Environmental Modelling and Software 134 (2020) 104853

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: http://www.elsevier.com/locate/envsoft

Hydrovise: A non-proprietary open-source software for hydrologic model and data visualization and evaluation

Navid Jadidoleslam *, Radoslaw Goska, Ricardo Mantilla, Witold F. Krajewski

IIHR-Hydroscience & Engineering, The University of Iowa, Iowa City, IA, 52242, USA

ARTICLE INFO

Keywords:
Web application
Data management
Environmental science
Hydrology
Evaluation

ABSTRACT

The authors developed a non-proprietary web-browser based open-source software that allows users to visualize and evaluate hydrologic space-time data in an interactive environment. Hydrovise is client-side browser-based software that interprets a configuration file to construct control elements in the Graphical User Interface for visualizations of space-time data and model simulation evaluations. It leverages the concept of three-dimensional data cubes that facilitate query in space, time, and variable dimension(s) without the requirement for a database system. Using a configuration file, users can define data sources as local file system resources and or external data sources (e.g., online data services). This capability makes Hydrovise a flexible and portable solution where users can share their hydrologic data in an interactive web environment. This paper provides the software description with four distinct example use cases including, but not limited to, time-series data visualization and evaluation, grid-based and river network-based data visualizations.

Software availability

Hydrovise is available under MIT open-source license terms and the source code and examples are available at the following GitHub repository:

- https://github.com/hydrovise/hydrovise

For documentation and tutorials, readers can refer to

- https://github.com/hydrovise/hydrovise/wiki

1. Introduction

The availability of data and computational resources have brought unprecedented opportunities to the hydrologic modeling community for understanding the hydrologic cycle. At the same time, it has introduced new challenges that demand efficient approaches for visualization, communication, and data evaluations. In recent years, there has been significant progress in developing new tools and technologies for visualization, analysis, and handling real-time data (e.g., Wong and Kerkez, 2016; Swain et al., 2016; Brendel et al., 2019). Open-Source Software (OSS) provide unique opportunities for software development and it has potential advantages over their commercial counterparts. An OSS can benefit from developer interest, user contributions/feedback, and frequent release (Ghapanchi et al., 2014) that could further improve the user experience and lead to a successful open-source solution. Midha and Palvia (2012) identify the most critical factors in the success of OSS as user base, language translations, responsibility assignment, and modularity, among others. Vitolo et al. (2015) provide an overview of the current web technologies used for environmental big data. Several OSS solutions used these technologies to develop web applications for water resources purposes (Swain et al., 2015). However, in most cases, solutions have several external dependencies and require expertise or at least adequate knowledge of multiple programming languages/frameworks. These dependencies make solutions fragile, as they complicate the deployment process, and, ultimately, decrease software sustainability. Murugesan and Gangadharan (2012) refer to software "sustainability" by quality attributes of a system such as modifiability, re-usability, and portability.

Swain et al. (2016) developed Tethys as a framework to lower the barrier in developing web applications for environmental data visualization and computations. Tethys uses Python (Python Foundation, 2016) as the primary programming language with its server-side deployment of Django (Django Software Foundation, 2018). It

* Corresponding author.
E-mail address: navid-jadidoleslam@uiowa.edu (N. Jadidoleslam).

https://doi.org/10.1016/j.envsoft.2020.104853
Accepted 24 August 2020
Available online 3 September 2020

leverages other third-party software tools that facilitate computing and visualizations. It is a flexible solution for developing web applications in this framework, but the user/developer requires more than basic knowledge of python and other third-party dependencies. Recently, Brendel et al. (2019) developed SHARKS for retrieval, visualization, and analysis of hydrologic and meteorological data. SHARKS uses the Shiny Apps framework and R programming language as its core programming language. The SHARKS web application allows users to analyze hydrologic data over a user-defined watershed.

Previous software solutions for environmental data have made the deployment of web applications easier for environmental scientists. However, deployment of these solutions could be challenging for a non-expert user because of various dependencies. Furthermore, they still rely on a programming languages that are not designed natively for web development. Therefore, for customized functionalities, users need to have at least a basic knowledge of JavaScript, CSS (Cascading Style Sheets), and HTML (HyperText Markup Language) that are core languages for web application development.

Many previous OSS solutions developed for environmental science depend on database systems to handle the space-time structure of the data. Swain et al. (2015) provide a detailed list of solutions that employ spatial database systems. The decision to use a database in an OSS solution depends on the objective and design of the OSS solution. For example, a database could be useful for complex queries and storing relational data. On the other hand, database may not be needed for hydrologic data visualizations. However, database as a hard requirement for a software solution adds further barrier for a user with limited database management expertise.

Data access using data services have become more popular among environmental data providers facilitating data sharing and access. For example, CUAHSI's (Consortium of Universities for the Advancement of Hydrologic Science) HydroClient provides access to data on an interactive web-based platform for the user-selected region (http://data.cuahsi.org). Data services deployed by NOAA (National Oceanic and Atmospheric Administration) is another example of a web service that allows users access to extensive environmental observations. In recent years, advances in web technologies have added new features facilitating the development of applications from native web programming languages such as JavaScript (Walker and Chapra, 2014). For example, standardization of JavaScript with modern ECMAScript standard (ECMA, 1999) significantly reduced compatibility issues on different browsers.

Our motivation in this work was twofold. First, we aimed to develop a web-based tool that facilitates the visualization and communication of space-time hydrologic data. Second, we intended to significantly lower code barriers for users with minimal web development knowledge. Therefore, we present a non-proprietary open-source software for hydrologic science that we coined Hydrovise. Hydrovise is a client-side application developed by leveraging available web tools and using existing web programming languages. It is configurable and deployable on a personal computer or a web server that does not require a supporting database server. Hydrovise allows users to visualize, evaluate, and share their hydrologic data in an interactive web environment by preparing the data, organizing the data, and preparing a configuration file for their project.

In the next section, we describe the main components of the developed software, data model, and extension modules. Then, we present example use cases common in the hydrologic community. In Section 4, we discuss the advantages and limitations of the current version of the developed software and its potential usages. Finally, we provide a summary and discussion on future developments.

2. Software description

Hydrovise is an open-source browser-based (client-side) software that lowers the code barrier in web-based hydrologic data visualization and evaluations. We selected JavaScript as its core programming language. Hydrovise is designed and developed as an entirely client-side application for increasing the portability of the solution, but it can also be deployed as Server-Client if additional server side functionalities are required. Hydrovise integrates the functionalities of multiple external libraries and internal custom modules. It consists of three main components: Modules, Configuration, and Graphical User Interface (GUI).

Modules are internal/custom libraries, including functions responsible for extension and orchestration of integrated third-party libraries and web tools in Hydrovise. Modules handle data acquisition, visualizations, and user interactions with the GUI. Hydrovise custom libraries control third-party libraries/web tools, and with the use of user-defined project configuration files generate interactive map based reports. We selected tools and libraries that are maintained regularly and are among the most popular open-source JavaScript libraries. Table 1 summarizes the external libraries and their usage in Hydrovise.

The project configuration file is a JavaScript Object Notation (JSON) file that contains a set of information/definitions about data sources (e.g., traces, space-time data, marker locations, and etc.), control elements in the GUI, and the styling of the visualizations.

The Graphical User Interface is a web browser-based interface created dynamically by the initialization module. The content of the GUI depends on the information stored in the project configuration file. It consists of a map, a canvas for visualization of time-series and spatial data, and other control elements that allow navigating data in different dimensions. Fig. 1 shows a schematic of the GUI and structure of the configuration file. The configuration objects are numbered with their corresponding elements on the GUI. Further details on the configuration file is provided in the Hydrovise Documentation.

Fig. 2 shows the Hydrovise workflow diagram. The elements with white and gray background correspond to pre-deployment and post-deployment stages, respectively.

Pre-deployment includes configuration and initialization. The GUI is generated dynamically by the initialization module using the configuration file. In other words, the absence or presence of control elements in the GUI depends on the information provided in the configuration file.

After initialization, user interactions are passed to modules by control elements in the GUI. Modules refer to requested data source using event information that corresponds to a specific element of the data cube (see Section 2.1) and data are visualized on the GUI.

2.1. Data model, structure and types

Hydrovise adopts the concept of data cubes as its data model. Initially, data cubes were introduced by Gray et al. (1997) for reducing the dimensions of data based on user queries. Maidment (2002) introduced the space-time-variable cube for referencing individual data to corresponding attributes. Later, Goodall et al. (2008) implemented this

Table 1
List of external libraries and basic usages in Hydrovise.

JS Library	Usage	Reference
Leaflet	Map and spatial data visualization	https://github.com/Leaflet/Leaflet
Plotly.js	Time-series visualization	https://github.com/plotly/plotly.js
Glify.js	WebGL extension for Leaflet	https://github.com/robertleeplummerjr/Leaflet.glify
jQuery	Creating forms Data acquisition	https://github.com/jquery/jquery
d3.js	Data acquisition	https://github.com/d3/d3
Papaparse.js	Data acquisition	https://github.com/mholt/PapaParse
JSZip	Unzipping data	https://github.com/Stuk/jszip
moment.js	Date and Time format	
math.js	Mathematical operations	https://github.com/josdejong/mathjs
numeric.js	Numerical analysis	https://github.com/sloisel/numeric
chroma.js	Color & styling	https://github.com/gka/chroma.js
toGeoJSON	Spatial data conversion	https://github.com/tmcw/togeojson

	example_config.json
-	{"page":{},
1	"data_part":{},
7	"traces":{},
7	"modTraces":{},
8	"timeSlider":{},
7	"horizontalGrid":{},
6,14	"mapMarkers":{},
2-5	"controls":{},
7,8	"plotlyLayout":{},
11	"mapLayers":{},
6,12,13,15	"map":{},
6	"customBaseMap":{},
9,16	"spatialData":{},
9,10	"timeSelectorLayout":{}}

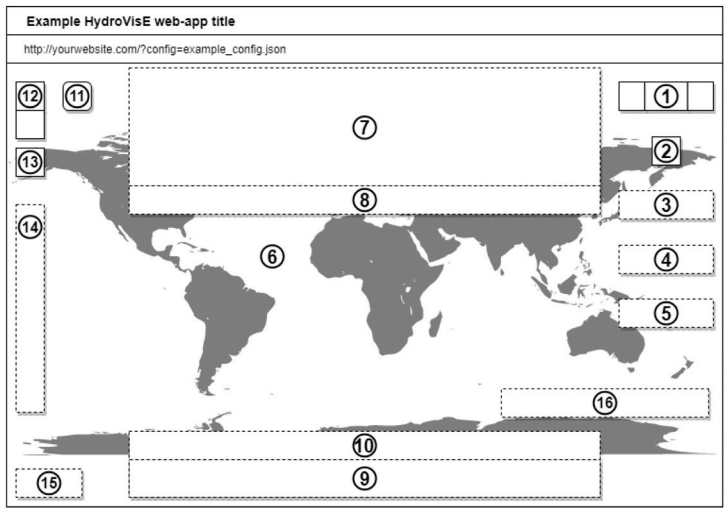

Fig. 1. A schematic of the Hydrovise Graphical User Interface (GUI) and configuration objects corresponding to elements in the GUI.

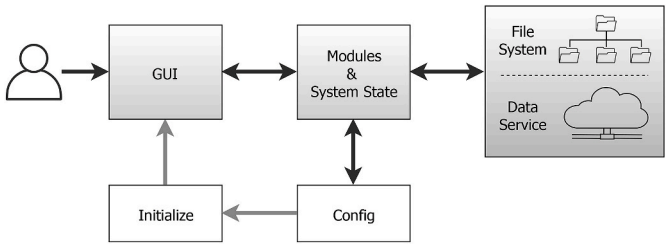

Fig. 2. Workflow diagram for Hydrovise. Gray arrows show the flow for initialization, and black arrows indicate post-deployment flow in Hydrovise.

concept in integrating time-series from different sources for the National Water Information System (NWIS). Fig. 3 shows a schematic of the data cubes where the user can query a chunk of data in a different dimension (s). The atomic element of the data cube can consist of a value, a vector, or a matrix.

Previous uses of the data cube concept were implemented in a database system (e.g., Microsoft SQL, PostgreSQL). This approach is efficient for executing complex queries (Goodall et al., 2008). However, the construction and management of the database, by itself, demands expertise that not every user may have. Hydrovise eliminates the restricting requirement of a database by using flat file system. For a file system implementation, space-time-variable cubes are constructed by organizing the files in folders and sub-folders to facilitate the navigation in selected dimension(s). This organization is defined in the configuration file by the user as a file path template for any given data source. Example file path template in the configuration file has the form

```
"template": {
"var": ["time", "variable", "comID"],
"path_format": "data/{0}{1}{2}.csv"}
```

where curly brackets are substituted with time, variable, and space (comID) that user selects on the GUI. The dynamic path creator uses the path template and user-selected variables to replace the placeholders in the template string and create the path to data source(s). A data source can be a local file, a custom data interface using a database query, or a web service.

Hydrovise can be used to visualize time-series of data associated to any spatial support extent including points, lines, or polygons. For simplicity, we selected the most common data types as the baseline data types. For space-time data (e.g., radar-based precipitation, Demir et al., 2015), users can define the underlying grid geometry and the corresponding timestamps for the data for visualizations. The data can be provided as Comma Separated Value (CSV) or binary files based on a common IDentifier (comID). Table 2 summarizes the data types and formats used for handling time-series, static, and space-time data. The users can also define static map layers (e.g. KML's) for adding contextual information to the map for the region of interest. Examples of these layers could be locations of hydrometeorological gauges, stream and river network, or administrative boundaries.

2.2. Map Markers

Users can inspect interactive locations on the map interface by hovering over an object that shows a tooltip. Tooltips can be customized to show user-defined attributes, metrics, or name of the observation site

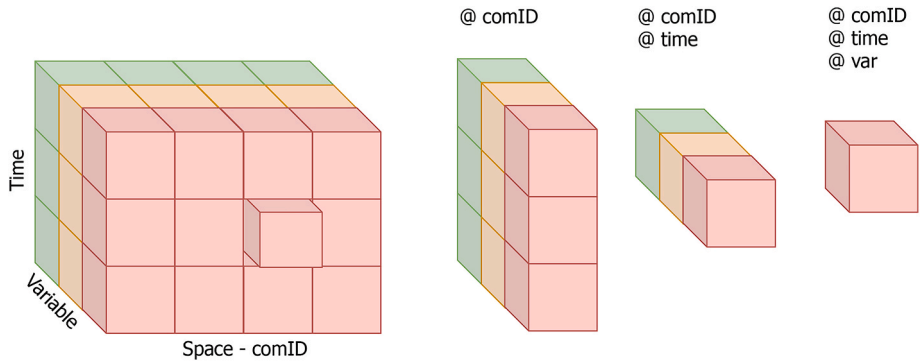

Fig. 3. A schematic of Space-Time-Variable data cubes implemented in Hydrovise.

Table 2
Supported data types in Hydrovise.

	Supported Data Type	Data Format	Description
Time-series	Univariate or Multivariate	CSV	Comma Separated Value
Spatial data	Vector	GeoJSON	Geographical data based on JavaScript Object Notation (Butler et al., 2008)
		KML	Keyhole Markup Language (Open Geospatial Consortium, 2008)
		KMZ	Zipped KML
	Raster	GeoTIFF	Georeference or geocoded raster imagery using Aldus-Adobe's public domain Tagged-Image File Format (Ritter and Ruth, 2000)
Space-time data feed		GeoTIFF/CSV/ binary (Float32)	CSV/binary formats require static geometry

(if available). This feature allows users to access information about locations on the map without plotting the time-series.

Interactive locations on the map are specified as Map Markers. Map Markers are a Feature Collection defined in a GeoJSON formatted file properly described in the configuration file. The underlying geometry could be a point, line, or polygon. These spatial features are sensitive to user-specified events (e.g., click, hover).

2.3. Visualizations

Spatial data (e.g. points, polygons, etc.) are visualized on the map using the Leaflet JavaScript library. Leaflet is an open-source JavaScript library for interactive maps. We selected Leaflet because of its large open-source community contributions, and capability to extend functionality. The basemaps for geographical locations or satellite imagery is provided in the configuration file as a tilemap data source. Leaflet library loads the map tiles depending on the zoom level and map extent. Hydrovise uses the Plotly JavaScript library to visualize time-series and available timestamps for space-time data. Plotly is a data visualization library with a wide range of chart types. We selected this library because of its user/developer community and documentation. The Plotly data canvas is interactive that allows users to zoom, pan, and inspect data for a given observation point. For space-time data visualizations (e.g. radar rainfall, satellite-based soil moisture), we use the Glify library developed as an extension for Leaflet to leverage WebGL web technology. WebGL employs Graphical Processor Unit (GPU) for rendering graphics in web browsers. It is a cross-platform API that uses the OpenGL Shading Language (GLSL) and runs in the HTML5 Canvas element. The visualization of space-time data is controlled by a panel that shows available timestamps for each space-time dataset. When a user selects a timestamp for the spatial data, data are visualized and added to the map inventory for additional layer control options. Data for a given space-time dataset could be a GeoTIFF raster file or geometry to be updated using CSV or Float32 binary file. Using the configuration file, users can add animation control and define a custom color legend for each dataset.

2.4. Extension modules

Extension modules extend the core functionality of Hydrovise. These modules are enabled using user-defined directives in the Hydrovise configuration file. We provide a list of implemented extension modules in the following sections.

2.4.1. Stage-discharge converter

The stage-discharge relationship or rating curves are used to estimate discharge (streamflow) from stage observations. We have developed an extension module that uses a non-linear least square method to fit a polynomial to stage-discharge data (rating curve) and use the fitted function to estimate discharge from stage observations or vice versa.

2.4.2. Flow categories

Flow categories or flood categories provide information on the severity of flow conditions. For example, in the United States, the National Weather Service (NWS) has classified flow stages into five categories. NWS defines these levels as Major, Moderate, Minor Flooding, action (Near Flood), and Normal (No Flooding) state (https://water.weather.gov/ahps/forecasts.php). The United Kingdom's flood information system (https://flood-warning-information.service.gov.uk) has a similar definition for flood levels. These levels serve as a means of communicating flow conditions to the general public and early warning purposes. We have included an extension module that allows users to incorporate flow level categories in streamflow and stage time-series visualizations. Users can provide information about flow categories, or historical flow conditions (e.g., flow return periods). They can visually inspect the hydrologic model predictions for flow categories.

2.4.3. Performance evaluations

Evaluation and validation provide insight into the strengths and limitations of models trying to replicate observed data. Performance evaluation of hydrologic model predictions helps identify the misrepresentations of the physical processes in a hydrologic model. However, this task can quickly become cumbersome as the number of dimensions increase. We refer to dimensions as different model state variables, model setups, input rainfall forcings, etc. Therefore, it is essential to use interactive tools that allow decision-makers/model developers to compare data with model outputs.

Using multiple performance metrics with a visual inspection of the predictions provides further confidence in model selection or performance assessment procedure (Bennett et al., 2013). We have developed an extension module that performs evaluations for the user-defined time period for selected performance metrics across the defined spatial domain. The module incorporates the most common performance evaluation metrics such as Kling-Gupta Efficiency (KGE) proposed by Gupta et al. (2009) and Nash–Sutcliffe Efficiency (NSE), among several others. Also, users can import their pre-computed performance metrics to visualize their model outputs and their performance.

2.4.4. Ensemble time-series navigator

We developed an extension module utilizing Plotly time-series canvas that allows users to navigate and visualize ensemble time-series for a user-selected location. This module shows the timestamp or the ensemble member number on time-series canvas. Users can add or remove multiple time-series to Plotly canvas dynamically.

2.4.5. Watershed boundary outline

This module allows users to define a data source for additional geometry related to a clicked spatial feature. This geometry could be a watershed boundary or relevant geometry to the selected spatial element from Map Markers spatial data. Users can define the data source for the geometry in the configuration file. The data source could be a path to the extracted watershed boundary or an extension module that extracts watershed boundaries from DEM (Digital Elevation Model) data (Sit et al., 2019).

2.5. Deployment

The minimum requirement for the implementation of Hydrovise is a HTTP server (e.g. Apache, Nginx), compatible data types, and a web browser. Potential users can copy the code from the GitHub repository (https://github.com/hydrovise/hydrovise) to a webserver directory. Hydrovise works as an interpreter of the configuration file, where it can

serve multiple projects defined as different configuration files. Therefore, providing the configuration file as a URL to the interpreter in the browser will load the GUI. This feature allows users to share their deployed web link with anyone with Internet access.

3. Example use cases

In this section, we provide four different example use cases of Hydrovise implemented by using different configuration files. Each example illustrates the different capabilities of the software.

3.1. Time-series data visualization and evaluation

This example use case illustrates visualization and evaluation of streamflow time-series. Live demos for this example use case is provided below:

1. Real-time and historical USGS streamflow data browser: http://hydrovise.com/app/?config=examples/ex1/config_usgs.json
2. Hydrologic model evaluations for SMAP satellite-based soil moisture assimilation: http://hydrovise.com/app/?config=examples/ex1/config.json.

We constructed a simple real-time and historical USGS (United States Geological Survey) streamflow data browser using a configuration file. The streamflow data are defined by creating a dynamic path to the USGS National Water Information System (NWIS) web service (USGS, 2019). Fig. 4 shows an example illustration of a real-time USGS streamflow data browser for a region of interest. Fig. 4a and b show stage (*ft*) and streamflow ($m^3.s^{-1}$) time-series using stage-discharge converter module. The data for USGS gauge observations are provided with a 15-min interval. The basin boundary for a selected gauge is highlighted as light gray. Following the NWS color convention, flood categories are shown with magenta, red, orange, and yellow colors defined as Major, Moderate, Minor Flooding and Near-Flooding stages, respectively.

As shown in Fig. 4, Map Inventory allows users to inspect the contextual map layers defined in the configuration, hide/show layer(s), and add their spatial data (vector/raster) to current view. The real-time streamflow data browser can be used for monitoring streamflow at the observation locations provided by the user. This example is useful for real-time hydrologic model evaluations, public, and emergency management offices that coordinate the preparations for flood damage mitigation.

The evaluation module developed for Hydrovise allows users to conduct hydrologic model performance assessments for a selected time period. Also, users can visualize pre-computed performance metrics by defining the data source in the configuration file.

For demonstration, we compare streamflow predictions from the HLM model (Krajewski et al., 2017) for two cases. The first case is referred to as the "open loop" model prediction that does not include any update to model states. The second case is a model prediction using data assimilation of SMAP satellite-based soil moisture in HLM's top layer. We use Ensemble Kalman Filter (Evensen, 2003) with time-dependent variance for perturbations of initial soil moistures, referred to as "EnKFV". We show streamflow predictions at the USGS gauge observation locations for the state of Iowa for the year 2015.

As shown in Fig. 5, Hydrovise populates summary maps of hydrologic model performance metrics for a selected year, statistical measure, and model setup. In this example, USGS locations are color-coded based on Kling-Gupta Efficiency. Note that locations without reference data for the selected year are shown as light gray circle markers. Streamflow predictions after SMAP satellite-based soil moisture assimilation (Fig. 5b) show improvement in majority of the USGS gauge locations compared to open loop model predictions (Fig. 5a).

Fig. 6 illustrates the time-series plot of the HLM model streamflow predictions for the two cases and USGS observation at Cedar River at Cedar Rapids. The color-coded circle markers shows the peak timing difference of the streamflow predictions using EnKFV and USGS observations.

3.2. Hydrologic assessment of precipitation forecasts

This is an example Hydrovise use case for visualization of streamflow forecasts. The live demo for two cases in this example could be accessed at:

- Streamflow forecast time-series for each issue time: http://hydrovise.com/app/?config=examples/ex2/config_issue_time.json.
- Streamflow time-series based on forecast lead time: http://hydrovise.com/app/?config=examples/ex2/config_lead_time.json.

Real-time flood forecasting models use Quantitative Precipitation Estimation (QPE) products for simulating historical streamflow and Quantitative Precipitation Forecast (QPF) products for their forecasts. Flood forecasting models obtain the precipitation forcing from atmospheric models. Because of the chaotic nature of atmospheric flows,

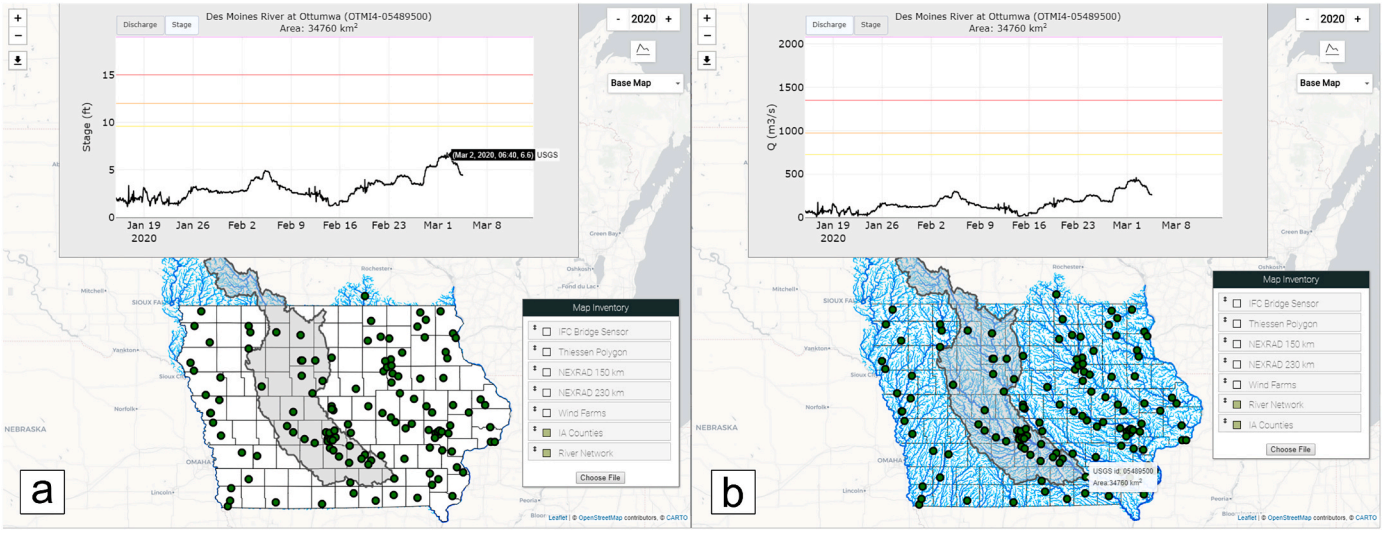

Fig. 4. Example real-time data browser for USGS observations with NWS flood categories and stage-discharge conversion. (a) Stage observations (*ft*) and (b) converted streamflow ($m^3.s^{-1}$) at the USGS gauge at Des Moines River at Ottumwa.

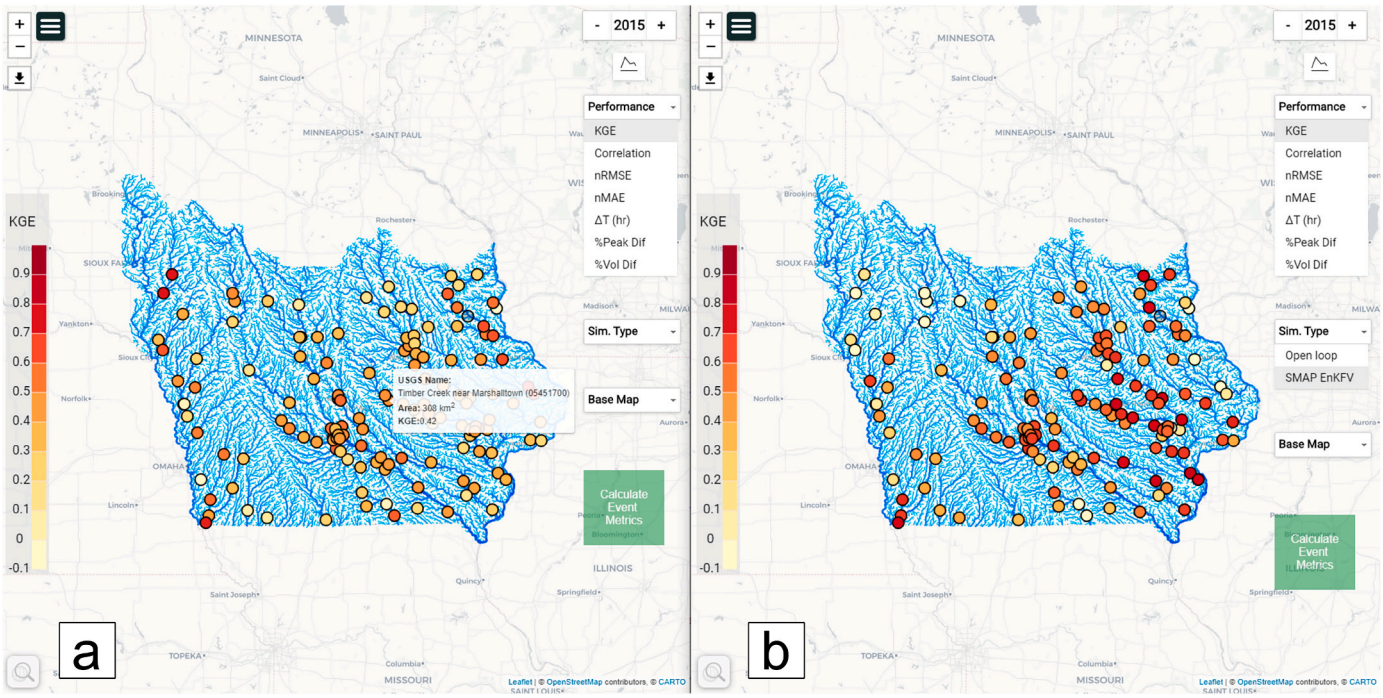

Fig. 5. Example performance metric summary map of Kling-Gupta Efficiency for streamflow predictions at the USGS gauge observation locations for (a) Open Loop and (b) SMAP satellite-based soil moisture assimilated cases.

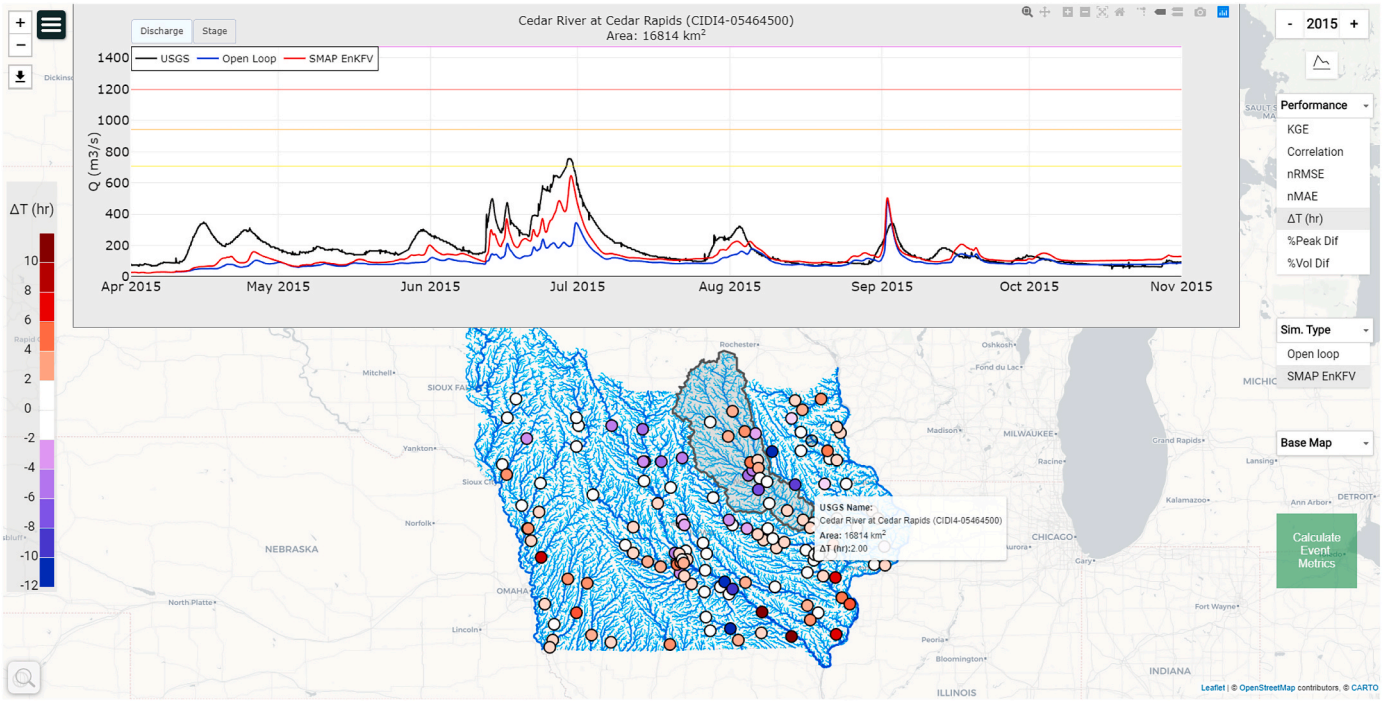

Fig. 6. Streamflow time-series plot for USGS observations (black), predictions from open loop (blue), and data assimilation of SMAP satellite-based soil moisture (red). The map shows color-coded USGS gauge locations using peak timing difference (ΔT) in hours with reference to the USGS gauge observations for the year 2015. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

atmospheric models use data assimilation techniques to account for new observations in model forecasts. Flood forecasting models update their streamflow forecasts as new precipitation forecasts are issued. Therefore, it is essential to understand the streamflow forecasts in the context of precipitation forecasts.

For this example, we used hourly accumulated MRMS (Multi Radar Multi Sensor) and IFC (Iowa Flood Center) QPE products for simulating historical streamflow. For streamflow forecasts, we use HRRR (High-Resolution Rapid Refresh) atmospheric model's (Benjamin et al., 2016) QPF. HRRR product is issued every hour with lead times starting from zero to 18 hour. Streamflow forecasts from HLM model are provided for the next five days starting from every precipitation forecast issue time.

We use the ensemble time-series navigator described in Section 2.4.4 to answer two key questions: How good were streamflow predictions driven by precipitation forecasts at different issue time? How do streamflow predictions evolve with the different precipitation forecast lead times?

To answer the first question, we organized streamflow predictions by precipitation forecast issue times. Fig. 7 shows USGS gauge observation and streamflow forecasts using MRMS and IFC QPE and HRRR QPF products for a user-selected issue time. The time-series show streamflow forecasts for the next five days from a selected forecast issue time. As shown in this figure, users can navigate to streamflow forecasts issued at any time for the selected USGS gauge location. The extension module allows users to add and remove time-series for a chosen product for visual comparisons with USGS gauge observations.

We answer the second question by organizing streamflow forecasts data by their lead time. Fig. 8 illustrates USGS gauge observations and streamflow forecast time-series for a 7-hour forecast lead time. Streamflow forecast time-series corresponding to different lead times can be added to the time-series plot shown in this figure. This extension module allows users to evaluate the predictions based on forecast lead time.

3.3. Grid-based hydrometeorological data visualization

This example illustrates the visualization of a space-time process on a regular grid. The live demo for this example use case is available at http://hydrovise.com/app/?config=examples/ex3/config.json.

As one of the critical hydrologic variables, soil moisture controls runoff production (e.g., Crow et al., 2018; Jadidoleslam et al., 2019b), and agricultural productivity (e.g., Hargreaves, 1975). SMOS (Soil Moisture Ocean Salinity) and SMAP (Soil Moisture Active Passive) are the two satellite missions that provide soil moisture information by using remote sensing techniques (e.g., Kerr et al., 2010; O'Neill et al., 2015).

In recent years, several studies have been conducted for validation, comparison, and evaluation of satellite-based soil moisture with field sensor observation networks (e.g., Colliander et al., 2019; Ma et al., 2019; Jackson et al., 2012; Tavakol et al., 2019). An interactive web-based tool can facilitate visualization and validation of satellite-based soil moisture products that could complement current validation studies. We provide an example use case of Hydrovise for visualization of satellite, ground-based soil moisture, and hydrologic model storage.

Fig. 9 shows soil moisture time-series from SMAP, SMOS, field sensor observations, and hydrologic model soil storage from the Hillslope Link Model (HLM) (Krajewski et al., 2017). SMAP data are posted on EASE v2 grid (Brodzik et al., 2012) which we use as the reference geometry shown with the light gray color. For comparison, SMOS satellite soil moisture data are gridded to the SMAP grid. Also, we show model storage as percentiles to visualize the soil moisture sub-grid variability originating from HLM model with higher spatial resolution. Iowa Flood Center operates a network of over 30 field soil moisture sensors that are shown with white circles. Field sensor soil moisture data are averaged for more than one collocated sensors in SMAP grid.

Fig. 9 shows that soil moisture oscillations from model and different products show good agreement for the selected grid highlighted on the map. The sub-grid variability decreases for higher median soil moisture values which is consistent with findings from previous studies (e.g., Famiglietti et al., 2008; Jadidoleslam et al., 2019a).

Fig. 10a illustrates SMAP satellite-based soil moisture estimation over the State of Iowa for a satellite overpass approximately at 6:00 p.m. on May 23, 2016. Fig. 10b shows hourly MRMS radar-based rainfall map for at 3:00 p.m. on May 23, 2016 on the same map. Higher soil moisture regions from SMAP in central Iowa could be described by antecedent rainfall. These figures illustrate the Hydrovise ability to display raster data.

3.4. River network-based data visualization

This example illustrates the visualization of a river network-based data. The live demo for this example use case is available at http://hydrovise.com/app/?config=examples/ex4/config.json. Portable version of this example use case is included in GitHub Examples folder. The live demo can be accessed at http://hydrovise.com/app/?config=examples/

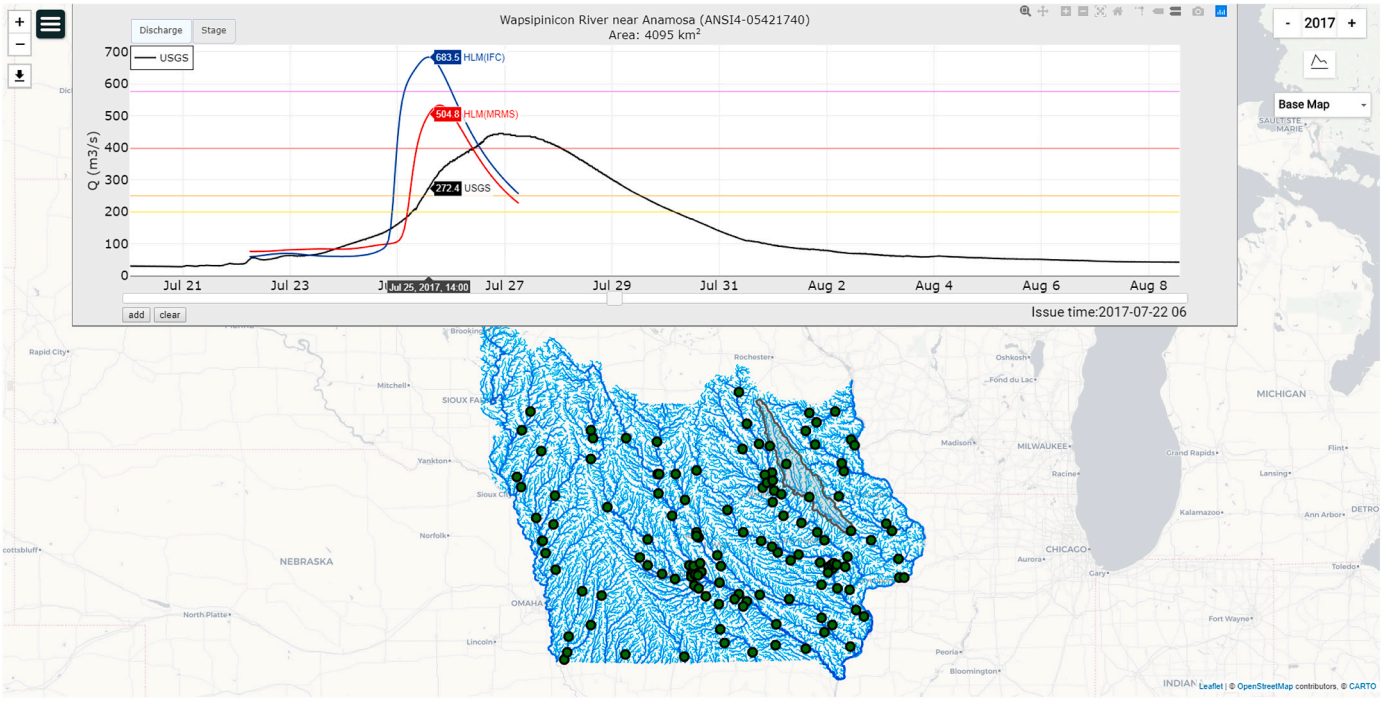

Fig. 7. USGS gauge observation (black) and streamflow forecast time-series from HLM (Hillslope Link Model) using IFC (blue) and MRMS (red) QPE and HRRR QPF issued at July 22, 2017, at 6:00 a.m. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

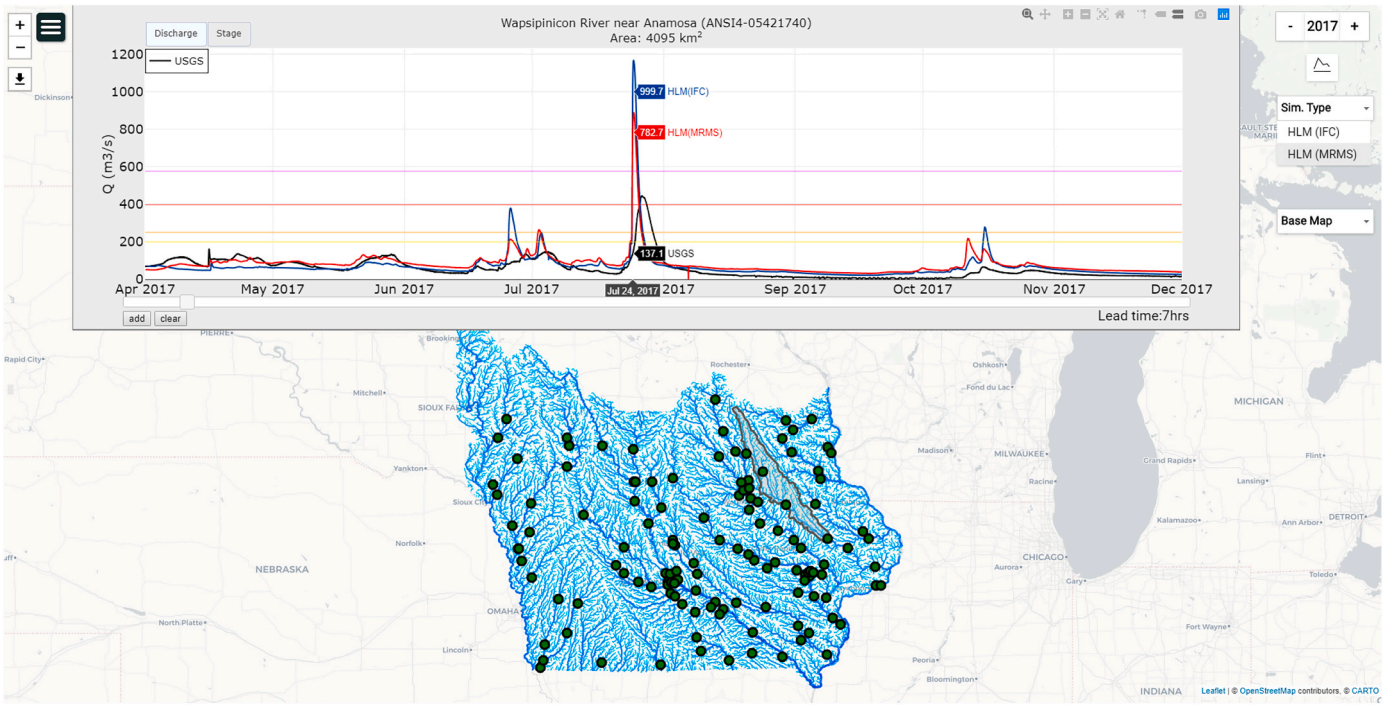

Fig. 8. Example time-series of USGS gauge observation (black) and streamflow forecast from HLM (Hillslope Link Model) using IFC (blue) and MRMS (red) QPE and HRRR QPF product with 7 h lead time for 2017 at Wapsipinicon River near Anamosa. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

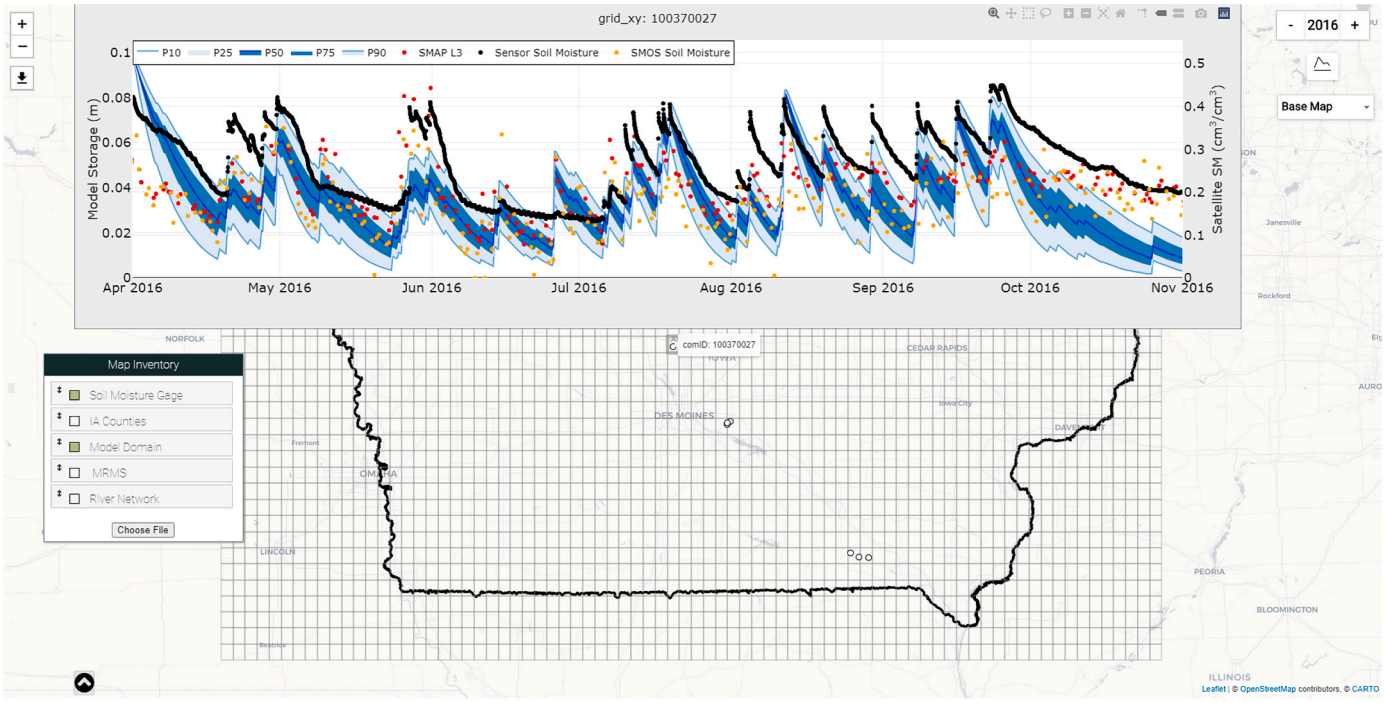

Fig. 9. Soil moisture time-series from field sensor observations (black), SMAP (red dots), and SMOS (orange dots) satellite-based soil moisture, and up-scaled model prediction percentiles for a selected pixel (shades of blue) on the EASE v2 grid (SMAP grid). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

ex4_min/config.json.

A river network has self-similar structure that describes surface water movement in a watershed or a region. River network-based visualizations can help users inspect the streamflow evolution in space and time. In this example, we use Hydrovise to visualize flood potential over state of Iowa's river network. Similarly, it is possible to visualize other variables such as water quality, and contaminant loads. Fig. 11 shows the river network for the state of Iowa and streamflow time-series for a user-selected river segment. The network includes more than 100,000 streams with Horton order three and above. Users can inspect

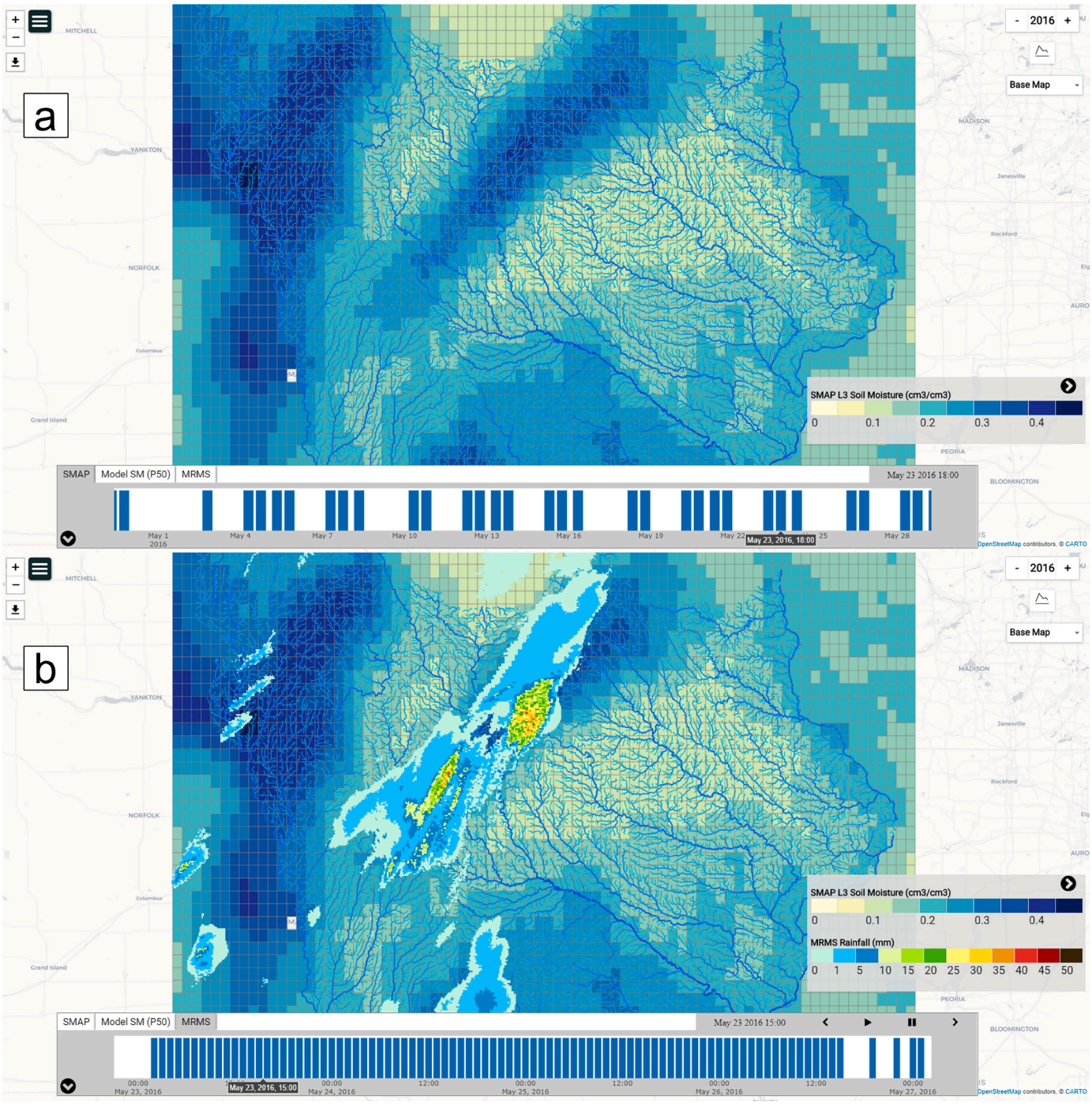

Fig. 10. Example grid-based visualization in Hydrovise for (a) SMAP satellite-based soil moisture 6:00 p.m. May 23, 2016 and (b) MRMS hourly radar-based rainfall for 3:00 p.m. May 23, 2016.

streamflow time-series for selected river segment(s) by providing time-series data source in the configuration file.

Flood potential index is an indicator for communicating the streamflow condition irrespective of upstream drainage area of a river segment. It is defined as the ratio of streamflow to mean annual flood (Quintero et al., 2020),

$$I = \frac{Q}{Q_{maf}} \tag{1}$$

where Q is streamflow ($m^3.s^{-1}$) and Q_{maf} is the mean annual flood for each river segment. Quintero et al. (2020) derived a power-law relationship for mean annual flood for the state of Iowa,

$$Q_{maf} = 3.12A^{0.57} \tag{2}$$

where A is the upstream drainage area of the river segment. We use Eqs. (1)-(2) to calculate flood potential maps.

Fig. 12 shows flood potential over the state of Iowa during the 2016 September flooding event. Also, hourly accumulated MRMS radar rainfall for September 22, 2016, at 3:00 a.m. is shown on map.

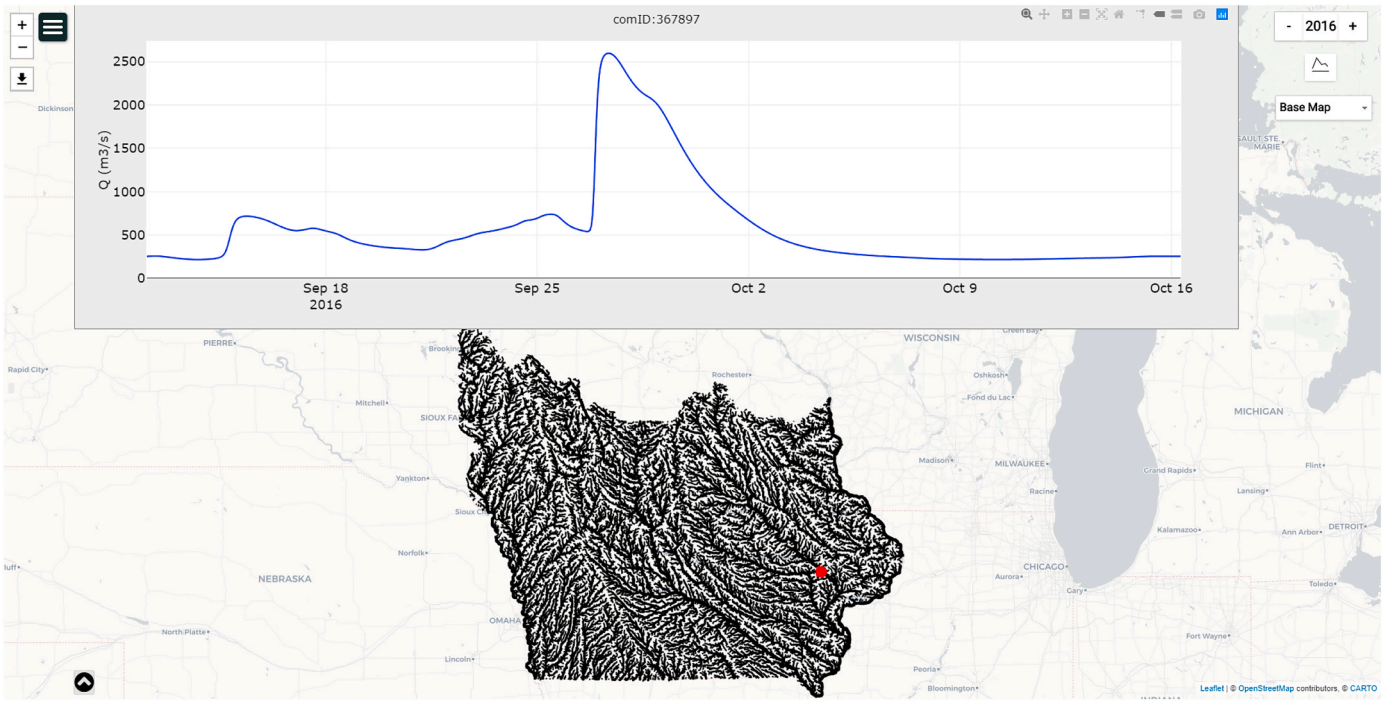

Fig. 11. Streamflow time-series from Hillslope Link Model for a user-selected river segment (red dot) on the river network. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

4. Discussion

The core concept incorporated in Hydrovise is the configuration file designed to include essential information about the map, data sources, visualization styling. This feature decreases development effort for web-based visualization and evaluation of hydrologic data. This also allows for substantial code re-use where one source code can serve multiple projects defined as different configuration files.

The target audience for Hydrovise is entry-level programmers, graduate students, research scientists, and decision-makers that have limited expertise in web development. Graduate students and researchers working on hydrology and water resources topics could use

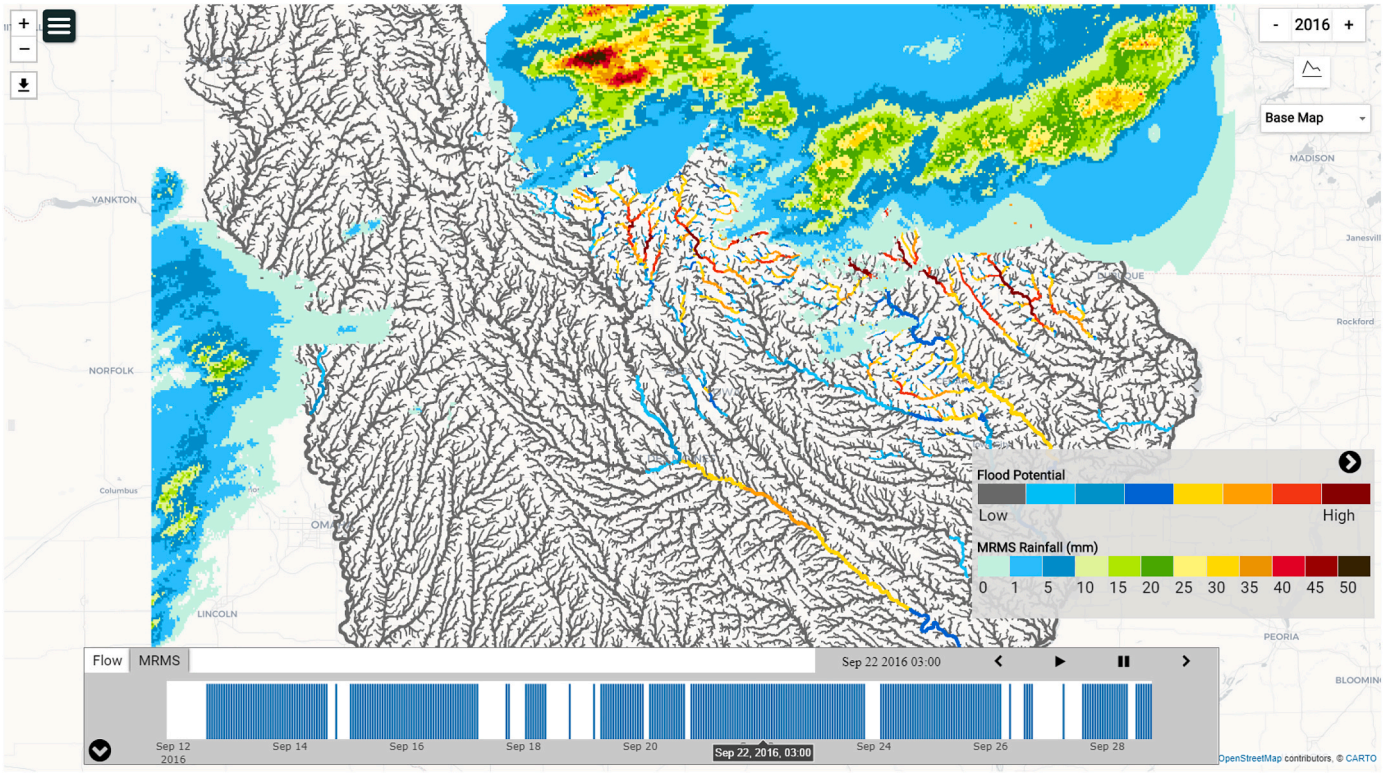

Fig. 12. River network-based flood potential and MRMS hourly-accumulated radar rainfall maps for a user-selected timestamp.

Hydrovise to communicate their results with their mentors or colleagues in an interactive web-based environment. Decision-makers (e.g., county emergency managers), who rely on hydrologic data during weather-related emergencies, could inspect the observation data and hydrologic model forecasts using Hydrovise. However, like any software, Hydrovise is also limited to its core objectives and scope. Therefore, for custom functionality or additional features, users must have basic knowledge of web development programming languages.

Extension modules presented in this work allows users to visualize additional data, gain insights on their hydrologic problem, and evaluate data for a selected time period. This ability helps users to identify spatial patterns through hydrologic data visualizations, evaluation, or validation. Jackson et al. (2019) provide a review of performance metrics for hydrologic predictions. In addition to data visualizations, Hydrovise incorporates multiple performance metrics for the evaluations, while new performance metrics can be added to the evaluation extension module. Users can also use pre-computed performance metrics calculated in Python or MATLAB by other open-source libraries (e.g., Jackson et al., 2019).

Database and web mapping server (e.g., GeoServer) implementations are useful for handling large space-time data visualizations and sharing. For example, the Iowa Flood Information System developed by the Iowa Flood Center uses a PostgreSQL database for storing and fetching data (Krajewski et al., 2017). Tethys uses database and web mapping servers for extensive environmental data (Swain et al., 2016). However, dependencies add layers of complexity to these solutions requiring more expertise. Also, it takes more effort to securely host and deploy the solutions that have more dependencies. On the other hand, a broad community of users in environmental science may not need these tools for their applications. Therefore, we used the concept of data cubes with a file system and a Dynamic Path Creator to handle multi-variate space--time data using the file system. Our approach is not a substitution for a database. It is an alternative for potential users that have limited knowledge (or time to learn) of database deployment. Therefore, we included the options to define internal/external data services as data sources. A data service can be used to access data in a database, or a chunk of data in other file formats such as HDF5, netCDF or other compressed data types given that user should have the expertise. We have not implemented these data types as they could have different formats and dimensions. This capability in Hydrovise makes the solution minimal, modular, and extendible. To reduce dependencies and data preparation workload, we selected the most common data types such as CSV, GeoTIFF, GeoJSON (Table 2) that could be easily written or read by users using free and open-source software such as QGIS, Python etc.

5. Summary and future developments

Our motivation in this work was to develop a practical tool for visualization of hydrologic data and the evaluation of hydrologic models that try to replicate observations. We developed an open-source software that lowers the barrier for implementation using a configuration, rather than code-based approach. Hydrovise allows users to conduct a qualitative and quantitative assessment on hydrologic data using its visualization and evaluation capabilities.

We have shown four different use cases for hydrologic data. The first example illustrated time-series data visualizations including real-time USGS streamflow data and hydrologic model visualizations and performance evaluations. Second, we provided a use case of a unique module that allows users to visualize forecast time-series. In the third example, we demonstrated grid-based multi-product soil moisture data visualization in which we used field sensors, satellite-based, and model soil moisture data. Finally, we visualized space-time network-based hydrologic data by using Hydrovise. These are only a few example use cases to demonstrate the capabilities and capacity of the developed software. Hydrovise can accommodate other environmental data that have an inherent space-time structure.

Open-access publications increase the transparency of the methods, results, and peer-review process in publications. However, due to the lack of easy-to-deploy tools and software, authors do not show the tendency to publish their companion dataset with their scientific findings. Hydrovise, as an entirely client-side and a portable solution, could be leveraged for sharing data in open-access journals along with published data for more transparency.

Hydrovise is an open-source software that could benefit from user community feedback, feature requests, and developer community for improvements and developments. These developments could include adding SQLite for portable serverless database as back-end and local file system, and SpatiaLite for adding geospatial analyses.

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.

Acknowledgements

This work is partially funded by NASA Science Utilization of the Soil Moisture Active-Passive Mission program Grant NASA Grant “NNX16AQ57G" and a project funded under the provisions of section 104 of the Water Resources Research Act of 1984 annual base grants (104b) distributed through the Iowa Water Center as “Graduate Student Supplemental Research Competition” with grant No. “2020IA095B" and Mid-America Transportation Center (MATC TRB RiP No. 91994-5) with contract number “69A3551747107”. We also gratefully acknowledge the user feedback provided by graduate students and research scientists working at the Iowa Flood Center at The University of Iowa.

References

Benjamin, S.G., Weygandt, S.S., Brown, J.M., Hu, M., Alexander, C.R., Smirnova, T.G., Olson, J.B., James, E.P., Dowell, D.C., Grell, G.A., Lin, H., Peckham, S.E., Smith, T.L., Moninger, W.R., Kenyon, J.S., Manikin, G.S., 2016. A North American hourly assimilation and model forecast cycle: the rapid refresh. Mon. Weather Rev. 144, 1669–1694. https://doi.org/10.1175/MWR-D-15-0242.1.

Bennett, N.D., Croke, B.F., Guariso, G., Guillaume, J.H., Hamilton, S.H., Jakeman, A.J., Marsili-Libelli, S., Newham, L.T., Norton, J.P., Perrin, C., Pierce, S.A., Robson, B., Seppelt, R., Voinov, A.A., Fath, B.D., Andreassian, V., 2013. Characterising performance of environmental models. Environ. Model. Software 40, 1–20. https://doi.org/10.1016/j.envsoft.2012.09.011.

Brendel, C.E., Dymond, R.L., Aguilar, M.F., 2019. An interactive web app for retrieval, visualization, and analysis of hydrologic and meteorological time series data. Environ. Model. Software 117, 14–28. https://doi.org/10.1016/j.envsoft.2019.03.003.

Brodzik, Mary J., Billingsley, Brendan, Haran, Terry, Raup, Bruce, Savoie, Matthew H., 2012. EASE-Grid 2.0: Incremental but significant improvements for earth-gridded data sets. ISPRS Int. J. Geo-Inf. 1 (1), 32–45. https://doi.org/10.3390/ijgi1010032.

Butler, H., Daly, M., Doyle, A., Gillies, S., Schaub, T., Schmidt, C., 2008. Geojson specification. Geojson.org.

Colliander, A., Cosh, M.H., Misra, S., Jackson, T.J., Crow, W.T., Powers, J., McNairn, H., Bullock, P., Berg, A., Magagi, R., Gao, Y., Bindlish, R., Williamson, R., Ramos, I., Latham, B., O'Neill, P., Yueh, S., 2019. Comparison of high-resolution airborne soil moisture retrievals to SMAP soil moisture during the SMAP validation experiment 2016 (SMAPVEX16). Remlote Sens. Environ. 227, 137–150. https://doi.org/10.1016/j.rse.2019.04.004.

Crow, W.T., Chen, F., Reichle, R.H., Xia, Y., Liu, Q., 2018. Exploiting soil moisture, precipitation, and streamflow observations to evaluate soil moisture/runoff coupling in land surface models. Geophys. Res. Lett. 45, 4869–4878. https://doi.org/10.1029/2018GL077193.

Demir, I., Conover, H., Krajewski, W.F., Seo, B.C., Goska, R., He, Y., Mceniry, M.F., Graves, S.J., Petersen, W., 2015. Data-enabled field experiment planning, management, and research using cyberinfrastructure. J. Hydrometeorol. 16, 1155–1170. https://doi.org/10.1175/JHM-D-14-0163.1.

Django Software Foundation, 2018. The Web Framework for Perfectionists with Deadlines. Django.

ECMA, 1999. 262: ecmascript language specification. http://www.ecma-international.org/ecma-262/8.0/index.html.

Evensen, G., 2003. The ensemble kalman filter: theoretical formulation and practical implementation. Ocean Dynam. 53, 343–367. https://doi.org/10.1007/s10236-003-0036-9.

Famiglietti, J.S., Ryu, D., Berg, A.A., Rodell, M., Jackson, T.J., 2008. Field observations of soil moisture variability across scales. Water Resour. Res. 44, 1–16. https://doi.org/10.1029/2006WR005804.

Ghapanchi, A.H., Wohlin, C., Aurum, A., 2014. Resources contributing to gaining competitive advantage for open source software projects: an application of resource-based theory. Int. J. Proj. Manag. 32, 139–152. https://doi.org/10.1016/j.ijproman.2013.03.002.

Goodall, J.L., Horsburgh, J.S., Whiteaker, T.L., Maidment, D.R., Zaslavsky, I., 2008. A first approach to web services for the National Water Information System. Environ. Model. Software 23, 404–411. https://doi.org/10.1016/j.envsoft.2007.01.005.

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., Pirahesh, H., 1997. Data cube: a relational aggregation operator generalizing group-by, cross-tab, and sub-totals. Data Min. Knowl. Discov. 1, 29–53. https://doi.org/10.1023/A:1009726021843.

Gupta, H.V., Kling, H., Yilmaz, K.K., Martinez, G.F., 2009. Decomposition of the mean squared error and NSE performance criteria: implications for improving hydrological modelling. J. Hydrol. 377, 80–91. https://doi.org/10.1016/j.jhydrol.2009.08.003.

Hargreaves, G.H., 1975. Moisture availability and crop production. Trans. ASAE 18, 980–984. https://doi.org/10.13031/2013.36722.

Jackson, E.K., Roberts, W., Nelsen, B., Williams, G.P., Nelson, E.J., Ames, D.P., 2019. Introductory overview: error metrics for hydrologic modelling – a review of common practices and an open source library to facilitate use and adoption. Environ. Model. Software 119, 32–48. https://doi.org/10.1016/j.envsoft.2019.05.001.

Jackson, T.J., Bindlish, R., Cosh, M.H., Zhao, T., Starks, P.J., Bosch, D.D., Seyfried, M., Moran, M.S., Goodrich, D.C., Kerr, Y.H., Leroux, D., 2012. Validation of soil moisture and Ocean Salinity (SMOS) soil moisture over watershed networks in the U.S. IEEE Trans. Geosci. Rem. Sens. https://doi.org/10.1109/TGRS.2011.2168533.

Jadidoleslam, N., Mantilla, R., Krajewski, W.F., Cosh, M.H., 2019a. Data-driven stochastic model for basin and sub-grid variability of SMAP satellite soil moisture. J. Hydrol. 576, 85–97. https://doi.org/10.1016/j.jhydrol.2019.06.026.

Jadidoleslam, N., Mantilla, R., Krajewski, W.F., Goska, R., 2019b. Investigating the role of antecedent SMAP satellite soil moisture, radar rainfall and MODIS vegetation on runoff production in an agricultural region. J. Hydrol. 579, 124210. https://doi.org/10.1016/j.jhydrol.2019.124210.

Kerr, Y.H., Waldteufel, P., Wigneron, J.P., Delwart, S., Cabot, F., Boutin, J., Escorihuela, M.J., Font, J., Reul, N., Gruhier, C., Juglea, S.E., Drinkwater, M.R., Hahne, A., Martin-Neira, M., Mecklenburg, S., 2010. The SMOS L: new tool for monitoring key elements ofthe global water cycle. Proc. IEEE 98, 666–687. https://doi.org/10.1109/JPROC.2010.2043032.arXiv:1504.03050.

Krajewski, W.F., Ceynar, D., Demir, I., Goska, R., Kruger, A., Langel, C., Mantilla, R., 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. Bull. Am. Meteorol. Soc. 98, 539–554. https://doi.org/10.1175/BAMS-D-15-00243.1.

Ma, H., Zeng, J., Chen, N., Zhang, X., Cosh, M.H., Wang, W., 2019. Satellite surface soil moisture from SMAP, SMOS, AMSR2 and ESA CCI: a comprehensive assessment using global ground-based observations. Remlote Sens. Environ. 231 https://doi.org/10.1016/j.rse.2019.111215.

Maidment, D., 2002. Arc Hydro : GIS for Water Resources. ESRI Press, Redlands, Calif.

Midha, V., Palvia, P., 2012. Factors affecting the success of open source software. J. Syst. Software 85, 895–905. https://doi.org/10.1016/j.jss.2011.11.010.

Murugesan, S., Gangadharan, G.R., 2012. Harnessing Green it: Principles and Practices. IEEE ComputerSociety/IEEE/Wiley. https://doi.org/10.1002/9781118305393.

O'Neill, P.E., Njoku, E.G., Jackson, T.J., Chan, S., Bindlish, R., 2015. SMAP Algorithm Theoretical Basis Document: Level 2 & 3 Soil Moisture (Passive) Data Products. Technical Report.

Open Geospatial Consortium, I., 2008. KML | Ogc®. URL. http://www.opengeospatial.org/standards/kml.

Python Foundation, 2016. About PythonTM | Python.org. URL: https://www.python.org/about/.

Quintero, F., Krajewski, W.F., Rojas, M., 2020. A flood potential index for effective communication of streamflow forecasts at ungauged communities. J. Hydrometeorol. 21, 807–814. https://doi.org/10.1175/jhm-d-19-0212.1.

Ritter, N., Ruth, M., 2000. GeoTIFF specifications. URL. http://geotiff.maptools.org/spec/geotiff1.html.

Sit, M., Sermet, Y., Demir, I., 2019. Optimized watershed delineation library for server-side and client-side web applications. Open Geospatial Data, Software. Stand. 4, 1–10. https://doi.org/10.1186/s40965-019-0068-9.

Swain, N.R., Christensen, S.D., Snow, A.D., Dolder, H., Espinoza-Dávalos, G., Goharian, E., Jones, N.L., Nelson, E.J., Ames, D.P., Burian, S.J., 2016. A new open source platform for lowering the barrier for environmental web app development. Environ. Model. Software 85, 11–26. https://doi.org/10.1016/j.envsoft.2016.08.003.

Swain, N.R., Latu, K., Christensen, S.D., Jones, N.L., Nelson, E.J., Ames, D.P., Williams, G.P., 2015. A review of open source software solutions for developing water resources web applications. Environ. Model. Software 67, 108–117. https://doi.org/10.1016/j.envsoft.2015.01.014.

Tavakol, Ameneh, Rahmani, Vahid, Quiring, Steven M., Kumar, Sujay V., 2019. Evaluation analysis of NASA SMAP L3 and L4 and SPoRT-LIS soil moisture data in the United States. Remote Sens. Environ. 229, 234–246. https://doi.org/10.1016/j.rse.2019.05.006.

USGS, 2019. USGS water data for the nation. URL. http://waterdata.usgs.gov/nwis.

Vitolo, C., Elkhatib, Y., Reusser, D., Macleod, C.J., Buytaert, W., 2015. Web technologies for environmental big data. Environ. Model. Software 63, 185–198. https://doi.org/10.1016/j.envsoft.2014.10.007.

Walker, J.D., Chapra, S.C., 2014. A client-side web application for interactive environmental simulation modeling. Environ. Model. Software 55, 49–60. https://doi.org/10.1016/j.envsoft.2014.01.023.

Wong, B.P., Kerkez, B., 2016. Real-time environmental sensor data: an application to water quality using web services. Environ. Model. Software 84, 505–517. https://doi.org/10.1016/j.envsoft.2016.07.020.