Generating interpretable rainfall-runoff models automatically from data

Travis Adrian Dantzer¹ and Branko Kerkez¹

¹University of Michigan-Ann Arbor

February 13, 2023

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 limits the discovery of insights and may impact trust. To that end, 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, which 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 solely from measurements. We apply this approach to fourteen 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 computational complexity. We discuss how the resulting parsimonious representation of watershed dynamics provides theoretical insight and computational efficiency to enable automated predictions across large areas.
Generating interpretable rainfall-runoff models automatically from data

Travis Adrian Dantzer\(^1\), Branko Kerkez\(^1\)

\(^1\)Civil and Environmental Engineering, University of Michigan
\(^1\)2350 Hayward St, Ann Arbor, Michigan, USA

Key Points:
- Existing process-based models are not designed to ingest large amounts of real-time data.
- Machine learning approaches can automatically ingest data to make predictions but lack interpretability.
- A new open-source method automatically creates interpretable models, as validated with data from many real catchments.

Corresponding author: Travis Adrian Dantzer, dantzert@umich.edu
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 limits the discovery of insights and may impact trust. To that end, 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, which 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 solely from measurements. We apply this approach to fourteen 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 computational complexity. We discuss how the resulting parsimonious representation of watershed dynamics provides theoretical insight and computational efficiency to enable automated predictions across large areas.

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 the calibration process 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 (Gayathri K. Devia, 2015). Many process-based models were created in the context of data scarcity and are therefore not readily amenable to data ingestion or data assimilation (A. Castelletti, 2012; Antonio Francipane, 2012; Singh, 2009; Silberstein, 2006). Machine learning approaches can automate data ingestion, (Mikhail Sarafanov, 2021; Randal S. Olson, 2016) but their opacity may limit insight and stakeholder trust (Jajarmizadeh, 2012). Both approaches require large amounts of computational power and data (R. Kumar, 2013; Moges B. Wagena, 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 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 latent state estimations within hydrologic systems using only rainfall and runoff measurements. The novelty of the approach lies in its ability to discover interpretable and fundamental equations governing hydrologic systems solely from data, while preserving the automaticity of data-driven methods.
2 Background

Prior efforts have pursued a combination of automaticity and interpretability through theoretical derivation, modification of existing process-based models, and explainable AI (XAI). The first two provide interpretability, but are somewhat manual. While XAI has mathematical interpretability, the parameters are not straightforwardly related to measurements or hydrologic concepts. (Angelov, 2021; Wenchong Tian, 2022)

A tractable system of differential equations representing a catchment can be constructed by derivation from first principles with appropriate simplifying assumptions. Interpretability is high as these models are rooted in physics and established domain principles. This approach is also computationally efficient, particularly if the derived algebraic relations have a closed-form solution (A. J. Jakeman, 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 focus on daily data, (Jung-Hun Song, 2019) which may not be granular enough to support the use cases of some stakeholders. Lastly, appropriate simplifying assumptions may differ by catchment and the data processing is often manual.

Reformulating an existing computational model can speed simulation or enable tools not compatible with the original form (Greg Welch, 1997; Hespanha, 2018; Matthew Bartos, 2021). If the original model was already calibrated, these approaches need little additional data (Sara C. Troutman, 2017) or computational power (Léonard Santos, 2018). Even with these barriers eliminated, a reliance on manual processes presents a bottleneck (Brandon P. Wong, 2018). Interpretability is good but coarse due to the abstractions and simplifications made in the reduction. Lastly, reductions in complexity may be limited.

A principal advantage of XAI is automaticity. Techniques such as dynamic mode decomposition (DMD) and Koopman operator construction offer structures ripe for mathematical interpretation (Wenchong Tian, 2022). However, this mathematical interpretability does not relate tractably to physical characteristics of the system or well understood hydrological concepts.

These three approaches have made progress towards combining automaticity and interpretability, but challenges remain. Two promising approaches for achieving these objectives are unit hydrographs – well established, but limited in their ability to capture complex dynamics – and differential equation discovery. To illustrate these approaches via example, we consider a real storm in a mid-size city in the Midwestern United States in the Summer of 2022 (Figure 1). The hydrograph is generated across roughly 30 km² of urbanized landscape and responds to rainfall events with a large initial peak, smaller delayed peak, and nonlinear recession. This illustrates a common but difficult model discovery problem, as a resulting model would need sufficient fidelity to represent superposition of hydrographs, while simultaneously having a simple-enough structure to be identified solely from data.

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 (Philip C. Bedient, 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 Figure 1. A unit hydrograph based on a gamma distribution (Mohammad Ali Ghorbani, 2017) can fit either the initial peak (Figure 1, first row, dot-dash red) or recession (Figure 1, first row, dotted cyan), but not both.

Alternatively, representing catchment response as a system of differential equations would be interpretable and computationally efficient, while still retaining a model struc-
Figure 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 fit 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).
Figure 2. Conceptual model of watershed dynamics approximated through data. (a) Only rainfall (P) and discharge (Q) are measured in the watershed. (b) Reservoirs are inferred to represent drainage processes and produce the indicated hydrographs in response to an impulse of rainfall. A function for the discharge (Q) is discovered from the rainfall transformations 1, 2, and 3.

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 (Brian M. de Silva, 2020; Steven L. Brunton, 2016). However, due to their instantaneous nature, differential equations cannot accommodate delays (Figure 1, second row).

In the following section, we introduce a new method (Figure 1, row 3) combining model discovery with an input transformation to recover the delay and complexity of watersheds in a parsimonious and interpretable model.

3 Methods

When generating runoff models directly from data, the delay between rainfall (P) and runoff (Q) can be understood as an observability gap. Instead of finding the optimal time offset between precipitation and discharge records, we represent delay as intervening unobservable reservoirs that receive rainfall and then contribute to discharge. Approximating these reservoirs provides a system of states that interact with each other instantaneously. In our formulation, the instantaneous interaction of these states is governed by a differential equation, which we seek to discover from data.

Modeling the delay and dispersion between rainfall and runoff as intervening unobservable reservoirs 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 (Sugawara, 1995) has a different topology, but evinces the same notion. The approach presented here is similar, but more flexible and structured with data ingestion in mind. The catchment is conceptualized as a network of parallel reservoirs (Figure 2) that all directly receive rainfall and directly contribute to discharge at the stream gauging station. In our formulation, the only sources of data to estimate the parameters of these intervening unobservable reservoirs are rainfall and streamflow. We achieve this by transforming the rainfall forward to estimate the unobservable reservoirs and then using those transformations to learn the differential equation that governs the discharge.
3.1 Watersheds as Sparsely Observed Dynamical Systems

Conceptualizing the rainfall transformation as a reservoir imposes constraints: (1) The transformation should only go forward in time, not backward. (2) The reservoirs should have only one maximum depth in response to a pulse of rainfall. (3) The levels in these reservoirs should change smoothly. (4) The area under the curve must be finite to reflect conservation of mass. Equation 1 fulfills all these criteria while providing sufficient flexibility to approximate complex responses.

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

Similar formulations have been used in literature to represent hydrographs as probability density functions (Mohammad Ali Ghorbani, 2017). This transformation will be used to estimate the unobservable states in the catchment.

The parameters of Equation (1) have straightforward meanings. Decreasing the \(\beta\) parameter delays and broadens the peak of the input transformation. The \(\alpha\) parameter controls asymmetry about the peak (skewness). The delay parameter \((d)\) shifts the transformation in time without affecting shape. The time to peak contribution (visual maximum) of an input transformation is \(T_p = (\alpha - 1)/\beta + d\). Note that \(\alpha \geq 1\) in this study. The similar expression \(T_{50} = \alpha/\beta + d\) 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 Figures 2 and 3.

Considering a catchment as a multiple input multiple output nonlinear dynamical system (MIMO NDS) gives the form:

\[
d\mathbf{X} = f(\mathbf{X}, \mathbf{U})
\]

where \(\mathbf{X}\) is a vector representing the current state of the catchment and \(\mathbf{U}\) is a vector of inputs that may impact \(d\mathbf{X}/dt\).

We divide \(\mathbf{X}\) into its observable (\(\mathbf{X}_o\)) and unobservable components (\(\mathbf{X}_{uo}\)).

In this study, we consider only observed liquid precipitation as an input, such that \(\mathbf{U} = r_o\).

We assume that the direct impact of \(r_o\) on \(X_o\) is negligible compared to the contribution of the drainage processes represented by the unobservable reservoirs. These reservoirs are denoted as \(X_{uo}\).

This yields:

\[
d\begin{bmatrix} X_o \\ X_{uo} \end{bmatrix} = \begin{bmatrix} f(X_o, X_{uo}) \\ f(X_o, X_{uo}, r_o) \end{bmatrix}
\]

\(X_{uo}\) is the link between \(r_o\) and \(X_o\), but \(X_{uo}\) is unknown. \(X_{uo}\) is approximated by transforming the rain such that \(X_{uo} = T(r_o)\) where \(T(r_o)\) is the forward propagation in time of Equation 1 scaled by the magnitude of the rain time series at each time step. As \(X_{uo}\) is a vector of time series, there may be an arbitrary number of rain transformations.

The fidelity of the approximation is evaluated by how parsimonious and generalizable of a link it creates between \(r_o\) and \(X_o\). As such, \(X_{uo}\) does not directly track to any measurement within the watershed, but is best understood as the unobservable reservoirs depicted in Figure 2. Then:
\( \frac{dX_o}{dt} = f(X_o, T(r_o)) \)

\( X_o \) is a scalar time series in this study, but it may be a vector. \( f \) is constrained as follows: (1) constant bias terms are omitted because \( \frac{dX_o}{dt} \) is not a function of time, (2) interaction terms are omitted because the instantaneous interaction term between rain and stage has no physically meaningful interpretation, and (3) only polynomial terms are included because they should be sufficient to capture the dynamics at the resolution we desire (by Taylor’s Theorem).

This gives the final form:

\( \frac{dX_o}{dt} = P_x(X_o) + P_r(T(r_o)) \)

where \( P_x \) and \( P_r \) are polynomials excluding the zero order term. \( P_x \) describes the shape of the recession curve as it is the autocorrelation on stage or discharge. \( P_r \) describes the contributions of the hypothetical reservoirs in Figure 2. Consult Figure 3 for an example of an equation of this form (left column).

Equation 5 can be expanded as:

\[ \begin{align*}
\frac{d}{dt} \begin{bmatrix} x_{o,1} \\
\vdots \\
x_{o,k} \\
\vdots \\
x_{o,n} 
\end{bmatrix} &= \sum_{i=1}^{q} \begin{bmatrix} p_{x,1,i}x_{o,1} + \sum_{j=1}^{m} p_{r,1,i,j}T(r_o)_j \\
\vdots \\
p_{x,k,i}x_{o,k} + \sum_{j=1}^{m} p_{r,k,i,j}T(r_o)_j \\
\vdots \\
p_{x,n,i}x_{o,n} + \sum_{j=1}^{m} p_{r,n,i,j}T(r_o)_j 
\end{bmatrix} \\
and m \text{ is the number of rainfall transformations, } |X_o| = n, \text{ and } q \text{ is the order of the differential equation. Note that } P_r \text{ is a three dimensional tensor in this case.}
\end{align*} \]

### 3.2 Generating the input transformations from data

Equation (5) is used to predict the derivative of the output. We denote \( x'_i \) as the measured derivative and \( y'_i \) as our estimate.

The Sparse Identification of Nonlinear Dynamics (SINDy) (Brian M. de Silva, 2020) algorithm is used. The algorithm chooses parameters of \( P_x \) and \( P_r \) to maximize the coefficient of determination (R²) between the observed and predicted derivative.

\[ \max_{P_x,P_r} \left[ 1 - \frac{\sum (x'_i - y'_i)^2}{\sum (x'_i - \bar{x'})^2} \right] \]

This optimization is the inner loop. The outer loop is the optimization of the rainfall transformations 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’s loop starts with one input transformation and evaluates increasing numbers of transformations until the maximum number is reached or the last transformation added less than 0.5% to the final R² score. If the returns of the last transformation added are marginal, it is removed and the model with one less transformation is returned as the final model. That is, if the \( j^{th} \) transformation produces marginal returns, the model with \( j \) rainfall transformations is the last to be evaluated and a model
Figure 3. Recovering a System of Parallel Reservoirs. Parallel reservoirs 1 (solid green) and 2 (solid purple) contribute to streamflow (solid cyan). Only precipitation (impulse, not pictured) and streamflow are observed. The unobservable reservoirs are approximated (dotted red and yellow) by transforming the precipitation. A differential equation is discovered (left column) to relate these transformations to streamflow (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 reservoir. Row 2. The algorithm finds an optimal representation of the system using a single reservoir. Row 3. Accuracy is below desired, so the algorithm adds a second reservoir (dotted yellow). Row 4. The levels of the unobservable reservoirs and the differential equation relating the reservoirs to the streamflow are recovered.
Algorithm 1. Optimize Input Transformations

1. for \((m = 1 ; m = m + 1 ; m \leq \text{m}_{\text{max}})\) :

2. \(c'_{i-1} = c_0 ; i = 0\)

3. for \((s = s_0 ; s > 1)\) :

4. Generate \(C(c'_{i-1}, s)\)

5. \(\forall C : Scores = \max_{P_x, P_r} \left[ 1 - \frac{\sum (x'_i - y'_i)^2}{\sum (x'_i - x)^2} \right] \)

6. \(c'_i = C[\arg\max\{Scores\}] ; i = i + 1\)

7. if \((c'_i == c'_{i-1}) : s = a \cdot s\)

8. End if \(\Delta R^2 < \Delta_{\text{min}} \text{ from last } m.\)

where:

- \(m\) is the number of input transformations,
- \(c_0\) is the initial set of input transformations,
- \(c'\) is the best performing set of input transformations for a given iteration,
- \(i\) is the iteration number,
- \(s\) determines how far away the compass search looks,
- \(s_0\) 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,
- \(\Delta_{\text{min}}\) is the minimum accuracy increase from an additional transformation.

with \(j-1\) transformations is returned. If \(m = \text{m}_{\text{max}}\) delivers non-marginal returns on accuracy the model with \(\text{m}_{\text{max}}\) transformations is returned.

(2): An initial guess is chosen for the input transformation. For records with more than one event (Figures 4 and 5) the first transformation is Equation 1 with parameters \((\alpha, \beta, d) = (1, 1, 0)\) while additional transformations are broader peaks centered at timesteps of maximum derivative in the output. For records of only one event (Figures 1 and 3), the first transformation is also based on the output derivative.

(3): Larger \(s\) corresponds to a larger perturbation. So the optimization begins looking far afield and converges to small steps.

(4): Eight input transformations are generated by perturbing the shape of the last iteration’s best performing transformation. The perturbations are scaled by \(s\).

(5): The input transformations are scored by how well SINDy performs using them.

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

(7): If the best performing input transformation is the same as last iteration, that means that none of the perturbations were helpful. \(s\) is reduced by the factor \(a\). If \(s\) is now less than one, we will exit the loop at (3).

A conceptual illustration of the algorithm is shown in Figure 3. In the figure, we imagine a hydrograph that is the outflow from two parallel reservoirs. As the steps of the algorithm iterate over the above approach, the algorithm recovers the dynamics of the two individual reservoirs and adds them together to reconstruct the cumulative signal.

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

\[ (8) : 3m + q(1 + m) \]

where \(m\) is the number of input transformations and \(q\) is the order of the ordinary differential equation. Each input transformation adds three parameters that describe its
shape. Increasing the order of the differential equation adds a term for the stage auto-
correlation and each input transformation.

The input transformation (Equation 1) could also be understood as the probabil-
ity density function of the time of travel for an individual drop of rain. We have presented
a more physical interpretation to support intuitive understanding.

### 3.3 Evaluation & Implementation

To evaluate the algorithm, we first apply it in a simulated setting to determine how
well it can replicate rainfall runoff models that are produced by a more complex, process-
based model. We then apply the algorithm to discover models from data at 14 US Geo-
ological Survey (USGS) stream gauges in the southern United States. Lastly, to inter-
pret how model parameters relate to watershed characteristics, we analyze 30 USGS gauges
using models with a linear differential equation and one input transformation. These are
the simplest models possible with this method, having only five parameters.

First, we apply our approach to data generated by a process-based urban water model
(Bryant E. McDonnell & Mullapudi, 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 with 420 storage nodes, 1200 junc-
tions, and 1800 subcatchments. The calibrated model was shared with us by a munic-
ipality and represents an urban stormwater system for roughly 100,000 people. While
EPA-SWMM may be simpler than some research-grade models, its explicit representa-
tion 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 underlying process model, but rather to evaluate if our proposed
framework can capture similar input-output dynamics, but with reduced model complex-
ity.

Three junctions are selected in the large urban watershed model. Two simulations
of the full software model are run using synthetic rainfall time series and the resulting
water levels are recorded. We train our approach on one set of rainfall runoff data for
each junction, and then evaluate using the other set (Figure 4). $R^2$ is used to measure
how accurately the reduced model predicts derivatives in the training storm. Mean ab-
solute error (MAE) provides a performance metric on the testing storm.

This same process is repeated, but using USGS stream stage observations. Four-
teen USGS stream gauges are chosen for analysis. The locations selected are predom-
inantly across the southern US to reduce the impact of snow melt, which is beyond the
scope of this paper. The selected locations are summarized in the supplementary infor-
mation section of this paper. We use stations that also collect precipitation at the same
location. Since precipitation at the pour point becomes a worse approximation of catchment-
wide precipitation as the contributing area increases, most of these catchments have a
contributing area under 150 km$^2$. For catchments this small, stage-discharge rating curves
are often unavailable. As such, the analysis is carried out on the stage (water level). In
pre-processing, a constant offset in stage measurements is removed by subtracting the
minimum stage value over the record to improve discovery of the underlying differential
equations. Our approach is trained on the sites using a 90%-10% training-testing split,
and evaluated with the same error metrics as the previous analysis. Results are presented
in Figure 5 and Table 1.
The last analysis examines how model parameters relate to watershed characteristics over 30 USGS sites, including the 14 mentioned above. We train the simplest models possible using this method, characterizing rainfall-runoff response using only five parameters. These are the two coefficients of the linear differential equation and the three parameters describing the rainfall transformation. The watershed characteristics examined are the sum of precipitation over the record and contributing area. The sum of precipitation expresses both data availability (number of storms) and the catchment’s aridity. The Pearson correlation coefficients between the model parameters and score and the watershed characteristics are presented in Table 2.

To support replicability, the complete algorithm and all code to generate the figures in this paper are implemented and freely shared as Python notebooks. In the accompanying code, the rainfall is transformed through Equation (1) and then fed into the model identification algorithm via the pySINDy library to learn differential equations relating the transformations to the output. The complete implementation of this toolchain, along with code to generate the figures in this paper, is documented at a public URL 1. All analyses took place on a laptop with 32GB RAM and an Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz 1.50 GHz processor.

4 Results and Discussion

4.1 Reduction of a process-based hydraulic and hydrologic model

As measured by fit metrics, our approach successfully replicated the dynamics and magnitude exhibited by the more complex physical model (figure 4). When trained on just one storm, the algorithm accurately predicted the depth at three locations in a subsequent storm. Compared to existing machine-learning based methods – which require large amounts of data to implicitly learn underlying dynamics – this relatively low data requirement is a major benefit of the approach. Since hydrologic dynamics are implicit in the formulation of our approach, the model output space is constrained and thus only needs to discover a limited number of parameters. In fact, the order of each differential equation selected by the algorithm ranged from one to three and only two or three input transformations were needed (Tables S1-S3). For example, one of the candidate models for Junction A uses a second order differential equation and two input transformations. This model of 12 parameters (see Equation 8) captures 96% of the variation in the training storm. Compared to the complex, process-based model – which has hundreds of junctions and catchments – this drastically reduces model state space and overall complexity needed to describe the data.

While beyond the scope of this paper, computational efficiency and surrogate modeling are a secondary benefit of the approach. The process-based model took two hours to run both storm events (training and testing) in Figure 4. In contrast, our algorithm took about fifteen seconds to predict the response to the testing storm after being trained. Training for Junctions A, B, and C took 1.9, 0.2, and 3.4 hours respectively. 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.

4.2 Predicting stream stage in real catchments

Table 1 summarizes performance statistics for the application of the approach to 14 USGS gauging stations, while three sites are highlighted in Figure 5 for illustrative

---

1 https://github.com/dantzert/automatic-rainfall-runoff-models
purposes. More details on the three sites of Figure 5 and training and testing figures for all 14 USGS sites are included in the supplemental information of this paper. The first two rows of Figure 5 illustrate two sites with good performance, as indicated by $R^2$ and MAE metrics. At these and other locations, the algorithm satisfactorily captured peak stage magnitudes, rates, and dynamics. 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.

To illustrate limitations of this approach, a third site is plotted in the final row of figure 5. This site has fluctuations which could be natural or caused by human intervention. Since these fluctuations are not driven by rainfall, our approach does not capture, nor seek to predict these variations. Nonetheless, the method still captures the recession trend and contributions of rainfall. As such, discretion should be used to determine if predictions demand representation of more complex processes. We excluded sites with significant snowmelt, but other sources of variation (e.g. dam releases, evapotranspiration) could still be present in the data. While beyond the scope of this paper, future work could include those processes more explicitly by including other other input data beyond precipitation (e.g. ambient temperature, solar radiation).

Overall, the method recovered more than half of the variation in the training records, as measured by $R^2$ (Table 1). Some of the variation in the training record was driven by noise in the sensor measurements, so the $R^2$ score likely understates how well the dynamics were captured. On the prediction task, MAE averaging less than twenty centimeters makes performance adequate for many practical scenarios.

4.3 Interpreting generated models

The models output by this approach have two components: a set of input transformations and a differential equation. The number of input transformations can be un-
Figure 5. Predicting Stage at USGS Gauging Stations (USGS, 2016) Based on Colocated Precipitation. Training is shown on the left, while testing is shown on the right.

Table 1. Performance summary for 14 USGS stream gauging stations.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Median</th>
<th>Standard Deviation</th>
<th>Maximum</th>
<th>Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training $R^2$</td>
<td>0.54</td>
<td>0.56</td>
<td>0.20</td>
<td>0.85</td>
<td>0.08</td>
</tr>
<tr>
<td>Testing MAE [m]</td>
<td>0.14</td>
<td>0.11</td>
<td>0.14</td>
<td>0.44</td>
<td>0.02</td>
</tr>
</tbody>
</table>
Table 2. Correlation coefficients of rainfall transformation parameters and model score to watershed aridity and contributing area. Models with linear differential equations and one rainfall transformation (total of five parameters) are trained on 30 USGS stations including the sites in Figure 5. Records are analyzed for October 15, 2022 to January 15, 2023.

<table>
<thead>
<tr>
<th></th>
<th>Total Precipitation</th>
<th>Contributing Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>-0.214</td>
<td>-0.072</td>
</tr>
<tr>
<td>$1/\beta$</td>
<td>0.441</td>
<td>-0.029</td>
</tr>
<tr>
<td>$d$</td>
<td>-0.262</td>
<td>-0.002</td>
</tr>
<tr>
<td>$R^2$ Score</td>
<td>0.592</td>
<td>-0.362</td>
</tr>
<tr>
<td>$T_p$</td>
<td>-0.24</td>
<td>-0.07</td>
</tr>
</tbody>
</table>

understood as the number of effective subcatchments in the contributing area. These effective subcatchments do not correspond to real geographic areas, but instead capture components of the overall rainfall-runoff response. The most intuitive way to examine these effective subcatchments is to visualize the shape of their response to an impulse of precipitation. Metrics such as $T_p$ and $T_{50}$ may also be helpful in understanding speed and shape of response. For example, a model may have two input transformations where the $T_p$ of one is twice that of the other. This can be understood as indicating two effective subcatchments that respond to rainfall at different rates.

The differential equation contains a polynomial on stage and the input transformations. The coefficients on powers of stage describe the shape of the recession curve. The coefficients on powers of the input transformations describe their contributions to streamflow. As rainfall transformations are generally not all in the same order of magnitude, the coefficients of different rainfall transformations cannot be directly compared to infer their relative importance.

To examine how model performance and parameters relate to catchment characteristics, Table 2 correlates model parameters to the total precipitation and contributing area of 30 USGS gauging sites. The overall model performance, as measured by $R^2$ score, was negatively correlated with contributing catchment area. Intuitively, a smaller contributing area makes precipitation at the gauging station a better representation of catchment-wide precipitation, and thus leads to better performance. As such, the performance of the proposed approach degrades as catchment scale goes up, unless more spatially-representative rainfall data are available. The positive correlation with total precipitation likely reflects the quantity of training available. The more storms in the record, the more data are available to train the model. While no "right" number of storms exists, ten or more events seems to be sufficient to train a model. However, consult the second row of Figure 5 for a model trained on only three storms. Number of storms aside, since a major benefit of the approach is its automated implementation, it can re-train periodically and improve as more data become available.

Another strong correlation is between the $\beta$ parameter and total precipitation. A larger $1/\beta$ reflects a rainfall transformation with a broader and delayed peak. Considering total precipitation over the record as a proxy for aridity, it follows that catchments with more rainfall should see a broadened and delayed peak due to groundwater and other catchment storage contributions to streamflow after rainfall events. Whereas losing streams in arid catchments (fewer storms in the data record) may have flashier storm responses more dominated by immediate runoff. Some parameters notably lack expected correlations. We posit that $d$ and $T_p$ would correlate positively with contributing area if spatially distributed rainfall data were used, rather than precipitation at the pour point. This should be tested as part of future efforts.
5 Limitations and Future Directions

With automaticity, accuracy, and interpretability, potential applications of this approach include low-cost predictive models for sensor networks and generalizing to ungauged catchments. As the method only requires precipitation and stage height, data needs are low compared to methods reliant on physical parameters that need to be measured in the field (e.g., cross sections, soil types). The small computational burden implies that a consumer laptop could update models rapidly and generate predictions daily for networks of hundreds of sensors. Since predictive models could then be deployed from precipitation data and stream gauging alone, the proposed approach can be quickly scaled, given the proliferation of low-cost and accessible sensor and wireless technologies.

Acquiring suitable precipitation data is a challenge of implementing this approach. This study used point precipitation measured at the stream gauging 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.

The input-output relationship between precipitation and stage could be improved significantly by automatically delineating the contributing area of each gauging station and sourcing spatially distributed precipitation data from weather stations or radar inferred rainfall. Coarse flow direction grids on the order of 30 arc-seconds delineate catchments with sufficient accuracy for this task. Multiple rainfall records could then be transformed to approximate the unobservable states. In catchments where snowmelt is significant, inferred snowmelt as a function of measurements including wind speed and air temperature could be transformed as just another time series input, similar to how rainfall is in this study.

Though rainfall transformation Equation 1 satisfies the criteria of our assumptions, other input transformations may be more effective. Decisions such as the sequence in which transformations are added, starting guesses for the transformations, and the restrictions on the SINDy algorithm could likely be improved. The characteristics of the optimization such as its convexity are not explored herein and may lead to even further improvements in the method.

From informal analyses, this method appears to be a more general discovery about latent state estimation in partially observable dynamical systems. Future studies will examine application and interpretation of this method in domains outside hydrology.

6 Conclusion

We have presented a method combining input transformations 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 was predicted at USGS 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. The algorithm may one day serve as the foundation of low-cost predictive networks using solely precipitation and stage data.

7 Open Research

Software: The United States Environmental Protection Agency’s Stormwater Management Model is available for free download at https://www.epa.gov/water-research/stormwater-management-model-swmm. All scripts used in analysis and figure creation are available at https://github.com/dantzert/automatic-rainfall-runoff-models. This study does not use any proprietary software and no registration or payment is required.
Data: Data for Figures 1 and 5 and Tables 1 and 2 are publicly available from the United States Geological Survey at https://maps.waterdata.usgs.gov/mapper/index.html. Alternatively, the scripts generating these tables and figures automatically fetch the data via REST API. The software model referenced in Figure 4 is not publicly available for privacy reasons.

Acknowledgments
We thank M. Tobias, B.E. Mason, J.Q. Schmidt, and L. Wojciechowski for their comments on the figures. This work was supported by the US National Science Foundation Grant 1750744 and the Michigan Department of Transportation grant OR21-003.

References


USGS. (2016). *National water information system*.

Generating interpretable rainfall-runoff models automatically from data

Travis Adrian Dantzer\textsuperscript{1}, Branko Kerkez\textsuperscript{1}

\textsuperscript{1}Civil and Environmental Engineering, University of Michigan
\textsuperscript{2}2350 Hayward St, Ann Arbor, Michigan, USA

Key Points:

• Existing process-based models are not designed to ingest large amounts of real-time data.
• Machine learning approaches can automatically ingest data to make predictions but lack interpretability.
• A new open-source method automatically creates interpretable models, as validated with data from many real catchments.

Corresponding author: Travis Adrian Dantzer, dantzert@umich.edu
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 limits the discovery of insights and may impact trust. To that end, 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, which 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 solely from measurements. We apply this approach to fourteen 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 computational complexity. We discuss how the resulting parsimonious representation of watershed dynamics provides theoretical insight and computational efficiency to enable automated predictions across large areas.

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 the calibration process 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 (Gayathri K. Devia, 2015). Many process-based models were created in the context of data scarcity and are therefore not readily amenable to data ingestion or data assimilation (A. Castelletti, 2012; Antonio Francipane, 2012; Singh, 2009; Silberstein, 2006). Machine learning approaches can automate data ingestion, (Mikhail Sarafanov, 2021; Randal S. Olson, 2016) but their opacity may limit insight and stakeholder trust (Jajarmizadeh, 2012). Both approaches require large amounts of computational power and data (R. Kumar, 2013; Moges B. Wagena, 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 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 latent state estimations within hydrologic systems using only rainfall and runoff measurements. The novelty of the approach lies in its ability to discover interpretable and fundamental equations governing hydrologic systems solely from data, while preserving the automaticity of data-driven methods.
2 Background

Prior efforts have pursued a combination of automaticity and interpretability through theoretical derivation, modification of existing process-based models, and explainable AI (XAI). The first two provide interpretability, but are somewhat manual. While XAI has mathematical interpretability, the parameters are not straightforwardly related to measurements or hydrologic concepts. (Angelov, 2021; Wenchong Tian, 2022)

A tractable system of differential equations representing a catchment can be constructed by derivation from first principles with appropriate simplifying assumptions. Interpretability is high as these models are rooted in physics and established domain principles. This approach is also computationally efficient, particularly if the derived algebraic relations have a closed-form solution (A. J. Jakeman, 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 focus on daily data, (Jung-Hun Song, 2019) which may not be granular enough to support the use cases of some stakeholders. Lastly, appropriate simplifying assumptions may differ by catchment and the data processing is often manual.

Reformulating an existing computational model can speed simulation or enable tools not compatible with the original form (Greg Welch, 1997; Hespanha, 2018; Matthew Barros, 2021). If the original model was already calibrated, these approaches need little additional data (Sara C. Troutman, 2017) or computational power (Léonard Santos, 2018). Even with these barriers eliminated, a reliance on manual processes presents a bottleneck (Brandon P. Wong, 2018). Interpretability is good but coarse due to the abstractions and simplifications made in the reduction. Lastly, reductions in complexity may be limited.

A principal advantage of XAI is automaticity. Techniques such as dynamic mode decomposition (DMD) and Koopman operator construction offer structures ripe for mathematical interpretation (Wenchong Tian, 2022). However, this mathematical interpretability does not relate tractably to physical characteristics of the system or well understood hydrological concepts.

These three approaches have made progress towards combining automaticity and interpretability, but challenges remain. Two promising approaches for achieving these objectives are unit hydrographs – well established, but limited in their ability to capture complex dynamics – and differential equation discovery. To illustrate these approaches via example, we consider a real storm in a mid-size city in the Midwestern United States in the Summer of 2022 (Figure 1). The hydrograph is generated across roughly 30 km$^2$ of urbanized landscape and responds to rainfall events with a large initial peak, smaller delayed peak, and nonlinear recession. This illustrates a common but difficult model discovery problem, as a resulting model would need sufficient fidelity to represent superposition of hydrographs, while simultaneously having a simple-enough structure to be identified solely from data.

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 (Philip C. Bedient, 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 Figure 1. A unit hydrograph based on a gamma distribution (Mohammad Ali Ghorbani, 2017) can fit either the initial peak (Figure 1, first row, dot-dash red) or recession (Figure 1, first row, dotted cyan), but not both.

Alternatively, representing catchment response as a system of differential equations would be interpretable and computationally efficient, while still retaining a model struc-
Figure 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 fit 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).
Figure 2. Conceptual model of watershed dynamics approximated through data. (a) Only rainfall (P) and discharge (Q) are measured in the watershed. (b) Reservoirs are inferred to represent drainage processes and produce the indicated hydrographs in response to an impulse of rainfall. A function for the discharge (Q) is discovered from the rainfall transformations 1, 2, and 3.

In the following section, we introduce a new method (Figure 1, row 3) combining model discovery with an input transformation to recover the delay and complexity of watersheds in a parsimonious and interpretable model.

3 Methods

When generating runoff models directly from data, the delay between rainfall (P) and runoff (Q) can be understood as an observability gap. Instead of finding the optimal time offset between precipitation and discharge records, we represent delay as intervening unobservable reservoirs that receive rainfall and then contribute to discharge. Approximating these reservoirs provides a system of states that interact with each other instantaneously. In our formulation, the instantaneous interaction of these states is governed by a differential equation, which we seek to discover from data.

Modeling the delay and dispersion between rainfall and runoff as intervening unobservable reservoirs 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 (Sugawara, 1995) has a different topology, but evinces the same notion. The approach presented here is similar, but more flexible and structured with data ingestion in mind. The catchment is conceptualized as a network of parallel reservoirs (Figure 2) that all directly receive rainfall and directly contribute to discharge at the stream gauging station. In our formulation, the only sources of data to estimate the parameters of these intervening unobservable reservoirs are rainfall and streamflow. We achieve this by transforming the rainfall forward to estimate the unobservable reservoirs and then using those transformations to learn the differential equation that governs the discharge.
3.1 Watersheds as Sparsely Observed Dynamical Systems

Conceptualizing the rainfall transformation as a reservoir imposes constraints: (1) The transformation should only go forward in time, not backward. (2) The reservoirs should have only one maximum depth in response to a pulse of rainfall. (3) The levels in these reservoirs should change smoothly. (4) The area under the curve must be finite to reflect conservation of mass. Equation 1 fulfills all these criteria while providing sufficient flexibility to approximate complex responses.

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

Similar formulations have been used in literature to represent hydrographs as probability density functions (Mohammad Ali Ghorbani, 2017). This transformation will be used to estimate the unobservable states in the catchment.

The parameters of Equation (1) have straightforward meanings. Decreasing the \(\beta\) parameter delays and broadens the peak of the input transformation. The \(\alpha\) parameter controls asymmetry about the peak (skewness). The delay parameter \((d)\) shifts the transformation in time without affecting shape. The time to peak contribution (visual maximum) of an input transformation is \(T_p = (\alpha - 1)/\beta + d\). Note that \(\alpha \geq 1\) in this study. The similar expression \(T_{50} = \alpha/\beta + d\) 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 Figures 2 and 3.

Considering a catchment as a multiple input multiple output nonlinear dynamical system (MIMO NDS) gives the form:

\[
\frac{dX}{dt} = f(X, U)
\]

where \(X\) is a vector representing the current state of the catchment and \(U\) is a vector of inputs that may impact \(\frac{dX}{dt}\).

We divide \(X\) into its observable \((X_o)\) and unobservable components \((X_{uo})\).

In this study, we consider only observed liquid precipitation as an input, such that \(U = r_o\).

We assume that the direct impact of \(r_o\) on \(X_o\) is negligible compared to the contribution of the drainage processes represented by the unobservable reservoirs. These reservoirs are denoted as \(X_{uo}\).

This yields:

\[
\frac{d}{dt} \begin{bmatrix} X_o \\ X_{uo} \end{bmatrix} = \begin{bmatrix} f(X_o, X_{uo}) \\ f(X_o, X_{uo}, r_o) \end{bmatrix}
\]

\(X_{uo}\) is the link between \(r_o\) and \(X_o\), but \(X_{uo}\) is unknown. \(X_{uo}\) is approximated by transforming the rain such that \(X_{uo} = T(r_o)\) where \(T(r_o)\) is the forward propagation in time of Equation 1 scaled by the magnitude of the rain time series at each time step. As \(X_{uo}\) is a vector of time series, there may be an arbitrary number of rain transformations.

The fidelity of the approximation is evaluated by how parsimonious and generalizable of a link it creates between \(r_o\) and \(X_o\). As such, \(X_{uo}\) does not directly track to any measurement within the watershed, but is best understood as the unobservable reservoirs depicted in Figure 2. Then:
\( \frac{dX_o}{dt} = f(X_o, T(r_o)) \)

\( X_o \) is a scalar time series in this study, but it may be a vector. \( f \) is constrained as follows: (1) constant bias terms are omitted because \( \frac{dX}{dt} \) is not a function of time, (2) interaction terms are omitted because the instantaneous interaction term between rain and stage has no physically meaningful interpretation, and (3) only polynomial terms are included because they should be sufficient to capture the dynamics at the resolution we desire (by Taylor’s Theorem).

This gives the the final form:

\( \frac{dX_o}{dt} = P_x(X_o) + P_r(T(r_o)) \)

where \( P_x \) and \( P_r \) are polynomials excluding the zero order term. \( P_x \) describes the shape of the recession curve as it is the autocorrelation on stage or discharge. \( P_r \) describes the contributions of the hypothetical reservoirs in Figure 2. Consult Figure 3 for an example of an equation of this form (left column).

Equation 5 can be expanded as:

\[
\frac{d}{dt} \begin{bmatrix} x_{o,1} \\ \vdots \\ x_{o,k} \\ \vdots \\ x_{o,n} \end{bmatrix} = \sum_{i=1}^{q} \begin{bmatrix} p_{x,1,i} x_{o,1}^i + \sum_{j=1}^{m} p_{r,1,i,j} T(r_o)_j \\ \vdots \\ p_{x,k,i} x_{o,k}^i + \sum_{j=1}^{m} p_{r,k,i,j} T(r_o)_j \\ \vdots \\ p_{x,n,i} x_{o,n}^i + \sum_{j=1}^{m} p_{r,n,i,j} T(r_o)_j \end{bmatrix}
\]

where \( m \) is the number of rainfall transformations, \( |X_o| = n \), and \( q \) is the order of the differential equation. Note that \( P_r \) is a three dimensional tensor in this case.

### 3.2 Generating the input transformations from data

Equation (5) is used to predict the derivative of the output. We denote \( x'_i \) as the measured derivative and \( y'_i \) as our estimate.

The Sparse Identification of Nonlinear Dynamics (SINDy) (Brian M. de Silva, 2020) algorithm is used. The algorithm chooses parameters of \( P_x \) and \( P_r \) to maximize the coefficient of determination (\( R^2 \)) between the observed and predicted derivative.

\[
\max_{P_x, P_r} \left[ 1 - \frac{\sum (x'_i - y'_i)^2}{\sum (x'_i - \bar{x}')^2} \right]
\]

This optimization is the inner loop. The outer loop is the optimization of the rainfall transformations 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’s loop starts with one input transformation and evaluates increasing numbers of transformations until the maximum number is reached or the last transformation added less than 0.5% to the final \( R^2 \) score. If the returns of the last transformation added are marginal, it is removed and the model with one less transformation is returned as the final model. That is, if the \( j \)th transformation produces marginal returns, the model with \( j \) rainfall transformations is the last to be evaluated and a model
Figure 3. Recovering a System of Parallel Reservoirs. Parallel reservoirs 1 (solid green) and 2 (solid purple) contribute to streamflow (solid cyan). Only precipitation (impulse, not pictured) and streamflow are observed. The unobservable reservoirs are approximated (dotted red and yellow) by transforming the precipitation. A differential equation is discovered (left column) to relate these transformations to streamflow (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 reservoir. **Row 2.** The algorithm finds an optimal representation of the system using a single reservoir. **Row 3.** Accuracy is below desired, so the algorithm adds a second reservoir (dotted yellow). **Row 4.** The levels of the unobservable reservoirs and the differential equation relating the reservoirs to the streamflow are recovered.
Algorithm 1: Optimize Input Transformations

1. for \( (m = 1 ; m = m + 1 ; m \leq m_{\text{max}}) \):
   2. \( c'_{i-1} = c_0 \); \( i = 0 \)
   3. for \( (s = s_0 ; s > 1) \):
      4. Generate \( C(c'_{i-1}, s) \)
      5. \( \forall C : \text{Scores} = \max_{P_x, P_r} \left[ 1 - \frac{\sum (x'_i - y'_i)^2}{\sum (x'_i - \bar{x}'_i)^2} \right] \)
      6. \( c'_i = C[\argmax \text{[Scores]}] \); \( i = i + 1 \)
      7. if \( (c'_i == c'_{i-1}) : s = a \cdot s \)
5. End if \( \Delta R^2 < \Delta_{\text{min}} \) from last \( m \).

where:
\( m \) is the number of input transformations,
\( c_0 \) is the initial set of input transformations,
\( c' \) is the best performing set of input transformations for a given iteration,
\( i \) is the iteration number,
\( s \) determines how far away the compass search looks,
\( s_0 \) 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,
\( \Delta_{\text{min}} \) is the minimum accuracy increase from an additional transformation.

with \( j-1 \) transformations is returned. If \( m = m_{\text{max}} \) delivers non-marginal returns on accuracy the model with \( m_{\text{max}} \) transformations is returned.

(2): An initial guess is chosen for the input transformation. For records with more than one event (Figures 4 and 5) the first transformation is Equation 1 with parameters \((\alpha, \beta, d) = (1, 1, 0)\) while additional transformations are broader peaks centered at timesteps of maximum derivative in the output. For records of only one event (Figures 1 and 3), the first transformation is also based on the output derivative.

(3): Larger \( s \) corresponds to a larger perturbation. So the optimization begins looking far afield and converges to small steps.

(4): Eight input transformations are generated by perturbing the shape of the last iteration’s best performing transformation. The perturbations are scaled by \( s \).

(5): The input transformations are scored by how well SINDy performs using them.

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

(7): If the best performing input transformation is the same as last iteration, that means that none of the perturbations were helpful. \( s \) is reduced by the factor \( a \). If \( s \) is now less than one, we will exit the loop at (3).

A conceptual illustration of the algorithm is shown in Figure 3. In the figure, we imagine a hydrograph that is the outflow from two parallel reservoirs. As the steps of the algorithm iterate over the above approach, the algorithm recovers the dynamics of the two individual reservoirs and adds them together to reconstruct the cumulative signal.

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

\[ (8) : 3m + q(1 + m) \]

where \( m \) is the number of input transformations and \( q \) is the order of the ordinary differential equation. Each input transformation adds three parameters that describe its
shape. Increasing the order of the differential equation adds a term for the stage auto-
correlation and each input transformation.

The input transformation (Equation 1) could also be understood as the probabil-
ity density function of the time of travel for an individual drop of rain. We have presented
a more physical interpretation to support intuitive understanding.

3.3 Evaluation & Implementation

To evaluate the algorithm, we first apply it in a simulated setting to determine how
well it can replicate rainfall runoff models that are produced by a more complex, process-
based model. We then apply the algorithm to discover models from data at 14 US Ge-
ological Survey (USGS) stream gauges in the southern United States. Lastly, to inter-
pret how model parameters relate to watershed characteristics, we analyze 30 USGS gauges
using models with a linear differential equation and one input transformation. These are
the simplest models possible with this method, having only five parameters.

First, we apply our approach to data generated by a process-based urban water model
(Bryant E. McDonnell & Mullapudi, 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 with 420 storage nodes, 1200 junc-
tions, and 1800 subcatchments. The calibrated model was shared with us by a munic-
ipality and represents an urban stormwater system for roughly 100,000 people. While
EPA-SWMM may be simpler than some research-grade models, its explicit representa-
tion 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 underlying process model, but rather to evaluate if our proposed
framework can capture similar input-output dynamics, but with reduced model complex-
ity.

Three junctions are selected in the large urban watershed model. Two simulations
of the full software model are run using synthetic rainfall time series and the resulting
water levels are recorded. We train our approach on one set of rainfall runoff data for
each junction, and then evaluate using the other set (Figure 4). $R^2$ is used to measure
how accurately the reduced model predicts derivatives in the training storm. Mean ab-
solute error (MAE) provides a performance metric on the testing storm.

This same process is repeated, but using USGS stream stage observations. Four-
teen USGS stream gauges are chosen for analysis. The locations selected are predom-
nantly across the southern US to reduce the impact of snow melt, which is beyond the
scope of this paper. The selected locations are summarized in the supplementary infor-
mation section of this paper. We use stations that also collect precipitation at the same
location. Since precipitation at the pour point becomes a worse approximation of catchment-
wide precipitation as the contributing area increases, most of these catchments have a
contributing area under 150 km$^2$. For catchments this small, stage-discharge rating curves
are often unavailable. As such, the analysis is carried out on the stage (water level). In
pre-processing, a constant offset in stage measurements is removed by subtracting the
minimum stage value over the record to improve discovery of the underlying differential
equations. Our approach is trained on the sites using a 90%-10% training-testing split,
and evaluated with the same error metrics as the previous analysis. Results are presented
in Figure 5 and Table 1.
The last analysis examines how model parameters relate to watershed characteristics over 30 USGS sites, including the 14 mentioned above. We train the simplest models possible using this method, characterizing rainfall-runoff response using only five parameters. These are the two coefficients of the linear differential equation and the three parameters describing the rainfall transformation. The watershed characteristics examined are the sum of precipitation over the record and contributing area. The sum of precipitation expresses both data availability (number of storms) and the catchment’s aridity. The Pearson correlation coefficients between the model parameters and score and the watershed characteristics are presented in Table 2.

To support replicability, the complete algorithm and all code to generate the figures in this paper are implemented and freely shared as Python notebooks. In the accompanying code, the rainfall is transformed through Equation (1) and then fed into the model identification algorithm via the pySINDy library to learn differential equations relating the transformations to the output. The complete implementation of this toolchain, along with code to generate the figures in this paper, is documented at a public URL 1. All analyses took place on a laptop with 32GB RAM and an Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz 1.50 GHz processor.

4 Results and Discussion

4.1 Reduction of a process-based hydraulic and hydrologic model

As measured by fit metrics, our approach successfully replicated the dynamics and magnitude exhibited by the more complex physical model (figure 4). When trained on just one storm, the algorithm accurately predicted the depth at three locations in a subsequent storm. Compared to existing machine-learning based methods – which require large amounts of data to implicitly learn underlying dynamics – this relatively low data requirement is a major benefit of the approach. Since hydrologic dynamics are implicit in the formulation of our approach, the model output space is constrained and thus only needs to discover a limited number of parameters. In fact, the order of each differential equation selected by the algorithm ranged from one to three and only two or three input transformations were needed (Tables S1-S3). For example, one of the candidate models for Junction A uses a second order differential equation and two input transformations. This model of 12 parameters (see Equation 8) captures 96% of the variation in the training storm. Compared to the complex, process-based model – which has hundreds of junctions and catchments – this drastically reduces model state space and overall complexity needed to describe the data.

While beyond the scope of this paper, computational efficiency and surrogate modeling are a secondary benefit of the approach. The process-based model took two hours to run both storm events (training and testing) in Figure 4. In contrast, our algorithm took about fifteen seconds to predict the response to the testing storm after being trained. Training for Junctions A, B, and C took 1.9, 0.2, and 3.4 hours respectively. 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.

4.2 Predicting stream stage in real catchments

Table 1 summarizes performance statistics for the application of the approach to 14 USGS gauging stations, while three sites are highlighted in Figure 5 for illustrative

---

1 https://github.com/dantzert/automatic-rainfall-runoff-models
Figure 4. Reduction of a process-based model. The left column shows training for three storage assets. The right column shows the trained model predicting the system’s response to a different storm event.

purposes. More details on the three sites of Figure 5 and training and testing figures for all 14 USGS sites are included in the supplemental information of this paper. The first two rows of Figure 5 illustrate two sites with good performance, as indicated by $R^2$ and $MAE$ metrics. At these and other locations, the algorithm satisfactorily captured peak stage magnitudes, rates, and dynamics. 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.

To illustrate limitations of this approach, a third site is plotted in the final row of figure 5. This site has fluctuations which could be natural or caused by human intervention. Since these fluctuations are not driven by rainfall, our approach does not capture, nor seek to predict these variations. Nonetheless, the method still captures the recession trend and contributions of rainfall. As such, discretion should be used to determine if predictions demand representation of more complex processes. We excluded sites with significant snowmelt, but other sources of variation (e.g. dam releases, evapotranspiration) could still be present in the data. While beyond the scope of this paper, future work could include those processes more explicitly by including other other input data beyond precipitation (e.g. ambient temperature, solar radiation).

Overall, the method recovered more than half of the variation in the training records, as measured by $R^2$ (Table 1). Some of the variation in the training record was driven by noise in the sensor measurements, so the $R^2$ score likely understates how well the dynamics were captured. On the prediction task, $MAE$ averaging less than twenty centimeters makes performance adequate for many practical scenarios.

4.3 Interpreting generated models

The models output by this approach have two components: a set of input transformations and a differential equation. The number of input transformations can be un
Figure 5. Predicting Stage at USGS Gauging Stations (USGS, 2016) Based on Colocated Precipitation. Training is shown on the left, while testing is shown on the right.

Table 1. Performance summary for 14 USGS stream gauging stations.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Median</th>
<th>Standard Deviation</th>
<th>Maximum</th>
<th>Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training $R^2$</td>
<td>0.54</td>
<td>0.56</td>
<td>0.20</td>
<td>0.85</td>
<td>0.08</td>
</tr>
<tr>
<td>Testing MAE [m]</td>
<td>0.14</td>
<td>0.11</td>
<td>0.14</td>
<td>0.44</td>
<td>0.02</td>
</tr>
</tbody>
</table>
Table 2. Correlation coefficients of rainfall transformation parameters and model score to watershed aridity and contributing area. Models with linear differential equations and one rainfall transformation (total of five parameters) are trained on 30 USGS stations including the sites in Figure 5. Records are analyzed for October 15, 2022 to January 15, 2023.

<table>
<thead>
<tr>
<th></th>
<th>Total Precipitation</th>
<th>Contributing Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>-0.214</td>
<td>-0.072</td>
</tr>
<tr>
<td>$1/\beta$</td>
<td>0.441</td>
<td>-0.029</td>
</tr>
<tr>
<td>$d$</td>
<td>-0.262</td>
<td>-0.002</td>
</tr>
<tr>
<td>$R^2$ Score</td>
<td>0.592</td>
<td>-0.362</td>
</tr>
<tr>
<td>$T_p$</td>
<td>-0.24</td>
<td>-0.07</td>
</tr>
</tbody>
</table>

understood as the number of effective subcatchments in the contributing area. These effective subcatchments do not correspond to real geographic areas, but instead capture components of the overall rainfall-runoff response. The most intuitive way to examine these effective subcatchments is to visualize the shape of their response to an impulse of precipitation. Metrics such as $T_p$ and $T_{50}$ may also be helpful in understanding speed and shape of response. For example, a model may have two input transformations where the $T_p$ of one is twice that of the other. This can be understood as indicating two effective subcatchments that respond to rainfall at different rates.

The differential equation contains a polynomial on stage and the input transformations. The coefficients on powers of stage describe the shape of the recession curve. The coefficients on powers of the input transformations describe their contributions to streamflow. As rainfall transformations are generally not all in the same order of magnitude, the coefficients of different rainfall transformations cannot be directly compared to infer their relative importance.

To examine how model performance and parameters relate to catchment characteristics, Table 2 correlates model parameters to the total precipitation and contributing area of 30 USGS gauging sites. The overall model performance, as measured by $R^2$ score, was negatively correlated with contributing catchment area. Intuitively, a smaller contributing area makes precipitation at the gauging station a better representation of catchment-wide precipitation, and thus leads to better performance. As such, the performance of the proposed approach degrades as catchment scale goes up, unless more spatially-representative rainfall data are available. The positive correlation with total precipitation likely reflects the quantity of training available. The more storms in the record, the more data are available to train the model. While no "right" number of storms exists, ten or more events seems to be sufficient to train a model. However, consult the second row of Figure 5 for a model trained on only three storms. Number of storms aside, since a major benefit of the approach is its automated implementation, it can re-train periodically and improve as more data become available.

Another strong correlation is between the $\beta$ parameter and total precipitation. A larger $1/\beta$ reflects a rainfall transformation with a broader and delayed peak. Considering total precipitation over the record as a proxy for aridity, it follows that catchments with more rainfall should see a broadened and delayed peak due to groundwater and other catchment storage contributions to streamflow after rainfall events. Whereas losing streams in arid catchments (fewer storms in the data record) may have flashier storm responses more dominated by immediate runoff. Some parameters notably lack expected correlations. We posit that $d$ and $T_p$ would correlate positively with contributing area if spatially distributed rainfall data were used, rather than precipitation at the pour point. This should be tested as part of future efforts.
5 Limitations and Future Directions

With automaticity, accuracy, and interpretability, potential applications of this approach include low-cost predictive models for sensor networks and generalizing to ungaged catchments. As the method only requires precipitation and stage height, data needs are low compared to methods reliant on physical parameters that need to be measured in the field (e.g., cross sections, soil types). The small computational burden implies that a consumer laptop could update models rapidly and generate predictions daily for networks of hundreds of sensors. Since predictive models could then be deployed from precipitation data and stream gauging alone, the proposed approach can be quickly scaled, given the proliferation of low-cost and accessible sensor and wireless technologies.

Acquiring suitable precipitation data is a challenge of implementing this approach. This study used point precipitation measured at the stream gauging 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.

The input-output relationship between precipitation and stage could be improved significantly by automatically delineating the contributing area of each gauging station and sourcing spatially distributed precipitation data from weather stations or radar inferred rainfall. Coarse flow direction grids on the order of 30 arc-seconds delineate catchments with sufficient accuracy for this task. Multiple rainfall records could then be transformed to approximate the unobservable states. In catchments where snowmelt is significant, inferred snowmelt as a function of measurements including wind speed and air temperature could be transformed as just another time series input, similar to how rainfall is in this study.

Though rainfall transformation Equation 1 satisfies the criteria of our assumptions, other input transformations may be more effective. Decisions such as the sequence in which transformations are added, starting guesses for the transformations, and the restrictions on the SINDy algorithm could likely be improved. The characteristics of the optimization such as its convexity are not explored herein and may lead to even further improvements in the method.

From informal analyses, this method appears to be a more general discovery about latent state estimation in partially observable dynamical systems. Future studies will examine application and interpretation of this method in domains outside hydrology.

6 Conclusion

We have presented a method combining input transformations 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 was predicted at USGS 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. The algorithm may one day serve as the foundation of low-cost predictive networks using solely precipitation and stage data.

7 Open Research

Software: The United States Environmental Protection Agency’s Stormwater Management Model is available for free download at https://www.epa.gov/water-research/stormwater-management-model-swmm. All scripts used in analysis and figure creation are available at https://github.com/dantzert/automatic-rainfall-runoff-models. This study does not use any proprietary software and no registration or payment is required.
Data: Data for Figures 1 and 5 and Tables 1 and 2 are publicly available from the United States Geological Survey at https://maps.waterdata.usgs.gov/mapper/index.html. Alternatively, the scripts generating these tables and figures automatically fetch the data via REST API. The software model referenced in Figure 4 is not publicly available for privacy reasons.

Acknowledgments
We thank M. Tobias, B.E. Mason, J.Q. Schmidt, and L. Wojciechowski for their comments on the figures. This work was supported by the US National Science Foundation Grant 1750744 and the Michigan Department of Transportation grant OR21-003.

References
USGS. (2016). *National water information system*.

-17-
Contents of this file

Figures S1 to S42
Tables S1 to S17

Introduction

- Figures S1 through S42 show training, testing, and combined images for the 14 USGS sites referenced in Table 1.
- Tables S1 through S3 are performance summaries for the SWMM junctions.
- S4 is the overall performance summary of all 14 USGS sites.
- The remaining tables are performance summaries for the USGS stations referenced in Table 1.
- In all tables the column “plot_scores” refers to a formula used to decide which candidate model’s prediction to show. That formula is:

  \[ \text{Plot score} = \frac{R^2}{\max(R^2)} + \frac{\min(MAE)}{MAE} + \frac{\min(RMSE)}{RMSE} \]

  Where the maximums and minimums are taken over the three candidate models.
- For additional details, please consult https://github.com/dantzert/automatic-rainfall-runoff-models
USGS 02299950 MANATEE RIVER AT SR 64 NEAR MYAKKA HEAD, FL | R^2 = 0.62 | MAE = 0.44 m

- Rainfall [cm/hr]
- Measured Stage
- Best Fit
- Prediction
Testing USGS 08178700 Salado Ck at Loop 410, San Antonio, TX | MAE = 0.04 m

Rainfall [cm/hr]

Measured Stage
Prediction

Time [days]

Stage [m]
<table>
<thead>
<tr>
<th>Junction A</th>
<th>training_time_minutes</th>
<th>predicting_time_seconds</th>
<th>differ_order</th>
<th>R2_score</th>
<th>training_mae_meters</th>
<th>training_rmse_meters</th>
<th>testing_mae_meters</th>
<th>testing_rmse_meters</th>
<th>number_of_reservoirs</th>
<th>plot_scores</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>26.42571582</td>
<td>10.9276583</td>
<td>1</td>
<td>0.9405798982</td>
<td>0.2041601358</td>
<td>0.2761453476</td>
<td>0.08163859026</td>
<td>0.1188054676</td>
<td>3</td>
<td>2.84302803</td>
</tr>
<tr>
<td>1</td>
<td>16.53518957</td>
<td>10.6684538</td>
<td>2</td>
<td>0.9595422332</td>
<td>0.1875750785</td>
<td>0.2620981983</td>
<td>0.05429203395</td>
<td>0.08305155108</td>
<td>2</td>
<td>2.994754363</td>
</tr>
<tr>
<td>2</td>
<td>65.64319534</td>
<td>17.2867854</td>
<td>3</td>
<td>0.9640528632</td>
<td>0.1874687578</td>
<td>0.2621506167</td>
<td>0.05743169644</td>
<td>0.0871961943</td>
<td>3</td>
<td>2.999800045</td>
</tr>
<tr>
<td>Junction B</td>
<td>training_time_minutes</td>
<td>predicting_time_seconds</td>
<td>diffולם order</td>
<td>R² score</td>
<td>training_mae_meters</td>
<td>training_rmse_meters</td>
<td>testing_mae_meters</td>
<td>testing_rmse_meters</td>
<td>number_of_reservoirs</td>
<td>plot_scores</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------</td>
<td>-------------------------</td>
<td>----------------</td>
<td>----------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>-------------------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
<td>3.950391307</td>
<td>11.0434098</td>
<td>1</td>
<td>0.6474274601</td>
<td>0.2802639775</td>
<td>0.3565523564</td>
<td>0.1123468343</td>
<td>0.1631472816</td>
<td>2</td>
<td>2.913702085</td>
</tr>
<tr>
<td>1</td>
<td>4.433427777</td>
<td>11.9807807</td>
<td>2</td>
<td>0.6633716266</td>
<td>0.2837134478</td>
<td>0.359745936</td>
<td>0.1211216658</td>
<td>0.1709895266</td>
<td>2</td>
<td>2.91730618</td>
</tr>
<tr>
<td>2</td>
<td>4.442820952</td>
<td>11.1198243</td>
<td>3</td>
<td>0.6617782041</td>
<td>0.2726149243</td>
<td>0.3440931733</td>
<td>0.104820948</td>
<td>0.1458541166</td>
<td>1</td>
<td>2.997697994</td>
</tr>
<tr>
<td>Junction C</td>
<td>training_time_mi</td>
<td>predicting_time_diffeq_order</td>
<td>R2_score</td>
<td>training_mae_m</td>
<td>training_rmse_m</td>
<td>testing_mae_m</td>
<td>testing_rmse_m</td>
<td>number_of_rese</td>
<td>plot_scores</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-----------------</td>
<td>-----------------------------</td>
<td>----------</td>
<td>----------------</td>
<td>----------------</td>
<td>---------------</td>
<td>---------------</td>
<td>----------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>35.44</td>
<td>16.477</td>
<td>1</td>
<td>0.649</td>
<td>0.048</td>
<td>0.07</td>
<td>0.021</td>
<td>0.032</td>
<td>4</td>
<td>2.696</td>
</tr>
<tr>
<td>1</td>
<td>64.37</td>
<td>16.542</td>
<td>2</td>
<td>0.749</td>
<td>0.233</td>
<td>0.246</td>
<td>0.288</td>
<td>0.295</td>
<td>3</td>
<td>1.409</td>
</tr>
<tr>
<td>2</td>
<td>46.81</td>
<td>16.446</td>
<td>3</td>
<td>0.792</td>
<td>0.045</td>
<td>0.066</td>
<td>0.024</td>
<td>0.036</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
### USGS Performance Summary

<table>
<thead>
<tr>
<th>USGS ID</th>
<th>R^2 (training)</th>
<th>mae (testing) [m]</th>
<th>complete record?</th>
<th>n = 24</th>
<th>mean</th>
<th>median</th>
<th>std dev</th>
<th>max</th>
<th>min</th>
</tr>
</thead>
<tbody>
<tr>
<td>9504420</td>
<td>0.03</td>
<td>0.04</td>
<td>0 R^2</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td>0.38625</td>
<td>0.03</td>
</tr>
<tr>
<td>9496500</td>
<td>0.19</td>
<td>0.02</td>
<td>0 MAE</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td>0.1470833333</td>
<td>0.52</td>
</tr>
<tr>
<td>8386505</td>
<td>0.08</td>
<td>0.02</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.355</td>
<td>0.02</td>
</tr>
<tr>
<td>8178700</td>
<td>0.3</td>
<td>0.04</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2491343709</td>
<td>0.03</td>
</tr>
<tr>
<td>8178500</td>
<td>0.08</td>
<td>0.1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1442063467</td>
<td>0.02</td>
</tr>
<tr>
<td>8177700</td>
<td>0.46</td>
<td>0.04</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.85</td>
<td>0.03</td>
</tr>
<tr>
<td>8072760</td>
<td>0.23</td>
<td>0.25</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.38625</td>
<td>0.03</td>
</tr>
<tr>
<td>8049700</td>
<td>0.2</td>
<td>0.31</td>
<td>0</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td>0.5378571429</td>
<td>0.55</td>
</tr>
<tr>
<td>8014800</td>
<td>0.28</td>
<td>0.52</td>
<td>0 MAE</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td>0.1435714286</td>
<td>0.44</td>
</tr>
<tr>
<td>7331200</td>
<td>0.32</td>
<td>0.03</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.555</td>
<td>0.08</td>
</tr>
<tr>
<td>7160350</td>
<td>0.41</td>
<td>0.05</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2047203936</td>
<td>0.08</td>
</tr>
<tr>
<td>7160290</td>
<td>0.3</td>
<td>0.05</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1357377076</td>
<td>0.02</td>
</tr>
<tr>
<td>7160280</td>
<td>0.85</td>
<td>0.04</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.5378571429</td>
<td>0.55</td>
</tr>
<tr>
<td>6932000</td>
<td>0.05</td>
<td>0.05</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1435714286</td>
<td>0.44</td>
</tr>
<tr>
<td>4174518</td>
<td>0.65</td>
<td>0.16</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2047203936</td>
<td>0.08</td>
</tr>
<tr>
<td>2308950</td>
<td>0.51</td>
<td>0.05</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1357377076</td>
<td>0.02</td>
</tr>
<tr>
<td>2307780</td>
<td>0.79</td>
<td>0.12</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.5378571429</td>
<td>0.55</td>
</tr>
<tr>
<td>2307674</td>
<td>0.59</td>
<td>0.11</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2047203936</td>
<td>0.08</td>
</tr>
<tr>
<td>2303205</td>
<td>0.39</td>
<td>0.11</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1357377076</td>
<td>0.02</td>
</tr>
<tr>
<td>2301740</td>
<td>0.75</td>
<td>0.14</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2047203936</td>
<td>0.08</td>
</tr>
<tr>
<td>2300033</td>
<td>0.61</td>
<td>0.38</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1357377076</td>
<td>0.02</td>
</tr>
<tr>
<td>2299950</td>
<td>0.62</td>
<td>0.44</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2047203936</td>
<td>0.08</td>
</tr>
<tr>
<td>2294898</td>
<td>0.52</td>
<td>0.31</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1357377076</td>
<td>0.02</td>
</tr>
<tr>
<td>training_time_minutes</td>
<td>predicting_time_seconds</td>
<td>diffeq_order</td>
<td>R2_score</td>
<td>training_mae_meters</td>
<td>training_rmse_meters</td>
<td>testing_mae_meters</td>
<td>testing_rmse_meters</td>
<td>number_of_reservoirs</td>
<td>plot_scores</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------</td>
<td>--------------</td>
<td>----------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>--------------------</td>
<td>--------------------</td>
<td>----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
<td>69.63216015</td>
<td>1</td>
<td>0.564347142</td>
<td>0.790387911</td>
<td>1.194614322</td>
<td>0.227669383</td>
<td>0.27304824</td>
<td>4</td>
<td>2.799641446</td>
</tr>
<tr>
<td>1</td>
<td>8.223851908</td>
<td>2</td>
<td>0.52266393</td>
<td>0.739596431</td>
<td>1.038476162</td>
<td>0.30687986</td>
<td>0.406966591</td>
<td>1</td>
<td>2.921142198</td>
</tr>
<tr>
<td>2</td>
<td>216.6762561</td>
<td>3</td>
<td>0.567408518</td>
<td>0.789284396</td>
<td>1.193836761</td>
<td>0.867604776</td>
<td>1.040041083</td>
<td>4</td>
<td>2.806911271</td>
</tr>
<tr>
<td>Training Time (Minutes)</td>
<td>Predicting Time (Seconds)</td>
<td>DiffEq Order</td>
<td>R^2 Score</td>
<td>Training MAE (Meters)</td>
<td>Training RMSE (Meters)</td>
<td>Testing MAE (Meters)</td>
<td>Testing RMSE (Meters)</td>
<td>Number of Reservoirs</td>
<td>Plot Scores</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------</td>
<td>--------------</td>
<td>-----------</td>
<td>----------------------</td>
<td>-----------------------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
<td>22.03419056</td>
<td>26.7791385</td>
<td>1</td>
<td>0.494386518</td>
<td>0.60852345</td>
<td>0.93649694</td>
<td>0.393374618</td>
<td>0.507762975</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>140.6438553</td>
<td>6.5949576</td>
<td>2</td>
<td>0.51647193</td>
<td>0.729910144</td>
<td>0.978652801</td>
<td>0.384315538</td>
<td>0.512278175</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>117.4022842</td>
<td>43.1927156</td>
<td>3</td>
<td>0.616811473</td>
<td>0.681782631</td>
<td>1.009006195</td>
<td>0.438295003</td>
<td>0.591562497</td>
<td>4</td>
</tr>
<tr>
<td>training_time_minutes</td>
<td>predicting_time_seconds</td>
<td>diffeq_order</td>
<td>R2_score</td>
<td>training_mae_meters</td>
<td>training_rmse_meters</td>
<td>testing_mae_meters</td>
<td>testing_rmse_meters</td>
<td>number_of_reservoirs</td>
<td>plot_scores</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------</td>
<td>--------------</td>
<td>-------------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>--------------------</td>
<td>--------------------</td>
<td>----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
<td>57.68201994</td>
<td>3.9978995</td>
<td>1</td>
<td>0.530289799</td>
<td>0.792500095</td>
<td>1.048040232</td>
<td>0.314813628</td>
<td>0.415922532</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>158.8627201</td>
<td>5.3731507</td>
<td>2</td>
<td>0.56748392</td>
<td>0.753416107</td>
<td>1.038362234</td>
<td>0.374008518</td>
<td>0.462054695</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>100.5352627</td>
<td>7.8529037</td>
<td>3</td>
<td>0.609916728</td>
<td>0.768629514</td>
<td>1.09027699</td>
<td>0.377144177</td>
<td>0.458868984</td>
<td>5</td>
</tr>
<tr>
<td>training_time_minutes</td>
<td>predicting_time_seconds</td>
<td>diffeq_order</td>
<td>R2_score</td>
<td>training_mae_meters</td>
<td>training_rmse_meters</td>
<td>testing_mae_meters</td>
<td>testing_rmse_meters</td>
<td>number_of_reservoirs</td>
<td>plot_scores</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------</td>
<td>--------------</td>
<td>----------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
<td>110.4133087</td>
<td>1</td>
<td>0.701245567</td>
<td>0.180912411</td>
<td>0.273863873</td>
<td>0.133464039</td>
<td>0.199992747</td>
<td>4</td>
<td>2.934200956</td>
</tr>
<tr>
<td>1</td>
<td>86.94275754</td>
<td>2</td>
<td>0.73059351</td>
<td>0.18454242</td>
<td>0.277453329</td>
<td>0.117312974</td>
<td>0.179146807</td>
<td>4</td>
<td>2.940690877</td>
</tr>
<tr>
<td>2</td>
<td>466.8289626</td>
<td>3</td>
<td>0.750636748</td>
<td>0.181050555</td>
<td>0.274246484</td>
<td>0.138409844</td>
<td>0.204648507</td>
<td>5</td>
<td>2.997841847</td>
</tr>
<tr>
<td>training_time_minutes</td>
<td>predicting_time_seconds</td>
<td>diffeq_order</td>
<td>R2_score</td>
<td>training_mae_meters</td>
<td>training_rmse_meters</td>
<td>testing_mae_meters</td>
<td>testing_rmse_meters</td>
<td>number_of_reservoirs</td>
<td>plot_scores</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------</td>
<td>--------------</td>
<td>----------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>-------------------</td>
<td>---------------------</td>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
<td>143.3632544</td>
<td>15.6687597</td>
<td>1</td>
<td>0.389524159</td>
<td>0.132798202</td>
<td>0.187853806</td>
<td>0.11008223</td>
<td>0.152782524</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>235.2317808</td>
<td>9.1287303</td>
<td>2</td>
<td>0.329547377</td>
<td>0.140906779</td>
<td>0.197983161</td>
<td>0.108028169</td>
<td>0.14801754</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>195.6291754</td>
<td>10.5107845</td>
<td>3</td>
<td>0.3357411157</td>
<td>0.141576721</td>
<td>0.1988133</td>
<td>0.110441275</td>
<td>0.149016079</td>
<td>1</td>
</tr>
<tr>
<td>training_time_minutes</td>
<td>predicting_time_seconds</td>
<td>diffeq_order</td>
<td>R2_score</td>
<td>training_mae_meters</td>
<td>training_rmse_meters</td>
<td>testing_mae_meters</td>
<td>testing_rmse_meters</td>
<td>number_of_reservoirs</td>
<td>plot_scores</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------</td>
<td>----------------------</td>
<td>--------------------</td>
<td>--------------------</td>
<td>----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
<td>25.70291675</td>
<td>10.9273106</td>
<td>1</td>
<td>0.541719739</td>
<td>0.118099433</td>
<td>0.169413436</td>
<td>0.095920079</td>
<td>0.182525704</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>64.57174859</td>
<td>11.8959675</td>
<td>2</td>
<td>0.590908912</td>
<td>0.11484768</td>
<td>0.168390056</td>
<td>0.107246263</td>
<td>0.195831601</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>74.16637148</td>
<td>12.8316491</td>
<td>3</td>
<td>0.592318226</td>
<td>0.11619152</td>
<td>0.170496275</td>
<td>0.111547337</td>
<td>0.198556561</td>
<td>2</td>
</tr>
<tr>
<td>training_time_minutes</td>
<td>predicting_time_seconds</td>
<td>diffeq_order</td>
<td>R2_score</td>
<td>training_mae_meters</td>
<td>training_rmse_meters</td>
<td>testing_mae_meters</td>
<td>testing_rmse_meters</td>
<td>number_of_reservoirs</td>
<td>plot_scores</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------</td>
<td>--------------</td>
<td>----------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>---------------------</td>
<td>---------------------</td>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>172.5051678</td>
<td>14.9613191</td>
<td>1</td>
<td>0.766351564</td>
<td>0.107049734</td>
<td>0.172792382</td>
<td>0.092755105</td>
<td>0.189322577</td>
<td>3</td>
<td>2.956833962</td>
</tr>
<tr>
<td>72.35168594</td>
<td>21.5800827</td>
<td>2</td>
<td>0.790113179</td>
<td>0.105648202</td>
<td>0.172979533</td>
<td>0.119810181</td>
<td>0.231026689</td>
<td>4</td>
<td>2.998918076</td>
</tr>
<tr>
<td>139.7131955</td>
<td>13.2118648</td>
<td>3</td>
<td>0.772537783</td>
<td>0.123601453</td>
<td>0.194059489</td>
<td>0.116545402</td>
<td>0.231713736</td>
<td>3</td>
<td>2.722914057</td>
</tr>
<tr>
<td>training_time_minutes</td>
<td>predicting_time_seconds</td>
<td>diffeq_order</td>
<td>R2_score</td>
<td>training_mae_meters</td>
<td>training_rmse_meters</td>
<td>testing_mae_meters</td>
<td>testing_rmse_meters</td>
<td>number_of_reservoirs</td>
<td>plot_scores</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------</td>
<td>--------------</td>
<td>----------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>-------------------</td>
<td>---------------------</td>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
<td>111.4936312</td>
<td>18.4043145</td>
<td>1</td>
<td>0.514410382</td>
<td>0.03721727</td>
<td>0.064279512</td>
<td>0.04897222</td>
<td>0.074398085</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>165.4588837</td>
<td>15.630961</td>
<td>2</td>
<td>0.414217696</td>
<td>0.040320777</td>
<td>0.070207419</td>
<td>0.061780644</td>
<td>0.086852333</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>8.108954182</td>
<td>17.0923622</td>
<td>3</td>
<td>0.432351293</td>
<td>0.038675296</td>
<td>0.068851212</td>
<td>0.040880606</td>
<td>0.071571101</td>
<td>2</td>
</tr>
</tbody>
</table>

165.4588837 15.630961 2 0.414217696 0.040320777 0.070207419 0.061780644 0.086852333 2
8.108954182 17.0923622 3 0.432351293 0.038675296 0.068851212 0.040880606 0.071571101 2

2.643823489 2
2.73638044 2

plot_scores
<table>
<thead>
<tr>
<th>training_time_minutes</th>
<th>predicting_time_seconds</th>
<th>diffeq_order</th>
<th>R2_score</th>
<th>training_mae_meters</th>
<th>training_rmse_meters</th>
<th>testing_mae_meters</th>
<th>testing_rmse_meters</th>
<th>number_of_reservoirs</th>
<th>plot_scores</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4.59337729</td>
<td>3.6692079</td>
<td>1</td>
<td>0.292620899</td>
<td>0.03501144</td>
<td>0.047807645</td>
<td>0.024756114</td>
<td>0.046892507</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>7.074180093</td>
<td>2.9127655</td>
<td>2</td>
<td>0.282585648</td>
<td>0.035864008</td>
<td>0.047304127</td>
<td>0.024923065</td>
<td>0.046640961</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>7.638881257</td>
<td>5.9220185</td>
<td>3</td>
<td>0.853268838</td>
<td>0.016997837</td>
<td>0.037212391</td>
<td>0.039573785</td>
<td>0.065937681</td>
<td>5</td>
</tr>
<tr>
<td>training_time_minutes</td>
<td>predicting_time_seconds</td>
<td>diffeq_order</td>
<td>R2_score</td>
<td>training_mae_meters</td>
<td>training_rmse_meters</td>
<td>testing_mae_meters</td>
<td>testing_rmse_meters</td>
<td>number_of_reservoirs</td>
<td>plot_scores</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------</td>
<td>--------------</td>
<td>----------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
<td>9.068420415</td>
<td>3.527419</td>
<td>1</td>
<td>0.498631145</td>
<td>0.046182452</td>
<td>0.067783975</td>
<td>0.043607296</td>
<td>3</td>
<td>2.406262413</td>
</tr>
<tr>
<td>1</td>
<td>11.33139328</td>
<td>3.4448095</td>
<td>2</td>
<td>0.407462079</td>
<td>0.029128633</td>
<td>0.056072738</td>
<td>0.05024803</td>
<td>1</td>
<td>2.742256</td>
</tr>
<tr>
<td>2</td>
<td>18.87106721</td>
<td>2.8787696</td>
<td>3</td>
<td>0.411781271</td>
<td>0.027606278</td>
<td>0.054803133</td>
<td>0.046441064</td>
<td>1</td>
<td>2.825823407</td>
</tr>
<tr>
<td>training_time_minutes</td>
<td>predicting_time_seconds</td>
<td>diffeq_order</td>
<td>R2_score</td>
<td>training_mae_meters</td>
<td>training_rmse_meters</td>
<td>testing_mae_meters</td>
<td>testing_rmse_meters</td>
<td>number_of_reservoirs</td>
<td>plot_scores</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------</td>
<td>--------------</td>
<td>----------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>---------------------</td>
<td>---------------------</td>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>13.06712616</td>
<td>11.7302312</td>
<td>1</td>
<td>0.371081155</td>
<td>0.021702646</td>
<td>0.048071941</td>
<td>0.018113637</td>
<td>0.045743178</td>
<td>2</td>
<td>2.806173345</td>
</tr>
<tr>
<td>63.54105863</td>
<td>17.1575676</td>
<td>2</td>
<td>0.438107226</td>
<td>0.022988033</td>
<td>0.053353194</td>
<td>0.034480068</td>
<td>0.062839341</td>
<td>2</td>
<td>2.796885297</td>
</tr>
<tr>
<td>33.78278193</td>
<td>18.4648517</td>
<td>3</td>
<td>0.460299459</td>
<td>0.022632639</td>
<td>0.052420315</td>
<td>0.040607381</td>
<td>0.073032533</td>
<td>2</td>
<td>2.875957127</td>
</tr>
<tr>
<td>training_time_minutes</td>
<td>predicting_time_seconds</td>
<td>diffeq_order</td>
<td>R2_score</td>
<td>training_mae_meters</td>
<td>training_rmse_meters</td>
<td>testing_mae_meters</td>
<td>testing_rmse_meters</td>
<td>number_of_reservoirs</td>
<td>plot_scores</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------</td>
<td>--------------</td>
<td>----------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>---------------------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
<td>101.1709667</td>
<td>22.2766977</td>
<td>1</td>
<td>0.304237001</td>
<td>0.037342754</td>
<td>0.093117189</td>
<td>0.04130702</td>
<td>0.065138283</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>111.5241065</td>
<td>16.5500243</td>
<td>2</td>
<td>0.344983325</td>
<td>0.041775114</td>
<td>0.099899755</td>
<td>0.060255618</td>
<td>0.090883363</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>113.9424707</td>
<td>23.2641689</td>
<td>3</td>
<td>0.399139498</td>
<td>0.045128483</td>
<td>0.101409801</td>
<td>0.072142142</td>
<td>0.102246311</td>
<td>4</td>
</tr>
<tr>
<td>training_time_minutes</td>
<td>predicting_time_seconds</td>
<td>diffeq_order</td>
<td>R2_score</td>
<td>training_mae_meters</td>
<td>training_rmse_meters</td>
<td>testing_mae_meters</td>
<td>testing_rmse_meters</td>
<td>number_of_reservoirs</td>
<td>plot_scores</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------</td>
<td>--------------</td>
<td>---------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>-------------------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>------------</td>
</tr>
<tr>
<td>0</td>
<td>130.6113464</td>
<td>0.8158995</td>
<td>1</td>
<td>0.022731462</td>
<td>0.05027511</td>
<td>0.067485458</td>
<td>0.01330096</td>
<td>0.016718513</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>178.0347716</td>
<td>7.6272461</td>
<td>2</td>
<td>0.052778706</td>
<td>0.061315144</td>
<td>0.077981858</td>
<td>0.010320474</td>
<td>0.014219624</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>167.9475081</td>
<td>7.9726236</td>
<td>3</td>
<td>0.075251536</td>
<td>0.062972572</td>
<td>0.080989331</td>
<td>0.016954483</td>
<td>0.021095785</td>
<td>4</td>
</tr>
</tbody>
</table>