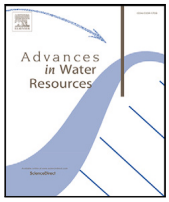
Advances in Water Resources 193 (2024) 104796

Contents lists available at ScienceDirect

Advances in Water Resources

journal homepage: www.elsevier.com/locate/advwatres



Generating interpretable rainfall-runoff models automatically from data

Travis Adrian Dantzer *, Branko Kerkez

Civil and Environmental Engineering, University of Michigan, 2350 Hayward St, Ann Arbor, Michigan, USA

ARTICLE INFO

Keywords:
Model discovery
Rainfall-Runoff
Dynamical systems
Explainable AI
Surrogate modeling
Data-driven

ABSTRACT

A sudden surge of data has created new challenges in water management, spanning quality control, assimilation, and analysis. Few approaches are available to integrate growing volumes of data into interpretable results. Process-based hydrologic models have not been designed to consume large amounts of data. Alternatively, new machine learning tools can automate data analysis and forecasting, but their lack of interpretability and reliance on very large data sets limits the discovery of insights and may impact trust. To address this gap, we present a new approach, which seeks to strike a middle ground between process-, and data-based modeling. The contribution of this work is an automated and scalable methodology that discovers differential equations and latent state estimations within hydrologic systems using only rainfall and runoff measurements. We show how this enables automated tools to learn interpretable models of 6 to 18 parameters solely from measurements. We apply this approach to nearly 400 stream gaging sites across the US, showing how complex catchment dynamics can be reconstructed solely from rainfall and runoff measurements. We also show how the approach discovers surrogate models that can replicate the dynamics of a much more complex process-based model, but at a fraction of the computational complexity. We discuss how the resulting representation of watershed dynamics provides insight and computational efficiency to enable automated predictions across large sensor networks.

Plain language summary

As the water sector adopts more sensors, few tools are available to deal with the resulting volumes of data. Experts have created valuable watershed models, but calibrating the models is labor intensive which limits use of real-time data. Machine learning can automate model building, but the resulting outputs are difficult to interpret. This paper presents a method that combines physics-based modeling with data to automatically build rainfall runoff models. Because of how computationally cheap and simple the model is, it has great potential for use in modeling and forecasting.

1. Introduction

Watershed data are increasingly available due to expanding sensor networks. This sudden surge of measurements has created new challenges in water management as quality controlling, processing, and integrating these data requires a great deal of highly skilled labor (Devia et al., 2015). Many process-based models were created in the context of time series data scarcity and are therefore not readily amenable to data ingestion or data assimilation (Castelletti et al., 2012; Francipane et al., 2012; Sorooshian et al., 2008; Silberstein, 2006). Machine learning approaches can automate data ingestion, Sarafanov et al. (2021) and Olson and Moore (2016) but their opacity may limit insight and stakeholder trust (Jajarmizadeh et al., 2012; Babovic and Abbott, 1997a). Both approaches require large amounts of computational power and data (Kumar et al., 2013; Wagena et al., 2020).

In light of these challenges, a parsimonious middle ground has been sought between pure process-based and data-driven modeling, posing the question: *in the age of ever increasing new data, how can the interpretability and comprehensibility of process-based models be preserved, while taking advantage of the scalability of data-driven methods*?

This paper introduces a new method to discover rainfall-runoff equations strictly from raw sensor data. The contribution of this work is an automated and scalable methodology, which discovers differential equations and estimates latent states within hydrologic systems using only rainfall and runoff measurements. These latent states represent transient storage in the catchment such as increased soil moisture, overland flow, and elevated groundwater levels. The novelty of the approach lies in its ability to discover fundamental equations governing hydrologic systems solely from data, while preserving the automaticity of data-driven methods.

* Corresponding author.
E-mail address: dantzert@umich.edu (T.A. Dantzer).

https://doi.org/10.1016/j.advwatres.2024.104796
Received 24 October 2023; Received in revised form 27 June 2024; Accepted 22 August 2024
Available online 28 August 2024

2. Background

Prior efforts have pursued a combination of automaticity and interpretability through modification of existing process-based models, theoretical derivation, and explainable AI (XAI) (Devia et al., 2015; Angelov et al., 2021; Jajarmizadeh et al., 2012; Babovic and Abbott, 1997b). Process-based hydrologic models are often derived from first principles and thus support interpretability, but their calibration and construction is highly manual. While XAI has made gains in the interpretability of machine learning methods, the resulting model parameters may still not be straightforwardly mapped to readily understood hydrologic concepts. This may make it difficult to understand or tune model behavior using domain knowledge (Angelov et al., 2021; Tian et al., 2022).

Methodological and computational advances have led to significant progress in hydrologic modeling, with the ability to represent more granular process and complexities. Once a model is developed and calibrated, it can be used to study scenarios by changing its many parameters and inputs. Reformulating an existing computational model can speed simulation, enable the use of tools not compatible with the original form, or support the interpretation of model features for domain understanding (Welch et al., 1995; Hespanha, 2018; Bartos and Kerkez, 2021; Alex et al., 2020). If the original model was already calibrated, these approaches need little additional data (Troutman et al., 2017) or computational power (Santos et al., 2018). Even with these barriers eliminated, a reliance on manual processes presents a bottleneck (Wong and Kerkez, 2018) given the large data sets and systems involved.

A tractable system of differential equations can also be used to represent catchment dynamics, and can be constructed by derivation from first principles with appropriate simplifying assumptions (Kirchner, 2009). The interpretability and tunability of differential equation-based models is good as these models are rooted in physics and established domain principles. Modeling hydrologic systems using differential equations is also computationally efficient, particularly if the derived algebraic relations have a closed-form solution (Jakeman and Hornberger, 1993). However, implementation may be difficult due to data requirements, time granularity, and limited automaticity. These approaches are typically demonstrated on unusually long and clean records provided from experimental catchments (Kirchner, 2009). They also tend to focus on daily data, Song et al. (2019) which may not be granular enough to support many important applications. Lastly, appropriate simplifying assumptions may differ by catchment, which makes data processing manual.

A principal advantage of machine learning-based methods is automaticity, but a core critique has focused on the "black box" nature of the resulting models. In the field of XAI, techniques such as dynamic mode decomposition (Schmid, 2022), Koopman operator construction (Mauroy et al., 2020), and others (Juang and Pappa, 1985; Ho and Kalman, 1966) offer structures ripe for mathematical interpretation (Tian et al., 2022). However, this mathematical interpretability may not relate tractably to physical characteristics of the system or well understood hydrological concepts. Genetic programming has also been a prominent approach within XAI (Babovic and Abbott, 1997a,b). This approach seeks to develop governing equations directly from data using an evolutionary approach (Babovic and Keijzer, 2002). More recent work has explored embedding expert knowledge to improve the accuracy and interpretability of the generated models (Babovic, 2009). The Long Short-Term Memory (LSTM) networks used for rainfall-runoff modeling in Kratzert et al. 2018 and 2019 provide some interpretability through examination of the evolution of the cell states throughout time (Kratzert et al., 2019, 2018). Work in Jiang et al. (2022) extended the interpretation of LSTMs to explicitly attribute flow peaks to snowmelt or rainfall events and examine characteristic differences between catchments. However, the models in Kratzert et al. (2019, 2018) have over 10,000 parameters which obscures the relation to catchment characteristics.

Hybrid modeling combines process-based modeling with data-driven approaches and often results in better, more consistent results (Kapoor et al., 2023; Fathian et al., 2019). One common approach within this vein is training ML models on the errors or residuals remaining between observations and process-based outputs (Schneider et al., 2022). To the extent that the process-based model predicts the observations, the results can be traced back to the model structure. That is, when the residuals the ML component is correcting for are small, interpretability is still similar to purely process-based models. Another approach is using domain knowledge to decompose a large problem into smaller subproblems which are easier to train data-driven models on. For example, predicting effluent contaminant concentrations from a resource recovery facility directly from the characteristics of the input may be more difficult than building a network of smaller models approximating intermediate processes within the facility. With this approach it may also be clearer which processes are poorly modeled and thereby causing error in the final prediction.

To illustrate the challenge of model selection via example, we consider a measured storm in a mid-size city in the Midwestern United States in the Summer of 2022 (Fig. 1). The hydrograph is generated across 30 km^2 of urbanized landscape and responds to rainfall events with a large initial peak, smaller delayed peak, and nonlinear recession. In pursuit of the "middle ground" we referred to in the introduction, we would like to have an automatically generated model with sufficient fidelity to represent these unusual dynamics, but which is still amenable to inspection and tuning. "Out of the box" solutions could include a machine-learning approach, which would need many more storm observations and a high parameter space to capture these dynamics. A process-based hydrologic model could capture these dynamics, but would need to be calibrated across many physiographic inputs and process parameters (infiltration, runoff, routing, etc.). Alternatively, a model could be derived from first principles and manually increased in complexity until the dynamics are adequately captured. To that end, two approaches which may supply the automaticity and interpretability we seek are unit hydrographs – well established, but limited in their ability to capture complex dynamics – and differential equation discovery.

The unit hydrograph approach reduces streamflow prediction to a transformation of the rainfall time series by assuming watersheds are linear and time invariant (LTI) dynamical systems (Bedient et al., 2008). This is computationally efficient, handles delay well, and can acceptably approximate watershed response in many cases. However, this approach is generally unable to represent complex responses such the example in Fig. 1. A unit hydrograph based on a gamma distribution (Ghorbani et al., 2017) can fit either the initial peak (Fig. 1, first row, dot-dash red) or recession (Fig. 1, first row, dotted cyan), but not both.

Alternatively, representing catchment response as a system of differential equations would be computationally efficient and present a model structure prevalent in the hydrology community. Model discovery automatically generates differential equations from data, and has been demonstrated for systems ostensibly more complex than watersheds (de Silva et al., 2020; Brunton et al., 2016). However, due to their instantaneous nature, differential equations cannot represent delayed causation (Fig. 1, second row).

Given these challenges, in the following section we introduce a new methodology (Fig. 1, third row) to recover the delay and dynamic complexity of watersheds solely from data, while providing the ability to evaluate the resulting model structure and parameters in the context of hydrologic domain knowledge.

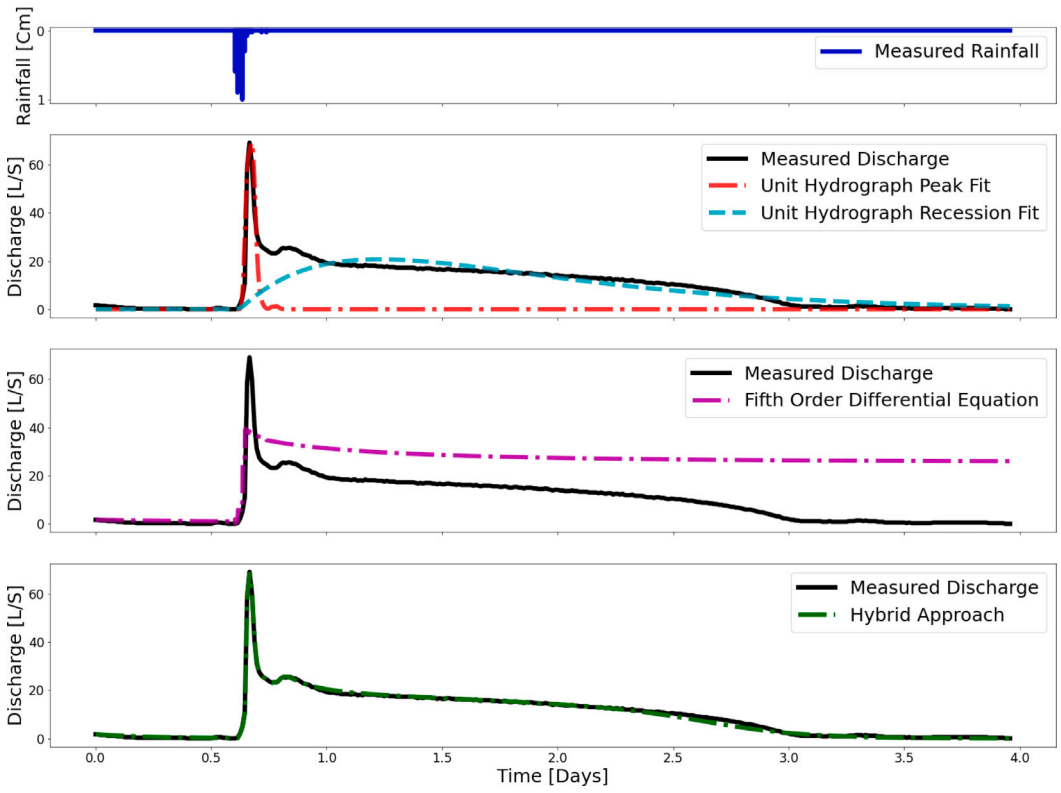


Fig. 1. Hybrid approach captures delay and complexity (USGS, 2016) Rainfall (blue) is the average of two rain gages located in and near the catchment of the stream. Discharge (solid black) is modeled by: transforming rainfall using unit hydrographs (dot-dash red and dotted cyan, row 1), discovering a differential equation relating rainfall and discharge (dot-dash magenta, row 2), and a new method combining these approaches (dot-dash green, row 3).

3. Methods

In this study, we introduce Model Discovery in Partially Observable Dynamical Systems (Dantzer, 2023b), which we also implement as an open-source software tool. We present the single input, single output hydrologic analogy of the method here for conciseness.

The delay between rainfall (P) and runoff (Q) is typically thought of in terms of time, but can also be conceptualized as a gap in observability. Though we perceive delayed causation, the processes between rainfall and runoff (e.g., soil saturation, flow routing) are actually instantaneous. It is our inability to fully observe the catchment that gives the *appearance* of delay. Instead of finding the time-shift between precipitation and discharge, we represent this delayed and diffused causation by optimally estimating the unit hydrographs of the effective subcatchments (unobserved states) within the watershed. We then learn a differential equation relating these subcatchment unit hydrographs to the measured hydrograph. Young's data-based mechanistic modeling approach has similar aims and results, but addresses partial observability by including pure delay terms in the differential equations used to model the system (Young, 2012, 2006). Genetic programming approaches also tend to represent delay as "pure" or "advective" (Babovic and Abbott, 1997b).

Modeling the delay and dispersion between rainfall and runoff as intervening unobservable states has origins at least as old as Nash's suggestion (Nash, 1959) of a cascade of linear reservoirs to approximate a unit hydrograph. Sugawara's Tank model (Lee et al., 2020; Chadalawada et al., 2020; Herath et al., 2021b,a) has a different topology, but evinces the same notion. The approach presented here is similar, but more dynamically flexible and amenable to real-time data ingestion. The catchment is thought of as being composed of effective subcatchments (Fig. 2) that represent not geographic areas, but constituent processes such as surface runoff, tributary contribution, or interflow. In our formulation, the only sources of data to estimate the parameters of these subcatchments are rainfall and streamflow. We achieve this by transforming the rainfall forward to estimate the unit hydrographs and then using those unit hydrographs to learn the differential equation that governs the discharge.

3.1. Watersheds as dynamical systems of subcatchments

We approximate the subcatchment unit hydrographs using a gamma probability density function (Eq. (1)). This function has been used for approximating unit hydrographs in prior studies (Ghorbani et al., 2017; Nadarajah, 2007; Haktanir and Sezen, 1990):

$$g(t+d,\alpha,\beta) = \frac{\beta^{\alpha} t^{\alpha-1} e^{-\beta t}}{(\alpha-1)!} \tag{1}$$

where β is the rate or inverse scale, α is the shape, and d is the location. These parameters have interpretable meanings in the context of hydrologic dynamics. Decreasing the β parameter delays and broadens the peak of the unit hydrograph. The α parameter controls asymmetry about the peak (skewness). The delay parameter (d) shifts the transformation in time without affecting shape. The time to peak is $T_p = \frac{\alpha-1}{\beta} + d$. Note that $\alpha \geq 1$ in this study. The similar expression $T_{50} = \frac{\alpha}{\beta} + d$ is the time at which half the total contribution has been made. T_{50} can be thought of as the mean or center of gravity of the transformations shown in dotted red and dotted yellow in Fig. 2.

The approximations of the subcatchment unit hydrographs are evaluated by how strongly they connect observed precipitation (p_o) with observed runoff (q_o) through the following differential equation:

$$\frac{dq_o}{dt} = f(q_o, p_o, T_i(p_o)) \tag{2}$$

where $\frac{dq_o}{dt}$ is the instantaneous rate of change of water level or discharge, $T_i(p_o)$ are the subcatchment unit hydrographs generated by transforming the input rainfall p_o, and f is some nonlinear function of compatible dimension. f is then constrained as follows: **(1)** constant

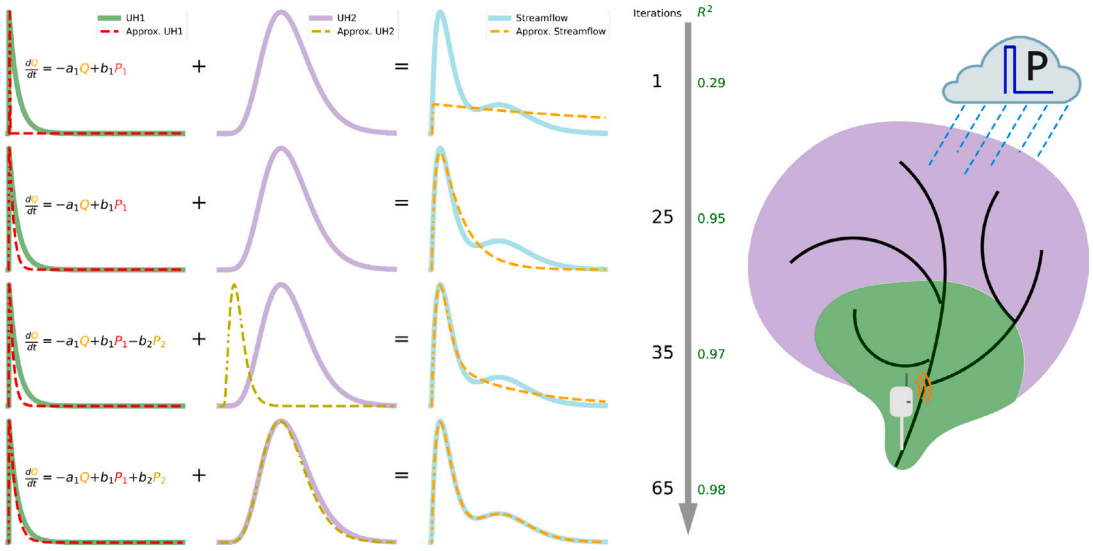


Fig. 2. Recovering the subcatchment hydrographs. Subcatchments 1 (first column solid green) and 2 (second column solid purple) contribute to streamflow (third column solid cyan). Only precipitation (impulse, not pictured) and streamflow are observed. The subcatchment unit hydrographs are approximated (dotted red first column and dotted yellow second column) by transforming the precipitation. A differential equation is discovered (left column) to relate these subcatchment hydrographs to streamflow (third column dotted orange). Iteration number and the coefficient of determination are indicated on the right. **Row 1.** The algorithm generates a starting guess for the first subcatchment. **Row 2.** The algorithm finds an optimal representation of the system using a single subcatchment. **Row 3.** Accuracy is below desired, so the algorithm adds a second subcatchment (second column dotted yellow). **Row 4.** The subcatchment hydrographs and the differential equation relating them to the streamflow are recovered.

bias terms are omitted because $\frac{dq_o}{dt}$ is not a function of time, **(2)** interaction terms are omitted because the instantaneous interaction between rain and stage has no physically meaningful interpretation, and **(3)** only polynomial terms are included because they are sufficient to capture the dynamics at the resolution we desire (Taylor's Theorem). This yields:

$$\frac{dq_o}{dt} = P_q(q_o) + P_p(p_o, T_i(p_o)) \tag{3}$$

where P_q and P_p are polynomials excluding the zero order term. P_q describes the shape of the recession curve as it is the autocorrelation on stage or discharge. P_p describes the contributions of the instantaneous precipitation p_o and subcatchment unit hydrographs $T_i(p_o)$.

The most complex models trained in this study will have polynomial order three and two effective subcatchment unit hydrographs. They have the form:

$$\begin{aligned}\frac{dq_o}{dt} =& a_1 q_o + a_2 q_o^2 + a_3 q_o^3 + b_{01} p_o + b_{02} p_o^2 + b_{03} p_o^3 \\ &+ b_{11} T_1(p_o) + b_{12} T_1(p_o)^2 + b_{13} T_1(p_o)^3 \\ &+ b_{21} T_2(p_o) + b_{22} T_2(p_o)^2 + b_{23} T_2(p_o)^3\end{aligned} \tag{4}$$

All results presented in this study will be produced by models which are no more than this ordinary differential equation and six unit hydrograph parameters that describe the shapes of T_1 and T_2. This model configuration has 18 parameters. The differential equations are integrated using the Explicit Runge–Kutta method of order 5 provided by Scipy's solve ivp function (Virtanen et al., 2020).

3.2. Approximating the subcatchment unit hydrographs using data

The Sparse Identification of Nonlinear Dynamics (SINDy) (de Silva et al., 2020) algorithm is used to predict the derivative of the output using Eq. (3). We denote x_i' as the measured derivative and y_i' as our estimate. The established SINDy implementation chooses the coefficients of the differential equation (P_q and P_p) to maximize the coefficient of determination (R^2) between the observed and predicted derivative.

$$\max_{P_q, P_p} \left[1 - \frac{\sum (x_i' - y_i')^2}{\sum (x_i' - \bar{x'})^2} \right] \tag{5}$$

Finding optimal differential equation coefficients (a_i, b_j) is the inner loop. The outer loop is the optimization of the subcatchment unit hydrographs (T_i) shown in Algorithm 1. As there is no analytical derivative this optimization is performed via compass search. The steps proceed as follows:

(1): The algorithm starts with one subcatchment and evaluates increasing numbers of subcatchments until the maximum number is reached or the last one added less than 0.5% to the R^2 score. If the returns of the last one added are marginal, it is removed and the model with one less subcatchment is returned as the final model. That is, if the jth subcatchment produces marginal returns, the model with j rainfall transformations is the last to be evaluated and a model with $j-1$ transformations is returned. If $m = m_{max}$ delivers non-marginal returns on accuracy, the model with m_{max} subcatchments is returned. Note that for this study we build models of one and two effective subcatchments and evaluate both.

(2): An initial guess is chosen for the subcatchment unit hydrograph. For records with more than one event (Fig. 3 and beyond) the first transformation is Eq. (1) with parameters $(\alpha, \beta, d) = (1, 1, 0)$ and additional transformations are broader peaks centered at timesteps 24, 48, and so on. For records of only one event (Figs. 1 and 2), the starting shapes are based on timesteps of maximum derivative in the output. By identifying timesteps with large derivatives we are attempting to identify distinct events or times of arrival in the delayed causation between input and output. As identifying these times of arrival is less clear when multiple driving events occur, a simpler heuristic is used in that case.

(3): Larger s corresponds to a larger perturbation. So the optimization begins looking at large perturbations and converges to small steps.

(4): Candidate unit hydrographs are generated by perturbing the shape of the last iteration's best performing transformation. The perturbations are scaled by s. In this paper we evaluate eight perturbations, but that number is not specific to this application. Rather, it is because we define the optimization space to have four axes: one for each of the three parameter values (α, β, d) and a fourth which changes α and β at the same time. This fourth direction changing α and β at the same time makes the distribution wider or narrower without affecting its center of mass.

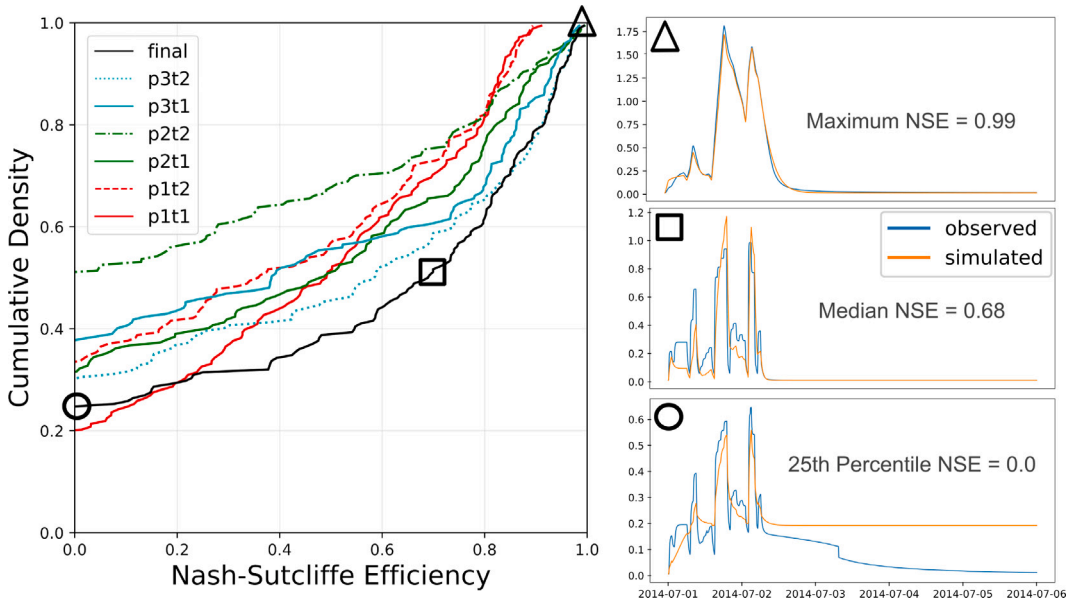


Fig. 3. Reduction of a process-based model. Models are evaluated at 178 junctions, comparing the predictions made by modpods to the process-based model's output for the testing storm. The cumulative density function of Nash Sutcliffe Efficiency is shown on the left. In the legend, p prefixes the polynomial order while t prefixes the number of subcatchments. The "final" model selects the p and t configuration that resulted in the highest training NSE score for each junction. On the right are evaluation simulations of junction depth in meters for models at the 25th percentile, median, and maximum of evaluation NSE (0.0, 0.68, and 0.99 respectively).

(5): The 9 input transformations (8 perturbations and the best from last time) are scored by how well the differential equation identification procedure performs as measured by R^2.

(6): The best performing input transformation is saved.

(7): If the best performing input transformation is the same as last iteration, none of the perturbations were helpful. s is reduced by the factor α. If s is now less than one, the algorithm will exit the loop at **(3)**. If the best performing input transformation is not the same as the last iteration, a step was taken that did improve performance. In that case, we return to **(4)**.

Algorithm 1. Approximate Subcatchment Unit Hydrographs

1. $for\ (m = 1\ ;\ m = m + 1\ ; m \leq m_{max})$:
   2. $c'_{i-1} = c_0\ ;\ i = 0$
   3. $for\ (s = s_o\ ;;\ s > 1)$:
      4. $Generate\ C(c'_{i-1}, s)$
      5. $\forall C\ :\ Scores = \max_{P_q, P_p}\left[1 - \frac{\sum(x'_i - y'_i)^2}{\sum(x'_i - x')^2}\right]$
      6. $c'_i = C[argmax[Scores]]\ ;\ i = i + 1$
      7. $if\ (c'_i == c'_{i-1})\ : s = a \cdot s$
8. $End\ if\ \Delta R^2 < \Delta_{min}\ from\ last\ m.$

where:
m is the number of input transformations,
c_0 is the initial set of input transformations,
c'_i is the best performing set of input transformations for iteration i,
i is the iteration number,
s determines how far away the compass search looks,
s_o is the initial spread of the compass search,
$a < 1$ determines how quickly s decays,
C is the 9 candidate transformation sets including the best from last iteration,
Δ_{min} is the minimum accuracy increase from an additional transformation.

A conceptual illustration of the algorithm is shown in Fig. 2. In the figure, a synthetic hydrograph is the outflow from two "subcatchments" which are series of linear reservoirs. As the algorithm iterates, it recovers the dynamics of the two subcatchments and combines them using a differential equation to reconstruct the output hydrograph.

Once Algorithm 1 terminates and the model is trained, the total number of parameters is:

$$3m + q(2 + m) \tag{6}$$

where m is the number of input transformations (subcatchments) and q is the order of the polynomial terms included in the differential equation. Each input transformation (m) adds three parameters that describe the shape of the subcatchment unit hydrograph. Increasing the polynomial order (q) adds a term for the output autocorrelation, instantaneous precipitation, and each precipitation transformation. As an example, Eq. (4) is the most complex model trained in this study and has six parameters describing the shapes of the two subcatchment unit hydrographs and twelve coefficients in the differential equation for a total of eighteen parameters. The simplest models are linear differential equations with one effective subcatchment and have six parameters total.

3.3. Evaluation and implementation

To evaluate our *modpods* approach, we first apply it in a simulated setting to determine how well it can replicate dynamics that are produced by a more complex, process-based model. We then apply the algorithm to discover models from measurements at 35 US Geological Survey (USGS) stream gages across the southern United States. To provide a comparison of performance, we also discover models at 348 sites within the Catchment Attributes and Meteorology for Large-sample Studies (CAMELS) dataset. This dataset is particularly valuable as it includes a large number of catchments of varying climate, size, and topography. Additionally, benchmark rainfall-runoff models including process-based and machine learning techniques are available (Addor et al., 2017b). Lastly, to interpret how model parameters relate to watershed characteristics and hydrologic signatures, we analyze correlations for the USGS and CAMELS datasets (Addor et al., 2017a).

3.3.1. Performance metrics and interpretation analysis

We train models with polynomial orders of 1, 2, or 3, with 1 or 2 subcatchments, for a total of six model configurations. These are denoted as "p-polynomial order, t-number of subcatchments". The

complexity of the model structure is defined by p and t, which can be thought of as hyperparameters. The only constraint on the coefficients of the differential equation is that the highest order output autocorrelation term is negative. This is to ensure that finite forcing produces a finite response.

The Nash–Sutcliffe Efficiency (NSE) – a common hydrologic error metric – is used as the primary measure of performance during evaluation. Additional error metrics including MAE, RMSE, and various bias measures are also included in the supplemental information. NSE is defined as one minus the ratio of the error variance of the model divided by the variance of the observed time series:

$$NSE = 1 - \frac{\sum(Q_{obs} - Q_{mod})^2}{\sum(Q_{obs} - \overline{Q_{obs}})^2} \tag{7}$$

A perfect model scores 1 while a model with error equal to the variance of the observed time series scores a 0. The NSE is measured for all model configurations. To select the hyperparameters p and t we choose the model configuration with the best training NSE for a given site or junction. That is, out of the six possible model configurations, we label the "final model" for each site or junction as the model configuration with the best training NSE. While machine-learning approaches typically use a three-way data split (training, validation, and testing), conceptual and process-based models often use a two-way split (e.g., calibration and validation in Newman et al. (2017)). While *modpods* is not a hydrologic model, the number of tunable parameters is more similar to conceptual hydrologic models than to machine learning approaches and we thereby use a two-way split. For example, the number of tunable parameters in Newman et al. (2017) varies from 2 to 13, close to our range of 6 to 18 parameters. Several studies have addressed the nuanced question of where and how to split data into training (calibration) and testing (validation) sets (Maier et al., 2023; Zheng et al., 2022; Chen et al., 2022). In this study our data splits are always wind-up, then train, then test as this arrangement best aligns with the use cases we anticipate detailed in the Future Directions section.

Finally, we carry out a two-fold interpretation analysis. Firstly, we correlate the final model parameters with physiographic features of the catchment to infer how changes in catchment composition may influence rainfall-runoff dynamics of the resulting *modpods* model. Secondly, we interpret the final model in terms of its hyperparameters to posit how the mathematical structure may be related to physical features of the underlying catchment.

3.3.2. Implementation

The complete implementation of this toolchain, example notebooks showing applications using several data sources, and code to generate the figures in this paper are freely shared (Dantzer, 2023a). That library depends on *modpods*, which we also share freely (Dantzer, 2023b). All analyses took place on a laptop with 32 GB RAM and an Intel(R) Core(TM) i7-1065G7 CPU @ 1.30 GHz 1.50 GHz processor.

4. Model reduction

4.1. Experiment setup

First, we apply our approach to data generated by a process-based urban water model (McDonnell et al., 2020; Huber, 1985). The motivation is two-fold: (1) using a process model guarantees causality between input and output, thus providing a controlled environment in which to test the baseline performance of the approach, and (2) it demonstrates a secondary benefit of the approach by performing model reduction (surrogate modeling) and showing how complex process models can be reduced to the formulation presented herein.

Focusing on an urban watershed in the Midwestern US, we replicate the junction depths of an EPA Stormwater Management Model (EPA-SWMM) with 420 storage nodes, 1200 junctions, and 1800 subcatchments. The calibrated model was shared with us by a municipality and represents an urban stormwater system for roughly 100,000 people. While EPA-SWMM may be simpler than some research-grade models, its explicit representation of important processes (evaporation, infiltration, and nonlinear routing over a large network) makes it sufficiently complex to test our method. The goal, as such, is not to test the accuracy of the process model, but rather to evaluate if our proposed framework can capture similar input–output dynamics, but with reduced model complexity.

178 junctions are selected in the large urban watershed model. Two five-day simulations of the full software model are run using synthetic rainfall time series and the resulting water levels are recorded at a one-minute timestep. We train our approach on one set of rainfall runoff data for each junction, and then evaluate using the other set.

4.2. Reduction of a process-based hydraulic and hydrologic model

The left side of Figs. 3 and 4 shows the cumulative density function of evaluation Nash Sutcliffe Efficiency computed in the analysis. Each line indicates the percentile of a given NSE score, such that better performance is positioned down and to the right while worse performance is up and to the left. For visual interpretation, the right side of the figure shows evaluation simulations for the "final" models at the 25th percentile, median, and maximum of evaluation NSE.

As measured by NSE and visual inspection, our approach was able to replicate the dynamics exhibited by the more complex physical model (Fig. 3). When trained on just one storm, the algorithm accurately predicted the junction depth in a subsequent storm. Compared to existing machine-learning based methods – which require large amounts of data to implicitly learn the underlying dynamics – this relatively low data requirement is a major benefit of the approach.

In general, as model complexity increases (higher order polynomials and more input transformations), model performance increases as well. However, there are tradeoffs in robustness and accuracy when choosing amongst model configurations. The simplest models (those with smaller p and t values, red, Fig. 3) are least likely to score NSEs below zero, but also have the lowest ceiling of performance. The models with third order polynomials (blue) achieve the best performance, but score below zero more often during evaluation. A frequent mode of failure is shown in the 25th percentile NSE simulation (Fig. 3, bottom right) where a spurious fixed depth is learned.

These differences in performance are straightforwardly tractable to the structure of the model. As recession limbs are seldom completely linear, the second and third order polynomials are better able to capture their shape. This may include learning a nonzero fixed depth as in the 25th percentile simulation in Fig. 4. In contrast, the linear models can only represent recession limbs as exponential decays to zero. However, this limitation does improve the consistency of the linear models, as they are not able to learn spurious fixed depths as in the 25th percentile simulation of Fig. 3. This tradeoff is visible where "p1t1" in solid red crosses "p3t2" in dotted blue at an NSE of about 0.35. Selecting the model with the best training NSE as the final model (black line Fig. 3) seems to be an effective heuristic for taking advantage of the enhanced accuracy of the nonlinear models when fit is good while defaulting to simpler approximations when fit is worse.

While beyond the core focus of this paper, computational efficiency and surrogate modeling are a secondary benefit of the approach. The process-based model took two hours each to run the training and testing storm events. In contrast, our models took about fifteen seconds to predict the response to the testing storm after being trained. Training all six model configurations took a median of one hour per junction. While training is expensive, it only needs to be done once. As such, our method may serve as a valuable surrogate modeling tool to either complement complex process models in high-performance-computing applications, or as a forecasting tool in real-time applications that require rapid predictions. Results here are reminiscent of Young (2006), where dominant mode analysis recovers 99.99% of the behavior of a

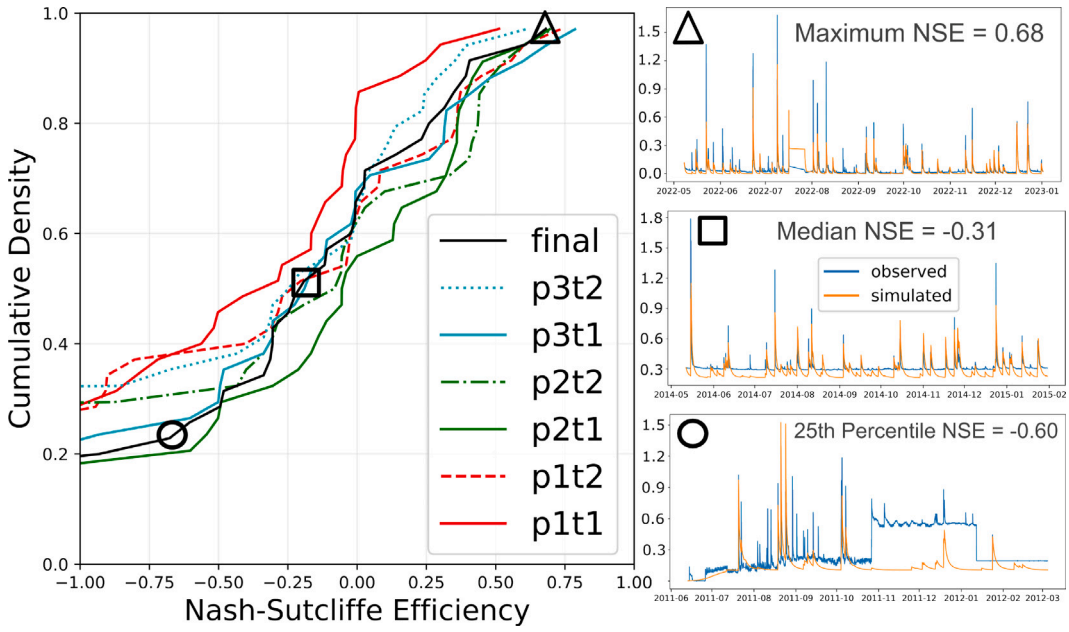


Fig. 4. Predicting stream stage in real catchments. Models are evaluated at 35 gaging stations over nine months. The cumulative probability distribution of Nash Sutcliffe Efficiency is shown on the left. In the legend, "p" prefixes the polynomial order while "t" prefixes the number of subcatchments. On the right are evaluation simulations of stage in meters for models at the 25th percentile, median, and maximum of NSE (−0.60, −0.31, and 0.68 respectively).

more complex model with significantly fewer parameters. This result reflects that in some highly parameterized process-based models "only a few of the model parameters have a statistically significant effect on the model output" (Young, 2006).

5. Streamgaging data

5.1. Experiment setup

We apply our approach on USGS precipitation and stream stage observations from 35 gaging stations at a 15-min timestep. These gaging stations are distributed across the southern continental United States with their locations and model performance summarized in SI Figure S3. Before applying the method to the more complex benchmarking data set, this set of gaging locations was selected across the southern US to: (1) study the performance of the method under limited impact of snow melt, the representation of which is beyond the scope of this paper, and (2) evaluate performance on minimally processed measurements that may contain noise or other artifacts. We use stations that also measure precipitation at the same location. Since precipitation at the pour point becomes a worse approximation of catchment-wide precipitation as the contributing area increases, we chose catchments with a contributing area under 100 km^2. As stage-discharge rating curves are often unavailable for catchments this small, the analysis is carried out on the stage (water level).

In pre-processing, any constant offset in stage measurements is removed by subtracting the minimum stage value over the record to improve discovery of the differential equations. Missing stage measurements are linearly interpolated while missing precipitation measurements are filled with zeros. No other pre-processing is performed. Models are evaluated over nine months. The first three months of data are used as windup to initialize the latent states and are excluded from training or testing. The dates and training record length are variable due to data availability. We examine data from January 1, 2005 to January 1, 2023 and use at most 7 years of data total. To ensure the system is causal, forcing data is shifted back one timestep relative to the response.

5.2. Predicting stream stage in catchments

The cumulative density function of Nash–Sutcliffe Efficiency for 35 USGS gaging stations is shown in Fig. 4. Unlike in the prior analysis, which focused on a noise-free and causal physical model, the data analyzed here were collected in the field and are thus subject to noise and various perturbations. Considering that the algorithm identified models entirely from these raw measurements, this demonstrates how complex hydrologic dynamics may be automatically discovered entirely from data without the need to manually develop and calibrate a more complex process model.

The maximum NSE simulation is particularly notable, as it achieves a score of 0.68 on raw data with only 12 total parameters (third order polynomial, one subcatchment) and *no hydrology-specific constraints*. The median and 25th percentile simulations show the importance of correctly defining baseflow. Though we accounted for a constant offset by subtracting the minimum value over the record, subtracting the first or second percentile may more effectively capture the stage to which streams decay, especially when diurnals or noise are present. The median simulation also highlights the incompleteness of numeric scores, as the model clearly has some predictive ability despite an NSE score of less than zero. We attempted to exclude sites with significant snowmelt, but other sources of variation (e.g. dam releases, diversions) are likely present in the data. SI Table S1 summarizes performance across an array of common hydrologic error metrics. Mean absolute and root-mean square error are generally on the order of 10 centimeters while biases are almost always positive.

To determine how much data this method needs to generate accurate models, we analyzed the correlation between Nash–Sutcliffe Efficiency and length of the training record. We examined models trained with between one and six years of data and found no correlation (r = 0%) between model score and training record length beyond one year. This lack of correlation may be due to how few degrees of freedom these models possess. They may become "saturated" after seeing as few as 10 storms. See supplementary information Figures S5 and S7 for more details.

Table 1 summarizes the correlations between linear models with one effective subcatchment (6 parameters) and watershed characteristics. These models have the form: $\frac{dq_o}{dt} = a_1 q_o + b_0 p_o + b_1 T_1(p_o)$. Hydrologically

Table 1
Correlation of model parameters with USGS metadata (with significance).

(r, p)	T_p	T_{50}	b_0	a_0	NSE
Area[a]	(0.54, 0.11)	(0.47, 0.17)	(−0.10, 0.79)	(0.24, 0.5)	**(−0.6, 0.07)**
Altitude	**(0.52, 0.01)**	**(0.40, 0.06)**	(0.07, 0.37)	(0.10, 0.54)	(0.07, 0.69)
Latitude	(0.21, 0.19)	(0.07, 0.61)	(−0.07, 0.12)	(0.08, 0.71)	**(0.33, 0.08)**
Longitude	(−0.23, 0.36)	(−0.13, 0.75)	(−0.19, 0.6)	(−0.14, 0.3)	(0.09, 0.75)

T_p is the time to peak contribution of the unit hydrograph $T_1(p_o)$.
T_{50} is the center of mass of the unit hydrograph $T_1(p_o)$.
b_0 is the coefficient multiplying instantaneous precipitation in the differential equation.
a_0 is the coefficient multiplying stage in the differential equation.
Bold entries are significant at $p = 0.10$.
[a] USGS lists "Drainage Area" and "Contributing Drainage Area" as separate metadata entries. This analysis uses "Contributing Drainage Area".

relevant characteristics available though the USGS REST API were latitude, longitude, altitude, and drainage area. Tables 1 and 2 use the simplest models so that all model parameters can be displayed in one table.

As expected, there is a positive correlation between contributing area and the delay (T_p and T_{50}) in rainfall-runoff response. A negative correlation with the coefficient multiplying instantaneous precipitation (b_0) indicates that streams increasingly tend to take more than one timestep (15 min) to respond as the contributing area increases. The positive correlation with the stage coefficient (a_0) implies that larger catchments have an autocorrelation on stage that is less negative. That is, recession occurs more slowly in larger catchments. NSE scores are worse in larger catchments as precipitation at the pour point becomes an increasingly bad approximation of precipitation over the entire catchment. As with any hydrologic modeling approach, applications at larger scales will benefit from spatially distributed rainfall data.

The positive correlation between altitude and time to peak (T_p) is against expectations. This may be resolved by the positive correlation between the coefficient multiplying instantaneous precipitation (b_0) and altitude. In headwater catchments there may be a more instantaneous response to rainfall, so b_0 is larger and the single subcatchment unit hydrograph may be pushed out to represent more delayed contributions. Given the wide variability in climate and topography across the southern half of the continental United States the meaning of the correlations with latitude and longitude are less clear. They are included here for completeness. Due to the relatively small sample size (n = 35), many correlations do not achieve statistical significance. Further studies with larger sample sizes would be necessary to rigorously support these correlations.

6. Benchmarking against process-based and machine-learning approaches

6.1. Experiment setup and benchmark description

We benchmark the performance of the proposed method on daily data from 348 sites within the National Center for Atmospheric Research (NCAR) CAMELS dataset spanning the continental United States. Locations and model performance are summarized in SI Figure S8. To that end, we use four well established process-based models as well as a variation of Long Short-Term Memory Networks (Entity Aware LSTM) (Kratzert et al., 2019). Entity Aware LSTM is chosen for comparison as it is an established machine learning method with state of the art performance. To train and evaluate, we follow the protocol detailed in Kratzert et al. (2018) by using 269 days of windup, training from October 1, 1999 to September 30, 2008 and evaluating from October 1, 1989 to September 30, 1999 (Kratzert et al., 2018). We build single input, single output models using surface water input (RAIM, comprises liquid precipitation and snowmelt) as forcing and observed runoff (OBS RUN) as the output (Newman et al., 2015). To ensure the system is causal, forcing data is shifted back one timestep relative to the response.

The process-based benchmark models are the Sacramento Soil Moisture Accounting Model (SAC-SMA), Variable Infiltration Capacity (VIC), mesoscale Hydrological Model (mHM), and Hydrologiska Byråns Vattenbalansavdelning (HBV). SAC-SMA has long been a key component of the US National Weather Service's River Forecast System and its source code is publicly accessible (Bowman et al., 2017). Variable Infiltration Capacity was developed in the early 1990s and has been used across scales to analyze trends, forecast, and assess impacts from climate change. Current notable uses include the University of Washington's drought monitoring program and forecasting systems (Hamman et al., 2018). The mHM uses multiscale parameter regionalization to address overparameterization, inadequate integration of spatial heterogeneity, and nontransferability of parameters across space and time (Mizukami et al., 2019; Samaniego et al., 2010). The HBV has been developed at the Swedish Meteorological and Hydrological Institute since the 1970s (Seibert and Vis, 2012) and has been applied for many uses including flash flood prediction (Grillakis et al., 2010).

6.2. Benchmarking using the CAMELS dataset

Fig. 5 benchmarks our modpods approach across 348 CAMELS sites. Four process-based benchmarks, the EA-LSTM, and modpods are plotted by their total number of parameters and requisite inputs. Maximum NSE when predicting ten years of daily discharge is shown as a measure of the upper bound of model fidelity. As illustrated in the figure, our method has a relatively small parameter and input space while achieving a maximum accuracy better than Variable Infiltration Capacity (VIC) and similar to the rest of the process-based models. This performance is achieved without any process-based constraints aside from assuming the system is causal and produces finite responses to finite forcings.

The most accurate result in Fig. 5 is the Entity Aware Long Short-Term Memory Network (Kratzert et al., 2019, 2018). Long Short-Term Memory networks are commonly applied to time series forecasting problems as they do not suffer from the *disappearing* or *exploding* gradients common in other neural networks (Jiang et al., 2022). In the implementation depicted in Fig. 5, various random seeds and ensembling are used to counteract overfitting. Each EA-LSTM model has 1617 randomly initialized learnable parameters without obvious analogs to physical processes. An ensemble of 8 members is used for the prediction for a total of 12,936 parameters.

To contrast the parameter space of the approaches, we consider the Sacramento Soil Moisture Accounting (SAC-SMA) Model coupled with the Snow-17 routine to represent the process-based models (Burnash, 1973; Newman et al., 2015). Snow-17 uses an air temperature index to calculate snow accumulation and ablation. SAC-SMA requires inputs of potential evapotranspiration and water surface input. Stream-flow routing is by a two-parameter instantaneous unit hydrograph model. When Snow-17 is coupled with SAC-SMA, 35 parameters are available for calibration, 20 of which are calibrated in the benchmark results plotted above. All of these parameters directly represent physical processes, some of which are measurable.

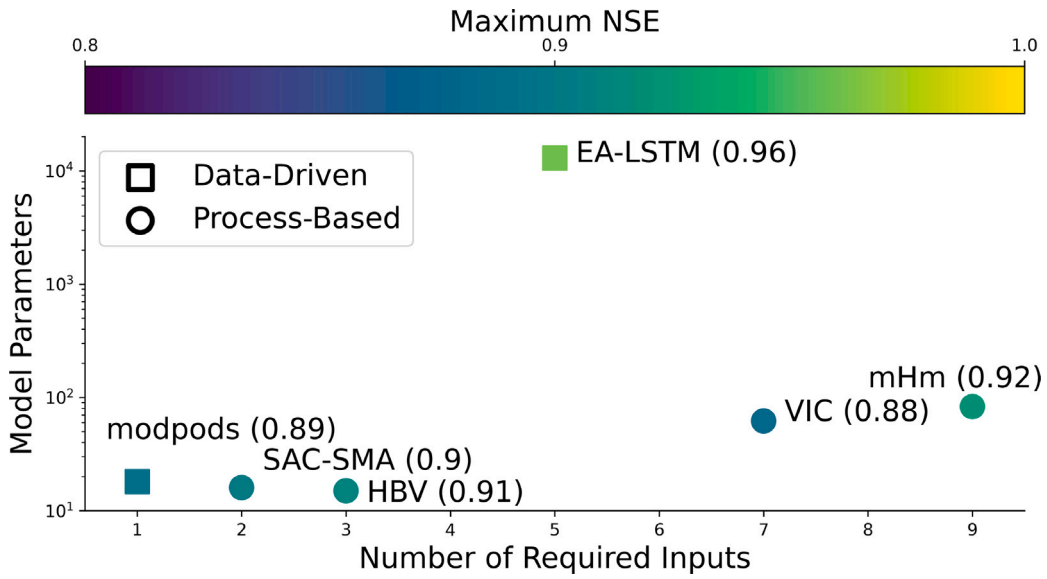


Fig. 5. Benchmarking on the CAMELS dataset. Maximum evaluation NSE across a number of data-driven and process-based approaches when predicting ten years of discharge measurements.

Our *modpods* models have between 6 and 18 parameters. These parameters either describe the shape of a unit hydrograph or are coefficients in a differential equation. The number of parameters is similar to SAC-SMA, but 700x fewer than EA-LSTM. Further, modpods models do not benefit from the constrained structure of explicitly process-based models or the ingestion of forcing variables besides surface water input, such as air temperature and potential evapotranspiration. While modpods uses only surface water input to predict discharge, all other models require at least potential evapotranspiration. Note the logarithmic scale on the y axis. While the EA-LSTM approach has the best NSE, it also has orders of magnitude more parameters than other approaches. While Fig. 5 illustrates common trade offs between model complexity and performance, the complexity of modpods situates it closer to the interpretable process-based models without sacrificing the automaticity offered by data-driven approaches. Note also that though the input transformations are referred to as "unit hydrographs" in this hydrologic application of the tool, these input transformations are potentially generalizable to other forms of causality in other domains. The data-driven approach and dynamical systems representation contrast with more complex process-based models in similar ways to the *data-based mechanistic* approach developed by Young and colleagues (Young, 2012).

Further details on comparative performance are provided in SI Table S2. Several metrics suggest this method has different systematic errors than other hydrologic models. While the benchmark models tend to have negative flow biases, *modpods* biases are almost always positive. As measured by α-NSE, *modpods* replicates the variability of the observed timeseries more closely than any of the benchmarks, but has bias (β-NSE) two orders of magnitude larger than the benchmark models. This may indicate that a relatively large portion of the error in the *modpods* models is due to a constant offset or baseflow, as in the Median NSE simulation of Fig. 4. This would align with results in Dantzer and Kerkez (2024) where definition of the percentile of flow which constitutes as baseflow is an important hyperparameter, ranging from 2% for natural streams to 10% for combined sewer systems which have dry weather flows. Models in this study take the minimum in the timeseries to approximate baseflow, which appears to be unrealistically low.

The *modpods* models appear well saturated after training on a decade of data, with SI Figure S10 showing the difference between training and evaluation NSE. In this study, some models achieved better NSE during the testing period than the training period. This potentially small data requirement contrasts with Kratzert et al. (2018) where 15 years of daily data is proposed as a lower bound of data requirements for the EA-LSTM. As such, while our approach is data-driven, it could be deployed rapidly after observing relatively few storms (SI Figure S7). Additional results from this experiment are detailed in SI Figures S8 through S12.

Table 2
Correlation of model parameters with catchment characteristics..

	T_p	T_{50}	b_0	a_0
Runoff ratio[a]	−0.09	−0.07	0.55	−0.01
FDC slope[b]	−0.14	−0.17	0.24	−0.22
High flow duration	0.19	0.27	−0.17	0.26
Fraction snow	0.26	0.35	0.00	0.24
Silt fraction	−0.07	−0.10	0.05	−0.20
Mean precipitation (P)	−0.16	−0.19	0.56	−0.10
Mean potential evapotranspiration (PET)	0.14	0.19	−0.22	0.14
Aridity[c]	0.25	0.30	−0.39	0.24

T_p is the time to peak contribution of the unit hydrograph.
T_{50} is the center of mass of the unit hydrograph.
b_0 is the coefficient multiplying instantaneous water surface input in the differential equation.
a_0 is the coefficient multiplying discharge in the differential equation.
[a] Runoff ratio is mean daily discharge divided by mean daily precipitation.
[b] FDC Slope is the slope of the flow duration curve between the log-transformed 33rd and 66th percentiles.
[c] The aridity index is mean PET/P with PET estimated by the Priestley–Taylor formulation calibrated for each catchment.

6.3. Correlations to catchment characteristics

To illustrate how model parameters relate to climate, topography, and common hydrologic signatures, Table 2 shows correlations between model parameters and catchment characteristics in the CAMELS dataset. As in Table 1, we consider a linear model with one sub-catchment that has a total of six parameters and report the Pearson correlation coefficient. We do not reproduce the full correlation table as CAMELS includes over 50 metadata variables, but that information is accessible via the repository.

The first two rows of Table 2 show logical correlations between model parameters and hydrologic signatures that could be considered measures of "flashiness" in flows. Catchments with high runoff ratios and large flow duration curve (FDC) slopes have quick rises and falls

in discharge. This is reflected by a quicker unit hydrograph (T_p and T_{50} smaller), a larger immediate contribution from water surface input (b_0 larger) and a quicker recession after storm events (a_0 more negative).

The next two rows are features associated with a slower, more diffusive runoff response. High flow duration measures the average duration of high flow events, which are periods with more than 9 times the median daily flow. Snowmelt is often (but not always) a slower process than rainfall. These features correlate with: a slower unit hydrograph (T_p and T_{50} larger), smaller immediate impact of water surface input (b_0 smaller), and a slower recession trend (a_0 less negative).

Concerning soils, the only clear trend is a flashier response in silty catchments. The last three rows correlating parameters with climatic conditions may be indicative of the effect of soil moisture deficit on runoff generation. Some of the stronger correlations in this table and the small number of parameters suggest that modpods models could be estimated from catchment characteristics and then manually tuned by experts. Alternatively, forecasted changes in climate could be translated to model parameters in scenario analysis. Some other expected correlations such as land use and contributing area were absent or inconsistent with hydrologic principles. These expected correlations may have been obscured due to the coarse timestep (daily), catchment-wide averaged forcing, and shifting the data to ensure causality.

7. Interpretation, limitations, and future directions

7.1. General interpretation of resulting models

Once a *modpods* model is discovered automatically from data, the resulting parameters and model structure can be used to infer properties of the catchment. The previous correlation analysis of this paper indicates that model parameters are related to physiographic features, which can be used to infer how changes in catchment composition may influence rainfall-runoff dynamics of the resulting *modpods* model. In process-based models, parameters can be changed to carry out scenario analyses or watershed planning. While beyond the scope of this paper, since resulting modpods parameters are correlated to physiographic features, future correlation analysis could be used to explore how catchment outflows may be affected by a changing landscape or to make predictions in ungaged basins.

Once a final model is selected, the number of effective subcatchments can be understood as a measure of the number or complexity of constituent processes producing runoff within a catchment. SI Figures S1, S4, and S9 show a roughly equal split between models with one and two effective subcatchments. Some models only need instantaneous precipitation and one effective subcatchment to adequately capture runoff generation processes. This may reflect catchments with homogeneous landuse, slope, and soils. On the other hand, catchments with strong diversity in slope and landuse may require more transformations. For example, a catchment with areas of steep, impervious surfaces as well as swamps would likely see distinct contributions from these areas. However, the granularity of this interpretation is limited by equifinality in the input–output behavior of watersheds. Especially when the dynamics are described in so few parameters, it may be difficult to discern how an increase in average catchment slope affects runofff response differently than an increase in imperviousness. This difficulty in uniquely identifying catchments from their precipitation response has long been a prominent point of discussion in the field (Beven, 2000; Sorooshian and Gupta, 1983).

There are also differences in performance amongst model configurations. Increasing the polynomial order allows additional degrees of freedom in the shape of the recession curve. SI Figures S1, S4, and S9 show that the overwhelming majority of sites achieve the best performance using a third order polynomial. This reflects that a linear exponential decay is not an accurate fit of the recession trend in most catchments, as receding limbs generally have some nonlinearity.

7.2. Limitations

Limitations of this study include the precipitation data used, the methods for defining baseflow, and the exclusion of important forcing variables. Acquiring suitable precipitation data is a challenge of implementing this approach, as it is for most approaches. The USGS gaging station experiment used point precipitation measured at the station, which is a poor approximation for the rain that falls on the entire catchment. This is especially true for large catchments as the pour point grows farther from the centroid with increasing contributing area. Correctly defining the equilibrium of the system is vital to the performance of any dynamical systems approach. The median simulation from Fig. 4 and results in SI Table S2 show that more work is needed to integrate rigorous and automated methods of estimating baseflow into this modeling approach. Processes by which water leaves a catchment besides discharge (e.g., evapotranspiration, deep groundwater storage) are ignored in this study by excluding forcing variables such as potential evapotranspiration and groundwater levels and not explicitly considering more lasting differences between catchments such as soil permeability. Though evapotranspiration rates and resulting soil moisture content may be implicitly modeled in recession rates, including a more comprehensive description of the water balance within the model may improve accuracy and interpretability.

7.3. Future directions

With its automaticity, accuracy, and correlation to catchment features, potential applications of this approach include low-cost predictive models for sensor networks and, possibly, predictions in ungauged catchments. The first application is detailed in Dantzer and Kerkez (2024) which builds multi-input models forced by rain and snowmelt data automatically sourced using the sensor's location.

The correlations in Tables 1 and 2 between model parameters and watershed characteristics suggest there may be potential for applying this method to ungauged basins by building regressions between watershed characteristics and model parameters. As there are fewer than twenty parameters in even the most complex models, experts could also manually tune these models after regression until they are satisfied that the model's behavior reflects their expectations of the ungauged catchment. Ease in tuning and interpretation also suggest use as an instructional tool.

In Dantzer and Kerkez (2024) the accuracy returns from including potential evapotranspiration were marginal. However, PET is clearly an important driver in the water budget and different model formulations may incorporate this information more effectively and improve results. For example, catchment-averaged soil moisture could be included as an output state to better capture seasonal differences in initial abstractions. As additional forcing variables are added to the model the number of parameters will increase. It will therefore likely be beneficial to incorporate domain knowledge through constraints on the coefficients of the differential equation that reduce the degrees of freedom and lower the chances of overfitting. A more informed approach to data-splitting may also increase the transferability and performance of models developed using the method detailed herein (Guo et al., 2020; Chen et al., 2022).

The algorithm presented is provided with limited formal system theoretic analysis. Decisions such as the sequence in which transformations are added, starting guesses for the transformations, and the restrictions on the differential equation discovery should be explored formally. The characteristics of the optimization such as its convexity are not explored herein and may lead to improvements in computational efficiency. Future studies could also examine application and interpretation of *modpods* in domains outside hydrology.

8. Conclusions

The purpose of this study was to demonstrate a new method combining unit hydrographs with differential equation discovery to build interpretable rainfall-runoff models automatically from precipitation and stage data. Using the approach, the outputs of a large process-based model were approximated accurately. Then, stage or discharge were predicted at nearly 400 stream gauging stations. This approach also provides a novel conceptual model of rainfall-runoff processes, as individual parameters of the identified models can be explored intuitively. Notable advantages of the data-driven approach include a relatively small parameter and input space while maintaining maximum fidelity equal to existing process-based models. The computational efficiency and limited data requirements of this approach suggest utility in providing predictions across large sensor networks.

Open research

[Software] - The United States Environmental Protection Agency's Stormwater Management Model is available for free download (Huber, 1985). We used the pyswmm interface to access results McDonnell et al. (2020). All scripts used in analysis and figure creation are also freely available (Dantzer, 2023a,b). This study does not use any proprietary software and no registration or payment is required.

[Data] - Data for Figs. 1 and 4 and Table 1 are publicly available from the United States Geological Survey (USGS, 2016). Alternatively, the scripts generating these tables and figures automatically fetch the data via REST API. The software model referenced in Fig. 3 was privately shared with us by a municipality. As identifying information is present in the model and stormwater networks are critical infrastructure, we are not able to freely share this information (Rossner et al., 2020). The NCAR CAMELS dataset used for Fig. 5 and Table 2 is also freely available (Addor et al., 2017b; Kratzert et al., 2019; Newman et al., 2015; Kratzert et al., 2018; Addor et al., 2017a).

CRediT authorship contribution statement

Travis Adrian Dantzer: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Branko Kerkez:** Conceptualization, Funding acquisition, Resources, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Travis Adrian Dantzer reports financial support was provided by Michigan Department of Transportation. Branko Kerkez reports financial support was provided by National Science Foundation. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All data and code are open-source and publicly available with the exception of the process-based model.

Acknowledgments

We thank M. Tobias, B.E. Mason, J.Q. Schmidt, and L. Wojciechowski for their comments on the figures. Excellent data availability and accessibility from Kratzert et al. (2019), the United States Geological Survey, and the National Center for Atmospheric Research was immensely helpful in this study. This work was supported by the US National Science Foundation Grant 1750744 and the Michigan Department of Transportation, United States grant OR21-003.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.advwatres.2024.104796.

References

Addor, N., Newman, A.J., Mizukami, N., Clark, M.P., 2017a. The CAMELS data set: catchment attributes and meteorology for large-sample studies [dataset]. Hydrol. Earth Syst. Sci. http://dx.doi.org/10.5194/hess-21-5293-2017.

Addor, N., Newman, A., Mizukami, M., Clark, M.P., 2017b. Catchment attributes for large-sample studies. [dataset]. http://dx.doi.org/10.5065/D6G73C3Q.

Alex, J., Hübner, C., Förster, L., 2020. Planning, testing and commissioning of automation solutions for waste water treatment plants using simulation. IFAC-PapersOnLine 53 (2), 16665–16670. http://dx.doi.org/10.1016/j.ifacol.2020.12.1084, Retrieved from https://www.sciencedirect.com/science/article/pii/S2405896320314592 (21st IFAC World Congress).

Angelov, P.P., Soares, E.A., Jiang, R., Arnold, N.I., Atkinson, P.M., 2021. Explainable artificial intelligence: an analytical review. WIREs Data Min. Knowl. Discov. 11 (5), e1424. http://dx.doi.org/10.1002/widm.1424, arXiv:https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1424. Retrieved from https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1424.

Babovic, V., 2009. Introducing knowledge into learning based on genetic programming. J. Hydroinform. 11 (3–4), 181–193. http://dx.doi.org/10.2166/hydro.2009.041, arXiv:https://iwaponline.com/jh/article-pdf/11/3-4/181/386361/181.pdf.

Babovic, V., Abbott, M.B., 1997a. The evolution of equations from hydraulic data Part I: Theory. J. Hydraul. Res. 35 (3), 397–410. http://dx.doi.org/10.1080/00221689709498420, arXiv:https://doi.org/10.1080/00221689709498420.

Babovic, V., Abbott, M.B., 1997b. The evolution of equations from hydraulic data Part II: Applications. J. Hydraul. Res. 35 (3), 411–430. http://dx.doi.org/10.1080/00221689709498421, arXiv:https://doi.org/10.1080/00221689709498421.

Babovic, V., Keijzer, M., 2002. Rainfall runoff modelling based on genetic programming. Hydrol. Res. 33 (5), 331–346. http://dx.doi.org/10.2166/nh.2002.0012, arXiv:https://iwaponline.com/hr/article-pdf/33/5/331/2707/331.pdf.

Bartos, M., Kerkez, B., 2021. Pipedream: An interactive digital twin model for natural and urban drainage systems. Environ. Model. Softw. 144, 105120.

Bedient, P.C., Huber, W.C., Vieux, B.E., 2008. Hydrology and Floodplain Analysis, 4th ed. Prentice Hall.

Beven, K.J., 2000. Uniqueness of place and process representations in hydrological modelling. Hydrol. Earth Syst. Sci. 4 (2), 203–213. http://dx.doi.org/10.5194/hess-4-203-2000, Retrieved from https://hess.copernicus.org/articles/4/203/2000/.

Bowman, A.L., Franz, K.J., Hogue, T.S., 2017. Case studies of a MODIS-based potential evapotranspiration input to the Sacramento Soil Moisture Accounting Model. J. Hydrometeorol. 18 (1), 151–158.

Brunton, S.L., Proctor, J.L., Kutz, J.N., 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. 113 (15), 3932–3937.

Burnash, R.J., 1973. A Generalized Streamflow Simulation System: Conceptual Modeling for Digital Computers. US Department of Commerce, National Weather Service, and State of California ....

Castelletti, A., Galelli, S., Ratto, M., Soncini-Sessa, R., Young, P.C., 2012. A general framework for dynamic emulation modelling in environmental problems. Environ. Model. Softw. 34, 5–18.

Chadalawada, J., Herath, H.M.V.V., Babovic, V., 2020. Hydrologically informed machine learning for rainfall-runoff modeling: A genetic programming-based toolkit for automatic model induction. Water Resour. Res. 56 (4), e2019WR026933. http://dx.doi.org/10.1029/2019WR026933, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2019WR026933. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019WR026933 (e2019WR026933 10.1029/2019WR026933).

Chen, J., Zheng, F., May, R., Guo, D., Gupta, H., Maier, H.R., 2022. Improved data splitting methods for data-driven hydrological model development based on a large number of catchment samples. J. Hydrol. 613, 128340. http://dx.doi.org/10.1016/j.jhydrol.2022.128340, Retrieved from https://www.sciencedirect.com/science/article/pii/S002216942200912X.

Dantzer, T.A., 2023a. Automatic rainfall runoff [software]. https://zenodo.org/badge/latestdoi/590232346.

Dantzer, T.A., 2023b. Model discovery in partially observable dynamical systems [software]. https://zenodo.org/badge/latestdoi/636838701. Retrieved from https://github.com/dantzert/modpods.

Dantzer, T.A., Kerkez, B., 2024. Automated hydrologic forecasting using open-source sensors: predicting stream depths across 200,000 km^2. Retrieved from https://ssrn.com/abstract=4760938 (Preprint not peer reviewed).

de Silva, B.M., Champion, K., Quade, M., Loiseau, J.-C., Kutz, J.N., Brunton, S.L., 2020. Pysindy: a python package for the sparse identification of nonlinear dynamics from data. arXiv preprint arXiv:2004.08424.

Devia, G.K., Ganasri, B.P., Dwarakish, G.S., 2015. A review on hydrological models. Aquat. Procedia 4, 1001–1007.

Fathian, F., Mehdizadeh, S., Sales, A.K., Safari, M.J.S., 2019. Hybrid models to improve the monthly river flow prediction: Integrating artificial intelligence and non-linear time series models. J. Hydrol. http://dx.doi.org/10.1016/j.jhydrol.2019.06.025.

Francipane, A., Ivanov, V.Y., Noto, L.V., Istanbulluoglu, E., Arnone, E., Bras, R.L., 2012. tRIBS-Erosion: A parsimonious physically-based model for studying catchment hydro-geomorphic response. Catena 92, 216–231.

Ghorbani, M.A., Singh, V.P., Sivakumar, B., H. Kashani, M., Atre, A.A., Asadi, H., 2017. Probability distribution functions for unit hydrographs with optimization using genetic algorithm. Appl. Water Sci. 7, 663–676.

Grillakis, M.G., Tsanis, I.K., Koutroulis, A.G., 2010. Application of the HBV hydrological model in a flash flood case in Slovenia. Nat. Hazards Earth Syst. Sci. 10 (12), 2713–2725. http://dx.doi.org/10.5194/nhess-10-2713-2010, Retrieved from https://nhess.copernicus.org/articles/10/2713/2010/.

Guo, D., Zheng, F., Gupta, H., Maier, H.R., 2020. On the robustness of conceptual rainfall-runoff models to calibration and evaluation data set splits selection: A large sample investigation. Water Resour. Res. 56 (3), e2019WR026752. http://dx.doi.org/10.1029/2019WR026752, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2019WR026752. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019WR026752 (e2019WR026752 2019WR026752).

Haktanir, T., Sezen, N., 1990. Suitability of two-parameter gamma and three-parameter beta distributions as synthetic unit hydrographs in Anatolia. Hydrol. Sci. J. 35 (2), 167–184. http://dx.doi.org/10.1080/02626669009492416.

Hamman, J.J., Nijssen, B., Bohn, T.J., Gergel, D.R., Mao, Y., 2018. The variable infiltration capacity model version 5 (VIC-5): infrastructure improvements for new applications and reproducibility. Geosci. Model Dev. 11 (8), 3481–3496. http://dx.doi.org/10.5194/gmd-11-3481-2018, Retrieved from https://gmd.copernicus.org/articles/11/3481/2018/.

Herath, H.M.V.V., Chadalawada, J., Babovic, V., 2021a. Genetic programming for hydrological applications: to model or to forecast that is the question. J. Hydroinform. 23 (4), 740–763. http://dx.doi.org/10.2166/hydro.2021.179, arXiv:https://iwaponline.com/jh/article-pdf/23/4/740/910365/jh0230740.pdf.

Herath, H.M.V.V., Chadalawada, J., Babovic, V., 2021b. Hydrologically informed machine learning for rainfall–runoff modelling: towards distributed modelling. Hydrol. Earth Syst. Sci. 25 (8), 4373–4401. http://dx.doi.org/10.5194/hess-25-4373-2021, Retrieved from https://hess.copernicus.org/articles/25/4373/2021/.

Hespanha, J.P., 2018. Linear Systems Theory. Princeton University Press.

Ho, L., Kalman, R.E., 1966. Editorial: Effective construction of linear state-variable models from input/output functions. Automatisierungstechnik http://dx.doi.org/10.1524/auto.1966.14.112.545.

Huber, W.C., 1985. Storm water management model (SWMM) [software]. doi:https://www.epa.gov/water-research/storm-water-management-model-swmm.

Jajarmizadeh, M., Harun, S., Salarpour, M., 2012. A review on theoretical consideration and types of models in hydrology. J. Environ. Sci. Technol. 5 (5), 249–261.

Jakeman, A., Hornberger, G., 1993. How much complexity is warranted in a rainfall-runoff model? Water Resour. Res. 29 (8), 2637–2649.

Jiang, S., Zheng, Y., Wang, C., Babovic, V., 2022. Uncovering flooding mechanisms across the contiguous United States through interpretive deep learning on representative catchments. Water Resour. Res. 58 (1), e2021WR030185. http://dx.doi.org/10.1029/2021WR030185, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2021WR030185. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2021WR030185 (e2021WR030185 2021WR030185).

Juang, J.N., Pappa, R.S., 1985. An Eigensystem Realiza"i0n Algorithm (Era) for Modal Parameter Identification and Model Reduction. Tech. Rep., NASA Langley Research Center.

Kapoor, A., Pathiraja, S., Marshall, L., Chandra, R., 2023. DeepGR4J: A deep learning hybridization approach for conceptual rainfall-runoff modelling. Environ. Model. Softw. http://dx.doi.org/10.1016/j.envsoft.2023.105831.

Kirchner, J.W., 2009. Catchments as simple dynamical systems: Catchment characterization, rainfall-runoff modeling, and doing hydrology backward. Water Resour. Res. 45 (2).

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., Herrnegger, M., 2018. Rainfall–runoff modelling using long short-term memory (LSTM) networks [dataset]. Hydrol. Earth Syst. Sci. http://dx.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 [dataset]. Hydrol. Earth Syst. Sci. http://dx.doi.org/10.5194/hess-23-5089-2019.

Kumar, R., Livneh, B., Samaniego, L., 2013. Toward computationally efficient large-scale hydrologic predictions with a multiscale regionalization scheme. Water Resour. Res. 49 (9), 5700–5714.

Lee, J.W., Chegal, S.D., Lee, S.O., 2020. A review of tank model and its applicability to various Korean catchment conditions. Water 12 (12), http://dx.doi.org/10.3390/w12123588, Retrieved from https://www.mdpi.com/2073-4441/12/12/3588.

Maier, H.R., Zheng, F., Gupta, H., Chen, J., Mai, J., Savic, D., Loritz, R., Wu, W., Guo, D., Bennett, A., Jakeman, A., Razavi, S., Zhao, J., 2023. On how data are partitioned in model development and evaluation: Confronting the elephant in the room to enhance model generalization. Environ. Model. Softw. 167, 105779. http://dx.doi.org/10.1016/j.envsoft.2023.105779, Retrieved from https://www.sciencedirect.com/science/article/pii/S1364815223001652.

Mauroy, A., Mezic, I., Susuki, Y., 2020. Koopman Operator in Systems and Control. Springer, doi:https://link.springer.com/content/pdf/10.1007/978-3-030-35713-9.pdf.

McDonnell, B.E., Ratliff, K., Tryby, M.E., Wu, J.J.X., Mullapudi, A., 2020. PySWMM: The python interface to stormwater management model (SWMM) [software]. J. Open Sour. Softw. http://dx.doi.org/10.21105/joss.02292.

Mizukami, N., Rakovec, O., Newman, A.J., Clark, M.P., Wood, A.W., Gupta, H.V., Kumar, R., 2019. On the choice of calibration metrics for "high-flow" estimation using hydrologic models. Hydrol. Earth Syst. Sci. 23 (6), 2601–2614. http://dx.doi.org/10.5194/hess-23-2601-2019, Retrieved from https://hess.copernicus.org/articles/23/2601/2019/.

Nadarajah, S., 2007. Probability models for unit hydrograph derivation. J. Hydrol. 344 (3–4), 185–189. http://dx.doi.org/10.1016/j.jhydrol.2007.07.004.

Nash, J., 1959. Systematic determination of unit hydrograph parameters. J. Geophys. Res. 64 (1), 111–115.

Newman, A.J., Clark, M.P., K. Sampson, A.W., L. E. Hay, A.B., Viger, R.J., Blodgett, D., Brekke, L., Arnold, J.R., Hopson, T., Duan, Q., 2015. Development of a large-sample watershed-scale hydrometeorological data set for the contiguous USA: data set characteristics and assessment of regional variability in hydrologic model performance [dataset]. Hydrol. Earth Syst. Sci. http://dx.doi.org/10.5194/hess-19-209-2015.

Newman, A.J., Mizukami, N., Clark, M.P., Wood, A.W., Nijssen, B., Nearing, G., 2017. Benchmarking of a physically based hydrologic model. J. Hydrometeorol. 18 (8), 2215–2225. http://dx.doi.org/10.1175/JHM-D-16-0284.1, Retrieved from https://journals.ametsoc.org/view/journals/hydr/18/8/jhm-d-16-0284_1.xml.

Olson, R.S., Moore, J.H., 2016. TPOT: A tree-based pipeline optimization tool for automating machine learning. Proc. Workshop Autom. Mach. Learn. 64, 66–74.

Rossner, G., Ress, E., Morley, K., 2020. Protecting the Water Sector's Critical Infrastructure Information. Tech. Rep., American Water Works Association, doi:https://www.awwa.org/Portals/0/AWWA/Government/ProtectingtheWaterSectorsCriticalInfrastructureInformation.pdf.

Samaniego, L., Kumar, R., Attinger, S., 2010. Multiscale parameter regionalization of a grid-based hydrologic model at the mesoscale. Water Resour. Res. 46 (5), http://dx.doi.org/10.1029/2008WR007327, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2008WR007327. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2008WR007327.

Santos, L., Thirel, G., Perrin, C., 2018. Continuous state-space representation of a bucket-type rainfall-runoff model: a case study with the GR4 model using state-space GR4 (version 1.0). Geosci. Model Dev. 11 (4), 1591–1605.

Sarafanov, M., Borisova, Y., Maslyaev, M., Revin, I., Maximov, G., Nikitin, N.O., 2021. Short-term river flood forecasting using composite models and automated machine learning: The case study of Lena River. Water 13 (24), 3482. http://dx.doi.org/10.3390/w13243482.

Schmid, P.J., 2022. Dynamic mode decomposition and its variants. Annu. Rev. Fluid Mech. http://dx.doi.org/10.1146/annurev-fluid-030121-015835.

Schneider, M.Y., Quaghebeur, W., Borzooei, S., Froemelt, A., Li, F., Saagi, R., Wade, M.J., Zhu, J.-J., Torfs, E., 2022. Hybrid modelling of water resource recovery facilities: status and opportunities. Water Sci. Technol. 85 (9), 2503–2524. http://dx.doi.org/10.2166/wst.2022.115, arXiv:https://iwaponline.com/wst/article-pdf/85/9/2503/1064960/wst085092503.pdf.

Seibert, J., Vis, M.J.P., 2012. Teaching hydrological modeling with a user-friendly catchment-runoff-model software package. Hydrol. Earth Syst. Sci. 16 (9), 3315–3325. http://dx.doi.org/10.5194/hess-16-3315-2012, Retrieved from https://hess.copernicus.org/articles/16/3315/2012/.

Silberstein, R., 2006. Hydrological models are so good, do we still need data? Environ. Model. Softw. 21 (9), 1340–1352.

Song, J.-H., Her, Y., Park, J., Kang, M.-S., 2019. Exploring parsimonious daily rainfall-runoff model structure using the hyperbolic tangent function and tank model. J. Hydrol. 574, 574–587.

Sorooshian, S., Gupta, V.K., 1983. Automatic calibration of conceptual rainfall-runoff models: The question of parameter observability and uniqueness. Water Resour. Res. 19 (1), 260–268. http://dx.doi.org/10.1029/WR019i001p00260, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/WR019i001p00260. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/WR019i001p00260.

Sorooshian, S., Hsu, K.-l., Coppola, E., Tomassetti, B., Verdecchia, M., Visconti, G., 2008. Hydrological Modelling and the Water Cycle: Coupling the Atmospheric and Hydrological Models. Vol. 63, Springer Science & Business Media.

Tian, W., Liao, Z., Zhang, Z., Wu, H., Xin, K., 2022. Flooding and overflow mitigation using deep reinforcement learning based on koopman operator of urban drainage systems. Water Resour. Res. 58 (7), e2021WR030939.

Troutman, S.C., Schambach, N., Love, N.G., Kerkez, B., 2017. An automated toolchain for the data-driven and dynamical modeling of combined sewer systems. Water Res. 126, 88–100.

USGS, 2016. National water information system [dataset]. doi:thttps://maps.waterdata.usgs.gov/mapper/index.html.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R.,

Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors, 2020. SciPy 1.0: Fundamental algorithms for scientific computing in python. Nat. Methods 17, 261–272. http://dx.doi.org/10.1038/s41592-019-0686-2.

Wagena, M.B., Goering, D., Collick, A.S., Bock, E., Fuka, D.R., Buda, A., Easton, Z.M., 2020. Comparison of short-term streamflow forecasting using stochastic time series, neural networks, process-based, and Bayesian models. Environ. Model. Softw. 126, 104669.

Welch, G., Bishop, G., et al., 1995. An introduction to the Kalman filter.

Wong, B.P., Kerkez, B., 2018. Real-time control of urban headwater catchments through linear feedback: Performance, analysis, and site selection. Water Resour. Res. 54 (10), 7309–7330.

Young, P.C., 2006. The data-based mechanistic approach to the modelling, forecasting and control of environmental systems. Annu. Rev. Control 30 (2), 169–182. http://dx.doi.org/10.1016/j.arcontrol.2006.05.002, Retrieved from https://www.sciencedirect.com/science/article/pii/S1367578806000496.

Young, P.C., 2012. Data-based mechanistic modelling: Natural philosophy revisited? In: Wang, L., Garnier, H. (Eds.), System Identification, Environmental Modelling, and Control System Design. Springer London, London, pp. 321–340. http://dx.doi.org/10.1007/978-0-85729-974-1_16.

Zheng, F., Chen, J., Maier, H.R., Gupta, H., 2022. Achieving robust and transferable performance for conservation-based models of dynamical physical systems. Water Resour. Res. 58 (5), e2021WR031818. http://dx.doi.org/10.1029/2021WR031818, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2021WR031818. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2021WR031818 (e2021WR031818 2021WR031818).