



Data assimilation for streamflow forecasting using Extreme Learning Machines and Multilayer Perceptrons

M.-A. Boucher¹, J. Quilty², and J. Adamowski³

¹Département de génie civil et de génie du bâtiment, Université de Sherbrooke, 2500 Boul. de l'Université, Sherbrooke,

Québec, J1R 2R2, Canada

²Department of Civil and Environmental Engineering, University of Waterloo, 200 University Ave W, Waterloo, Ontario,

N2L 3G1, Canada

³Department of Bioresource Engineering, McGill University, 2111 Lakeshore, Ste-Anne de Bellevue, Quebec, H9X3V9,

Canada

Key Points:

- Neural network-based methods are proposed for assimilating state variables in conceptual hydrological models
- Both Multilayer Perceptrons and Extreme Learning Machines are shown to generate accurate starting points for streamflow forecasts
- Multilayer Perceptron ensembles provided more reliable estimates of state variable uncertainty than Extreme Learning Machine ensembles

Corresponding author: Marie-Amélie Boucher, marie-amelie.boucher@usherbrooke.ca

This article has been accepted for publication and undergone full peer review but has not been through the copyediting, typesetting, pagination, and proofreading process which may lead to differences between this version and the Version of Record. Please cite this article as doi: 10.1029/2019WR026226

17 **Abstract**

18 Data assimilation allows for updating state variables in a model to represent the initial
 19 condition of a catchment more accurately than the initial Open Loop simulation. In hy-
 20 drology, data assimilation is often a pre-requisite for forecasting. According to *Hornik*
 21 [1991], artificial neural networks can learn any nonlinear relationship between inputs and
 22 outputs. Here, we hypothesize that neural networks could learn the relationship between
 23 the simulated streamflow (from a hydrological model) and the corresponding state va-
 24 riables. Once learned, this relationship can be used to obtain corrected state variables by
 25 applying it to observed rather than simulated streamflow. Based on this, we propose a no-
 26 vel, ensemble-based, data assimilation approach. As a proof of concept and to verify the
 27 above mentioned hypothesis, we used an international testbed comprising four hydrologi-
 28 cally dissimilar catchments. We applied the new data assimilation method to the lumped
 29 hydrological model GR4J, which has two state variables. Within this framework, we com-
 30 pared two types of neural networks, namely Extreme Learning Machine and the Multilayer
 31 Perceptron. Using well-known metrics such as the Continuous Ranked Probability Score,
 32 we compared the assimilated streamflow series with the Open Loop streamflow series and
 33 with the observed streamflow. We show that neural networks can be successfully used for
 34 data assimilation, with a noticeable improvement over the Open Loop simulation for all
 35 catchments.

36 **1 Introduction**

37 Data assimilation refers to any process that allows for updating state variables in a
 38 model to represent reality more accurately than the initial (or "Open Loop") simulation. In
 39 hydrology, data assimilation is often a pre-requisite for forecasting, in order for the fore-
 40 casts to start from the best possible initial state of the model. The most rudimentary form
 41 of data assimilation, called direct insertion, consists in replacing one or more simulated
 42 state variables by observations when they become available. For instance, many hydrolo-
 43 gical models compute snow water equivalent as a state variable, but this state variable can
 44 also be measured. Direct insertion attributes full credibility to the measurements and none
 45 to the open loop simulation.

46 However, because measurements are also uncertain, there exist systematic data assi-
 47 milation methods that seek a compromise between the Open Loop simulation and in-situ
 48 measurements. Some of the most well-known data assimilation methods in hydrology are
 49 the Ensemble Kalman Filter [EnKF, *Evensen*, 1994] and the particle filter [PF, *Doucet*
 50 *et al.*, 2000]. So far, there is no consensus as to the best method, or how to implement
 51 these methods efficiently to a variety of catchments that have different characteristics [*Liu*
 52 *et al.*, 2012].

53 Artificial neural networks (or simply neural networks, NN) can learn any nonlinear
 54 relationship between inputs and outputs *Hornik* [1991]. Here, we hypothesize that an en-
 55 semble of neural networks could learn the relationship between the simulated streamflow
 56 (from a hydrological model) and the corresponding state variables. Once learned, this re-
 57 lationship could be applied to observed rather than simulated streamflow, and therefore
 58 be used to obtain corrected state variables. The ensemble of neural networks, if properly
 59 constructed, could also capture the uncertainty related to the state variables. Based on
 60 this, we propose a novel, ensemble-based, data assimilation approach. This approach could
 61 be applied using any type of machine learning method, although this research focuses on
 62 neural networks due to their widespread use in hydrology [*Abrahart R.J. et al.*, 2012]. In
 63 particular, we use two different types of neural networks : Multilayer Perceptrons and Ex-
 64 treme Learning Machines. The Multilayer Perceptron (MLP) is the most popular type of
 65 neural network in hydrology, whereas Extreme Learning Machine (ELM) is a newer type
 66 of neural network that has the advantage of being faster and simpler to implement. Contra-
 67 rarily to MLP, ELM does not require iterative training [calibration, *Huang et al.*, 2012].

Machine learning (ML) techniques have been used to solve hydrology-related problems for more than 25 years, the most common application being streamflow forecasting [e.g., *Boucher et al.*, 2010; *Abrahart R.J. et al.*, 2012; *Lima et al.*, 2016]. As noted in *Abrahart R.J. et al.* [2012], with very few exceptions, operational forecasting agencies do not use ML techniques for streamflow forecasting. One possible explanation is the "black-box" nature of ML-based rainfall-runoff models. Indeed, operational forecasters and water resources managers are held accountable for their decisions. They must be able to support those decisions on sound reasoning based on hydrological processes, and it is important to "get the right answer for the right reason" [Kirchner, 2006]. Still, even some recent papers in hydroinformatics continue to advocate ideas such as : «The primary reason to promote the use of data-driven models lies in that many data-driven models are capable of capturing non-linear processes numerically *without fully understanding the underlying physical processes involved*»[*Kasiviswanathan et al.*, 2016, *italics ours*].

In the present research, ML techniques do not replace conceptual or physics-based models. Rather, they improve their estimates of state variables. Hence, ML supports the hydrological model rather than replacing it. That way, operational agencies could benefit from the powerful capabilities of ML to improve their forecasts, without ever sacrificing process understanding.

Through this type of combined use of ML techniques and conceptual (or, eventually, physics-based) hydrological models, similar to those in operation in many forecasting agencies, we hope to bring together two communities that typically evolve in parallel to one another.

To the best of our knowledge, only very few studies have explored the learning ability of neural networks for data assimilation purposes in the context of hydrological forecasting. There are numerous examples in the literature where neural networks have been used to process remotely sensed data by converting this information into input variables that are more readily usable in atmospheric and/or hydrological models. For instance, *Rodríguez-Fernández et al.* [2019] used neural networks to convert SMOS brightness temperatures into soil moisture data to be used in land surface models. From a hydrological perspective, this type of study could be categorized more accurately as pre-processing rather than data assimilation.

In hydrology, *Siek and Solomatine* [2011] used a nonlinear auto-regressive neural network with exogenous inputs (NARX) to combine observations with outputs from a storm surge forecasting model. Although they successfully improved forecasts, this technique left the open-loop state variables unchanged. Therefore, their method can be seen as a post-processing tool rather than a data assimilation method. In operational hydrology, where meteorological forecasts provide insights regarding future weather, state variables represent the "starting point" for the forecasting period, and it is important that they are as accurate as possible.

Similarly, in a recent paper, *Snauffer et al.* [2018] used MLP to combine six snow-related products into one final snow water equivalent (SWE) gridded product covering the entire province of British Columbia (Canada). The resulting SWE maps (or grids) can be used to assist hydrological forecasting and decision-making. They compared those grids to in situ data obtained from manual snow surveys (386 sites) and found that merging the three most informative data sets (namely MERRA, GLDAS2 and ERA-Land) via MLP led to the best results when compared to in situ observations. The merged grids also outperformed the individual grids based on the three different products. Although the appellation "assimilation" is used throughout the paper from *Snauffer et al.* [2018], their method does not update the model's state variables using the latest observations. Hence, the method could perhaps again be more accurately categorized as a post-processing method. Another similar study can be found in *Chang et al.* [2014], where ML techniques were used to merge precipitation observations from multiple sensors.

In Cintra *et al.* [2013], Härter and de Campos Velho [2008] and Härter and de Campos Velho [2012], neural networks were trained to emulate the results obtained by a Kalman filter rather than being developed as a data assimilation technique in itself, in contrast to this study where neural networks are used to directly assimilate state variables of a hydrological model. Furthermore, these earlier studies focused on meteorological variables and models rather than this study, which focuses on streamflow and corresponding hydrological models.

Wahle *et al.* [2015] were the first to demonstrate the capability of neural networks for data assimilation in wave modeling. In their study, the COSYNA WAM Cycle 4.5.4 model was used to simulate wave height, direction, period and other characteristics for the German Bight area in the North Sea. Similarly to an atmospheric model or a hydrological model, the COSYNA WAM Cycle 4.5.4 model uses state variables to describe the state of the system at each time step of the simulation. Wahle *et al.* [2015] first used MLP to model a relationship between the wave model's state variables and the simulated wave parameters. Then, once this relationship was correctly learned, they inverted the neural network so that state variables could be computed from wave parameters. One key element of their approach is that a MLP was trained using simulated streamflow, which was derived from the Open Loop state variables. Consequently, the "true" relationship between state variables and wave parameters were learned. Thus, assuming that model structure is correct and that state variable uncertainty is much greater than model structure uncertainty, the inverse neural network model fed with observed wave parameter values can be used to estimate the "true" value of state variables. However, the relationship between the model's state variables and the outputs is not objective : there exist an ensemble of possible state vectors that could lead to the same model output. For this reason, Wahle *et al.* [2015] explain that they had to use two inverse neural networks instead of one. The first inverse model was used to derive boundary values, which were then used to further constrain the second inverse model, aimed at correcting state variables. Wahle *et al.* [2015] are not the first to apply neural networks to inverse problem solving, but, to the best of our knowledge, they are the first to explicitly aim at using neural networks to update the state variables of a model, similar to the models used in hydrology.

Aires *et al.* [2001] proposed an approach based on MLP with a "first guess" to reconstruct atmospheric and surface parameters from satellite-based microwave observations and a radiative transfer model. The "first guess" is an initial rough estimate of the parameters they are attempting to retrieve, which can, for instance, be modeled by the radiative transfer model. This introduction of a first guess is very useful to constrain the solution in the context of ill-posed inverse problems, where more than one set of inputs can lead to the same output(s).

Based on this literature review, this study is the first to propose the use of neural networks (and, in general, machine learning) for data assimilation in operational hydrology. The main purpose of our proposed approach is to update the state variables of a conceptual rainfall runoff model using the most recent hydro-meteorological observations. Contrary to Cintra *et al.* [2013], Härter and de Campos Velho [2008] and Härter and de Campos Velho [2012], we do not emulate another data assimilation method. Instead, we use an ensemble of neural networks to directly derive a (new) data assimilation method. In contrast to Siek and Solomatine [2011] and Snauffer *et al.* [2018], our method updates the state variables of the model and is not an *a posteriori* merging of different sources of information. Lastly, in comparison to the approach proposed in Wahle *et al.* [2015], our method allows for estimating the uncertainty of state variables, which is known to be an important source of uncertainty when forecasting lead times are shorter than the response time of a catchment [e.g., Thiboult *et al.*, 2016].

Therefore, the main goal of this study is to provide a proof of concept of our new data assimilation method and verify the following two hypotheses :

172 1. Neural networks can be used to accurately perform data assimilation (updating
 173 the state variables according to available hydro-meteorological observations) in a
 174 simple rainfall-runoff model.

175 2. Extreme Learning Machines outperform Multilayer Perceptrons for data assimila-
 176 tion in terms of performance assessment metrics typically used by the hydrologi-
 177 cal ensemble forecasting community (CRPS, reliability diagram, etc.).

178 In order to verify hypothesis 1, we compare both the Open Loop simulation (without
 179 data assimilation) and the simulation including data assimilation to corresponding stream-
 180 flow observations, on four very different catchments via performance assessment metrics
 181 (CRPS, reliability diagrams, etc.). To verify hypothesis 2, we use two different types of
 182 neural networks, namely Extreme Learning Machines and Multilayer Perceptrons, to solve
 183 identical data assimilation problems and we compare their performance (also against the
 184 series of observed streamflow via the same performance assessment metrics).

185 In the following sections, we first provide a description of the experimental set-up,
 186 including the rainfall-runoff model we use, the snow accumulation model, the different
 187 types of neural networks, as well as the performance assessment metrics. Details about the
 188 catchments and data are given in section 3. Experiment results are presented and discussed
 189 in section 4. Finally, our conclusions and recommendations for future work are given in
 190 section 5.

191 2 Experimental setup

192 The experimental setup used for this study is schematically represented on Figure
 193 1. It is first broadly described here, and then each component is further detailed in sub-
 194 sections 2.1 to 2.4.

195 There are two main methodological steps involved in this study. The first one consists
 196 of training the neural networks until they accurately model the relationship between hydro-
 197 meteorological variables (such as previous days' streamflow and meteorological variables
 198 at several lag times) and the corresponding state variables of the hydrological model (here
 199 GR4J, see section 2.1). This training or learning process is illustrated on the left hand side
 200 of Figure 1. The hydrological model has to be previously calibrated before the neural net-
 201 works can be trained. The main assumption for this first methodological step is that the
 202 relationship between the modeled streamflow and corresponding state variables is the same
 203 as the relationship between the *observed* streamflow and the corresponding, but unknown,
 204 state variables that we wish to estimate. In order to account for the fact that this relation-
 205 ship cannot be modelled with certainty, an ensemble of 100 neural networks is used rather
 206 than a single one. Each neural network forming the ensemble is initialized independently.
 207 The distribution of the ensemble members should therefore reflect a portion of the uncer-
 208 tainty related to state variables.

209 The second main methodological step is data assimilation as such. At that point, the
 210 relationship between streamflow and state variables is assumed to be known. Then, one
 211 only has to provide *observed*, rather than simulated, streamflow as input to the neural net-
 212 works to obtain an estimate of the corresponding state variables. This process is illustrated
 213 on the right hand side of Figure 1. Those new, hopefully corrected, state variables, can
 214 then be passed back to the hydrological model, jointly with the appropriate inputs, to pro-
 215 duce a series of assimilated streamflows. Since, for GR4J, the true value of state variables
 216 cannot be known (see section 2.1), it is only possible to assess the performance and ef-
 217 fectiveness of the proposed data assimilation methodology through a comparison of the
 218 observed streamflow series with corresponding simulated streamflow after data assimila-
 219 tion.

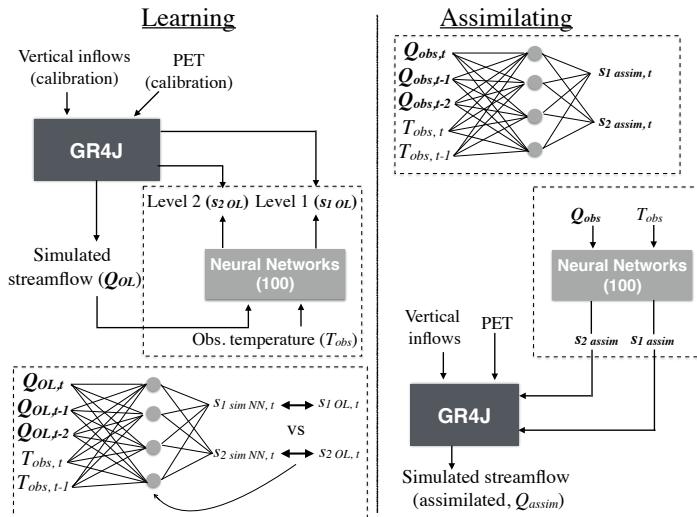


FIGURE 1. Scheme of the proposed methodology

220

221

2.1 The GR4J model

222 The hydrological model used in this study is the *modèle du génie rural à 4 para-*
 223 *mètres journalier* [GR4J, Perrin et al., 2003]. GR4J is possibly the (or at least one of the)
 224 simplest existing continuous hydrological models. It is precisely for this reason that it was
 225 chosen for this study. Since the objective of this paper is to test and propose a new data
 226 assimilation strategy based on neural networks, it was decided to keep the general mode-
 227 ling framework as simple as possible (i.e., with very few parameters to calibrate and few
 228 state variables to assimilate). This simplicity allows us to focus on the data assimilation
 229 technique and to test it on many catchments (see section 3). In addition, the GR4J model
 230 has been shown to perform very well in a wide variety of hydro-meteorological condi-
 231 tions. [e.g., Westra et al., 2014].

232 GR4J is a conceptual, lumped, bucket-type model with two reservoirs : a production
 233 reservoir which allows for the computation of runoff from precipitation and a routing re-
 234 servoir which aims at reproducing the response time of the basin by retaining a portion of
 235 the previously generated runoff. Routing also involves a unit hydrograph for the transfer of
 236 water from the production reservoir to the routing reservoir. The four parameters of GR4J
 237 are : the respective capacities of those two reservoirs (in mm), the base time of the unit
 238 hydrograph (in hours), and a "water exchange coefficient" (unit-less). The latter allows for
 239 water exchange between neighbouring catchments through groundwater. The calibration
 240 of the four parameters of GR4J was performed using the University of Arizona Shuffled
 241 Complex Algorithm [Duan et al., 1993] on a portion of the data set (see section 3). Since
 242 this paper focuses only on state variable uncertainty and data assimilation, only a single
 243 parameter set is used. However, since GR4J has only four parameters, it is expected that
 244 equifinality is not a major concern.

245 The model has two state variables, which are the level of water in the production
 246 reservoir and in the routing reservoir. Together they define the hydrological state of the
 247 catchment at each time step during a simulation. Contrarily to typical state variables found
 248 in physics-based models (e.g., snow water equivalent, soil moisture), the two state va-
 249 riiables of GR4J are quantitative but conceptual representations of a catchment's physical
 250 processes and cannot be directly measured. For this reason, the performance assessment of
 251 the proposed data assimilation method has to be performed on streamflow series (a mea-
 252 surable quantity) generated by the model from corrected state variables.

253 **2.2 Computation of vertical inflows using the degree-day snow module Cema-**
 254 **Neige**

255 As such, GR4J requires only two inputs, which are daily rainfall and potential evapo-
 256 transpiration. It was designed for snow-free catchments. However, it can be coupled
 257 with a snow module in order to (1) separate total precipitation as rain and snow, (2) store
 258 water as snow over the winter period and (3) gradually melt snow when temperature rises
 259 in the spring. In this study, the CemaNeige [Valery, 2010] snow module is used for catch-
 260 ments where a fraction of precipitation falls as snow. The output of CemaNeige are refer-
 261 red to as "vertical inflows", that is, the sum of rain and snowmelt. CemaNeige produces
 262 two additional state variables that could, in principle, also be considered in the data as-
 263 similation process. However, preliminary tests showed that there are specific challenges
 264 related to the correction of those two state variables (please see the Conclusion for further
 265 discussion on this topic). Considering this, it was decided that the data assimilation me-
 266 thod should be first developed without involving the state variables related to snow. Hence,
 267 the CemaNeige snow module was used to compute the vertical inflows for snow domi-
 268 nated catchments in a preliminary step, which were later used instead of precipitation as
 269 inputs to GR4J.

270 CemaNeige is a degree-day model that requires additional input data compared to
 271 GR4J. It is not distributed but it requires the catchment to be divided into five zones of
 272 equal area according to elevation. Snow melt rate is then computed as a function of al-
 273 titude according to those zones. The median altitude of each altitude zone is one of the
 274 additional inputs required by CemaNeige. CemaNeige also requires a correction factor
 275 for both precipitation and temperature with altitude, the mean annual snow accumulation
 276 on the basin (in mm), the snow melting point temperature in °C, and the minimal snow
 277 melt rate in mm/°C. It also has two parameters that need to be calibrated. The first one is
 278 a weighting coefficient used in the updating of the calorific deficit of snow at each time
 279 step. It weights the relative influence of the previous calorific deficit and air temperature.
 280 This coefficient is bounded by [0, 0.99]. If it is zero, then the calorific deficit of the snow
 281 pack is always equal to air temperature. The other parameter for CemaNeige is the degree-
 282 day melt rate, in mm/°C.

283 **2.3 Neural network methods for data assimilation**

284 **2.3.1 Multilayer Perceptrons**

285 The data assimilation methodology proposed in this paper relies on neural networks.
 286 There exist many different types of neural networks, but one of the most popular among
 287 hydrologists (and in general) remains the Multilayer Perceptron [MLP, Rosenblatt, 1958]
 288 and its variants. The MLP consists of three or more layers. In our case, there are three
 289 layers. The first one is constituted here of hydro-meteorological inputs. They are passed
 290 on to the second layer, which comprises a certain number of neurons (see section 2.3.4).
 291 Each input vector is connected to each neuron of the hidden layer. Those neurons are in
 292 fact linear or nonlinear functions, called "activation functions" that are applied to weighted
 293 and summed input vectors. The weighted sum in the hidden layer is given by equation (1)

$$\zeta_{j,t} = a_j + \sum_{i=1}^n (W_{j,i} \cdot X_{i,t}) \quad (1)$$

294 where $X_{i,t}$ is the i^{th} input at time t , n is the number of inputs, $W_{j,i}$ is the j^{th} input weight
 295 connected to the i^{th} input, and a_j is the j^{th} hidden neuron bias, added to the weighted
 296 sum. The weighted sum of inputs is passed on to the activation function of neuron j , to
 297 generate the hidden layer output vector at time t , $C(\zeta_{j,t})$, given by (in this case) a sigmoid
 298 tangent function as in equation (2)

$$C(\zeta_{j,t}) = \frac{2}{1 + e^{-2\zeta_{j,t}}} - 1 \quad (2)$$

Finally, the outputs of the neurons in the hidden layer are weighted and summed in a similar fashion as in equation (1) to generate the MLP outputs. However, the activation function for the neurons of the output layer is linear. There are as many neurons in the output layer as there are state variables in the hydrological model (hence two for GR4J). The MLP outputs can be determined as in equation (3)

$$Y_{k,t} = b_k + \sum_{j=1}^m (\beta_{k,j} \cdot C(\zeta_{j,t})) \quad (3)$$

where b_k is the k^{th} output layer bias, $\beta_{k,j}$ is the k^{th} output layer weight applied to the j^{th} hidden layer output, m is the total number of hidden neurons and $Y_{k,t}$ is the k^{th} output of the MLP at time t .

The weights and biases for MLP are initially unknown and have to be optimized iteratively to minimize an objective function. Here, the objective function is the root mean squared error (RMSE) of the state variables modeled by the MLP compared to the state variables simulated by GR4J. The optimization algorithm is Levenberg-Marquardt [Levenberg, 1944; Marquardt, 1963]. At each iteration of the training process, weights and biases are modified until state variables modelled by the MLP, represented by the vector Y , closely match those simulated by GR4J.

For each catchment, the available data base was separated in three portions (calibration, validation and test, see Table 2 in section 3). The training was performed on the calibration data set, and the validation data set was used to monitor over training. However, as demonstrated in *Boucher et al.* [2010] when the training is performed with the aim of obtaining a reliable ensemble, it is better to under-train each ensemble member (each MLP) to ensure that the resulting confidence intervals respect their nominal coverage. Therefore, this study varies the number of training epochs (iterations) used in MLP to explore its effect on data assimilation performance.

In this study, the MLP is compared to Extreme Learning Machines, a newer type of neural network that is rapidly gaining popularity for applications such as streamflow forecasting [*Yaseen et al.*, 2019].

2.3.2 Extreme Learning Machines

Extreme Learning Machines share the same architecture as the MLP. However, the main difference between ELM and MLP is that the ELM is does not require iterative training (i.e., training by backpropagation of network errors is not used to "tune" the network parameters) and therefore, greatly reduces the computational demand during network training. Instead, ELM uses randomly generated hidden layer parameters (i.e., input weights and biases), passes model inputs into the hidden layer to generate the hidden layer outputs (similar to MLP, see equation 2), and then (in contrast to MLP) the output layer weights are uniquely determined by taking the product of the Moore-Penrose inverse of the hidden layer output(s) and the target variable(s) [*Huang et al.*, 2012]. The determination of the output layer weights in ELM is given as [*Huang et al.*, 2012] (equation 4) :

$$\beta_{k,j} = T_{k,t} \cdot G(\zeta_{j,t})^\dagger \quad (4)$$

where \dagger represents the Moore-Penrose inverse, $T_{k,t}$ is the k^{th} target variable at time t and

$$G(\zeta_{j,t}) = \frac{1}{1 + e^{-\zeta_{j,t}}} \quad (5)$$

337 is the sigmoid function (the most commonly adopted activation function used within ELM).

338 Another small (but noteworthy) difference between the ELM and MLP is that the
 339 bias term is generally excluded from the output layer in ELM. Therefore, to determine the
 340 ELM output, equation 3 is used without including the bias term and swapping $C(\zeta_{j,t})$ for
 341 $G(\zeta_{j,t})$.

342 For both MLP and ELM and for each catchment (see section 3), an ensemble of 100
 343 neural networks was built by randomly generating the network parameters (weights and
 344 biases) before model training. This procedure for generating the ensemble follows *Boucher*
 345 *et al.* [2010], except that the inputs of the neural networks are not bootstrapped. Therefore,
 346 the ensembles account for parametric uncertainty in the estimation of state variables, since
 347 each member of the ensemble is a MLP or an ELM initialized with different weights and
 348 biases. Exploring other sources of uncertainty in MLP and ELM is left to future work.

349 For ELM, only the input weights and hidden layer biases were randomly generated
 350 (since the output layer weights are uniquely determined via least-squares). While there has
 351 been some ‘controversy’ over the original naming of the ELM and whether it originally
 352 existed first under the names functional link network (FLN) or random vector functional
 353 link networks [RVFL, *Wang and Wan*, 2008], our reason behind adopting ELM is related
 354 to the wide availability of its source code¹ under this name. The original ELM source
 355 code was modified for the purposes of this study. For the interested reader, *Rizk and Awad*
 356 [2019] highlight a number of similarities between ELM and other approaches proposed
 357 earlier in the literature.

358 There are many examples of where ELM has been used with success in hydrology
 359 [see for instance *Lima et al.*, 2017; *Atiquzzaman and Kandasamy*, 2018; *Amaranto et al.*,
 360 2018].

361 The choice of input variables used within MLP and ELM depended on the hydroclimatic
 362 regime of each catchment, as will be explained in the following sub-section.

363 2.3.3 Input variable selection

364 The standard approach to input variable selection (IVS) for neural network-based
 365 models in the hydrologic community follows what is termed sequential forward selection.
 366 In the sequential forward selection approach to IVS, one begins with a null input variable
 367 set and sequentially adds the “best” input, one step at a time, until some pre-defined stop-
 368 ping criterion is met, returning all inputs selected up until the termination point. Many
 369 approaches to sequential IVS exist. In general, there have been a wide number of IVS ap-
 370 proaches adopted in the hydrology literature. For instance, *Fleming et al.* [2015] developed
 371 ensembles of MLPs in a flood forecast model by using a different set of input variables
 372 for each MLP, corresponding to different physical conceptualizations of hydrologic mecha-
 373 nisms responsible for flood generation in the study basin. *Fleming and Goodbody* [2019]
 374 used a genetic algorithm for optimal feature creation and selection in a ML metasystem
 375 for seasonal water supply forecasting. *Amaranto et al.* [2018] used a constrained input va-
 376 riable selection approach coupled with several ML methods (MLP, ELM, support vector
 377 regression, etc.) for semi-seasonal groundwater forecasting. Their approach is rule-based,
 378 adopting process knowledge to constrain input variable selection to only consider phy-
 379 sically plausible model input combinations. *Beuzen and Simmons* [2019] developed an
 380 open-source Bayesian Networks approach for input variable selection and demonstrated

1. https://www.ntu.edu.sg/home/egbhuang/elm_codes.html

381 the effectiveness of their method for predicting beach erosion caused by coastal storms. To
 382 predict total dissolved solids in the Sufi Chai river in Iran, Yousefi *et al.* [2018] used MLP
 383 and Garrison's method to identify the most useful model inputs based on a large number
 384 of source water quality, climatic, and hydrometric variables.

385 Some of the most popular approaches for IVS are linear and partial correlation se-
 386 lection followed by information-theoretic approaches based on mutual information [May
 387 *et al.*, 2011], both of which are examples of sequential forward selection.

388 Mutual information can be used to estimate general (linear or nonlinear) multiva-
 389 riate dependencies between two or more variables and is therefore of great utility in the
 390 study of hydrological systems that are often characterized by nonlinear relationships bet-
 391 ween system processes. For this reason, mutual information has become a popular method
 392 for IVS. While mutual information-based IVS approaches can detect nonlinear dependen-
 393 cies between multiple variables, they are unable to detect redundancy or conditional rele-
 394 vancy amongst a set of variables [Hejazi and Cai, 2009] - another situation often met in
 395 the study of hydrologic systems. To overcome this deficiency of mutual information, IVS
 396 methods based on conditional mutual information (CMI) [Cover and Thomas, 2006] have
 397 grown in popularity [Sharma, 2000; Galelli *et al.*, 2014].

398 CMI allows for the estimation of mutual information between two or more variables,
 399 conditional on a set of side conditions (i.e., other variables that have been identified as
 400 potentially useful in explaining some properties of a target process) and can detect redun-
 401 dancy and conditional relevancy amongst a set of variables.

402 Arguably, one of the most popular CMI-based approaches to IVS is the Partial Mu-
 403 tual Information Selection (PMIS) algorithm [Galelli *et al.*, 2014]. Partial mutual infor-
 404 mation is simply another term used for CMI. However, as noted in an earlier study, PMIS
 405 can be very computationally demanding as it requires the calibration of two nonlinear re-
 406 gression models whose performance is highly dependent on parametric settings [Li *et al.*,
 407 2015]. Instead of using PMIS, this study adopts a more straightforward approach to CMI-
 408 based IVS that uses K nearest-neighbours-based estimates of CMI instead of nonlinear
 409 regression methods (as in PMIS). The K nearest-neighbours CMI-based approach to IVS
 410 requires specification of only a single parameter, K (the number of nearest neighbours).
 411 This method was adopted as it was shown in an earlier study to be more computationally
 412 efficient than PMIS and provide nearly the same level of performance on several bench-
 413 mark IVS problems [Quilty *et al.*, 2016].

414 For sake of brevity, the theoretical underpinnings of the K nearest-neighbour CMI-
 415 based IVS approach can be found in earlier studies [Frenzel and Pompe, 2007; Kugiumtzis,
 416 2013].

417 In the context of this experiment, the calculation of information-theoretic measures
 418 was performed using TIM, an open-source C++ library². The CMI was calculated from a
 419 pool of potential input variables that include daily streamflow, daily average air tempera-
 420 ture and daily vertical inflow, all for 1 to 15 days prior to the assimilation, using $K = 5$.

421 2.3.4 Neural network architecture

422 Although MLP and ELM share many similarities, MLP need to be trained for a cer-
 423 tain number of epochs in order to reach good performances, while this is not the case for
 424 ELM. Hence, one of the challenges in this comparison is to ensure that the final architec-
 425 tures for both types of neural networks is carefully selected so as to not favor one type
 426 of neural networks over the other. Therefore, both ELM and MLP were trained using the
 427 same input data for each catchment. For the MLP, the number of training epochs was also

2. <http://www.cs.tut.fi/~7Etimhome/tim/tim.htm>

428 varied between 1 and 200 as this was found to provide suitable performance. Increasing
 429 the number of training epochs would further increase computation time without a justifi-
 430 able 'payback' in model performance.

431 Note that this experiment was carried out in Matlab, using toolboxes developed du-
 432 ring a previous study by Dr. Kris Villez (MLP) and from the second author (ELM), based
 433 on the original ELM source code³.

434 2.4 Performance assessment

435 2.4.1 Benchmarks

436 The benchmarks for assessing the performance of the proposed data assimilation
 437 method are the Open Loop simulation and the Ensemble Kalman Filter [EnKF, *Evensen*,
 438 2003]. The former is the series of simulated streamflow values obtained by running GR4J
 439 with observed vertical inflows and potential evapotranspiration, without any data assimi-
 440 lation. The latter is a now well-known data assimilation technique in hydrology. To avoid
 441 diverting the paper into a lengthy explanation about the technical details of the implemen-
 442 tation of an EnKF, we instead refer the interested reader to some of the numerous papers
 443 that have been written on the topic, such as, for instance, the comprehensive explanation
 444 from *Mandel* [2006], *Evensen* [2003] or *Thiboult and Anctil* [2015]. The EnKF requires
 445 the computation of a Kalman gain, which is then used as a weight to perform a compro-
 446 mize between a model simulation originating from a particular state vector (a member of
 447 the ensemble) and the observed streamflow value.

448 Also, it is important to mention that we used the HOOPLA (HydrOlOgical Pre-
 449 diction LABoratory) framework from *Thiboult et al.* [2019] to perform data assimilation
 450 with the EnKF. HOOPLA⁴ is a modular framework that allows for different designs of
 451 ensemble forecasting systems in order to account for different types of uncertainties. In
 452 HOOPLA, there is the option to apply the EnKF to a given hydrological model, previously
 453 calibrated, for any catchment. Additionally, HOOPLA suggests default parameters for the
 454 EnKF based on previous studies [*Thiboult and Anctil*, 2015; *Liu et al.*, 2012; *Reichle et al.*,
 455 2002]. Those parameters define the standard deviation of the normal distribution that re-
 456 presents the uncertainty around model inputs and outputs. Although they can be fine-tuned
 457 to each catchment as in *Thiboult and Anctil* [2015], here we chose to use the default pa-
 458 rameters in order for the EnKF to be as "standard" as possible for the purpose of compa-
 459 rison. Those default parameters are standard deviations that correspond to 10% of the ob-
 460 served value for the observed streamflow and potential evapotranspiration, 50% for rainfall
 461 and 2 degrees Celcius for temperature [after *Liu et al.*, 2012]. The size of the ensemble is
 462 also 100, to be comparable with that of MLP and ELM.

463 2.4.2 Scores and performance assessment tools

464 Since GR4J is a deterministic model and we chose to use only one parameter set,
 465 the Open Loop simulation is deterministic (not an ensemble). The simulation after data
 466 assimilation is, however, an ensemble comprising of 100 members. Fortunately, the Conti-
 467 nuous Ranked Probability Score [CRPS, *Matheson and Winkler*, 1976] is the probabilis-
 468 tic equivalent of the mean absolute error (MAE) [*Gneiting and Raftery*, 2007]. This allows
 469 for a direct and fair comparison of hydrologic simulations before and after data assimi-
 470 lation. However, since both the MAE and the CRPS values are proportional to the mean
 471 streamflow value, they cannot be used directly to compare performances between catch-
 472 ments. The CRPS can be computed from equation (6)

3. https://www.ntu.edu.sg/home/egbhuang/elm_codes.html

4. <https://github.com/AntoineThiboult/HOOPLA>

$$CRPS(F, Q_{obs}) = \frac{1}{M} \sum_{t=1}^M \int_{-\infty}^{+\infty} (F_t(Q) - H(Q \geq Q_{obs,t}))^2 dQ \quad (6)$$

473 where $F_t(Q)$ is the forecast for streamflow Q at time t , expressed as a cumulative
 474 distribution function, $Q_{obs,t}$ is the corresponding observation and H is the Heaviside func-
 475 tion. The latter equals 0 for predicted values smaller than the observed streamflow and 1
 476 otherwise. M is the total number of forecast-observation pairs.

477 The logarithmic score [Good, 1952] also allows for the evaluation of forecasts' qua-
 478 lity, but penalizes a lack of reliability more heavily than the CRPS, as the logarithmic
 479 score is a local score (also see section 4.3 for further discussion). The logarithmic score
 480 is given by equation (7)

$$LogScore(f, Q_{obs}) = \frac{1}{M} \sum_{t=1}^M \log(f_t(Q_{obs,t})) \quad (7)$$

481 where f_t is the forecast for streamflow Q at time t , expressed as a probability den-
 482 sity function. Therefore, $f_t(Q_{obs,t})$ is the probability density corresponding to the obser-
 483 ved streamflow value at time t .

484 Since, according to Gneiting and Raftery [2007], reliability is the most important
 485 quality to be achieved in ensemble forecasting, particular attention is paid to that attri-
 486 butive. The reliability diagram [Wilks, 1995] allows for its assessment in a visual way. It
 487 consists in plotting the effective probability for different confidence intervals computed
 488 from the forecasts (or simulations) against their nominal probability. For instance, if the
 489 95% confidence interval computed using the ensemble simulations only includes 90% of
 490 the observations, it means that simulations are unreliable (under-dispersed, in this case).
 491 For perfectly reliable ensemble simulations, the nominal and effective confidence levels
 492 should correspond exactly and the reliability diagram should lie on the diagonal. This is
 493 rarely the case in practice, because most forecasting (or simulation) systems account for a
 494 single source of uncertainty or only a few of them.

495 Since this study focuses on the development of a new data assimilation method and
 496 on the assessment of its efficiency, there are no real forecasts involved. Forecasting stream-
 497 flow would be the next logical step, but here, the goal is strictly to assess the performance
 498 of data assimilation. However, in order to quantify the persistence of the data assimilation,
 499 a "perfect forecasts" experiment was also performed.

500 "Perfect forecasts" based on meteorological observations for the next D days, that
 501 are provided to GR4J as inputs. What differentiates perfect forecasts from simulations is
 502 that GR4J is initialized with assimilated states at the beginning of each "forecasting" per-
 503 iod. Then, for the next D days, the model runs without data assimilation. This allows one
 504 to see how the model's performance evolves as a function of lead-time, as the effect of
 505 data assimilation "fades away".

506 To summarize, there are three types of model runs that are explored in our case
 507 study : 1) the Open Loop simulation, in which no data assimilation is involved; 2) the si-
 508 mulation with continuous data assimilation, in which state variables are corrected at every
 509 time step (day); and 3) the perfect forecasts, which are initialized using assimilated states,
 510 but then the state variables are left free to "drift" (i.e., are not updated) for the next D
 511 days.

512 3 Catchments database

513 In order to validate the robustness of the proposed data assimilation methodology,
 514 a database of catchments under contrasting hydro-climatic conditions was built. Table 1
 515 indicates the geographical locations of the catchments. It also provides basic descriptive
 516 statistics regarding meteorological inputs and streamflow. Mistassibi is a Nordic catchment
 517 with a strong annual cycle marked by an annual spring freshet. The Main River catch-
 518 ment at Schwuerbitz and the Ourthe catchment also receive some precipitation in the form
 519 of snow, but their climates are much more temperate than Mistassibi. Schwuerbitz is the
 520 name of the gauging station for the upper part of the Main River catchment. Hereafter, it
 521 will only be referred to as "Schwuerbitz". The Los Idolos catchment is located in a much
 522 warmer climate than the other three catchments and does not receive any snow.

523 All four catchments are uninfluenced (i.e., no dams, natural streamflow regime).
 524 Precipitation and temperature observations are available from ground stations, but unfor-
 525 tunately, the spatial density as well as the data quality of those stations is variable. For
 526 instance, for the Mistassibi catchment, kriged precipitation and temperature data (10km
 527 spatial resolution) was available. For the Los Idolos catchment, measurements from seven
 528 ground stations within the basin were available, but only one among the seven had data
 529 throughout the entire time period. This time period was chosen according to the availabi-
 530 lity of streamflow data. In all cases, precipitation and temperature data were averaged over
 531 the catchment.

532 Potential evapotranspiration (PET) was estimated by the empirical method of *Oudin*
 533 [2004], based on mean daily air temperature, the Julian day and the latitude of the
 534 centroid of the catchment. *Oudin* [2004] found that this simple model provided the best
 535 results among 27 empirical PET formulas in a large scale study involving 308 catchments.

536 **TABLE 1.** Hydro-climatic information regarding the catchments used in this study

Name	Area (km ²)	Location	Mean an. rainfall (mm)	Mean an. snowfall (mm)	Avg. min. annual temperature (°C)	Avg. max. annual temperature (°C)	Avg. min. annual streamflow (mm/day)	Avg. max. annual streamflow (mm/day)
Mistassibi	8684	Canada	717	259	-42	30	0.36	10.03
Schwuerbitz	2419	Germany	853	22	-16	31	0.22	10.01
Ourthe	1597	Belgium	1004	18	-10	19	0.17	8.82
Los Idolos	455	Mexico	1036	NA	8	37	0.09	28.07

537 Table 2 details how the database for each catchment was separated into calibration
 538 (training), validation, and test partitions, which varied according to the availability of data
 539 within each catchment.

540 **TABLE 2.** Separation of the database in three portions for calibration, validation and test.

Name	Calibration	Validation	Test
Mistassibi	1970/01/01 - 2005/12/31	2006/01/01 - 2011/12/31	2012/01/01 - 2015/12/31
Schwuerbitz	1970/01/01 - 2005/12/31	2006/01/01 - 2011/12/31	2012/01/01 - 2015/12/31
Ourthe	2000/01/01 - 2005/12/31	2006/01/01 - 2008/12/31	2009/01/01 - 2010/12/31
Los Idolos	1964/01/01 - 1999/12/31	2000/01/01 - 2006/12/31	2007/01/01 - 2011/12/31

541 **4 Results**

542 **4.1 Input variable selection**

543 Table 3 shows the input variables that were selected according to the procedure de-
 544 tailed in section 2.3.3. They represent the input variable combinations that were found to
 545 provide the largest CMI for each catchment. In this table, Q identifies streamflow, T iden-
 546 tifies air temperature and P identifies vertical inflow. The subscript refers to time. For ins-
 547 tance, Q_{t_0-2} represents the observed streamflow two days before the moment data assimila-
 548 tion is performed.

549 **TABLE 3.** Input variables selected and maximum number of hidden neurons for each catchments

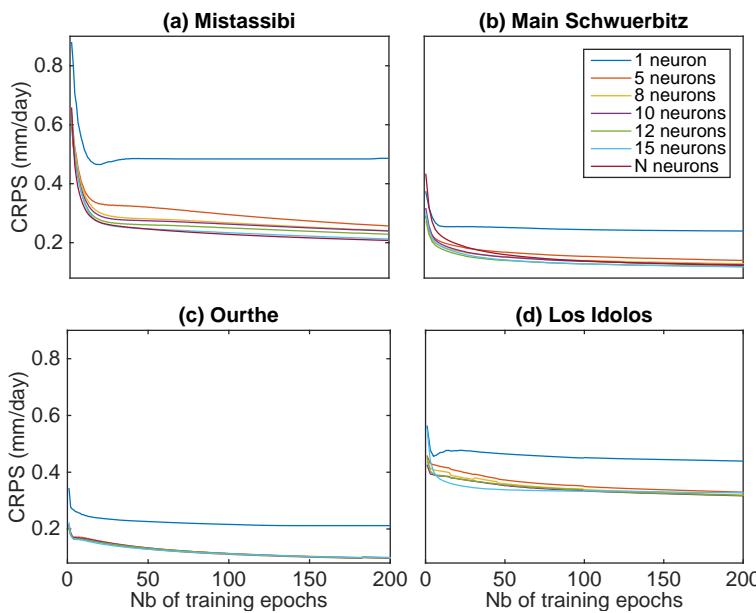
Catchment	Inputs (same for ELM and MLP)	Max. Nb. of hidden neurons (N)
Mistassibi	Q_{t_0-2} to Q_{t_0} , T_{t_0-5} to T_{t_0}	28
Main - Schwuerbitz	Q_{t_0-10} to Q_{t_0} , T_{t_0-1} and T_{t_0}	40
Ourthe	Q_{t_0-3} to Q_{t_0} , T_{t_0-1} to T_{t_0}	19
Los Idolos	Q_{t_0-5} to Q_{t_0} , T_{t_0-2} to T_{t_0} and P_{t_0-5} to P_{t_0}	46

550 **4.2 Determining network architecture and training epochs**

551 An important initial step in building MLP and ELM models is to determine the best
 552 architecture (i.e., number of hidden neurons) as well as the number of epochs to use for
 553 training the MLP. The number of hidden neurons for both MLP and ELM was made to
 554 vary between one and a maximum number defined as "one plus three times the number of
 555 input variables". This is a variant of the formula proposed by *Hecht-Nielsen* [1990], allo-
 556 wing us to test more possibilities, as our earlier experiments indicated that ELM typically
 557 require more hidden neurons than MLP. We wanted to test the same possibilities for both
 558 ELM and MLP to ensure a fair comparison between both methods.

559 Figure 2 shows the CRPS as a function of the training epochs for MLP, by varying
 560 the number of neurons in the hidden layer. These results were obtained in simulation mode,
 561 meaning that data assimilation is performed at every time step (here every day) and the
 562 model also simulates streamflow daily. This represents the best case scenario, where the
 563 model is always initiated from the best estimate of the initial state. This is very similar to
 564 a "nowcasting" situation. For Figures 2 to 8 in this section, results are shown only for the
 565 validation portion of the database.

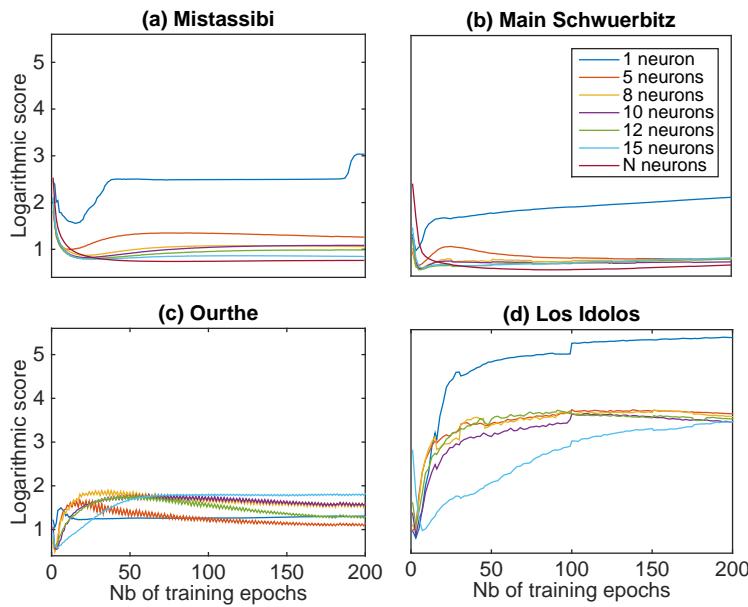
566 In the legend, N refers to the maximal number of hidden neurons (see Table 3). For
 567 the Mistassibi catchment, the lowest (best) CRPS was obtained for the maximal number of
 568 hidden neurons, and the maximum number of training epochs. This is, however, compu-
 569 tationally demanding. In addition, the CRPS provides only a partial portrait of the overall
 570 quality of the ensemble simulation. For the Schwuerbitz catchment, between 10 and 15
 571 hidden neurons provide the best results. For Ourthe, the CRPS appears somewhat insensi-
 572 tive to the number of hidden neurons, whereas for the Los Idolos catchment, 15 neurons
 573 leads to the best CRPS, even with very few training epochs (less than 20).



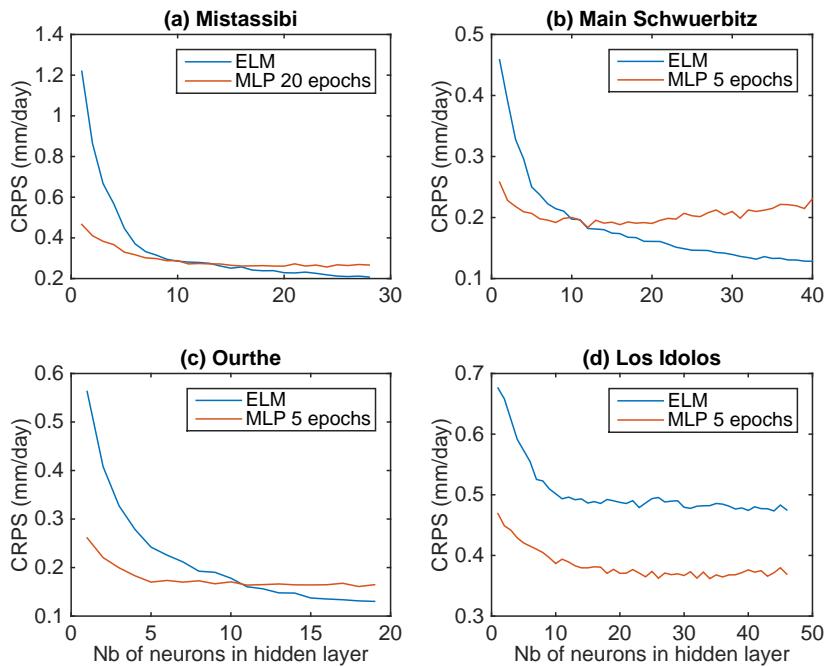
574 **FIGURE 2.** CRPS of simulations after data assimilation using MLP for different number of hidden neurons
575 and training epochs.

576 Figure 3 provides a different perspective of the situation. This figure shows the lo-
577 garithmic score as a function of training epochs and number of hidden neurons. The lo-
578 garithmic score and the CRPS each focus on different attributes of the forecast. The former
579 is highly sensitive to observations that fall outside the predictive distribution, as it pena-
580 lizes low probability events heavily [Gneiting and Raftery, 2007]. Reliability is therefore
581 more important than resolution (also termed sharpness) for the logarithmic score, whereas
582 resolution and reliability are both equally important for the CRPS. The logarithmic score
583 generally starts to increase past 20 epochs of training for the Mistassibi catchment and five
584 for the other three catchments.

587 Figure 4 compares MLP with ELM, using the number of training epochs for MLP
588 suggested by Figures 2 and 3. In general, this figure shows that, except for the Los Idolo-
589 los catchment, ELM can outperform MLP provided they have sufficient number of hid-
590 den neurons. However, for the Los Idolos catchment, MLP perform noticeably better than
591 ELM. The performance of MLP either reaches a plateau or begins to degrade (CRPS in-
592 creases) with the number of hidden neurons. This is not surprising, as an optimal MLP
593 design is the combination of a specific number of hidden neurons and training epochs.
594 The number of required training epochs is proportional to the number of hidden neurons.
595 So increasing the number of neurons without increasing the number of training epochs
596 unsurprisingly decreases performances.



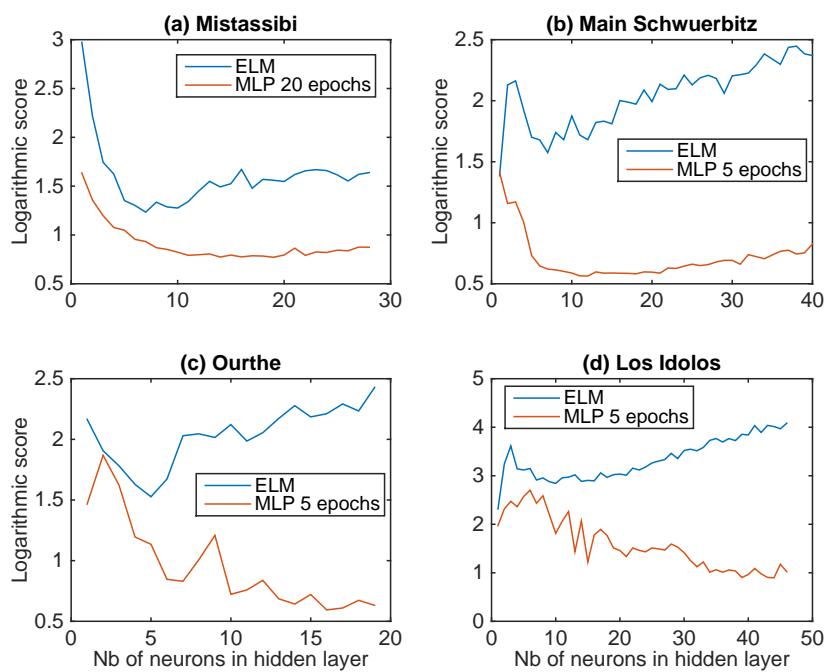
585 **FIGURE 3.** Logarithmic scores of simulations after data assimilation using MLP for different number of
586 hidden neurons and training epochs.



597 **FIGURE 4.** Comparison of the CRPS of MLP and ELM as a function of the number of hidden neurons

598 Figure 5 shows the results from a similar experiment using the logarithmic score
599 instead. According to this score, MLP always outperform ELM, and the difference bet-
600 between the two increases with the number of hidden neurons. This is a possible indication

601 that MLP forecasts are more reliable than ELM since the logarithmic score emphasizes
 602 this attribute (more so than the CRPS).



603 **FIGURE 5.** Comparison of the logarithmic score of MLP and ELM as a function of the number of hidden
 604 neurons

605 To further examine this issue of reliability, Figures 6 and 7 show the reliability dia-
606 grams for the Mistassibi and Schwuerbitz catchments, for a varying number of training
607 epochs (for MLP) and hidden neurons. The same figures were plotted for the two other
608 catchments and are included in the Supporting Information. The thick diagonal black line
609 represents perfectly reliable forecasts. Indeed, one can see that ELM (black dashed line)
610 are consistently under-dispersed, which supports the conclusions from Figure 5 that MLP
611 is more reliable than ELM. Adding more hidden neurons does not significantly change the
612 result. For the Mistassibi catchment, Figure 6 shows that 10 hidden neurons and five to
613 12 training epochs provide reliable simulations (panel e, very close to the diagonal). Se-
614 venteen (17, panel g) hidden neurons with 5 training epochs also provide equally reliable
615 simulations. However, the more parsimonious option should be preferred to speed up mo-
616 del output calculations. According to the same guiding principles, the best option for the
617 Schwuerbitz catchment is 15 hidden neurons and 12 training epochs.

618 Reliability for the Ourthe and Los Idolos catchments is generally lower. For instance,
619 for the Ourthe catchment, with less than 10 hidden neurons, both MLP and ELM provide
620 very similar results in terms of reliability (not shown here, see Supporting Information),
621 and all simulations are under-dispersed. For MLP, between 15 and 19 hidden neurons and
622 five training epochs seem to work best. Conclusions for the Los Idolos catchment are si-
623 milar and the best scenario for MLP is 20 hidden neurons and five epochs.

Accepted Article

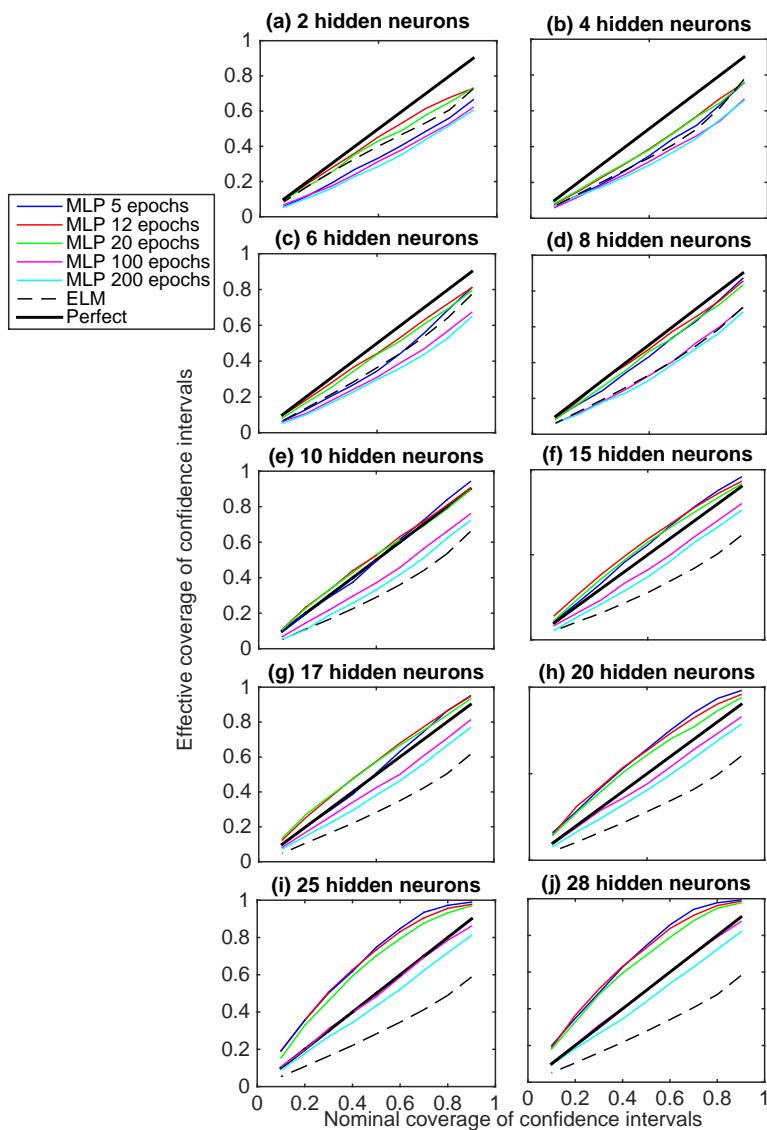


FIGURE 6. Reliability diagrams for the Mistassibi catchment, as a function of the number of hidden neurons and training epochs

To further investigate the impact of modifying the number of hidden neurons in both MLP and ELM, Figure 8 shows the ratio of missed events for MLP trained for the optimal number of epochs as determined from the above, compared with MLP trained for 100 epochs and with ELM, by considering a varying number of neurons in the hidden layer. The red curve confirms that MLP with 10 or more hidden neurons, trained for five or 20 epochs, offer good performance (few missed events) for the catchments under study. Another conclusion from this figure is that ELM with fewer hidden neurons will have a lower rate of missed events compared to ELM with 20 or more hidden neurons. This is linked to the lack of reliability (under-dispersion) of ELM-based simulations as observed in Figures 6 and 7. Figure 4 also indicates that a small number of hidden neurons for ELM is sufficient to obtain a low CRPS.

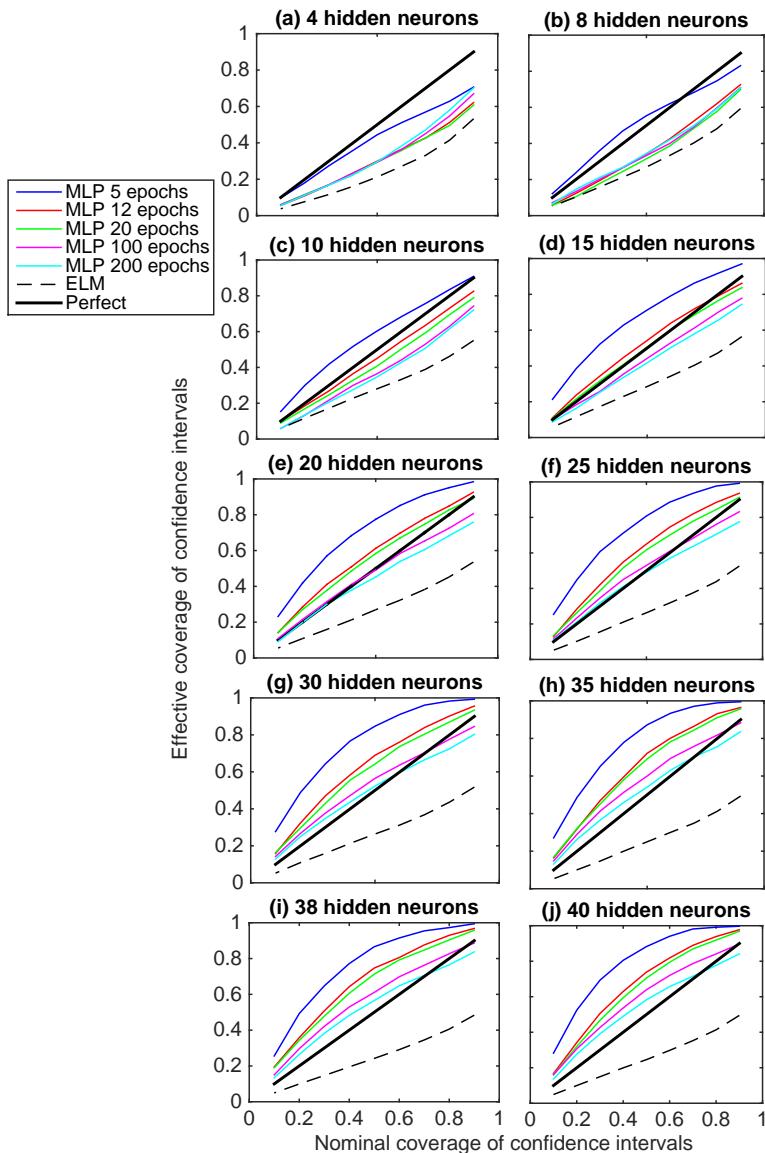
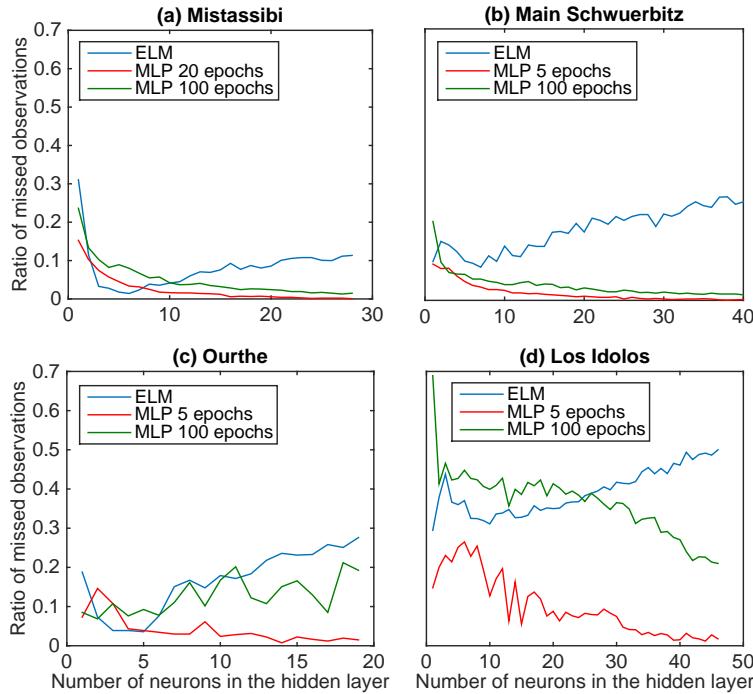


FIGURE 7. Reliability diagrams for the Schwuerbitz catchment, as a function of the number of hidden neurons and training epochs

The final network architectures and number of training epochs was set according to the CRPS and the logarithmic score (Figures 2 to 5), keeping in mind that the ensembles must be reliable (i.e., by considering the respective reliability diagrams for MLP and ELM). This was done by comparing the different metrics, which sometimes did not agree (for Schwuerbitz, in particular).

For the ELM, according to the CRPS (Figure 4), one should always use the maximum number of hidden neurons. However, according to the logarithmic score, as well as to maximize the reliability and minimize the number of missed events, one should use a much smaller number of hidden neurons.



639 **FIGURE 8.** Ratio of missed events as a function of the number of neurons in the hidden layer for (a) Mistassibi
 640 sibi, (b) Schwuerbitz, (c) Ourthe and (d) Los Idolos

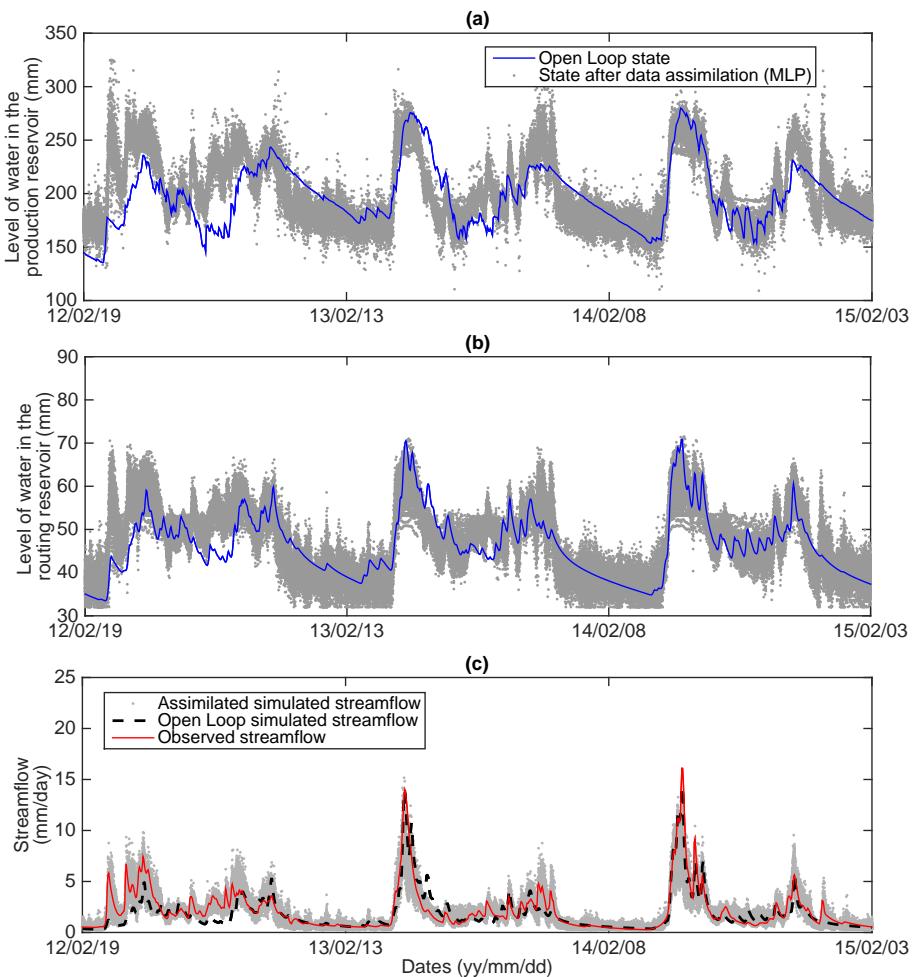
650 The following architectures presented in Table 4 were used in a pseudo-forecasting
 651 framework, described in the next sub-section, as well as in the time series plots provided
 652 in Figures 9 and 10. Once the optimal architecture has been identified (and training per-
 653 formed in case of MLP), applying the method for data assimilation for one time step, as one
 654 would do in an operational setting, is very fast. It varies between 0.02 and 0.04 seconds,
 655 depending on the number of inputs, type of neural network, and architecture.

656 **TABLE 4.** Final architectures for the neural networks

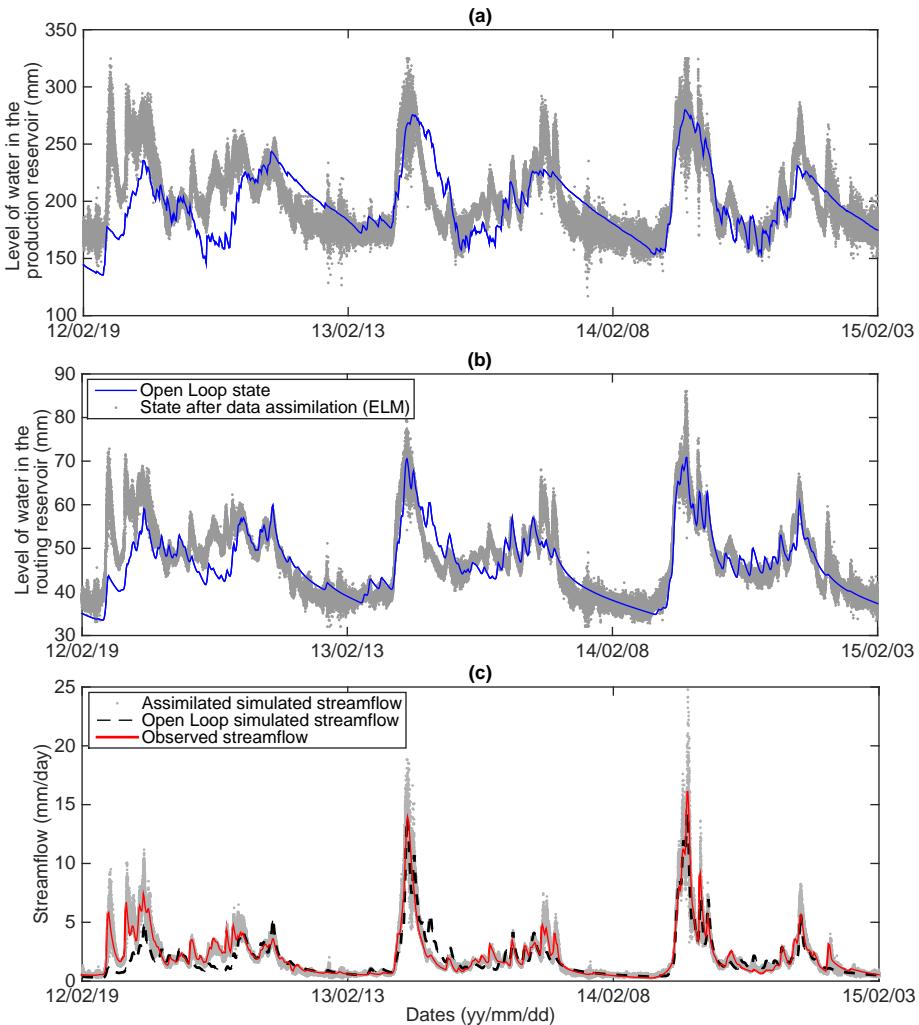
Catchment	MLP	ELM
Mistassibi	20 epochs and 10 neurons (Fig. 4 and 6)	6 neurons (Fig. 6(e) and 8)
Main - Schwuerbitz	5 epochs and 8 neurons (Fig. 7)	8 neurons (Fig. 5 and 8)
Ourthe	5 epochs and 19 neurons (Fig. 8)	5 neurons (Fig. 5 and 8)
Los Idolos	5 epochs and 20 neurons (Fig. 4)	10 neurons (Fig. 5 and 8)

657 Figures 9 and 10 show the state variables before and after data assimilation using
 658 MLP and ELM, respectively. Since GR4J's state variables do not correspond to observable
 659 quantities, they cannot be measured and therefore cannot be directly compared to observa-
 660 tions. However, it is possible to assess the performance of the data assimilation indirectly
 661 by comparing the streamflow simulations before and after data assimilation with the cor-
 662 responding observations. Here, the Open Loop simulation is the dashed line in panel (c)
 663 of both figures, while the observed streamflow is shown in red. Since the proposed data
 664 assimilation method relies on an ensemble of 100 neural networks, streamflow simulations
 665 include 100 members (grey dots).

666 From Figures 9 and 10, it is clear that the proposed data assimilation method effec-
 667 tively brings the assimilated streamflow simulations closer to the corresponding observa-
 668 tions, while at the same time providing an estimate of the uncertainty related to the state
 669 variables. From Figure 6, it can be seen that this uncertainty estimation is reliable when
 670 an appropriate MLP architecture is chosen. It can also be seen that, despite the care taken
 671 to select appropriate network architectures and training parameters according to different
 672 measures of quality, ELM provide narrower uncertainty distributions than MLP, which
 673 could lead to them being less reliable in practice.



674 **FIGURE 9.** Times series of the open loop (blue) and assimilated state variables (grey) for the test portion
 675 of the database using MLP, for the Mistassibi catchment : (a) the level of water in the production reservoir
 676 of GR4J, (b) the level of water in the routing reservoir of GR4J. Panel (c) shows the open loop simulated
 677 streamflow (dashed black line) compared to the observations (red) and the assimilated streamflow ensembles
 678 (grey).



679 **FIGURE 10.** Times series of the Open Loop (blue) and assimilated state variables (grey) for the test portion
 680 of the database using ELM, for the Mistassibi catchment : (a) the level of water in the production reservoir
 681 of GR4J, (b) the level of water in the routing reservoir of GR4J. Panel (c) shows the Open Loop simulated
 682 streamflow (dashed black line) compared to the observations (red) and the assimilated streamflow ensembles
 683 (grey).

684 4.3 Perfect forecasts

685 In order to assess the persistence of the proposed data assimilation method in time,
 686 we apply it to a "perfect forecasts" framework. This does not involve using precipita-
 687 tion and temperature forecasts as inputs to GR4J as one would do in a real forecasting
 688 context. Instead, it is considered that future precipitation and temperature inputs are per-
 689 fectly known, and thus correspond to the observed values without any uncertainty (the
 690 uncertainty on state variables remains, though). This amounts to performing data assimi-
 691 lation once every D days, and then continuing to simulate streamflow with the model to
 692 see how long the effect of data assimilation will persist. Here we chose a horizon of ten
 693 days ($D=10$), as it is typical of many short-term operational forecasting systems. Of note,
 694 we also include the performance of the EnKF to demonstrate the utility of our proposed
 695 neural network-based data assimilation methodology against a widely used approach.

Figure 11 shows the logarithmic score as a function of time, for ELM (blue), MLP (red), and also for the generic EnKF (green). The logarithmic score at time t_0 , immediately after data assimilation, is shown for comparison. This value of the score for t_0 corresponds to the best estimate of the state variables according to the proposed method. Therefore, the score is expected to deteriorate (increase) with the forecasting horizon. This is exactly what this figure shows, but we can also observe differences between the performance of the ELM and MLP-based data assimilation. For all four catchments, MLP reaches a lower (better) logarithmic score than ELM. There are also differences in the behaviour of individual catchments. For Mistassibi, the effect of data assimilation has very little persistence : the logarithmic score increases (deteriorates) rapidly after t_0 . However, for the Los Idolos catchment, the persistence is much higher and the effect of data assimilation lasts for up to three days with the ELM-based data assimilation method (blue line). A similar effect is observed for the Schwuerbitz and Ourthe catchment with the MLP-based method. This suggests that data assimilation could be performed less often than daily for those catchments; for example, once every two or three days. However, the detailed verification of the optimal time frequency for data assimilation is left to future studies. Regarding the comparison with EnKF, this figure shows that, according to the logarithmic score, the proposed method is superior to the generic (i.e., not fine-tuned) EnKF.

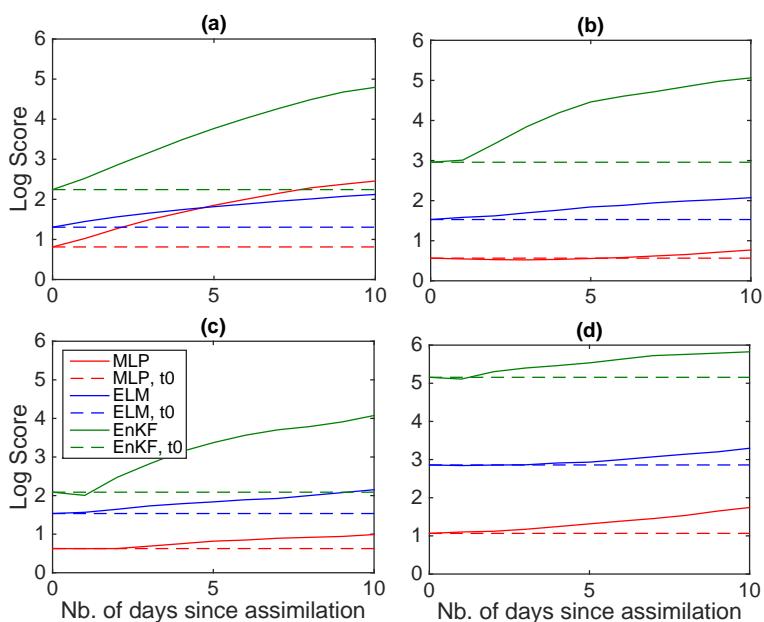


FIGURE 11. Evolution of the logarithmic score in a "perfect forecasts" situation for (a) Mistassibi, (b) Schwuerbitz, (c) Ourthe and (d) Los Idolos. Results from the ELM-based data assimilation are shown in blue, results from the MLP-based data assimilation are in red and results for the EnKF are in green. The dashed line represents the base line, which is the logarithmic for time t_0 .

The CRPS was also computed in this "perfect forecasts" framework and the results are displayed on Figure 12. Interestingly, according to the CRPS, EnKF-based forecasts are almost always initially better (lower score) than MLP or ELM (except for Los Idolos, for which MLP perform better). However, the effect of data assimilation fades away more quickly. It is important to stress that, as mentioned in section 2.4.1, this is a generic version of the EnKF, which could be improved. Here, the spread of the EnKF-based ensemble for both state variables and for streamflow (not shown, an example is provided in the Supporting Information) is much less than illustrated in Figures 9 and 10. It is the main reason why data assimilation with the EnKF fades away more quickly : ensemble members rapidly become very similar, which lowers the reliability.

It may seem surprising at first to notice that for the Schwuerbitz and Los Idolos catchments, the CRPS improves (decreases) with the forecasting horizon. This behaviour is different from what is observed with the logarithmic score in Figure 11. However, one must remember that while the logarithmic score is local, the CRPS is a global score that is affected both by the reliability and sharpness of the forecasts. The CRPS can be decomposed into its main constituents to better understand this behaviour.

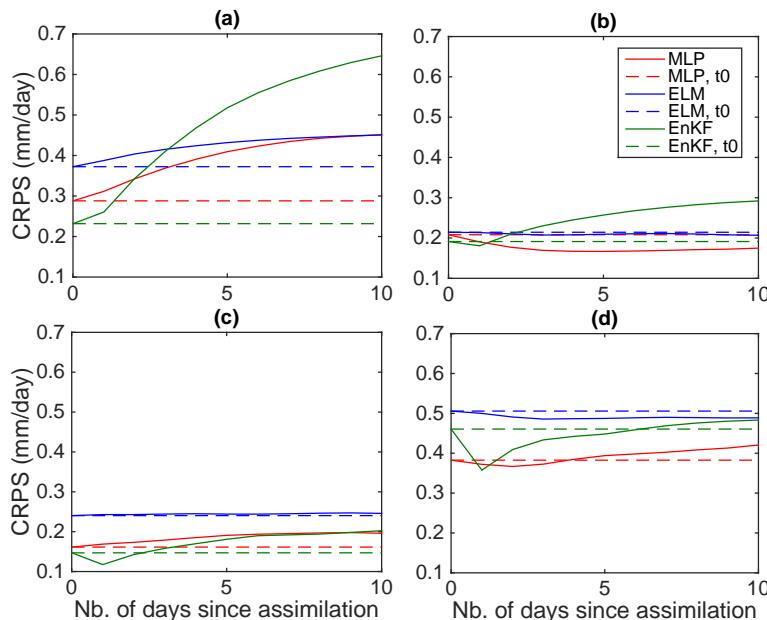
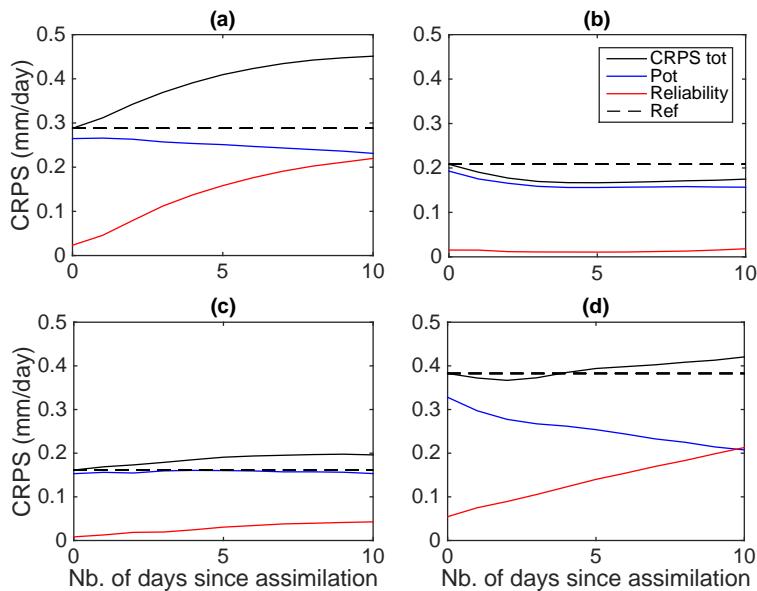


FIGURE 12. Evolution of the CRPS in a "perfect forecasts" situation for (a) Mistassibi, (b) Schwuerbitz, (c) Ourthe and (d) Los Idolos. Results from the ELM-based data assimilation are shown in blue, results from the MLP-based data assimilation are in red and results for the EnKF are in green. The dashed line represents the base line, which is the logarithmic for time t_0 .

Figure 13 shows the evolution of the CRPS and its decomposition into reliability and potential components, only for MLP-based perfect forecasts (corresponding to red lines in Figures 11 and 12). The potential CRPS is the value that could be achieved if the forecasts were perfectly reliable, for the same level of sharpness. This potential CRPS decreases (improves) with lead time for all catchments. This means that, as the lead time evolves, the forecasts becomes sharper. Indeed, as the forecasting horizon increases, the state variables are recomputed by the model (GR4J) and become increasingly similar to one another. This reduces the spread of the ensemble (improves sharpness/potential CRPS) with the lead time. However, this also makes the forecasts less reliable, as the ensembles cover less possibilities. This can be seen in Figure 13, by the increasing trend for all red

748 lines. This general behaviour is observed for all four catchments, except Schwuerbitz.
 749 For this catchment, the improvement of the potential component of the CRPS dominates
 750 over the reliability component, which remains approximately unchanged across horizons.
 751 Hence, the total CRPS improves. Also, note that the behaviour of the reliability com-
 752 ponent of the CRPS is very similar to the behaviour of the logarithmic score (Figure 11).
 753 This stresses the fact that the logarithmic score places more emphasis on forecast reliabi-
 754 lity than sharpness. Example hydrographs for lead times 1-day, 3-day, 6-day, and 10-day
 755 are provided for Schwuerbitz and Mistassibi in the Supporting Information, to help the
 756 reader observe this decrease in ensemble spread (increased sharpness) with lead time, in-
 757 cluding its adverse effects on reliability.



758 **FIGURE 13.** Evolution of the CRPS (black) and its decomposition into reliability (red) and potential (blue)
 759 components in a "perfect forecasts" situation, for MLP-based data assimilation only, for (a) Mistassibi, (b)
 760 Schwuerbitz, (c) Ourthe and (d) Los Idolos. The dashed line represents the base line, which is the CRPS for
 761 time t_0 .

762

5 Conclusion

763 To the best of our knowledge, this study is the first to propose neural networks (and
 764 machine learning in general) as a new method for data assimilation in hydrological fore-
 765 casting. We believe that this is a very promising avenue for machine learning in operatio-
 766 nal hydrology. Indeed, in this paper, machine learning techniques do not replace concep-
 767 tual or physics-based models. Such models are of great importance for process unders-
 768 tanding and to justify decisions based on forecasts in operational hydrology. Rather, we
 769 propose that machine learning techniques could support conceptual and/or physics-based
 770 models through data assimilation.

771 In our framework, the machine learning model plays exactly the same role as any
 772 other data assimilation method would (e.g., ensemble Kalman filter, particle filter or other).
 773 The goals of ensemble-based data assimilation are to (1) update the state variables to im-
 774 prove upon the first estimate obtained by running the hydrological model in Open Loop
 775 (which can be considered a first guess as in *Aires et al. [2001]*) and (2) to estimate state
 776 variable uncertainty.

777 With this study, we wanted to verify two hypotheses stated in the Introduction. The
 778 first one was verified by showing that neural networks can indeed be used to accurately
 779 perform data assimilation in hydrological models. This was demonstrated using a simple
 780 rainfall-runoff model (here GR4J). It was shown that the proposed data assimilation me-
 781 thod can be accurate and reliable for catchments of various sizes and under different cli-
 782 mates (but not necessarily with the same level of performance). The varying level of per-
 783 formance can be attributed to many different factors, and perhaps the first one that comes
 784 to mind is the varying quality of data. Specifically, the input data quality for the Los Ido-
 785 los catchment is deemed much less than for the other catchments (less ground stations and
 786 some stations where relocated). In addition, this study focused on only one conceptual mo-
 787 del. It is possible that this model may not be the best choice for all four catchments. It
 788 would be fairly straightforward to apply the proposed data assimilation technique to other
 789 lumped conceptual models. Extending it to more complex, physics-based, or even distribu-
 790 ted models is certainly possible, and likely of interest to the hydrological modeling com-
 791 munity, but would require much more work.

792 In this study, snow-related state variables were left out of the data assimilation pro-
 793 cess. One of those state variables is the percentage of the catchment covered with snow.
 794 Possible values obviously should remain between 0 and 100%, but it is difficult to constrain
 795 the outputs of neural networks between fixed boundaries, especially when each network
 796 has multiple outputs, each with different limits. Furthermore, there is a period of time du-
 797 ring the year where the percentage of the catchment covered by snow should be exactly
 798 zero. To include those constraints into MLP or ELM was found to be difficult (not trivial).
 799 However, because of the importance that accurate estimation of snow water equivalent
 800 can have, especially for Nordic watersheds, the inclusion of snow-related state variables
 801 in the proposed data assimilation method is, in our opinion, the obvious choice in terms
 802 of priorities for further studies. Among the possibilities to explore in order to assimilate
 803 snow-related state variables, some newer machine learning methods allow the enforcement
 804 of a non-negativity constraint [e.g., *Cannon, 2018; Fleming and Goodbody, 2019*]

805 Going back to the second hypothesis we wanted to verify, it turned out to be the
 806 opposite of that which was expected. ELM did not outperform the MLP for data assimi-
 807 lation. ELM led to less reliable estimations of the state variables' uncertainty. However,
 808 since ELM is a linear least-squares method, it is trivial (and computationally inexpensive)
 809 to update its output weights once new data is collected. Since this is exactly the premise
 810 on which data assimilation is based, the ELM method can be modified using "online se-
 811 quential" learning so that the parameters within ELM (output weights) can be made time-
 812 varying [*Seok Lim et al., 2013*], which could lead to further improvements in performance.

813 Additionally, MLP and ELM are not the only possibilities for a machine learning-based
 814 data assimilation framework. For example, the idea of using recurrent neural networks as in *Härter and de Campos Velho [2012]* is very interesting. In contrast to MLP,
 815 recurrent neural networks assume a correlation between sequential inputs, which is naturally the case for state variables, streamflow, and other hydro-meteorological variables (air
 816 temperature, soil moisture, etc.). The idea of using recurrent neural networks for data as-
 817 similation purposes is also advocated for theoretically in *Duane [2017]*. Other machine
 818 learning techniques could also be tested : Long-short term memory networks [LSTM, e.g.,
 819 *Kratzert et al., 2018*] (a new variation of recurrent neural networks that seek to mitigate
 820 the "vanishing gradient problem"), support vector regression, random forests, eXtreme
 821 Gradient Boosting, etc.

822 Finally, since this study was oriented toward developing the method and testing its
 823 efficacy (namely, performance and efficiency) as a proof of concept, we did not include
 824 a rigorous comparison with several other popular techniques in hydrology. Now that it is
 825 known that the proposed data assimilation methodology is effective, it would be interesting
 826 to pursue a large-scale comparison and explore whether data assimilation based on neural
 827 networks can outperform more traditional techniques, such as the EnKF and its variants or
 828 various types of particle filters, and compare their respective computational requirements.
 829

830 Acknowledgments

831 We would like to thank all the people and institutions who so kindly provided data for this
 832 study : Rodolfo Alvarado Montero (Main), Albrecht Weerts (Ourthe), Annie Poulin (Los
 833 Idolos) and the Direction de l'Expertise Hydrique et Atmosphérique at Quebec's provin-
 834 cial government (Mistassibi). We are also thankful for Kriz Villez who provided the Mat-
 835 lab codes for the Multilayer Perceptrons and to François Anctil and Gregory Seiller who
 836 provided the Matlab codes for GR4J and CemaNeige. Data is available in the Harvard Da-
 837 taverse, at https://dataverse.harvard.edu/dataverse/WRR_Boucher_etal2019/.
 838

839 Références

- 840 Abrahart R.J. et al. (2012), Two decades of anarchy ? emerging themes and outstanding
 841 challenges for neural network river forecasting, *Progress in Physical Geography*, 36(4),
 842 480–513, doi :doi:10.5194/hess-2017-169.
- 843 Aires, F., C. Pringent, W. Rossow, and M. Rothstein (2001), A new neural network ap-
 844 proach including first guess for retrieval of atmospheric water vapor, cloud liquid water
 845 path, surface temperature, and emissivities over land from satellite microwave observa-
 846 tions, *Journal of Geophysical Research*, 106(D14), 14,887–14,907.
- 847 Amaranto, A., F. Munoz-Arriola, G. Corzo, D. Solomatine, and G. Meyer (2018), Semi-
 848 seasonal groundwater forecast using multiple data-driven models in an irrigated cro-
 849 pland, *Journal of Hydroinformatics*, 20(6), 1227–1246, doi :10.2166/hydro.2018.002.
- 850 Atiquzzaman, M., and J. Kandasamy (2018), Robustness of Extreme Learning Machine in
 851 the prediction of hydrological flow series, *Computers and Geosciences*, 120, 105–114.
- 852 Beuzen, T., and J. Simmons (2019), A variable selection package driving Netica with Py-
 853 thon, *Environmental Modelling and Software*, 115, 1–5, doi :10.1016/j.envsoft.2019.01.
 854 018.
- 855 Boucher, M.-A., J.-P. Laliberté, and F. Anctil (2010), An experiment on the evolution of
 856 an ensemble of neural networks for streamflow forecasting, *Hydrology and Earth System
 857 Sciences*, 14(3), 603–612.
- 858 Cannon, A. (2018), Non-crossing nonlinear regression quantiles by monotone composite
 859 quantile regression neural network, with application to rainfall extremes, *Environmental
 860 Research and Risk Assessment*, 32(11), 3207–3225.
- 861 Chang, F.-J., Y.-M. Chiang, M.-J. Tsai, M.-C. Shieh, K. Hsu, and S. Sorooshian (2014),
 862 Watershed rainfall forecasting using neuro-fuzzy networks with the assimilation of

- 863 multi-sensor information, *Journal of Hydrology*, 508, 374–384.
- 864 Cintra, R., H. de Campos Velho, and H. Furtado (2013), Neural network for performance
865 improvement in atmospheric prediction systems : Data assimilation, 1st BRICS Coun-
866 tries & 11th CBIC Brazilian Congress on Computational Intelligence, Recife, Brasil.
867 Porto de Galinhas Beach.
- 868 Cover, T. M., and J. A. Thomas (2006), *Elements of Information Theory*, 2nd edition, 776
869 pp., Wiley-Blackwell, New York.
- 870 Doucet, A., S. Godsill, and C. Andrieu (2000), On sequential Monte Carlo sampling me-
871 thods for Bayesian filtering, *Statistics and Computing*, 10(3), 197–208.
- 872 Duan, Q., V. Gupta, and S. Sorooshian (1993), Shuffled complex evolution approach for
873 effective and efficient global minimization, *Journal of Optimization Theory and Applica-*
874 *tions*, 76(3), 501–521.
- 875 Duane, G. (2017), "FORCE" learning in recurrent neural networks as data assimilation,
876 *Chaos*, 27(12), doi :<https://doi.org/10.1063/1.4990730>.
- 877 Evensen, G. (1994), Sequential data assimilation with a nonlinear quasi-geostrophic mo-
878 del using Monte Carlo methods to forecast error statistics, *Journal of Geophysical Re-*
879 *search : Oceans*, 99(C5), 10,143–10,162.
- 880 Evensen, G. (2003), The ensemble Kalman filter : Theoretical formulation and practical
881 implementation, *Ocean Dynamics*, 53(4), 343–367.
- 882 Fleming, S., and A. Goodbody (2019), A machine learning metasystem for robust probabi-
883 listic nonlinear regression-based forecasting of seasonal water availability in the us west,
884 *IEEE Access*, 7, 119,943–119,964.
- 885 Fleming, S., D. Bourdin, D. Campbell, R. Stull, and T. Gardner (2015), Development and
886 operational testing of super-ensemble artificial intelligence flood-forecast model for a
887 pacific northwest river, *Journal of the American Water Resources Association*, 51(2),
888 502–512, doi :[10.1111/jawr.12259](https://doi.org/10.1111/jawr.12259).
- 889 Frenzel, S., and B. Pompe (2007), Partial mutual information for coupling analysis of
890 multivariate time series, *Physical Review Letters*, 99(20), 204,101–1–204,101–4, doi :
891 [10.1103/PhysRevLett.99.204101](https://doi.org/10.1103/PhysRevLett.99.204101).
- 892 Galelli, S., G. B. Humphrey, H. R. Maier, A. Castelletti, G. C. Dandy, and M. S. Gibbs
893 (2014), An evaluation framework for input variable selection algorithms for environ-
894 mental data-driven models, *Environmental Modelling and Software*, 62, 33 – 51, doi :
895 <https://doi.org/10.1016/j.envsoft.2014.08.015>.
- 896 Gneiting, T., and A. E. Raftery (2007), Strictly proper scoring rules, prediction, and esti-
897 mation, *Journal of the American Statistical Association*, 102(477), 359–378.
- 898 Good, I.-J. (1952), Rational decisions, *Journal of the Royal Statistical Society, Series B*,
899 14(1), 107–114.
- 900 Härter, F., and H. de Campos Velho (2008), New approach to applying neural network in
901 nonlinear dynamic model, *Applied Mathematical Modelling*, 32(12), 2621–2633.
- 902 Härter, F., and H. de Campos Velho (2012), Data assimilation procedure by recurrent neu-
903 ral network, *Engineering Applications of Computational Fluid Mechanics*, 6(2), 224–233.
- 904 Hecht-Nielsen, R. (1990), *Neurocomputing*, 433 pp., Addison-Wesley, Reading, MA.
- 905 Hejazi, M. I., and X. Cai (2009), Input variable selection for water resources systems
906 using a modified minimum redundancy maximum relevance (mMRMR) algorithm, *Ad-*
907 *vances in Water Resources*, 32(4), 582–593, doi :[10.1016/j.advwatres.2009.01.009](https://doi.org/10.1016/j.advwatres.2009.01.009).
- 908 Hornik, F. (1991), Approximation capabilities of multilayer feedforward networks, *Neural
909 Networks*, 4(2), 251–257.
- 910 Huang, G.-B., H. Zhou, X. Ding, and R. Zhang (2012), Extreme learning machine for re-
911 gression and multiclass classification, *Transactions on Systems, Man and Cybernetics
912 Part B*, 42(2), 513–529, doi :[10.1109/TSMCB.2011.2168604](https://doi.org/10.1109/TSMCB.2011.2168604).
- 913 Kasiviswanathan, J., K.S.and He, K. Sudheer, and J. Tay (2016), Potential application of
914 wavelet neural network ensemble to forecast streamflow for flood management, *Journal
915 of Hydrology*, 536, 161–173, doi :[10.1016/j.jhydrol.2016.02.044](https://doi.org/10.1016/j.jhydrol.2016.02.044).

- 916 Kirchner, J. (2006), Getting the right answer for the right reasons : linking measurements,
 917 analyses, and models to advance the science of hydrology, *Water Resources Research*,
 918 42(3), doi :10.1029/2005WR004362.
- 919 Kratzert, F., D. Klotz, C. Brenner, S. K., and M. Herrnegger (2018), Rainfall-runoff mo-
 920 delling using long short-term memory (LSTM) networks, *Hydrology and Earth System
 921 Sciences*, 22(11), 6005–6022.
- 922 Kugiumtzis, D. (2013), Direct-coupling information measure from nonuniform embedding,
 923 *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 87(6), 1–14, doi :
 924 10.1103/PhysRevE.87.062918.
- 925 Levenberg, K. (1944), A method for the solution of certain non-linear problems in least
 926 squares, *Quarterly Journal of Applied Mathematics*, 2(2), 164–168.
- 927 Li, X., A. C. Zecchin, and H. R. Maier (2015), Improving partial mutual information-
 928 based input variable selection by consideration of boundary issues associated with band-
 929 width estimation, *Environmental Modelling and Software*, 71, 78–96, doi :10.1016/j.
 930 envsoft.2015.05.013.
- 931 Lima, A., A. Cannon, and W. Hsieh (2016), Forecasting daily streamflow using online
 932 sequential extreme learning machines, *Journal of Hydrology*, 537, 431–443.
- 933 Lima, A., W. Hsieh, and A. Cannon (2017), Variable complexity online sequential extreme
 934 learning machine, *Journal of Hydrology*, 555, 983–994.
- 935 Liu, Y., A. Weerts, M. Clark, H. Hendricks Franssen, S. Kumar, H. Moradkhani, D.-J.
 936 Seo, D. Schwandenb erg, P. Smith, A. van Dijk, M. can Velzen, N. ans He, H. Lee,
 937 S. Noh, O. Rakovec, and P. Restrepo (2012), Advancing data assimilation in operational
 938 hydrologic forecasting : progresses, challenges, and emerging opportunities, *Hydrology
 939 and Earth System Science*, 16(10), 3863–3887, doi :10.5194/hess-16-3863-2012.
- 940 Mandel, J. (2006), Efficient implementation of the ensemble kalman filter, *Tech. rep.*, Uni-
 941 versity of Colorado at Denver and Health Sciences Center, Denver.
- 942 Marquardt, D. (1963), An algorithm for least-squares estimation of nonlinear parameters,
 943 *SIAM Journal of Applied Mathematics*, 11(2), 431–441.
- 944 Matheson, J. E., and R. L. Winkler (1976), Scoring rules for continuous probability distri-
 945 butions, *Management Science*, 22(10), 1087–1096.
- 946 May, R. J., G. Dandy, and H. Maier (2011), Review of input variable selection methods
 947 for artificial neural networks, *Artificial Neural Networks-Methodological Advances and
 948 Biomedical Applications*, pp. 19–44.
- 949 Oudin, L. (2004), Recherche d'un modèle d'évapotranspiration potentielle pertinent
 950 comme entrée d'un modèle pluie-débit global, Ph.D. thesis, ENGREF (AgroParisTech).
- 951 Perrin, C., C. Michel, and V. Andreassian (2003), Improvement of a parsimonious model
 952 for streamflow simulation, *Journal of Hydrology*, 279(1-4), 275–289.
- 953 Quilty, J., J. Adamowski, B. Khalil, and M. Rathinasamy (2016), Bootstrap rank-ordered
 954 conditional mutual information (broCMI) : A nonlinear input variable selection method
 955 for water resources modeling, *Water Resources Research*, 52(3), 2299–2326, doi :10.
 956 1002/2015WR016959.
- 957 Reichle, R., J. Walker, R. Koster, and P. Houser (2002), Extended versus ensemble Kal-
 958 man filtering for land data assimilation, *Journal of Hydrometeorology*, 3, 728–740.
- 959 Rizk, Y., and M. Awad (2019), On extreme learning machines in sequential and time se-
 960 ries prediction : A non-iterative and approximate training algorithm for recurrent neural
 961 networks, *Neurocomputing*, 325, 1–19.
- 962 Rodríguez-Fernández, N., P. de Rosnay, C. Albergel, P. Richaume, F. Aires, C. Prigent,
 963 and Y. Kerr (2019), SMOS neural network soil moisture data assimilation in a land sur-
 964 face model and atmospheric impact, *Remote Sensing*, 11(11), 1334–1334.
- 965 Rosenblatt, F. (1958), The perceptron : A probabilistic model for information storage and
 966 organization in the brain, *Psychological Review*, 65(6), 386–408.
- 967 Seok Lim, J., S. Lee, and H. S. Pang (2013), Low complexity adaptive forgetting factor
 968 for online sequential extreme learning machine (OS-ELM) for application to nonstatio-
 969 nary system estimations, *Neural Computing and Applications*, 22(3-4), 569–576.

- 970 Sharma, A. (2000), Seasonal to interannual rainfall probabilistic forecasts for improved
 971 water supply management : Part 1 - A strategy for system predictor identification, *Journal of Hydrology*, 239(1-4), 232–239, doi :10.1016/S0022-1694(00)00346-2.
- 972 Sieck, M., and D. Solomatine (2011), Real-time data assimilation for chaotic storm surge
 973 model using NARX neural network, *Journal of Coastal Research*, 64, 1189–1194.
- 974 Snauffer, A., W. Hsieh, A. Cannon, and M. Schnorbus (2018), Improving gridded snow
 975 water equivalent products in british columbia, canada : multi-source data fusion by neu-
 976 ral network models, *The Cryosphere*, 12(3), 891–905.
- 977 Thibault, A., and F. Anctil (2015), On the difficulty to optimally implement the ensemble
 978 kalman filter : An experiment based on many hydrological models and catchments,
 979 *Journal of Hydrology*, 529(Part 3), 1147–1160.
- 980 Thibault, A., F. Anctil, and M.-A. Boucher (2016), Accounting for three sources of uncer-
 981 tainty in ensemble hydrological forecasting, *Hydrology and Earth System Science*, 20(5),
 982 1809–1825, doi :doi:10.5194/hess-20-1809-2016.
- 983 Thibault, A., C. Poncelet, and F. Anctil (2019), User manual : Hoopla version 1.0.2, *Tech.
 984 rep.*, Université Laval, Quebec City, Quebec, Canada.
- 985 Valery, A. (2010), Modélisation précipitations-débit sous influence nivale elaboration
 986 d'un module neige et évaluation sur 380 bassins versants, Ph.D. thesis, L'Institut des
 987 Sciences et Industries du Vivant et de l'Environnement (AgroParisTech).
- 988 Wahle, K., J. Staneva, and H. Guenther (2015), Data assimilation of ocean wind waves
 989 using neural networks. a case study for the german bight, *Ocean Modelling*, 96(Part 1),
 990 117–125.
- 991 Wang, L., and C. Wan (2008), Comments on "the extreme learning machine", *IEEE Trans-
 992 action on Neural Networks*, 19(8), 1494–1495.
- 993 Westra, S., M. Thyer, M. Leonard, D. Kavetski, and M. Lambert (2014), A strategy for
 994 diagnosing and interpreting hydrological model nonstationarity, *Water Resources Re-
 995 search*, 50(6), 5090–5113.
- 996 Wilks, D. (1995), *Statistical Methods in the Atmospheric Sciences*, 467 pp., Academic
 997 Press, San Diego.
- 998 Yaseen, Z., S. Sulaiman, R. Deo, and K.-W. Chau (2019), An enhanced extreme learning
 999 machine model for river flow forecasting : State-of-the-art, practical applications in wa-
 1000 ter resource engineering area and future research direction, *Journal of Hydrology*, 569,
 1001 387–408.
- 1002 Yousefi, P., G. Naser, and H. Mohammadi (2018), Surface water quality model : Impacts
 1003 of influential variables, *Journal of Water Resources Planning and Management*, 144(5),
 1004 1–10, doi :10.1061/(ASCE)WR.1943-5452.0000900.