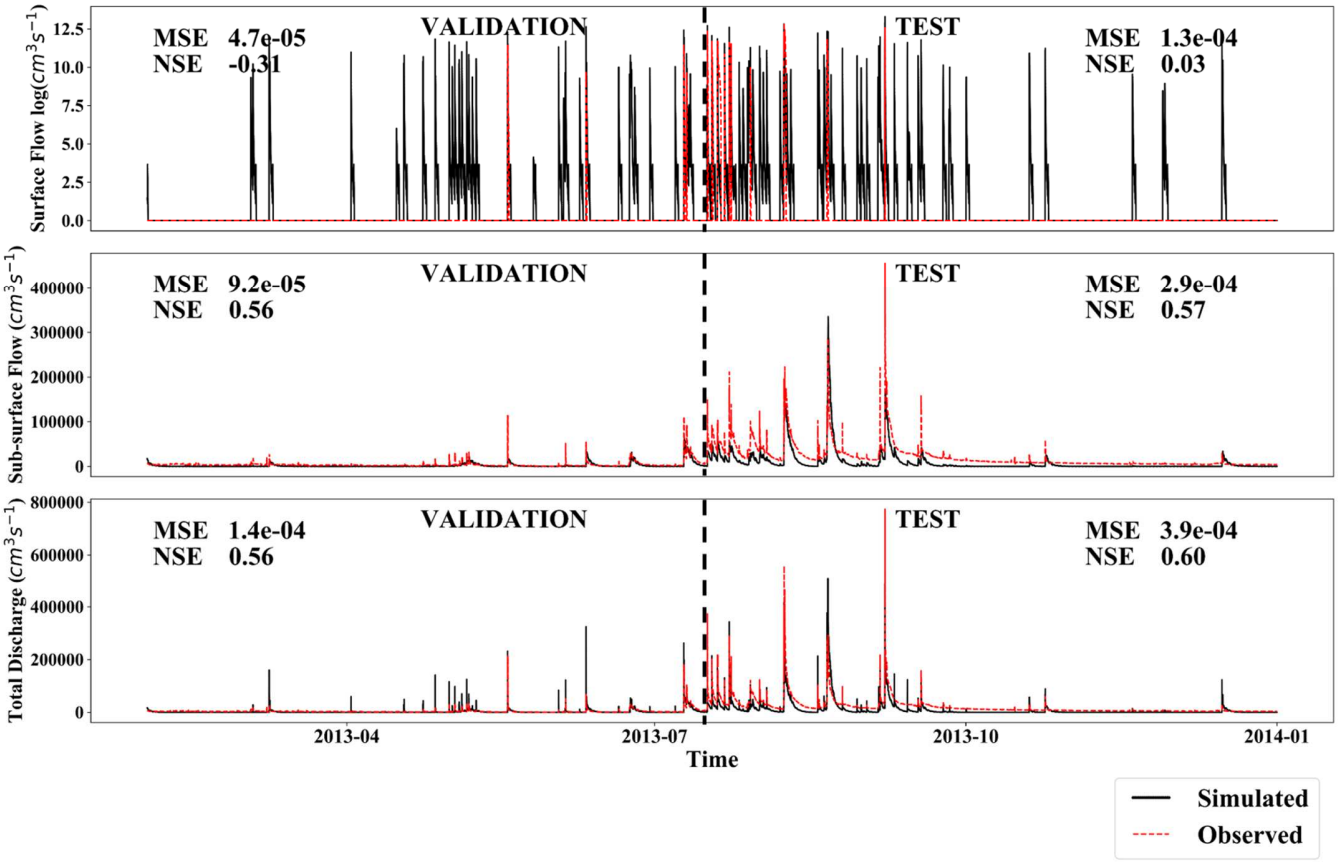

Figure 6. Prediction of surface runoff (m^3s^{-1}), sub-surface flow (m^3s^{-1}), and total discharge (m^3s^{-1}) using calibrated HSPF model.

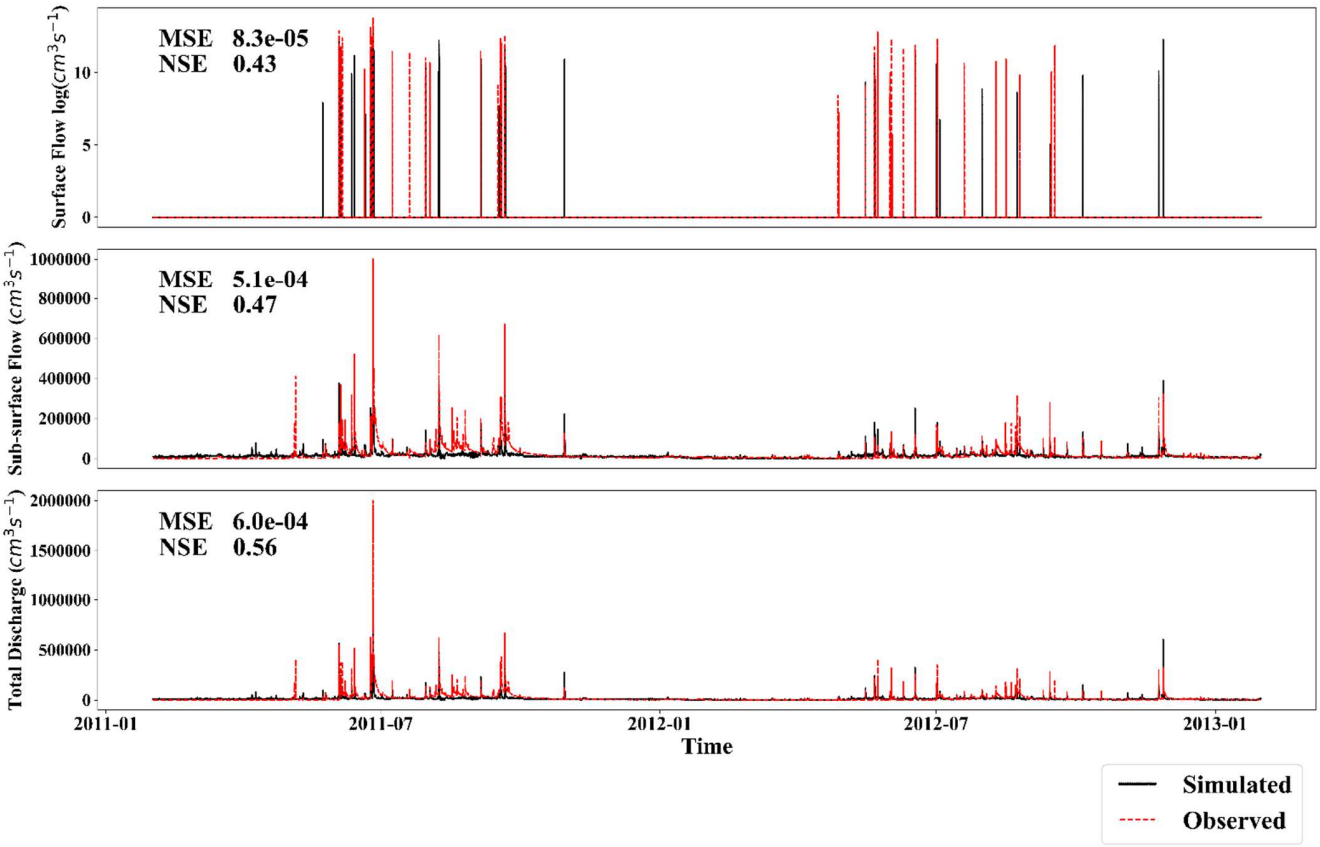

Figure 7 Calibration of surface runoff (m^3s^{-1}), sub-surface flow (m^3s^{-1}), and total discharge (m^3s^{-1}) using simple LSTM model.

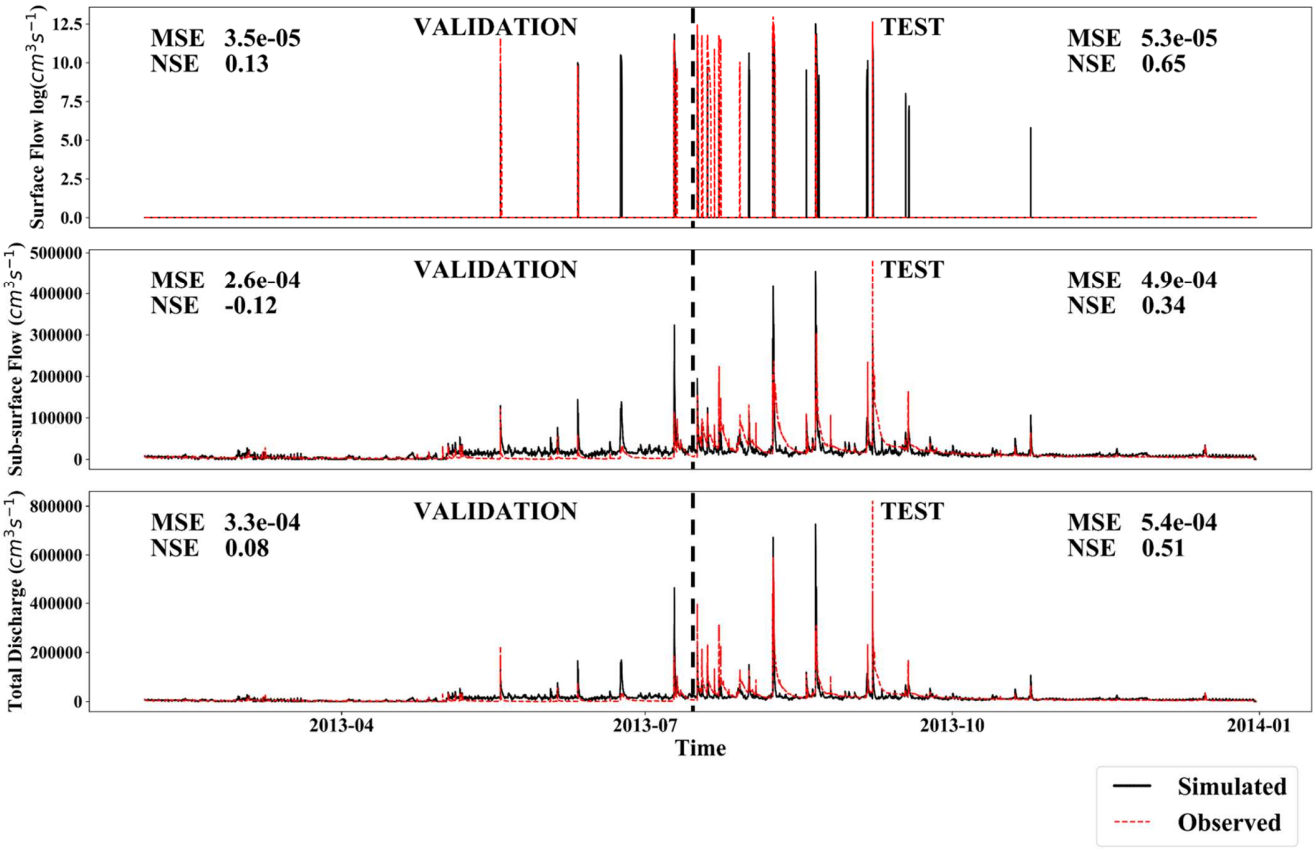

Figure 8 Prediction surface runoff (m^3s^{-1}), sub-surface flow (m^3s^{-1}), and total discharge (m^3s^{-1}) using calibrated simple LSTM model.

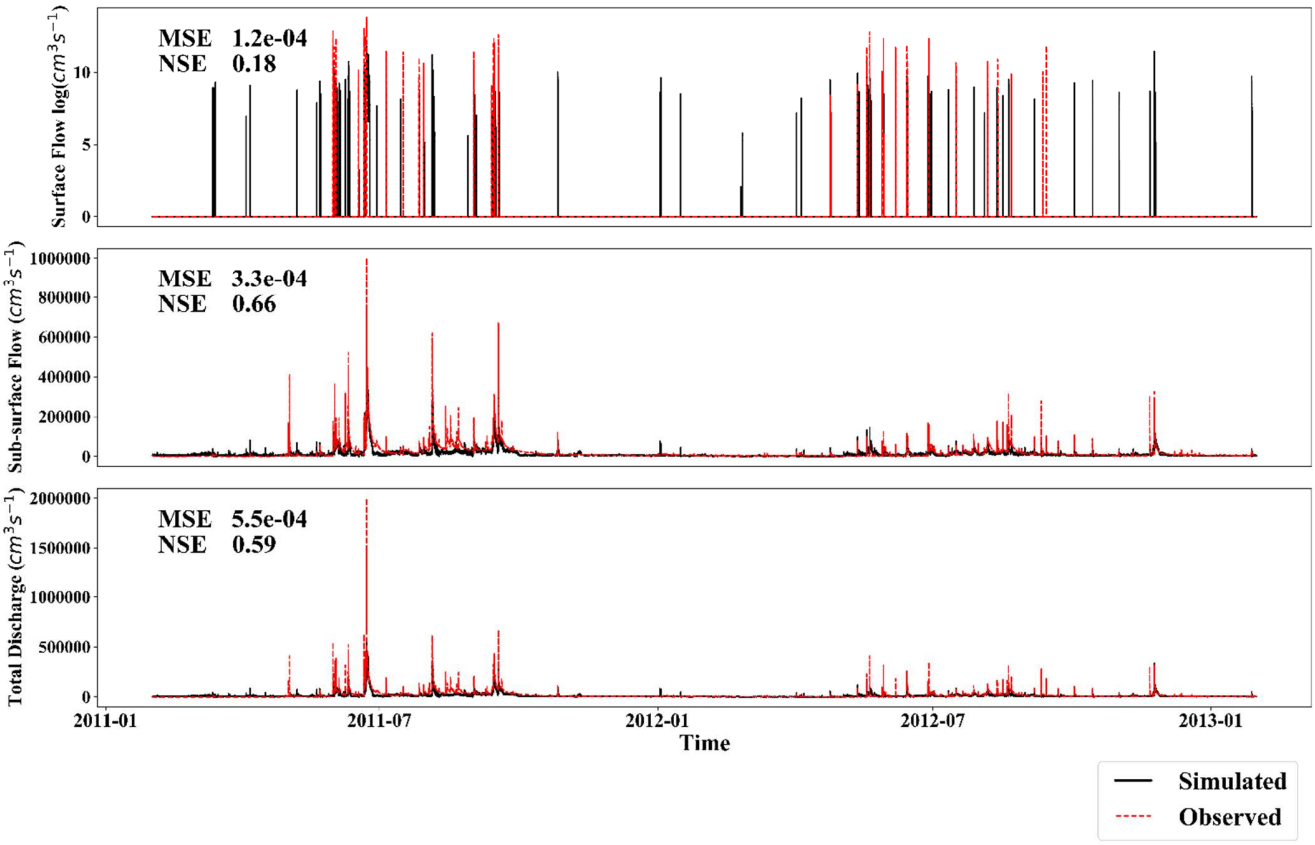

Figure 9 Calibration of surface runoff (m^3s^{-1}), sub-surface flow (m^3s^{-1}), and total discharge (m^3s^{-1}) using HRU-based LSTM model.

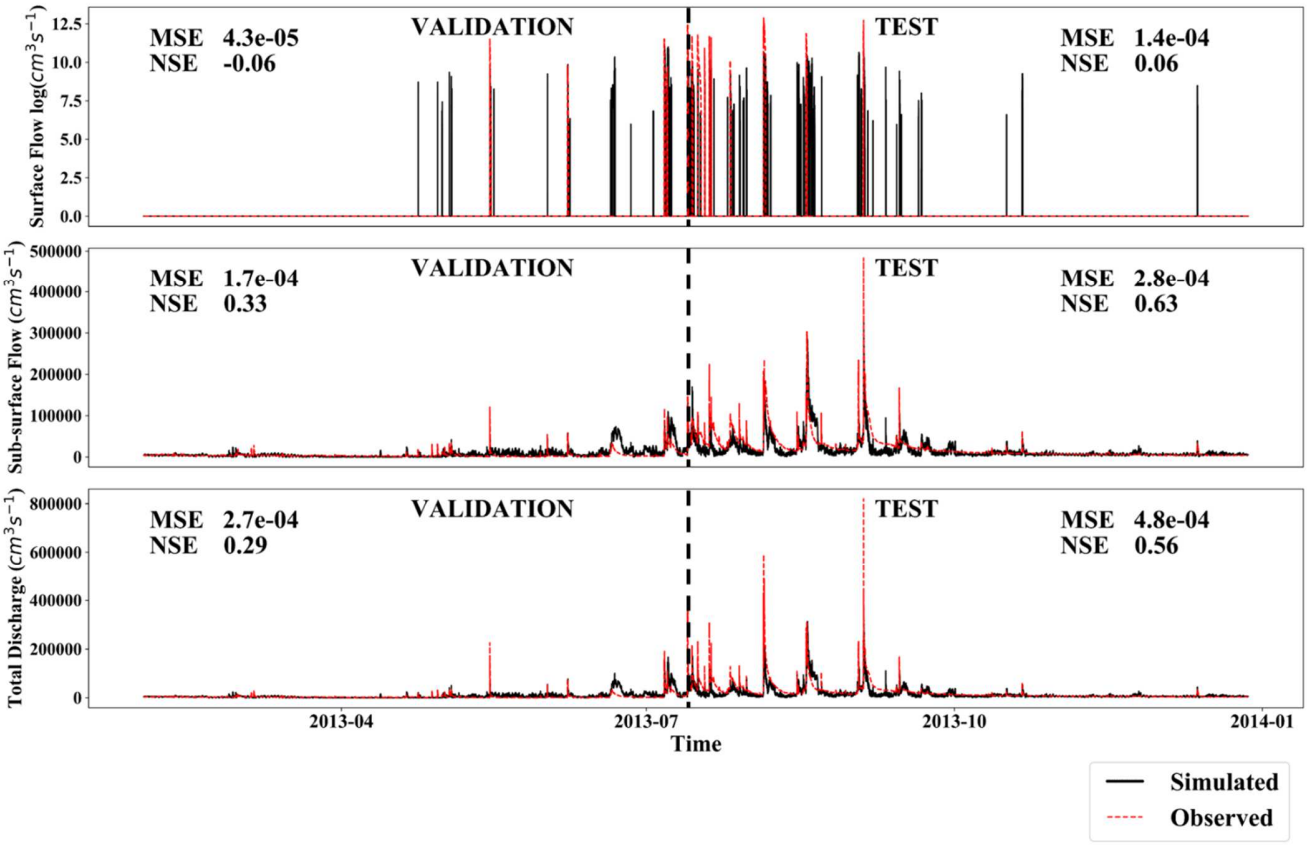

Figure 10 Prediction of surface runoff (m^3s^{-1}), sub-surface flow (m^3s^{-1}), and total discharge (m^3s^{-1}) using calibrated HRU-based LSTM model.

Table 1 Hyper-parameters used for building simple and HRU-based LSTM models.

Parameter	**Value for simple model**	**Values for HRU-based model**
Activation function	Rectified Linear Unit	Rectified Linear Unit
Batch size	64	64
Learning rate	1e-5	1e-5
Sequence length	13 h	80 h
Hidden units	128	128
Input data	Precipitation moving average, air temperature, precipitation, wind speed, solar radiation	Precipitation moving average, air temperature, precipitation, wind speed, solar radiation, potential evapotranspiration
HRU specific data	None	Rainfall, distance to outlet, curve number
Calibration epochs	214	66

Table 2 Results of sensitivity analysis of parameters for sub-surface flow and total discharge.

Sub-surface Flow				Total Discharge			
Parameters	**Land use**	**Sensitivity Rank**	**Calibrated values**	**Parameters**	**Land use**	**Sensitivity Rank**	**Calibrated values**
AGWETP	Fallow	1	0.066	UZSN	Forest	1	2.0
AGWETP	Teak	2	0.0	UZSN	Fallow	2	0.05
AGWETP	Annual crops	3	0.2	INTFW	Forest	3	10.0
AGWETP	Forest	4	0.133	INTFW	Fallow	4	10.0
INFILT	Fallow	5	0.5	UZSN	Teak	5	1.35
INFILT	Forest	6	0.336	INFILT	Forest	6	0.336
INFILT	Annual crops	7	0.5	INFILT	Fallow	7	0.5
INFILT	Teak	8	0.336	INTFW	Teak	8	4.0
BASETP	Forest	9	0.066	UZSN	Annual crops	9	2.0
BASETP	Annual crops	10	0.0	INFILT	Teak	10	0.336
DEEPFR	Fallow	11	0.16	INTFW	Annual crops	11	7.0
DEEPFR	Teak	12	0.0	INFILT	Annual crops	12	0.5

Table 3. Performance matrix of HSPF, simple NN, and HRU-based LSTM model. Bold numbers represent values that fall under the category of 'satisfactory' after Moriasi et al., (2015). MSE is measured in units of meter cube per second (m^3s^{-1}).

Model Type	Flow Type	Training			Validation			Test		
		MSE	**NSE**	**PBIAS**	**MSE**	**NSE**	**PBIAS**	**MSE**	**NSE**	**PBIAS**
HSPF	Surface runoff	1.6e-4	-0.27	45.8	4.7e-5	-0.31	69	1.3e-4	0.03	26
	Sub-surface flow	5.4e-4	0.35	-70.8	9.2e-5	**0.56**	-54	2.9e-4	**0.57**	-55
	Total discharge	5.7e-4	0.39	-66	1.4e-4	**0.56**	-47	3.9e-4	**0.60**	-52
Simple	Surface runoff	8.3e-5	0.43	**-5.6**	3.5e-5	0.13	-53	5.3e-5	**0.65**	**-3.2**
LSTM	Sub-surface flow	5.1e-4	0.47	**-14**	2.6e-4	-0.12	64	4.9e-4	0.34	-16
	Total discharge	6.0e-4	**0.56**	**-14.4**	3.3e-5	0.08	57	5.4e-5	**0.51**	**-15**
HRU-based	Surface runoff	1.2e-4	0.18	-63	4.3e-5	-0.06	54	1.4e-4	0.06	-67
LSTM	Sub-surface	3.3e-4	**0.66**	-22	1.7e-4	0.33	10	2.8e-4	**0.63**	17
	Total	5.5e-4	**0.59**	-23	2.7e-4	0.29	**7**	4.8e-4	**0.56**	-19

Table 4 Hyper-parameters and their optimization. The optimization of these parameters was carried out for 5 scenarios. During optimization, parameters such as learning rate, hidden units, and NN layers were varied in the ranges given Table S3. Validation MSE represents the MSE value to which the algorithm converged for each sequence length.

Sequence length	**Learning rate**	**Hidden units**	**NN layers**	**Normalization**	**Loss calculation method**	**Type of NN cell**	**Activation function**	**Validation MSE** (m^3s^{-1})
20	1.164e-6	128	4	True	Weighted	LSTM	ReLu	6.3e-3
30	3.07e-4	128	3	True	Normal	LSTM	ReLu	6.0e-3
40	1.123e-5	128	1	True	Weighted	LSTM	ReLu	6.2e-3
50	1.s2-5	128	1	True	Weighted	LSTM	Tanh	6.5e-3
60	5.441e-5	128	4	True	Normal	LSTM	ReLu	6.4e-3