

Position Paper

Automated hydrologic forecasting using open-source sensors: Predicting stream depths across 200,000 km²[☆]Travis Adrian Dantzer^{*}, Branko Kerkez

University of Michigan Civil and Environmental Engineering, 2350 Hayward St, Ann Arbor, 48104, MI, USA

ARTICLE INFO

Keywords:

Model discovery
Wireless sensor networks
Open-source method
Hydrological prediction
System identification
Gray-box modeling

ABSTRACT

Wireless sensor networks support decision-making in diverse environmental contexts. Adoption of these networks has increased dramatically due to technological advances that have increased value while lowering cost. However, real-time information only allows for reactive management. As most interventions take time, predictions across these sensor networks enable better planning and decision making. Prediction models across large water level and discharge sensor networks do exist. However, they have limitations in their accessibility, automaticity, and data requirements. We present an open-source method for automatically generating computationally cheap rainfall-runoff models for any depth or discharge sensor given only its measurements and location. We characterize reliability in a real-world case study across 200,000 km², evaluate long-term accuracy, and assess sensitivity to measurement noise and errors in catchment delineation. The method's accuracy, computational efficiency, and automaticity make it a valuable asset to support operational decision making for diverse stakeholders including bridge inspectors and utilities.

1. Introduction

Wireless sensor networks support decision making by diverse means in varied environmental contexts. Real-time data helps people make better decisions in the moment. Additionally, the historical records these networks generate can improve understanding of a system's behavior and inform long-term management. Adoption of these networks has increased in the past several decades, largely due to technological advances which have given these networks greater value and lower cost (Bellini et al., 2022; Madushanki et al., 2019). Electronic components including solar panels, microprocessor boards, connectivity solutions, and sensors have become less expensive with better performance and reliability (Shalf, 2020). Programming and networking IoT devices is also easier than it used to be (Currie, 2021; Choi, 2021). Once these networks are up and running, some tools exist to automate transforming measurement data into information through processing, analysis, and visualization (Krishnan and Wu, 2019; Feurer et al., 2022; Chakraborty and Kundan, 2021; Schmidt and Kerkez, 2023). While adoption and ease-of-use have increased, many organizations still experience significant barriers in deploying and maintaining wireless sensor networks, as well as extracting insight from the data (Lin et al., 2023).

Furthermore, as the purpose of collecting data is producing information that improves decision making (Aven, 2010) and many interventions take time (e.g., flood evacuations), there is greater value in the sensor network if users also have access to predictions. Many organizations also experience barriers here, as current methods for generating predictions are not highly automated or accessible (Krajewski et al., 2017; Maidment, 2016).

Motivated by the increasing variability in precipitation (both droughts and extreme storms) caused by climate change, we consider sensor networks and predictive models for surface water. Water level and flow sensor networks are typically used for flood warning systems (Kruger et al., 2016; Normand, 2021; Kalyanapu et al., 2023), reservoir management, utility operation, and to inform ecological conservation efforts (Mason et al., 2023). However, the adoption barriers previously identified contribute to dramatic inequities in flood protection. While flood mortality has decreased in high and middle income nations since 1975, mortality has increased in low-income nations (Jonkman et al., 2024). The contribution of this work is a rapid and automated prediction engine for water level and flow sensor networks which could lower barriers in the availability of flood warning systems. We also detail enhancements to an open-source framework for

[☆] This work was supported by the US National Science Foundation Grant 1750744 and the Michigan Department of Transportation, United States grant OR21-003.

^{*} Corresponding author.

E-mail address: dantzer@umich.edu (T.A. Dantzer).

URL: <https://sites.google.com/umich.edu/travis-adrian-dantzer> (T.A. Dantzer).

<https://doi.org/10.1016/j.envsoft.2024.106137>

Received 20 February 2024; Received in revised form 31 May 2024; Accepted 30 June 2024

Available online 8 July 2024

1364-8152/© 2024 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

wireless sensor networks and a case study of this prediction engine serving bridge inspectors across 68 water level sensors spanning 200,000 km² in the Midwestern United States (<http://maps.open-storm.org/>).

2. Background

Sensor network architectures and prediction engines enabling broader adoption and thus enhancing equity in flood protection would be designed for accessibility, automaticity, data-frugality, and noise tolerance. Though supercomputers and highly-trained hydrologists are available to build predictive models at some agencies, universities and national research centers, this may not be true of the many utilities, conservation agencies, and other organizations who use (or would like to use) models built on wireless sensor network data. Expertise and computational resources are even less available in the developing nations far more strongly impacted by floods and droughts (Tshimanga et al., 2016; Bubola, 2023; Marshall, 2023). An *accessible* solution would lower barriers in expertise and expense, enabling adoption in resource-constrained settings. An *automated* solution would bridge the gap between raw sensor data and hydrologic models, thus enhancing adoption by improving scalability, consistency, and speed of deployment. *Data-frugality* (Gadzinski and Castello, 2020) means generating accurate and reliable models with little data. Many methods for rainfall-runoff modeling (Kratzert et al., 2018; Boughton, 2007) rely on long, clean records, while the data many organizations have is short and noisy. Indeed, production of these long and clean records is in itself a significant effort (Neal et al., 2011). This has tended to limit forecasts to locations with long-term monitoring programs which have been in place for many years, while end-users likely desire forecasts within the first few months of deploying their sensor networks. Many excellent and valuable frameworks exist for creating sensor networks and generating predictions, but they may have some drawbacks in accessibility, automaticity, or data-frugality. A framework which had these qualities would enhance flood protection equity by easing adoption of predictive sensor networks in the low-income nations most adversely impacted by droughts and floods (Jonkman et al., 2024).

The National Flood Interoperability Experiment produced near real-time, high spatial resolution flood forecasting for 2.7 million locations across the continental United States (Maidment, 2016). This work also integrated these forecasts into inundation mapping and local emergency response. However, it relied on process-based models developed and tuned by expert hydrologists and took 10 min to run a single timestep on a supercomputing cluster (Texas Advanced Computing Center). It also integrated water level and discharge data exclusively from the United States Geological Survey (USGS) streamgaging network. The USGS streamgaging network produces high quality data at a corresponding cost (Normand, 2021) of roughly ten times that of low-cost alternatives (Bartos et al., 2018). While the process-based method grounds results in first principles, this approach is not necessarily accessible, automated, or data-frugal.

Researchers at the Iowa Flood Center (IFC) (Krajewski et al., 2017) present an excellent and more accessible solution by using less expensive water level sensors (Kruger et al., 2016). The Iowa Flood Center's process-based model is atypical in that it is not calibrated to the measurements of that sensor network.

“Contrary to the dominant culture in the hydrologic model development community and the past practice of the NWS (National Weather Service), we claim that large-scale, high-resolution distributed models cannot be calibrated. There are simply too many degrees of freedom and too many sources of uncertainty. Therefore, tuning model parameters must be replaced with changing the model components and/or structure”. (Krajewski et al., 2017)

While this argument against calibration is compelling, manual restructuring of the model in lieu of calibration may make the IFC's approach considerably less automated than some others. Like the National Water Model, computations are run on multiple high performance computing clusters.

If services such as the previous two are not available, scientists, utilities and other organizations may opt to build their own sensor networks and prediction engines. However, despite recent advances, building a new sensor network is still difficult and expensive for organizations not experienced with these tools and technologies. Design, fabrication, and deployment of sensor networks requires resources and expertise in embedded systems design, low-level programming, cloud services, and manufacturing. Once measurements are transmitting, hosting and quality control present new challenges.

Once a new network is operational, there are further difficulties in developing prediction engines. Creating a new process-based model for each sensor in a network is generally expensive in labor and computation (Lepore et al., 2013), though this could be data-frugal depending on the approach. Creating a machine-learning model (Pandhiani et al., 2020; Dazzi et al., 2021; Bartoletti et al., 2018) would be accessible and automated so long as sufficient computational resources are available. However, these approaches generally rely on much clean data — often decades (Kratzert et al., 2018). Hybrid modeling combines process-based modeling with data-driven approaches and often results in better, more consistent results (Kapoor et al., 2023; Fathian et al., 2019). However, developing hybrid models entails the expertise and computational expense of developing *both* a process-based *and* a data-driven model. The model discovery approach developed in Dantzer (2023a) fulfills our criteria in accessibility and data-frugality, but requires the appropriate weather data to be provided.

Given the above challenges, automated solutions to enable highly localized stream level and flood forecasts would integrate:

- Sensors that measure and report hydrologic variables of interest
- Storage and hosting solutions that efficiently scale with network expansion
- Automated model discovery and calibration
- Automated data services to pull in physiographic and weather data needed to train models and make predictions
- Visualization and alerting interfaces to present the forecasts and alerts to users

3. Methods

Here, we present an end-to-end architecture, sensors included, for automated hydrologic forecasting. Once a sensor is deployed, the system automatically fuses the sensor data with physiographic features of the landscape to train a forecast model. The architecture is comprised of the following components (Fig. 1):

3.1 Open-source wireless sensors for the measurement of hydrologic variables, specifically water levels. A cloud-hosted middleware, for provisioning sensors, hosting timeseries measurements and metadata, and providing quality control.

3.2 A hydrology-informed data mask that automatically aggregates weather data for a given sensor location. A prediction engine that automatically discovers models and writes forecasts to a server.

3.3 Visualization and alerting services for desktop and mobile contexts.

As the data mask and prediction engine (3.2) are the primary contribution, this paper will focus on their formulation, implementation, and evaluation. The other components of the architecture will be covered more briefly while highlighting recent improvements. Note that this architecture is highly modular and each component could be replaced or customized with minimal impact on the other components. That is, different sensors or communication methods could be used in place of 3.1, and different visualization and alerting endpoints could be substituted for 3.3.

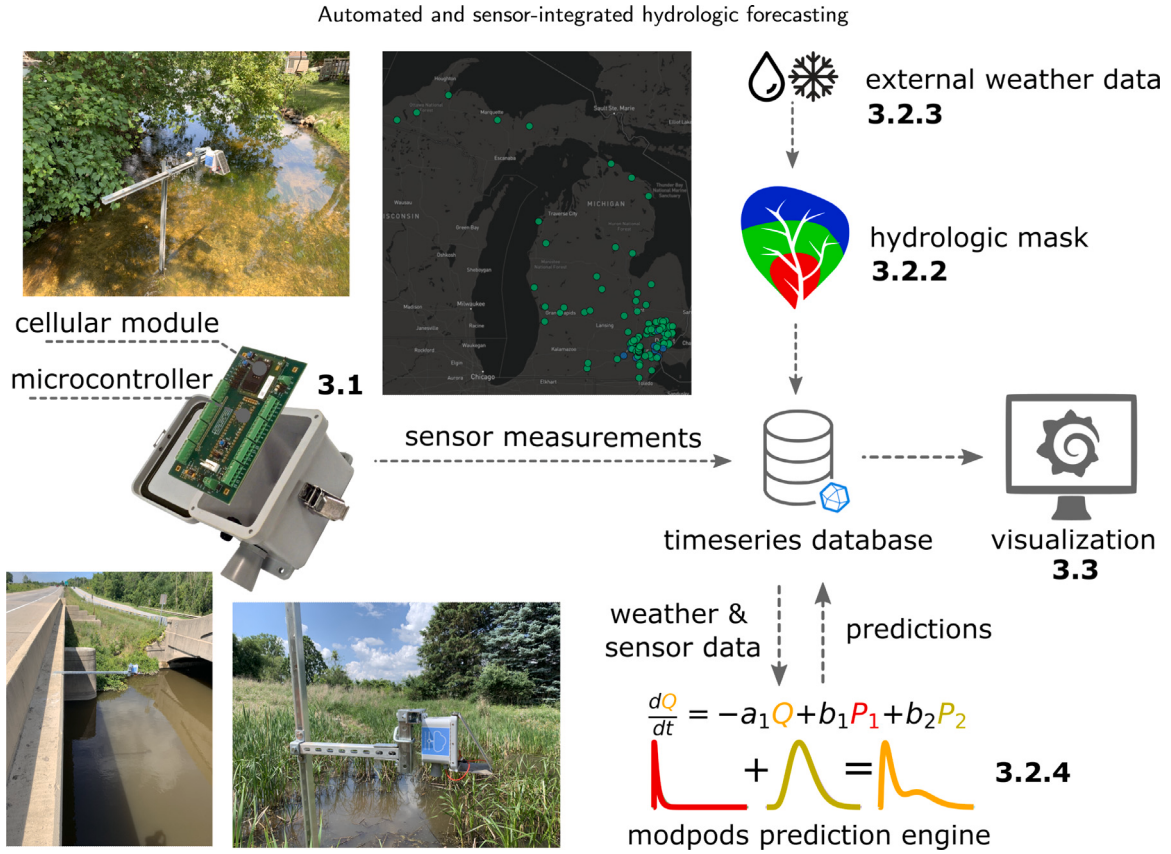


Fig. 1. Network architecture. The *open-storm* stack augmented with a hydrologic data-masking service and *modpods*-driven prediction engine. Dashed arrows indicate data transfers.

3.1. Open-source wireless sensor network

Open-storm (Bartos et al., 2018) has been providing open-source resources to enhance the accessibility of wireless water sensor networks since 2014. Here we give an overview of the architecture and describe several recent enhancements to the embedded operating system, hardware, and cloud services used in the sensor network operated by the Digital Water Lab (DWL) at the University of Michigan.

Open-storm **sensor nodes** are custom low-power embedded computers which connect to online databases using cellular networks. A microcontroller is programmed with a non-preemptive operating system which wakes the device from sleep, downloads instructions from the server, records sensor readings and triggers control assets, transmits data to the server, and returns the device to sleep. The *open-storm* printed circuit board (PCB) was updated and accommodates diverse sensors and actuators through a Cypress PSoC5LP microcontroller. The system-on-chip (SoC) design allows control over analog and digital components, enhancing the platform's flexibility for integrating new sensors and actuators. Connectivity is provided by a 4G/LTE-capable Telit cellular modem. Though many types of sensors and actuators (e.g., weirs, valves, soil moisture, optical rain) are available in the open-storm stack, this work focuses on nodes equipped with Maxbotix ultrasonic range finders that measure the distance between the sensor node and the water surface. This depth-sensor configuration is the default and requires minimal user setup beyond specifying a server endpoint to transmit data to. In the Digital Water Lab network, those distance readings are referenced to an elevation survey and reported to end-users as water depth above streambed or water surface elevation relative to the NAVD88 datum (Fig. 2).

Table 1 includes a detailed cost breakdown of a Digital Water Lab depth sensor node. After including labor costs for assembly and deployment, the complete wireless sensor node costs under 1,500 USD.

Much of the material cost does not come from the core functionality, but from components that increase longevity and durability to decrease operational costs. Indeed, a less robust version of the sensor node could be assembled for about 300 USD by using the current microcontroller board and cellular modem along with less expensive alternatives for the enclosure, sensor, and power supply. The solar charging panels and rechargeable lithium ion batteries are not the cheapest options available for powering a microcontroller, but they have allowed many devices to be deployed for multiple years without changing batteries. Similarly, the structural steel and submersible enclosure keep the sensor nodes protected, dry, and securely attached to poles and bridges. These and other features of the sensor nodes enable a small lab to keep maintenance costs near just 500 USD per year per site. Table 2 shows that much of the cost of running the network comes from technician visits to field sites to repair and replace sensor nodes. These operational cost centers are the primary motivation for the extensive investment in longevity and durability of each node shown in Table 1.

Using cellular networks is not a universal standard in wireless sensor networks for water, as Wi-Fi, mesh, and LoRaWAN are also popular protocols (Karegar et al., 2022; Montestruque et al., 2008; Pearson et al., 2019). Principal critiques of using cellular for IoT applications are its cost and power consumption. However, the sparse distribution of the open-source sensors (average density of one per 2000 km²) and deployment in rural and forested areas makes mesh and line-of-sight protocols less practicable options. The high power consumption of cellular can be accommodated by maintaining very low sleep currents (μA) and short duty cycles (Bartos et al., 2018; Moreno et al., 2019). The current iteration achieves a sleep current of 50 micro-amps at 3.7 V in its deep sleep mode. The enhanced power efficiency of the updated system has led to some devices operating for over three years without battery replacements.

Table 1

Detailed cost breakdown for an *open-storm* depth sensor node at the time of writing. All values are in USD. For more details, see <https://github.com/open-storm/docs.open-storm.org/wiki/Parts-list>.

Component	Vendor	Price per node	Details
Printed circuit board & microcontroller	Cypress PSoC	100.00	Custom-printed
Solar Panel	Voltaic Systems	35.00	5 Watt 6 Volt (SKU: P105)
Battery	Tenergy	68.99	13.7V 156000 mAh, ID 31059
Cellular Modem	Telit	100.00	Telit LE910
Enclosure and cable glands	McMaster-Carr	181.61	ID: 7740K11 and 7310K11
Distance Sensor	MaxBotix	119.95	MB7388
Structural	McMaster-Carr	61.75	5 ft steel strut channel (3310T64)
Nuts, bolts, washers	McMaster-Carr	30.00	Various
Total cost		697.30	

Table 2

Operational costs for the *open-storm* depth sensor node network at the time of writing. All values are in USD.

Component	Vendor	Cost per node per year
Cloud hosting	Amazon Web Services	32.38
Cellular Service	Twilio/KORE	55.44
Technician Labor	Internal	166.20
Management Labor	Internal	57.24
Vehicle fuel and supplies	Various	207.72
Total cost		518.98

Open-Storm's **hosting services** are built around InfluxDB, a time series-optimized database facilitating efficient data storage and retrieval. InfluxDB's RESTful APIs allow for seamless interaction between sensors and external applications, supporting both data input and output. InfluxDB is implemented in open source, but can now also be purchased as a hosted commercial services on [InfluxData.com](https://influxdata.com), which significantly reduces the server maintenance burden on users. InfluxDB's primary role is to store sensor data transmitted via HTTP Post requests and enable adaptive sampling and real-time control through cloud-stored device settings, accessible to sensor nodes during server communication. This system supports bidirectional communication with field nodes and allows remote customization of measurement and transmission frequencies, thereby reducing the need for site visits. Actuator settings such as valve open percentage are controlled automatically and remotely via the same method. Beyond environmental monitoring, the sensor node is also equipped with internal diagnostics tools that track the device's operational health, including battery levels, cellular signal strength, and network connection attempts. An optional data quality module can also be activated to refine data and detect sensor defects or obstructions (Schmidt and Kerkez, 2023). More details regarding the nodes, available sensors and actuators, and the cloud architecture are available at <https://www.digitalwaterlab.org/build>.

3.2. Prediction engine

Given its automaticity, computational efficiency, and data-frugality, we build on the approach developed for rainfall-runoff modeling in Dantzer (2023a). Here we generalize Model Discovery in Partially Observable Dynamical Systems (*modpods*, Dantzer (2023b)) to the multiple input case and develop software to automatically source the appropriate weather data given only a sensor's location. This enables automated deployment of computationally efficient predictive models given only a sensor measurement record and that sensor's location.

The *modpods*-based prediction engine takes in a depth or discharge time series (retrieved via RESTful API) and the corresponding sensor location. Then, the contributing area is delineated and weather data is sourced using publicly available datasets. Models are then trained. Once a day, trained models are fed weather forecasts and the resulting predictions are pushed to decision support dashboards. Computational expense is minimal as it takes thirty seconds to generate a one-week prediction on a consumer laptop. While accuracy is evaluated using

historical data, we also present the use case of providing predictions for 68 of the 100+ low-cost sensors within a network which spans 200,000 km² across the US state of Michigan (Fig. 1 or see <http://maps.open-storm.org/>). While the prediction engine is compatible with data generated from any kind of sensor, we present the case study network to ease adoption by illustrating an open-source, end-to-end architecture.

3.2.1. Data requirements

A digital elevation model of North America from USGS Hydrosheds (Lehner et al., 2011) is stored locally on a laptop. A coarse flow direction grid with a resolution of 300 m is used because there is no process-based modeling. The resolution need only be accurate enough to sample the right weather data. Weather data (liquid precipitation, snow depth, air temperature, and wind speed) are provided by the open-source project Meteostat (Lamprecht, 0000). The end-user need only provide a time-series record of sensor measurements and the location of the sensor. The measurements targeted for prediction in this study are water level and discharge, but could be other parameters (e.g., Total Suspended Solids) which have precipitation- and snowmelt-driven dynamics.

3.2.2. Hydrology-informed data filtering

Using the sensor's location and the Hydrosheds flow direction grid, the contributing area of the catchment with pour point at the sensor location is delineated using *pysheds* (Bartos, 0000). The flow distance for each grid cell is then calculated and used to divide the catchment (Fig. 1, 3.2.2) into regions with short, medium, and long flow paths to the sensor. These flow-distance regions will be used to aggregate rainfall. Once the total catchment is delineated and the flow distance regions are defined, the flow direction grid is no longer used. No other information about the catchment (e.g., average slope, landuse, topographic roughness index) is used by the model.

3.2.3. Spatially distributed rainfall and snowmelt

Evenly spaced points within each region are used to sample rainfall intensities at one-hour frequency. Those intensities are then averaged across each flow distance region. Snowfall is excluded because it generally does not immediately contribute to runoff. As weather stations are not ubiquitous and climate models often have spatial resolution of ten kilometers or more (NOAA, 0000), small catchments will often have identical data between different flow-distance regions. In this case the redundant data is omitted.

For estimating snowmelt, factors including albedo, insolation intensity, and humidity are important (Colbeck, 1988). However, to keep the model structure simple we coarsely represent the radiative and advective processes by making snowmelt a function of wind speed and air temperature per (Hasebe and Kumekawa, 1995). Because snow depth is only available at a daily resolution through Meteostat, it is linearly interpolated from daily to hourly frequency. As the underlying data is daily, the catchment is not segmented into flow distance regions for snowmelt estimates and these are instead aggregated across the entire catchment.

3.2.4. Model training

Here we provide a brief summary of the rainfall-runoff method we use to take historical sensor measurements and public weather data and automatically discover hydrologic forecast models. For details on the original, single-input algorithm, please consult (Dantzer, 2023a).

The *modpods* model has two components: unit hydrographs describing the delay between rainfall or snowmelt and the impact on the sensor measurement, and a differential equation that describes the magnitude of those contributions. The unit hydrograph is a classic approach in hydrology (Nash, 1959) that generates streamflow from precipitation by assuming the catchment is a linear and time invariant dynamical system (Singh et al., 2014). The differential equation incorporating these unit hydrographs has the form:

$$\frac{dq_o}{dt} = f(q_o, p_o, T(p_o)) \quad (1)$$

where q_o is the observed water level or discharge, p_o is the estimated rainfall or snowmelt, $\frac{dq_o}{dt}$ is the rate of change in water level or discharge, and $T(p_o)$ is the transformation of p_o corresponding to the unit hydrograph learned for that input.

The differential equations are linear, quadratic, or cubic polynomials without interaction terms. The coefficients on p_o and $T(p_o)$ terms are constrained to be positive. The highest order autocorrelation (e.g., q_o^3 in a cubic polynomial) is constrained to be negative to ensure the dynamical system is bounded-input, bounded-output (BIBO) stable (Hespanha, 2009). Lower-order autocorrelations are permitted to be positive so that any fixed depth or baseflow can be correctly modeled as a nonzero equilibrium of the system. For example, if a quadratic model has $\frac{dq_o}{dt} = +a_0q_o - a_1q_o^2 + \dots$ there will be a stable equilibrium at $q_o^* = \frac{a_0}{a_1}$. As an example, one of the model configurations used in this study is a linear differential equation without snow, which has the form:

$$\begin{aligned} \frac{dq_o}{dt} = & -a_0q_o + b_0p_{rain,close} + b_1T_{r,c}(p_{rain,close}) \\ & + b_2p_{rain,mid} + b_3T_{r,m}(p_{rain,mid}) \\ & + b_4p_{rain,far} + b_5T_{r,f}(p_{rain,far}) \end{aligned} \quad (2)$$

where $T()$ are transformations of the observed forcing corresponding to the unit hydrographs learned for that input, and a_i, b_i are learned coefficients. The models are trained in two loops. The inner loop optimizes the coefficients of the differential equation for a given set of unit hydrographs. The outer loop changes the shapes of the unit hydrographs to increase the training accuracy of the differential equation as measured by the coefficient of determination (R^2).

These models have between 6 and 48 parameters describing the coefficients of the differential equation and the shapes of the unit hydrographs. This parsimony makes training cheap, reduces the length of data record needed, and increases tolerance of noise. The representation as differential equations and unit hydrographs grounds the method in basic concepts of hydrology. Our prior work (Dantzer, 2023a) and this analysis (SI Figures 3, 6, 10, and 13) indicate a lack of the overfitting characteristic of many machine learning approaches. As overfitting is not a concern, no regularization or ensembling is performed.

In this work we have extended the approach in Dantzer (2023a) to accommodate multiple inputs, making it more appropriate for application in larger watersheds where spatial rainfall variability may be important. The *modpods* engine is implemented in python and can be deployed as a microservice (e.g., AWS lambda) or scheduled script on a small server or consumer laptop.

3.3. Predictions and visualizations

For the open-source sensor network, all trained models are loaded once per day and given historical and forecast weather with which to make a one-week prediction. A validation prediction over the past week is used to quantify the uncertainty of the prediction and generate bounds around the central estimate corresponding to the mean and

maximum absolute percentage error over the validation interval. The predictions and weather data are then pushed to the server and fetched at the visualization endpoints (Fig. 2). To see example live dashboards, please visit <https://www.digitalwaterlab.org/mdot>.

The visualization interface built on Grafana (Fig. 2) is the primary way users interact with measurements and predictions. Grafana is an open-source analytics and monitoring platform, known for its ability to visualize and explore metrics from various data sources in a customizable dashboard format. It is widely used for tracking and visualizing time series data, such as performance metrics and IoT sensor data. Grafana's hosted services (Grafana.com) offer a user-friendly solution, eliminating the need for installation and setup. This cloud-based approach simplifies the use of Grafana, allowing users to focus on data analysis and visualization without the complexities of managing server infrastructure. Grafana also features a built-in InfluxDB data source template which makes the generation of dashboards relatively quick and easy. Alerts can also be easily configured and delivered via channels including email and SMS.

3.4. Implementation and evaluation

The complete implementation of the *modpods* prediction toolchain, code to generate the figures in this paper, and pictures of sensor nodes from the Digital Water Lab network are freely shared (Dantzer, 2023c). Some of the resources concerning the sewer system are redacted or excluded for privacy and infrastructure security protection. That library depends on *modpods*, which we also share freely (Dantzer, 2023b). The predictions are served from a laptop with 32 GB RAM and an Intel(R) Core(TM) i7-1065G7 CPU @1.30 GHz 1.50 GHz processor. Resources concerning the sensor network are available at <https://www.digitalwaterlab.org/build>.

To evaluate the accuracy that can be expected in the long term, we use records of stage (water surface elevation) and discharge at 118 USGS streamgaging stations. It would be desirable to evaluate the long-term prediction accuracy of the method on the same low-cost sensor network in which it is currently implemented. However, accuracy evaluations of hydrologic models typically occur over *at least* one full year of data, while many of the sensors in the open-source sensor network to which we are serving predictions have only been deployed for several months. The shortest training record is 58 days, while the median is 455 days and the maximum is 1029 days. In light of this, our evaluation of the architecture on the open-source sensor network will be in computational expense (Table 4), training accuracy (Fig. 3), and network reliability.

Another point of evaluation is how sensitive the method is to noise in the data record and errors in the catchment delineation. Combined sewer systems provide a natural laboratory for these questions because their contributing areas are far more difficult to delineate than surface waters and dry weather sewage flows constitute significant “noise” when attempting to learn precipitation-driven dynamics. We use flow and depth data provided by a partner utility at 19 locations within their network. To align results with a highly automated use-case, data preprocessing is minimal in all evaluations.

To better understand the impact of inaccuracies in rainfall on model performance, we augment the forcing data retrieved by the software tool with additional rainfall data for both of the historical datasets. For the USGS dataset this is precipitation measured at the gaging station, which approximates catchment-wide precipitation in small catchments. For the sewer dataset, the extra rainfall data comes from two or three rainfall gages identified by the utility as being relevant to the flow and depth sensors.

We use Nash–Sutcliffe Efficiency (NSE) as our measure of performance throughout as it is dimensionless, correlates well to visual fit, and is a common error metric in hydrology. NSE is one minus the



Fig. 2. Sample decision support dashboard. Estimated snowmelt and rainfall (across all flow-distance regions) are indicated on top of the graph in purple and blue respectively. Historical water surface elevation is in green. Predictions with uncertainty estimates are provided in yellow.

ratio of the error variance of the model divided by the variance of the observed time series:

$$NSE = 1 - \frac{\sum (Q_{obs} - Q_{mod})^2}{\sum (Q_{obs} - \bar{Q}_{obs})^2} \quad (3)$$

A perfect model scores 1 while a model with error equal to the variance of the observed time series scores 0. Of the three model configurations trained (linear, quadratic, and cubic), the configuration with the best NSE when simulating over the training record for a site is chosen as the “final” model. Each model configuration is individually optimized for R^2 on predicting the derivative, *then* training NSE between the predicted and measured depth/discharge is used to select the final model configuration. That final model’s NSE score on the evaluation dataset is what is shown in Figs. 4 and 5, as well as the SI Figures. Though the equations for R^2 and NSE are the same, the distinction is made to clarify when training fit on derivative (R^2) or simulation accuracy on measurement (NSE) are being discussed. Root mean square error (RMSE) and mean absolute error (MAE) are also presented (Tables 3, 5, and 7) to further contextualize results.

3.4.1. Evaluating the complete toolchain

Training accuracy is evaluated for 68 sensors within the open-source network by forward-simulating the models given the initial measurement and comparing to the data on which the model was trained. Missing sensor measurements are linearly interpolated while missing precipitation measurements are filled with zeros. The data is then resampled to hourly resolution. Additionally, any constant offset in stage or discharge is removed by subtracting the second percentile from the entire record. To ensure the system is causal, forcing data is shifted back one timestep relative to the response. If this is not done, then responses may occur in the same timestep as their forcing for small catchments. No other preprocessing is performed. The preprocessing for the USGS experiment is the same. Because the forecasting step is fast, computational expense is evaluated by training time. Network results are presented in terms of up-time and financial efficiency.

3.4.2. Evaluating long-term accuracy in surface water systems

We apply the prediction engine to stage, discharge, and precipitation observations from 118 USGS streamgaging stations at a 15-minute timestep. The data is resampled to hourly frequency to match the finest frequency available from the weather database (Lamprecht, 0000). The locations span the continental United States are summarized in SI Figures 1, 4, 8, and 11. Models are trained for between 3 and 12 years and evaluated over one year. One year of rainfall and snowmelt is used to wind up the models. The evaluation year always comes last. All data falls between June 1, 2008 and June 1, 2023.

3.4.3. Evaluating sensitivity to noise and delineation errors

We also evaluate models trained on depth and discharge measurements from 19 sensors in a combined sewer. Combined sewers provide a fundamentally harder case for two primary reasons: (1) contributing area is far less directly related to topography than for surface waters, and (2) dry-weather flows significantly obscure the precipitation response signal in the discharge. We included this dataset not only to demonstrate utility to use cases outside of surface water gages, but also to “stress test” the approach with more difficult catchment delineations and noisier data.

One year of hourly data from January 1, 2021 to December 31, 2021 are retrieved from the Supervisory Control and Data Acquisition (SCADA) system. One month (January) is used for windup, six months (February through July) are used for training, and five months (August through December) are used for evaluation. As in the previous experiment, missing stage and discharge measurements are linearly interpolated while missing precipitation measurements are filled with zeros. Any constant offset in stage or discharge is removed by subtracting the tenth percentile from the entire record. A larger percentile is subtracted from the data because it is “noisier” due to dry-weather sewage flows. To remove the constant offset in perfectly clean data, one would subtract the minimum. As before, the forcing data is shifted back one timestep relative to the response. No other preprocessing is performed.

3.4.4. Effect of additional rainfall data

In the USGS experiment, models are trained using only the meteorological weather data as well as using both the meteorological weather data and the rain gage at the streamgaging station. In the sewer experiment there are three configurations: meteorological only, rain gages only, and meteorological and rain gages.

4. Results and discussion

4.1. Network reliability and performance

While not the focus of this paper, we give a summary of the open source sensor network deployment. This deployment was achieved rapidly, as forty sensors were built and deployed by the first author and two undergraduate students in the Summer of 2021. The hydrologic sensor network and data services were energy efficient and reliable. Sensor nodes consumed only 50 μ A in their sleep state. The cellular modem's transmission, operating on a slow power cycle, was able to activate within a minute. During transmission the current consumption averaged 200 mA with a peak of 2 A. The network's sustainability was enhanced by solar power. With low power consumption, a measurement cycle of 10 min, and transmission cycle of 60 min, the devices were able to operate through a Michigan winter and recharge the lithium ion battery consistently. Transmissions were reliable (95% packet throughput) and aided by a buffering system in the devices that allowed for data to be sent later if connection failed during a given transmission attempt. There were a few network outages due to an issue with the domain service provider, but not the sensors or data services themselves. Maintenance of the devices was infrequent and generally only required because of physical damage to the node or sensor obstructions. The major challenge with the ultrasonic sensor data was noise caused by physical obstructions (e.g., plants and bridge decks) that blocked the sensor from making a measurement of water levels. The automated quality control system (Schmidt and Kerkez, 2023) allowed these issues to be detected and addressed by trimming plants or adjusting the sensor's location on the bridge. At the time of writing, the complete wireless sensor package (sensors, hardware, batteries, enclosures, etc.) could be built for under 1500 USD (parts and labor). We estimate the entire 100+ sensor network to cost less than 100k USD per year to maintain.

4.2. Low-cost sensor network training accuracy

The left side of Fig. 3 shows the cumulative density function of training Nash Sutcliffe Efficiency. The line indicates the percentile of a given NSE score, such that better overall performance lies in the bottom right, while worse performance goes to the top left. As numeric score metrics are always incomplete, the right side of the figure shows selected simulations to contextualize these scores.

Training records vary from several years (25th percentile simulation, Fig. 3) to under a year (maximum) and from relatively clean (maximum) to obstructed and noisy (median). It seems the models are sufficiently complex to capture rainfall-runoff processes when sensor data is relatively clean and weather data is accurate (maximum). The prediction engine is also resilient to high noise levels, still finding a reasonable representation of the dynamics even when the observations are corrupted with measurement noise (median). Models also behave predictably due to their first-principles-based construction. Per correspondence and conversation with Michigan Department of Transportation (MDOT) bridge inspectors, prediction accuracy (Fig. 2) is sufficient at most sites to support their operational decision-making of whether to plan to inspect or close a bridge during or after a storm. During one particularly severe storm in the upper peninsula of Michigan, a bridge inspector saw the predicted rise in river level and drove out to temporarily remove the sensor node from the bridge to prevent it from being struck by woody debris.

Table 3

Error metrics for training simulations on the open-source sensor network corresponding to Fig. 3.

	NSE	MAE (m)	RMSE (m)
min	-1.41	0.02	0.04
25%	-0.02	0.07	0.09
median	0.25	0.12	0.18
75%	0.41	0.18	0.24
max	0.87	0.71	0.93

Table 3 gives further context for the results in Fig. 3 by providing dimensional error metrics MAE and RMSE. Simulation accuracy on the training data is generally good, as mean absolute error remains within 12 centimeters for the majority of sites. Some sites have very good fits, as the minimum mean absolute error is only 2 centimeters, less than an inch.

4.3. Computational expense

Table 4 characterizes the training times for the models trained for predictions on open-source sensors, as well as the USGS and sewer experiments. The number of data points in the training record strongly determines training expense. The USGS experiment used 3–12 years of hourly data which is between 26 and 105 thousand points. The sewer experiment used six months of hourly data to train, which is about 4400 data points. The Digital Water Lab models trained on between 2 and 34 months of hourly data, which is 1500 to 24,800 data points.

The dimensionality of the models also varied. Some of the USGS streamgaging stations had contributing areas small enough and climate warm enough that only one liquid precipitation input was used for forcing. In contrast, sewer experiments including both meteorological and rain gage data had 7 or 8 forcing variables. Computational expense was generally greatest in the USGS experiment because of the length of the record. Though dimensionality was higher in the sewer experiment, the short record resulted in quicker training.

Once the models were trained for the Digital Water Lab sensors, running the one-week prediction took about thirty seconds per site on a consumer laptop with 32 GB RAM and an Intel(R) Core(TM) i7-1065G7 CPU @ 1.30 GHz 1.50 GHz processor. The computational expense for training and predicting on the open-source sensor nodes is minimal and suggests good scalability and cost profiles.

4.4. Long-term accuracy in surface water systems

The left side of Figs. 4 and 5 show the cumulative density function of evaluation Nash Sutcliffe Efficiency, while the right side shows selected simulations to contextualize these scores.

The maximum accuracy simulation shows a good prediction of the discharge, demonstrating that model complexity is sufficient for at least some catchments. The median and 25th percentile simulations suggest much of the error may be due to inaccuracies in weather data. In the median simulation, rainfall and snowmelt seem to be over-reported in August of 2022 and under-reported in the Spring of 2023. When forcing magnitude is correct, the model seems to model the dynamics accurately (small storm in December of 2022). However, this error could also stem from insufficient representation of soil moisture dynamics. Most of the error in the 25th percentile simulation occurs when rainfall data indicates a storm within the catchment, but the discharge record does not corroborate this (August–October 2022). Additional results for this experiment are presented in SI Figures 1 through 14. These include plots similar to Fig. 4 for stage and discharge with and without raingages, site maps, and the returns to accuracy from increased training period length. More error metrics characterizing this experiment are presented in Table 5.

We found no correlation ($R^2 = 0.64\%$) between the training period length and the evaluation NSE score (SI Figure 3). This suggests

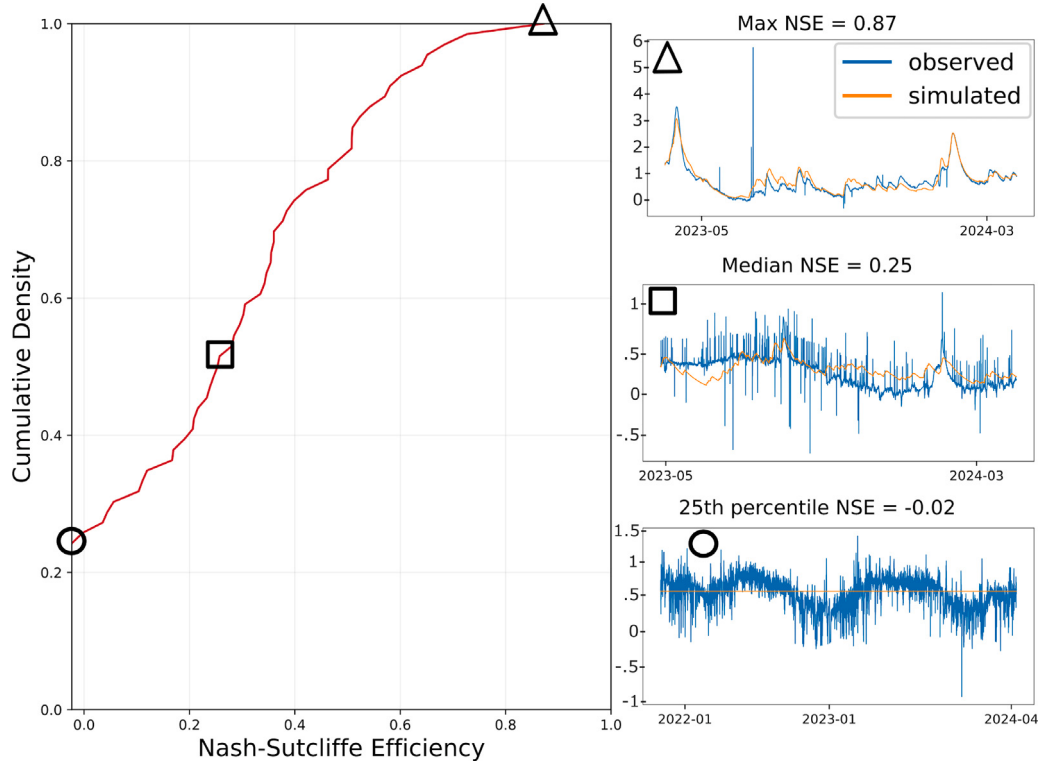


Fig. 3. Training water level models across a low-cost wireless sensor network. Training accuracy is shown for 68 locations within the wireless sensor network. The cumulative probability distribution of Nash Sutcliffe Efficiency is shown on the left. On the right are measured and simulated stage in meters for sites with training accuracy at the 25th percentile, median, and maximum of NSE (-0.02 , 0.25 , and 0.87 respectively).

Table 4

Computational expense of training models [hours/sensor].

	First Quartile	Median	Third Quartile	Data Points (thousands)	Max Inputs
USGS	1.51	3.99	9.28	26–105	6
Combined Sewer	0.14	0.26	0.47	4.4	8
Digital Water Lab	0.24	0.69	1.45	1.5–24.8	5

Table 5

Error metrics for evaluation simulations on USGS flow stations corresponding to Fig. 4.

	NSE	MAE (m^3/s)	RMSE (m^3/s)
min	-26.8	0.05	0.26
25%	-0.76	1.19	2.09
median	-0.13	3.71	7.35
75%	0.15	10.68	17.11
max	0.78	167.1	261.35

models are quickly saturated, supporting the desire for a data-frugal modeling approach. We believe this is due to the small number of parameters (always less than 50) in these models. This result contrasts with (Boughton, 2007) where datasets shorter than five years are associated with considerably larger errors than longer records.

We are not aware of a benchmark toolchain providing the same functionality to the system described in this work: receiving only measurements and location and automatically creating and deploying a hydrologic prediction model. Despite the lack of close analogs, we present performance of some representative hydrologic prediction models in Table 6, while noting the substantial differences that make a comparative analysis of accuracy difficult. A comprehensive and direct accuracy comparison with state-of-the-art approaches is provided in Dantzer (2023a). However, that study details a single-input, single-output model without the automated weather data retrieval service.

Table 6 shows performance tends to be better under greater aggregation of the time series and shorter prediction windows. Typically, daily data is fit more easily than hourly as the aggregation throughout time reduces the variance in the time series. Despite this, the method proposed here achieves better performance than the models predicting daily flows evaluated in Fisaha Unduche and Zhu (2018). Shorter prediction windows tend to have better error metrics than longer windows because error accumulates throughout a simulation. This is provided the time series does not vary so little that carrying forward the last measurement (persistence) is an accurate estimator. The accumulation of error over a simulation can be substantial, as in Demir et al. (2022) where NSE scores decline from 0.97 at 6 h to just 0.6 when predicting 5 days out. The importance of accurate weather data is visible in the 25th percentile simulation of Fig. 4 where several storms which do not actually fall on the catchment (as judged by the discharge measurements) are reported as falling within the catchment using historical precipitation records. The accuracy limitations imposed by imperfect knowledge of historical precipitation and the difficulty of accurately forecasting weather are also noted in Krajewski et al. (2017).

4.5. Sensitivity to noise and delineation errors

Fig. 5 shows that maximum accuracy is similar to the surface water case, indicating this approach has potential to be applied to combined and separated sewer systems. The noise tolerance is visible as predictions cut through the oscillations induced by dry-weather flows.

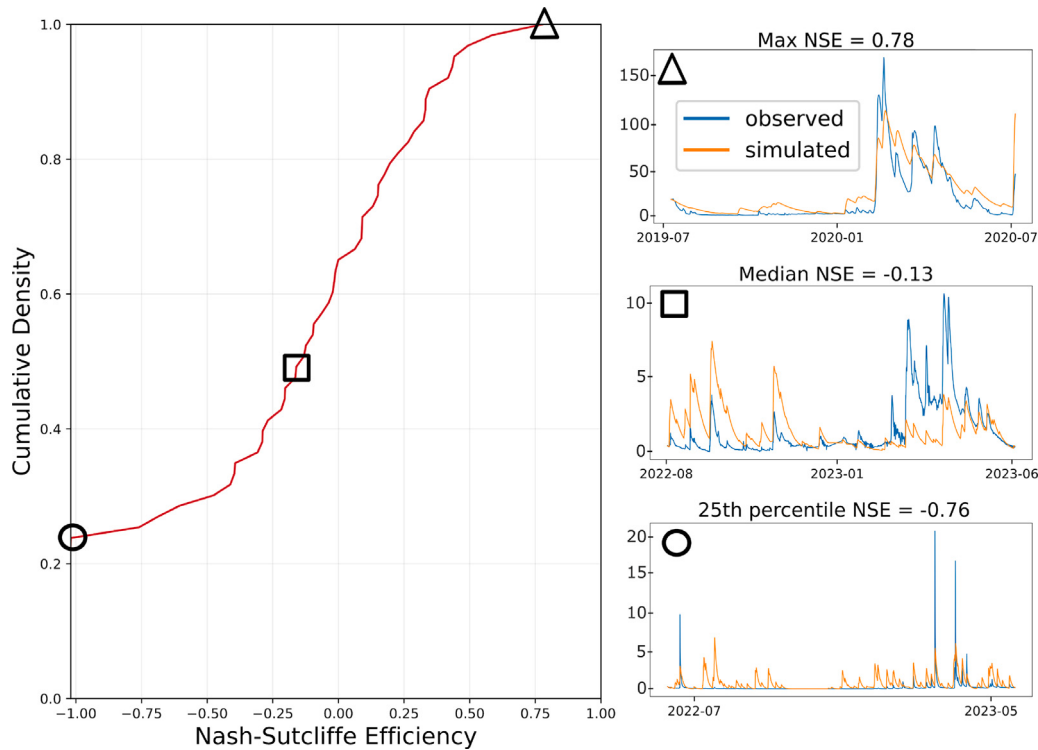


Fig. 4. Predicting discharge at federal streamgages. Models predicting discharge without using station rain gages are evaluated at 118 streamgaging stations. The cumulative density function of Nash Sutcliffe Efficiency is shown on the left. On the right are evaluation simulations of discharge in cubic meters per second for models at the first quartile, median, and maximum of NSE (-0.76 , -0.13 , and 0.78 respectively).

Table 6
Comparative accuracy of hydrologic prediction models.

Citation	Frequency	Prediction Horizon	Number of parameters	Source of weather data	Max NSE	Median NSE
Demir et al. (2022) Table 4, Fig. 5 - Ridge Regression	Hourly	5 days	Not provided	Manually curated	0.9	0.6
This work, Fig. 4	Hourly	1 year	6 to 48	Automatically derived from location	0.78	-0.13
Fisaha Unduche and Zhu (2018) Table 5 - Performance across four process-based models (WATFLOOD, HSPF, HBV-EC, HEC-HMS)	Daily	10 years	Varying	Manually curated	0.48	-0.52
Kratzert et al. (2018) Entity-Aware Long-Short Term memory network	Daily	10 years	12,936	Manually curated	0.97	0.74

Because these models are time invariant and not driven with dry-weather flow data they are not designed to represent these time-driven dynamics in the data and thus ignore them. This stands in contrast to a machine-learning approach, which would have sufficient parameters to attempt to fit any pattern which appears in the data. Here that would be undesirable.

The median simulation shows consistent performance despite an order of magnitude difference in the scale of flows. The model even provides reasonable dynamics to fill the gap in measurement when a sensor error occurs during a large storm. The 25th percentile simulation shows a sensor where flows are primarily dry-weather and dynamics do not seem to be precipitation-driven. SI Figures 15 through 22 show performance under varying targets (flow or depth) and forcing data.

Table 7 shows the MAE and RMSE for the evaluation simulations of the prediction models. Though performance is generally good, there is an outlier in the maximum error simulation. This is due to the model erroneously learning a very large baseflow.

Echoing the results of the previous section, it seems that models quickly converge to reasonable approximations of the rainfall-runoff dynamics in a catchment. The models shown in Fig. 5 have been trained on only six months of data comprising about a dozen storms. Reasonable performance with a very limited training record stands in contrast

Table 7
Error metrics for evaluation simulations on combined sewer flow measurements corresponding to Fig. 5.

	NSE	MAE (m^3/s)	RMSE (m^3/s)
min	$-1\text{E}12$	0.001	0.003
25%	-1.48	0.04	0.074
median	0.07	0.36	0.72
75%	0.48	2.78	4.54
max	0.72	18 464.16	51 054.97

to popular machine-learning approaches (Boughton, 2007) prone to overfitting or failing to converge to a reasonable representation of the dynamics when data is not plentiful.

Though our principal goal in evaluating the method on a combined sewer system was to assess sensitivity to noisy data and errors in catchment delineation, the performance suggests utility in supporting predictive control of urban drainage systems. Estimating future depths and flows from rainfall forecasts (input modeling) is a primary difficulty of applying model predictive control (Lund et al., 2018). This toolchain

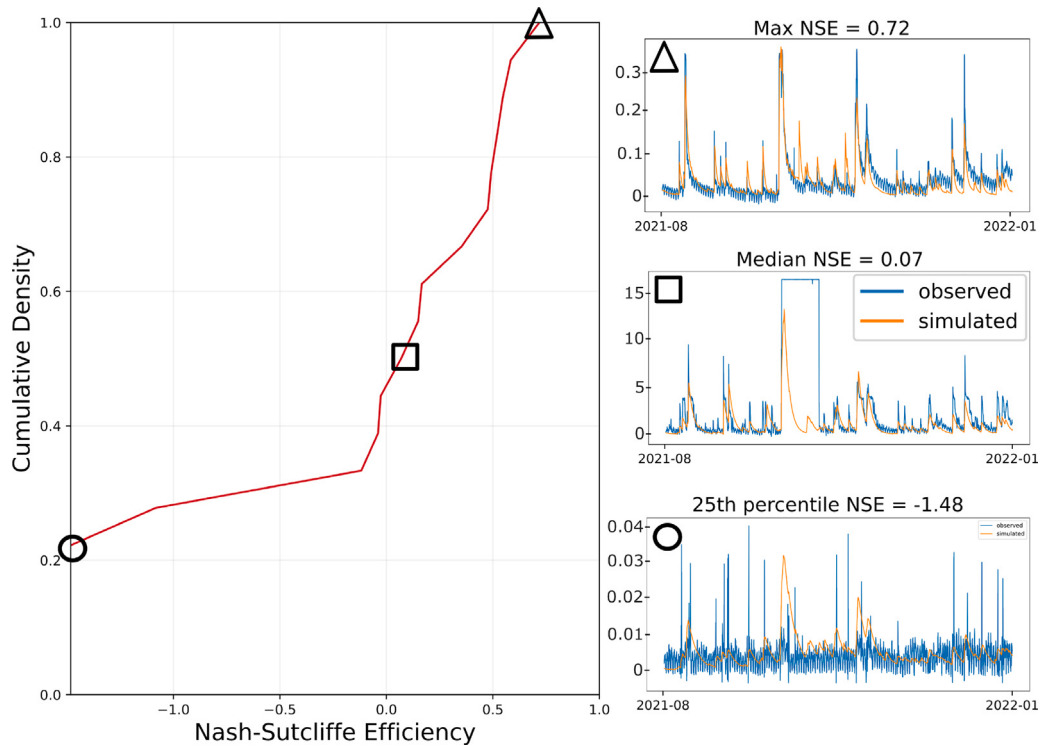


Fig. 5. Predicting discharge in a combined sewer. Models are evaluated in predicting flow using meteostat and raingage data at 19 sensor locations within the combined sewer system. The cumulative density function of Nash Sutcliffe Efficiency is shown on the left. On the right are evaluation simulations of discharge in cubic meters per second for models at the first quartile, median, and maximum of NSE (-1.48 , 0.02 , and 0.72 respectively).

presents an automated and computationally efficient method for predicting depths and flows at sensors upstream of control assets in an urban drainage system.

4.6. Effect of additional rainfall and potential evapotranspiration data

We found that including additional rain gages had good returns to accuracy in the aggregate, but decreased accuracy in some cases (SI Figures 7, 14, 18, and 22). Fig. 4 does not use raingage data while Fig. 5 does. Even with these rain gages included, we suspect inaccuracies in weather data still constitute a large part of the remaining error as suggested in Krajewski et al. (2017). Error could also be due to other factors including inaccuracy in the catchment delineation, insufficient complexity in the model structure, or the need for more forcing variables such as potential evapotranspiration (PET). We did evaluate the inclusion of PET as a forcing variable and found the performance improvement marginal. The marginal utility of PET to prediction accuracy may be because much of the climate's impact on typical soil moisture levels is already captured in the nonlinear recession rates. For example, ephemeral streams in arid areas may have typically quicker recession rates than large streams in humid areas with significant baseflow. This omission does however fail to represent seasonal variations in soil moisture content which can lead to significantly different storm responses (e.g., dry soil in late summer opposed to wet soils early in the spring). Though we have found it to have marginal predictive utility thus far, potential evapotranspiration may improve results if incorporated in a different formulation. For example, the model could be reformulated to have two outputs or states (discharge and soil moisture) which also affect each other. A similar idea is explored in Kirchner (2009). The automated method for sourcing weather data given only the sensor's location should be sufficient for many applications, and external data can be easily pulled into the model if it is found to improve performance.

5. Conclusions

We have presented an end-to-end toolchain for hydrologic forecasting, sensors included. Though the maximum accuracy results are encouraging, consistency could be improved. This may be done by using a finer digital elevation model at greater computational expense for the initial delineation. This would not severely increase the computational burden as the delineation only needs to occur once for each site, not every day. Better parameterizing snowmelt with a more nuanced process description could also yield better and more consistent accuracy. More accurate weather data is also likely to improve accuracy. The spatially compact, severe, and unpredictable thunderstorms characteristic of many temperate climates are particularly confounding. Upstream river level measurements could also be included as additional forcing to the model. This may reduce vulnerabilities to unobservability and uncertainty in weather data, but could also over-parameterize the model as upstream river levels are conceptually included in the weather forcing transformation terms in Eqs. (1) and (2).

We are currently serving daily predictions to various users including bridge inspectors, utilities, and conservation agencies. Usability studies on the interface (Fig. 2) could assess and improve the decision-support value provided by the network and forecasts. Lastly, most low-cost sensors focus on water level, while stakeholders are often concerned with discharge. Finding an automated and scalable approach for estimating discharge rating curves could enhance this work by providing discharge predictions wherever low-cost water level sensors can be deployed.

Existing prediction engines for large sensor networks have shortcomings in their accessibility, automaticity, and data-frugality. These barriers prevent their extension to the low-resource communities most adversely impacted by floods. We have presented a new software tool that automatically generates computationally cheap rainfall-runoff models given only a sensor's measurement record and location. Performance on surface waters and sewers compares favorably to benchmarks and suggests this tool could be a valuable asset to support operational

decision making. We also detail enhancements to an existing, open-source framework for wireless sensor networks. This work has the potential to enhance equity in flood protection by lowering barriers in the financial expense and modeling expertise required to deploy predictive wireless sensor networks.

CRediT authorship contribution statement

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

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Most data and all software is available. Some access restricted for privacy/security reasons.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.envsoft.2024.106137>.

References

- Aven, T., 2010. On how to define, understand and describe risk. *Reliab. Eng. Syst. Saf.* <http://dx.doi.org/10.1016/j.res.2010.01.011>.
- Bartoletti, N., Casagli, F., Marsili-Libelli, S., A. Nardi, L.P., 2018. Data-driven rainfall/runoff modelling based on a neuro-fuzzy inference system. *Environ. Model. Softw.* <http://dx.doi.org/10.1016/j.envsoft.2017.11.026>.
- Bartos, M., 0000. Pysheds: Simple and fast watershed delineation in python, URL <https://github.com/mbartos/pysheds>.
- Bartos, M., Wong, B., Kerkez, B., 2018. Open storm: A complete framework for sensing and control of urban watersheds. *Environ. Sci.: Water Res. Technol.*
- Bellini, P., Nesi, P., Pantaleo, G., 2022. IoT-enabled smart cities: A review of concepts, frameworks and key technologies. *Appl. Sci.* <http://dx.doi.org/10.3390/app12031607>.
- Boughton, W.C., 2007. Effect of data length on rainfall-runoff modelling. *Environ. Model. Softw.* <http://dx.doi.org/10.1016/j.envsoft.2006.01.001>.
- Bubola, E., 2023. Catastrophic congo flooding kills more than 400 people. *New York Times* URL.
- Chakraborty, M., Kundan, A.P., 2021. *Monitoring Cloud-Native Applications*. Springer, URL https://link.springer.com/chapter/10.1007/978-1-4842-6888-9_6.
- Choi, B., 2021. *Python Network Automation Labs: Ansible, pyATS, Docker, and the Twilio API*. Springer, URL https://link.springer.com/chapter/10.1007/978-1-4842-6806-3_16.
- Colbeck, S.C., 1988. *Snowmelt Increase Through Albedo Reduction*. Technical Report, Cold Regions Research and Engineering Lab Hanover NH.
- Currie, E.H., 2021. *PSoc Creator*. Springer, URL https://link.springer.com/chapter/10.1007/978-3-030-70312-7_7.
- Dantzer, T.A., 2023a. Automatic rainfall runoff [software]. <https://zenodo.org/badge/latestdoi/590232346>.
- Dantzer, T.A., 2023b. Model discovery in partially observable dynamical systems [software]. <https://zenodo.org/badge/latestdoi/636838701>.
- Dantzer, T.A., 2023c. Rainfall-runoff-anywhere [software]. https://github.com/dantzer/rainfall_runoff_anywhere_public.
- Dazzi, S., Vacondio, R., Mignosa, P., 2021. Flood stage forecasting using machine-learning methods: A case study on the Parma River (Italy). *Water* <http://dx.doi.org/10.3390/w13121612>.
- Demir, I., Xiang, Z., Demiray, B., Sit, M., 2022. WaterBench-iowa: A large-scale benchmark dataset for data-driven streamflow forecasting. *Earth Syst. Sci. Data* 14, 5605–5616. <http://dx.doi.org/10.5194/essd-14-5605-2022>.
- Fathian, F., Mehdizadeh, S., Sales, A.K., Safari, M.J.S., 2019. Hybrid models to improve the monthly river flow prediction: Integrating artificial intelligence and non-linear time series models. *J. Hydrol.* <http://dx.doi.org/10.1016/j.jhydrol.2019.06.025>.
- Feurer, M., Eggenberger, K., Falkner, S., Lindauer, M., Hutter, F., 2022. Auto-sklearn 2.0: Hands-free automl via meta-learning. *J. Mach. Learn. Res.* URL <https://www.jmlr.org/papers/volume23/21-0992/21-0992.pdf>.
- Fisaha Unduche, D.S., Zhu, E., 2018. Evaluation of four hydrological models for operational flood forecasting in a Canadian prairie watershed. *Hydrol. Sci. J.* 63 (8), 1133–1149. <http://dx.doi.org/10.1080/02626667.2018.1474219>, arXiv:10.1080/02626667.2018.1474219.
- Gadzinski, G., Castello, A., 2020. Fast and frugal heuristics augmented: When machine learning quantifies Bayesian uncertainty. *J. Behav. Exp. Finance* 26, 100293. <http://dx.doi.org/10.1016/j.jbef.2020.100293>, URL <https://www.sciencedirect.com/science/article/pii/S2214635019302357>.
- Hasebe, M., Kumekawa, T., 1995. Estimation of snowmelt volume using air temperature and wind speed. *Environ. Int.* [http://dx.doi.org/10.1016/0160-4120\(95\)00048-P](http://dx.doi.org/10.1016/0160-4120(95)00048-P).
- Hespanha, J.P., 2009. *Linear Systems Theory*. Princeton University Press.
- Jonkman, S.N., Curran, A., Bouwer, L.M., 2024. Floods have become less deadly: an analysis of global flood fatalities 1975–2022. *Nat. Hazards* 120, 6327–6342.
- Kalyanapu, A., Owusu, C., Wright, T., Datta, T., 2023. Low-cost real-time water level monitoring network for falling water river watershed: A case study. *Geosciences* <http://dx.doi.org/10.3390/geosciences13030065>.
- Kapoor, A., Pathiraja, S., Marshall, L., Chandra, R., 2023. DeepGR4J: A deep learning hybridization approach for conceptual rainfall-runoff modelling. *Environ. Model. Softw.* <http://dx.doi.org/10.1016/j.envsoft.2023.105831>.
- Karegar, M.A., Kusche, J., Geremia-Nievinski, F., Larson, K.M., 2022. Raspberry pi reflector (RPR): A low-cost water-level monitoring system based on GNSS interferometric reflectometry. *Water Resour. Res.* 58 (12), <http://dx.doi.org/10.1029/2021WR031713>, e2021WR031713.
- Kirchner, J.W., 2009. Catchments as simple dynamical systems: Catchment characterization, rainfall-runoff modeling, and doing hydrology backward. *Water Resour. Res.* 45 (2), W02429. <http://dx.doi.org/10.1029/2008WR006912>.
- Krajewski, W.F., Ceynar, D., Demir, I., Goska, R., Kruger, A., Carmen Langel, R.M., Niemeier, J., Quintero, F., Seo, B.C., Small, S.J., Weber, L.J., Young, N.C., 2017. Real-time flood forecasting and information system for the state of iowa. *Am. Meteorol. Soc.* <http://dx.doi.org/10.1175/BAMS-D-15-00243.1>.
- Kratzert, F., Klotz, D., Brenner, C., Schulz, K., Herrnegger, M., 2018. Rainfall-runoff modelling using long short-term memory (LSTM) networks [Dataset]. *Hydrol. Earth Syst. Sci.* <http://dx.doi.org/10.5194/hess-22-6005-2018>.
- Krishnan, S., Wu, E., 2019. AlphaClean: Automatic generation of data cleaning pipelines. *arXiv:1904.11827*.
- Kruger, A., Krajewski, W.F., Niemeier, J.J., Ceynar, D.L., Goska, R., 2016. Bridge-mounted river stage sensors (BMRS). *IEEE Access* 4, 8948–8966. <http://dx.doi.org/10.1109/ACCESS.2016.2631172>.
- Lamprecht, C.S., 0000. Meteostat, URL <https://meteostat.net/>.
- Lehner, B., Verdin, K., Jarvis, A., 2011. New global hydrography derived from spaceborne elevation data. In: *Eos, Transactions American Geophysical Union* Volume 89, Issue 10 P. 93-94. <http://dx.doi.org/10.1029/2008EO100001>.
- Lepore, C., Arnone, E., Noto, L.V., Sivandran, G., Bras, R.L., 2013. Physically based modeling of rainfall-triggered landslides: A case study in the Luquillo forest, Puerto Rico. *Hydrol. Earth Syst. Sci.* <http://dx.doi.org/10.5194/hess-17-3371-2013>.
- Lin, S.S., Zhu, K.Y., Zhang, X.H., Liu, Y.C., Wang, C.Y., 2023. Development of a microservice-based storm sewer simulation system with IoT devices for early warning in urban areas. *Smart Cities* 6 (6), 3411–3426. <http://dx.doi.org/10.3390/smartcities6060151>, URL <https://www.mdpi.com/2624-6511/6/6/151>.
- Lund, N.S.V., Falk, A.K.V., Borup, M., Madsen, H., Mikkelsen, P.S., 2018. Model predictive control of urban drainage systems: A review and perspective towards smart real-time water management. *Crit. Rev. Environ. Sci. Technol.* 48 (3), 279–339. <http://dx.doi.org/10.1080/10643389.2018.1455484>.
- Madushanki, A.A.R., Halgamuge, M.N., Wirasagoda, W.A.H.S., Syed, A., 2019. Adoption of the internet of things (IoT) in agriculture and smart farming towards urban greening: A review. *Int. J. Adv. Comput. Sci. Appl.* URL https://www.researchgate.net/profile/Malka-Halgamuge/publication/332762725_Adoption_of_the_Internet_of_Things_IoT_in_Agriculture_and_Smart_Farming_towards_Urban_Greening_A_Review/links/5cc84bd4458156cd7bc0e27/Adoption-of-the-Internet-of-Things-IoT-in-Agriculture-and-Smart-Farming-towards-Urban-Greening-A-Review.pdf.
- Maidment, D.R., 2016. Conceptual framework for the national flood interoperability experiment. *JAWRA J. Am. Water Resour. Assoc.* 53 (2), 245–257. <http://dx.doi.org/10.1111/1752-1688.12474>.
- Marshall, M., 2023. Libya floods: how climate change intensified the death and devastation. *Nature* <http://dx.doi.org/10.1038/d41586-023-02899-6>.
- Mason, B.E., Schmidt, J., Kerkez, B., 2023. Measuring city-scale green infrastructure drawdown dynamics using internet-connected sensors in detroit. *Environ. Sci.: Water Res. Technol.* 9 (12), 3213–3226.
- Montestruque, L., Lemmon, M., EmNet, L., 2008. Csonet: A metropolitan scale wireless sensor-actuator network. In: *International Workshop on Mobile Device and Urban Sensing*. MODUS.
- Moreno, C., Aquino, R., Ibarreche, J., Pérez, I., Castellanos, E., Álvarez, E., Rentería, R., Anguiano, L., Edwards, A., Lepper, P., Edwards, R.M., Clark, B., 2019. RiverCore: IoT device for river water level monitoring over cellular communications. *Sensors* 19 (1), 127. <http://dx.doi.org/10.3390/s19010127>, URL <https://www.mdpi.com/1424-8220/19/1/127>.

- Nash, J.E., 1959. Systematic determination of unit hydrograph parameters. *J. Geophys. Res.* 64.
- Neal, C., Reynolds, B., Norris, D., Kirchner, J.W., Neal, M., Rowland, P., Wickham, H., Harman, S., Armstrong, L., Sleep, D., Lawlor, A., Woods, C., Williams, B., Fry, M., Newton, G., Wright, D., 2011. Three decades of water quality measurements from the upper severn experimental catchments at Plynlimon, Wales: an openly accessible data resource for research, modelling, environmental management and education. *Hydrol. Process.* URL https://seismo.berkeley.edu/~kirchner/reprints/2011_98_Neal_Plynlimon_weekly_data.pdf.
- NOAA, 0000. NOAA quantitative precipitation forecast, URL <https://www.wpc.ncep.noaa.gov/html/fam2.shtml#qpf>, The NOAA QPF is currently produced on a 20 km X 20 km grid.
- Normand, A.E., 2021. U.S. Geological Survey (USGS) Streamgaging Network: Overview and Issues for Congress. Technical Report, Congressional Research Service, URL <https://crsreports.congress.gov/product/pdf/R/R45695>.
- Pandhiani, S.M., Sihag, P., Shabri, A.B., Singh, B., Pham, Q.B., 2020. Time-series prediction of streamflows of Malaysian rivers using data-driven techniques. *J. Irrig. Drain. Eng.* 146 (7), 04020013. [http://dx.doi.org/10.1061/\(ASCE\)IR.1943-4774.0001463](http://dx.doi.org/10.1061/(ASCE)IR.1943-4774.0001463).
- Pearson, B., Kearns, T., Slawewski, T., Stubbs, B., Herzog, M., Paige, K., Fitch, D., 2019. Making lake erie smart by driving innovations in technology and networking. *Front. Mar. Sci.* 6, 731.
- Schmidt, J.Q., Kerkez, B., 2023. Machine learning-assisted, process-based quality control for detecting compromised environmental sensors. *Environ. Sci. Technol.* <http://dx.doi.org/10.1021/acs.est.3c00360>.
- Shalf, J., 2020. The future of computing beyond Moore's law. *Philos. Trans. A* <http://dx.doi.org/10.1098/rsta.2019.0061>.
- Singh, P.K., Mishra, S.K., Jain, M.K., 2014. A review of the synthetic unit hydrograph: from the empirical UH to advanced geomorphological methods. *Hydrol. Sci. J.* <http://dx.doi.org/10.1080/02626667.2013.870664>.
- Tshimanga, R.M., Tshitenge, J.M., Kabuya, P., D. Alsdorf, G.M., Kibukusa, G., Lukanda, V., 2016. Flood Forecasting. Academic Press, <http://dx.doi.org/10.1016/B978-0-12-801884-2.00004-9>.